MODULAR CIRCUIT TECHNOLOGY

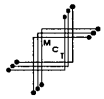
USER'S MANUAL MOD-EMUP-A PC-BASED UNIVERSAL PROGRAMMER and TESTER

Made in Taiwan

MOD-EMUP-A

Universal Programmer

Owner's Reference Guide



MODULAR CIRCUIT TECHNOLOGY

Credits

AMD is a trademark of Advanced Micro Devices, Inc.

Fujitsu is a trademark of Fujitsu Ltd.

Hitachi is a trademark of Hitachi America, Ltd.

IBM, PC and AT are trademarks of International Business Machines Corporation.

MCT and the MCT logo are trademarks of Modular Circuit Technology.

MMI is a trademark of Monolithic Memories, Inc.

MS-DOS is a trademark of Microsoft Corporation.

National Semiconductor is a trademark of National Semiconductor Corporation.

PC-DOS is a trademark of International Business Machines Corporation and Microsoft Corporation.

Signetics is a trademark of Philips Corporation.

Texas Instruments is a trademark of Texas Instruments, Inc.

All other brand and product names are trademarks or registered trademarks of their respective holders.

Contents

Gang-Adaptor

Tester

Setup, File, and Utility

Technical Reference

1

Introduction

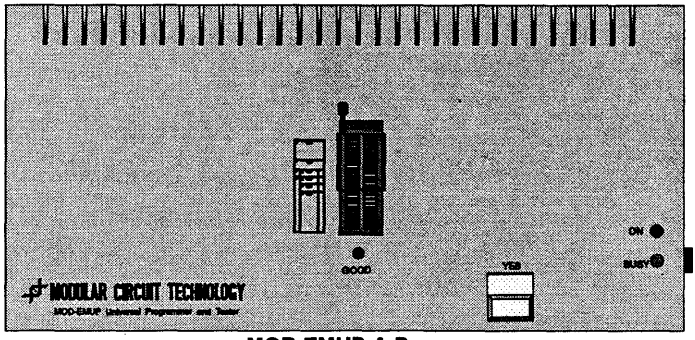
What is the MOD-EMUP-A Universal Programmer?

The MOD-EMUP-A is a high-performance universal programmer and tester, combining functions that are usually only found in stand-alone systems costing thousands of dollars. When used to its maximum potential, it can perform the functions of a variety of different units; thus, it is a very economical solution for most engineers. Furthermore, the MOD-EMUP-A has a complete line of specialized adaptors, giving it the flexibility to take full advantage of the industry's latest technology. Some of its functions are:

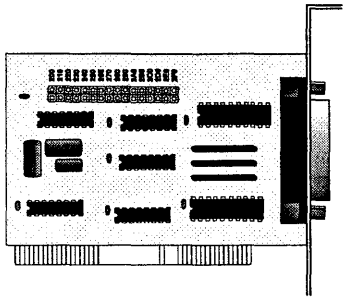
- Programs EPROMs, EEPROMs and CMOS EPROMs from 16Kb to 4Mb capacity
- Full-featured PLD programmer, from PLDs, EPLDs, EEPLDs, FPLs, PALs, GALs, PEELs and logic sequencers
- IC tester function tests TTL, CMOS, static-column, dynamic RAM, SIPP, and SIMM
- Bipolar PROM programming capabilities
- Serial PROM programming capabilities
- Wide-range of microprocessor programming
- Capability to program PLCC, PGA, and other package types with use of adaptors

What's Included

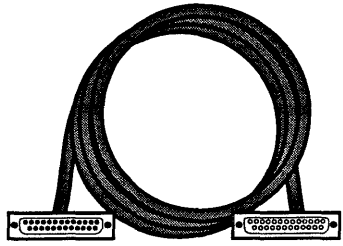
Your MOD-EMUP-A should come with the following:



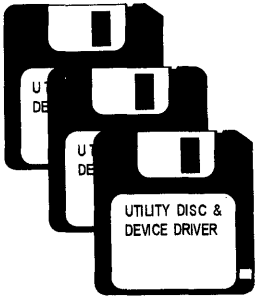
MOD-EMUP-A Programmer



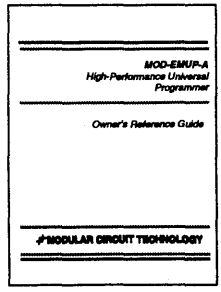
Host Adapter Card



Adapter Cable



3 Utility Diskettes



MOD-EMUP-A User Guide

Figure 1-1. Package Contents

Conventions Used

The following conventions are used throughout this guide:

- Commands you are to enter from your computer's keyboard appear in bold lowercase Courier font. For example:

  install c: ~

  You can enter commands in lower-, upper-, or mixed-case characters, unless otherwise directed.

- Messages and other information you see on your computer screen appear in mixed case Courier font. For example:

  File Name:

- Keys you are to press appear in angle brackets. For example, <Enter> denotes the Enter key on your keyboard.

- Combination keystrokes are indicated by a series of keys, which are not separated by spaces. For example:

  Press <Ctrl><Alt><Del> means to press and hold the Ctrl and Alt keys, and press the Delete key.

- Hexadecimal numbers are followed by a uppercase "H", as in 80H or A23H.

Note: Notes provide advice or suggestions or call your attention to important information.

Tech Tip: Tech Tips are helpful hints about working with your MOD-EMUP-A and any associated hardware and software.

Caution: Cautions alert you to operations that could cause damage to the MOD-EMUP-A programmer or your system.

Warning: Warnings alert you to operations that could cause injury to you or someone else.

System Requirements

To use the Universal Programmer, your system must have:

- ❑ An 8-bit or 16-bit ISA bus slot
- ❑ 3.5-inch floppy disk drive
- ❑ 512Kb available base memory
- ❑ 8MHz bus speed or slower

Tech Tip: If you receive an out of memory error, try preforming a clean-boot (restarting using a bootable DOS diskette). This will prevent any device drivers from being loaded and will maximize your available base memory. To create a "clean" bootable diskette using DOS version 5.0 or higher:

1. Insert a diskette into your A: drive.
2. At the DOS prompt type:

   ```
   format a: /s
   ```

   This command will format the diskette and copy the startup files from your operating system onto the diskette.

For more information on creating a bootable diskette, refer to your DOS manual.

2

Installation

This section contains step-by-step instructions for installing and configuring your MOD-EMUP-A adaptor card.

Setting the Jumper for the MOD-EMUP-A I/O Address

The adaptor card for your MOD-EMUP-A programmer is designed to give great flexibility in port address assignments. The default jumper setting of 2E0 (shown in Figure 2-1) should work for most standard installations.

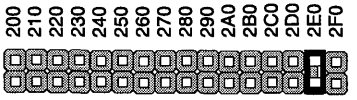


Figure 2-1. Jumper Settings for MOD-EMUP-A Adapter Card

Tech Tip: If jumper 2E0 does not work with your system, try another jumper setting, starting with jumper 200.

Taking Precautions

Before performing any of the procedures listed in this section, be sure to review the following precautions. These guidelines will help protect both you and your system.

Power Sources

To avoid exposure to electric shock and damage to your system:

1. Turn off all power to your computer.
2. Unplug all cables connected to it, including the computer's power cord.

Tech Tip: Be sure to label all cables, adapter cards, slots, and connectors so you'll know where to reinstall/reconnect them when you reassemble your system.

Electrostatic Discharge

Static electricity that builds up in your body can cause serious damage to electronic components. To protect your system, follow these static reducing guidelines:

1. Do not remove an electronic component from its static-shielded bag until you are ready to install it.
2. Wear a wrist grounding strap. If a wrist grounding strap is not available, ground yourself frequently by touching an unpainted metal portion of your computer case or power supply for a few seconds.
3. Don't wear tennis shoes or shoes with rubber soles.
4. Use anti-static padding on all working surfaces.
5. Avoid static-inducing carpeted areas.
6. If an electronic component is passed from one person to another, the two should touch hands (without holding the device) first.

Installing the MOD-EMUP-A Adapter Card

Warning: Turn all power to your computer and the MOD-EMUP-A programmer OFF, and unplug all cables and power cords.

Open your computer case according to the documentation that came with your case or computer. If the documentation does not contain the appropriate instructions, contact the dealer from whom you purchased the case/computer.

Tech Tip: Be sure to label all cables, adapter cards, slots, and connectors so you'll know where to reinstall/reconnect them when you reassemble your system.

1. Turn off all power to your system, and unplug all cables, including the computer's power cord.
2. Locate an empty 8-bit or 16-bit expansion slot on the motherboard. Carefully plug the MOD-EMUP-A into the slot using a slight rocking motion.

   The edge connector (the gold *fingers*) on the card should be firmly seated in the slot.
3. Secure the adaptor card to the computer chassis using the retaining screws.
4. Close the computer case, and secure it according to the documentation provided with your case/computer.

Connecting the MOD-EMUP-A

After the adaptor card has been installed, the programmer can be connected to your system using the 25-pin connector cable provided with your MOD-EMUP-A.

Caution: Do not use a standard serial cable to connect the adaptor to the MOD-EMUP-A. Many serial cables only have 9 wires, even though they may have 25 pins on each end.

You may safely turn on the MOD-EMUP-A programmer after the adaptor card, cable, and programmer are properly connected to your system. Be sure to check the LEDs on the MOD-EMUP-A for

proper operation when power is applied. Refer to Figure 2-2 for the location of the On/Off switch and LEDs.

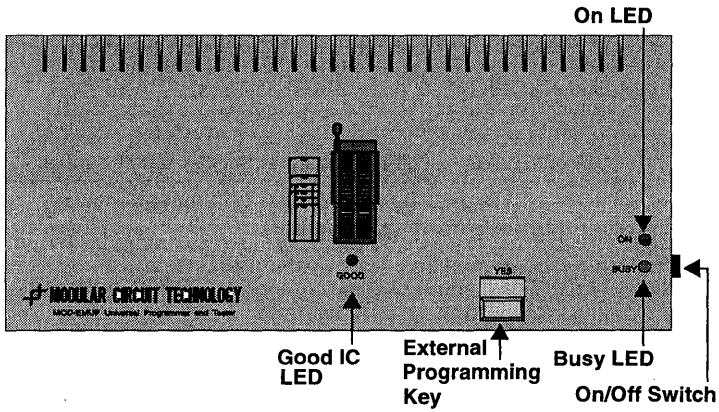


Figure 2-2. MOD-EMUP-A LED Configuration When Power is On

The MOD-EMUP-A Software

The utility software for the MOD-EMUP-A comes on three (3) diskettes. These utilities control the MOD-EMUP-A in different ways.

Note: In an effort to keep the MOD-EMUP-A software as current as possible, software revisions are done periodically. The software listing should reflect your software closely, although some differences may be apparent.

Disk1

EPP1024B	EXE	39,184	9-12-94	3:13p
EPP1024W	EXE	40,304	9-27-94	11:24a
BPPGM	EXE	49,072	11-01-94	5:28p
SEEP1	EXE	75,536	3-22-95	11:35a
SEEP2	EXE	32,496	9-15-94	5:31p
MPU1	EXE	35,312	9-14-94	4:52p
EEP1	EXE	52,800	10-17-94	1:48p
PGM48	EXE	31,104	9-15-94	12:10p
PGM51	EXE	41,056	3-26-96	11:03a
PGMZ8	EXE	33,856	9-16-94	3:01p
EPLD1	EXE	40,352	9-16-94	3:31p
FPLP1	EXE	37,024	9-16-94	9:02a
FPLP2	EXE	31,488	9-16-94	9:20a
GAL1	EXE	34,672	3-18-96	10:10a
GAL2	EXE	40,304	9-04-95	11:06a
GAL3	EXE	30,640	9-15-94	1:52p
PALP1	EXE	43,376	9-15-94	2:46p
PALP2	EXE	38,000	9-15-94	3:22p
PALP3	EXE	37,232	7-26-95	9:10a
PALP4	EXE	33,808	9-23-94	11:49a
PALP5	EXE	34,944	7-25-95	9:15p
PEEL1	EXE	40,928	3-22-95	11:48a
PEEL2	EXE	26,336	9-15-94	5:41p
EPP512	EXE	42,800	7-26-95	9:46a
README01	DOC	71,707	1-22-94	8:39p
25 file(s)			1,014,331 bytes	
			437,248 bytes free	

Disk2

ICTEST	EXE	38,080	11-17-94	11:20a
DS5000	EXE	30,960	9-14-94	1:44p
CMOS40	LIB	11,342	7-29-91	10:02a
CMOS45	LIB	3,657	2-06-90	12:16p
TTL74	LIB	29,152	11-03-93	3:55p
CD4040	VEC	224	11-07-91	9:34a
TTL06	VEC	64	11-07-91	9:34a
PA7024	EXE	29,952	9-15-94	2:53p
PA7128	EXE	30,240	9-15-94	3:16p
PA7140	EXE	30,432	9-15-94	3:36p
5AC324	EXE	27,968	9-16-94	3:10p
COP800	EXE	33,680	9-14-94	5:31p
7C361	EXE	25,648	9-14-94	11:49a
MAX	EXE	36,688	9-16-94	4:44p
PIC16	EXE	39,024	11-02-94	10:15a
PLS105	EXE	34,016	9-16-94	9:43a
ST62	EXE	32,208	9-14-94	2:41p
Z86E08	EXE	31,680	9-15-94	4:04p
PIC17	EXE	34,688	10-18-94	9:15a
17P10X	EXE	32,112	9-14-94	2:02p
EPC	EXE	32,688	9-16-94	5:48p
7C33X	EXE	32,752	9-14-94	9:59a
PALTEST	EXE	32,416	9-15-94	5:19p
README02	DOC	16,508	1-22-94	9:57p
24 file(s)			646,179 bytes	
			805,376 bytes free	

Disk3

PROG	DEV	139,350	7-25-95	5:17p
DIAG03B	EXE	25,920	12-05-95	9:42a
BIN2HEX	EXE	12,945	9-06-93	9:23a
CUT2	EXE	9,263	5-03-94	1:54p
CUT4	EXE	9,483	5-03-94	1:54p
DUMP	EXE	9,141	8-25-93	2:39p
EDBIN	EXE	23,498	3-26-91	3:21p
HEXBIN	EXE	13,471	8-17-93	12:12p
HEXBIN2	EXE	17,625	8-24-94	10:38a
REVERT	EXE	8,693	9-14-93	2:29p
SEP2MB	EXE	11,101	7-27-90	4:02p
SERPC	EXE	24,716	12-08-93	5:03p
SHUFF2	EXE	9,145	8-17-93	12:17p
SHUFF4	EXE	9,515	8-17-93	12:17p
SPLIT2	EXE	9,165	8-17-93	12:14p
SPLIT216	EXE	9,267	8-17-93	12:16p
SPLIT3	EXE	9,243	8-24-93	11:40a
SPLIT4	EXE	9,431	8-17-93	12:15p
SPLIT8	EXE	10,141	8-17-93	12:16p
ACCESS	EXE	39,632	2-15-95	2:03p
README03	DOC	1,840	1-22-94	9:59p
21 file(s)			412,585 bytes	
			1,039,872 bytes free	

Before installing the software to your system, it is a good idea to make backup copies. Use the backup copies rather than the master diskettes for installing the software. The diskettes are not copy-protected. To make a backup copy, use the DOS DISKCOPY command. See your DOS manual for details.

Installing the Software

To install the software onto your system:

1. Create a directory on your drive by typing the following:

   `C:\ mkdir emup <Enter>`

2. Copy the MOD-EMUP-A program into the newly created emup directory on your hard drive by inserting each program diskette into your floppy drive A: and typing:

   `A:\> copy *.* C:emup <Enter>`

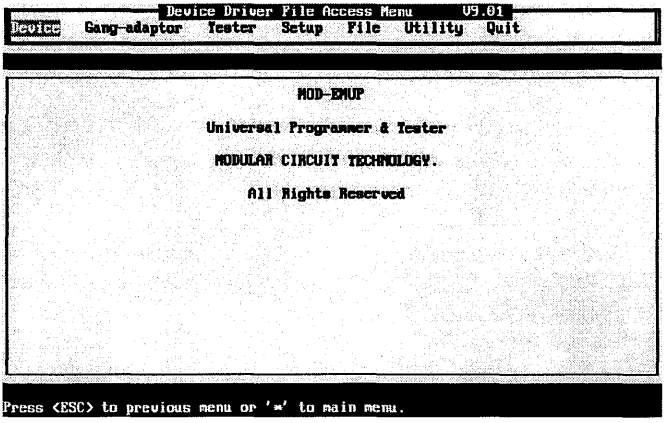
Launching the Software

Once the installation is complete, change back to your C: drive and type:

`C:\cd emup <Enter>`

`C:\EMUP> access <Enter>`

The following screen will appear:



Note: If you do not have a hard drive, you will be limited to executing individual files from diskette. Be sure to use the backup copies rather than the originals as your working diskettes. For a listing of manufacturers, device types, algorithms, and corresponding device driver program names, see the section "Cross Reference" on page 6-2.

There are seven sub-menus available from the main MOD-EMUP-A menu: "Device", "Gang-adaptor", "Tester", "Setup", "File", "Utility", and "Quit". The following sections look at the functions listed under each of these options. Examples of programming and testing different types of chips are also provided.

Placing Chips on the MOD-EMUP-A

The MOD-EMUP-A has a ZIF (Zero Insertion Force) socket designed to accept DIP style chips of up to 40 pins. The MOD-EMUP-A case provides an illustration showing the proper alignment of chips next to the programming socket (see Figure 2-3).

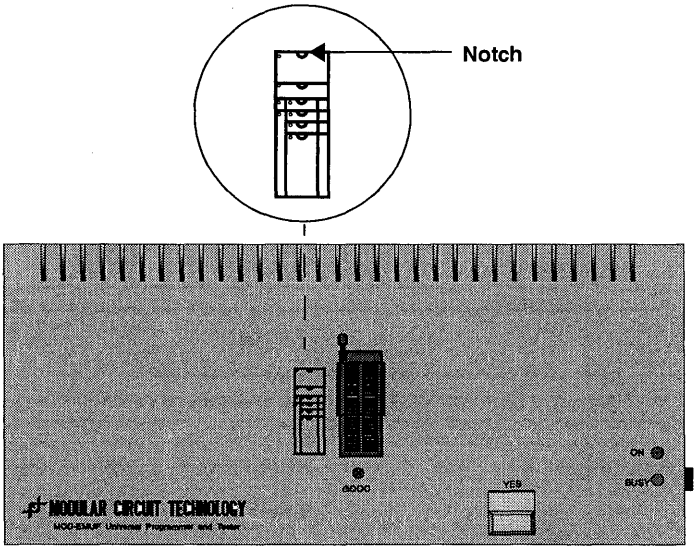


Figure 2-3. Chip Orientation on the MOD-EMUP-A

Note: Note the "Notch" on the case drawing. This corresponds with the "Notch" on most IC chips, and indicates the location of Pin 1.

To insert a chip, lift the lever 90 degrees (into the upright position) and insert the connectors of the chip into the MOD-EMUP-A

socket. Be sure the "bottom" edge (opposite the notch) of the chip is flush against the bottom of the socket. When the chip is seated properly in the socket, gently push the socket lever forward 90 degrees to secure the connection with the chip.

MOD-EMUP-A External Programming Key

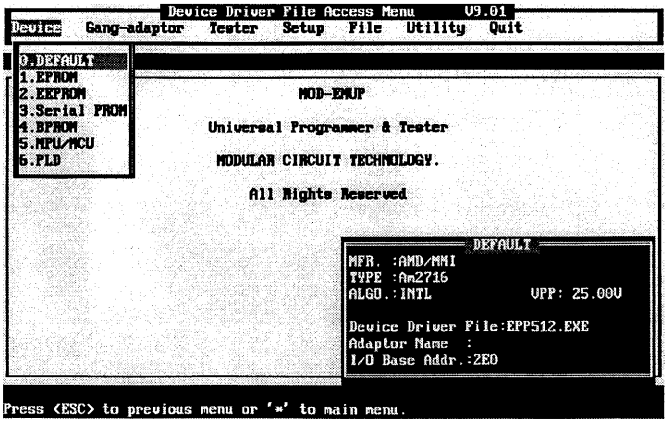
There is only one external key on the MOD-EMUP-A. It is labeled YES (see Figure 1-1 on page 1-2). Depressing this key is the same as depressing the "Y" key on the keyboard. Whenever you are prompted to press "Y", pressing YES on the MOD-EMUP-A programmer will have the same effect.

User Notes

3

Device

Choosing "Device" from the Access Main Menu will call a sub-menu with the choices: "DEFAULT", "EPROM", "EEPROM", "Serial PROM", "BPROM", "MPU/MCU", and "PLD". These devices are programmed by choosing a manufacturer and device type. Once you have selected the manufacturer and device type, the Programming Main Menu will appear. The series of screens for each device are similar, varying only by providing information specific to the device that you select for programming.

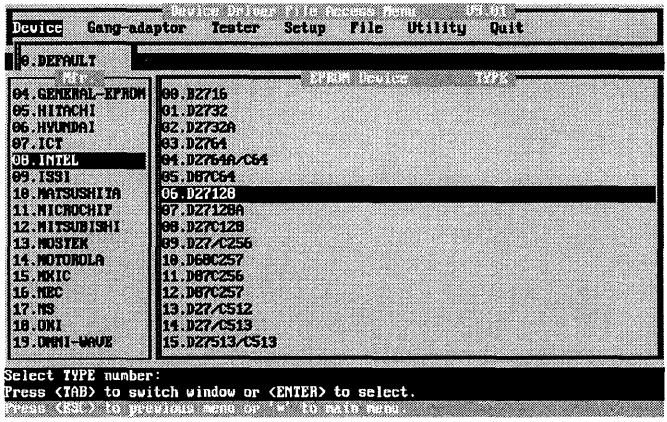


Default

This option skips the manufacturer and device list, taking you directly to the programming screen. If you have already selected a manufacturer and device during a previous programming session, the MOD-EMUP-A software defaults to these settings.

EPROM

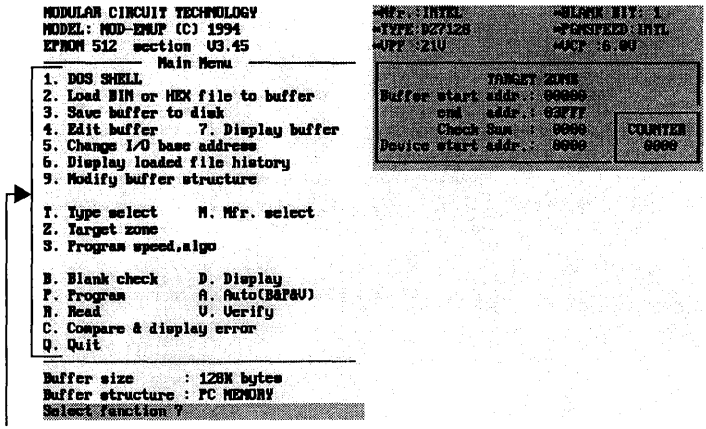
Choosing EPROM from the device list will bring up the following screen:



This screen provides a list of manufactures and EPROM device types. To program an Intel D2716 for example, scroll down the manufacturer list using the arrow keys. When the highlight bar is over the manufacturer Intel, press <Enter>. The highlight bar will move to the next field, allowing you to scroll down the device list to the D2716, and press <Enter>. Once you have selected a manufacturer and device, the Programming Main Menu screen will appear.

Programming Main Menu

The following screen is the Programming Main Menu for the EPROM. Each device has slight menu option variations, depending on the specific device you are programming.



Programming Main Menu Options

The Programming Main Menu allows you to perform a variety of functions which are explained in the following sections.

DOS Shell

Choosing the "DOS Shell" option from the Programming Main Menu will cause the software to search for COMMAND.COM on the boot disk drive. The following screen will appear:

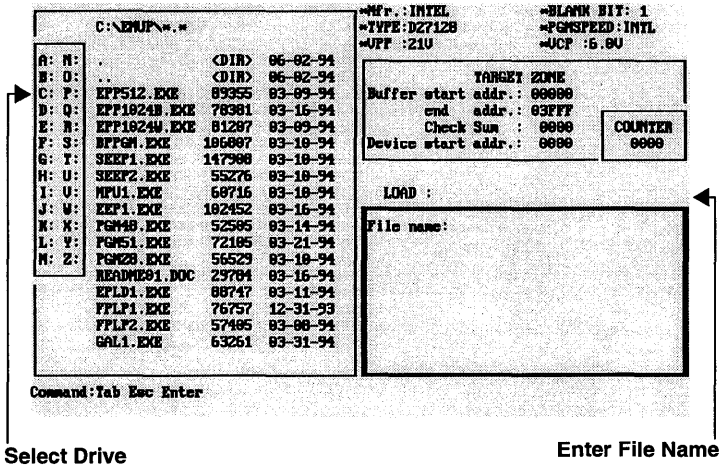
```
Type EXIT to return to Programmer menu

Microsoft(R) MS-DOS(R) Version 6.20
            (C)Copyright Microsoft Corp 1981-1993.

C:\EMUP>>
```

The software will execute the COMMAND.COM command and pass control back to DOS, allowing you to perform DOS operations while running the MOD-EMUP-A Access Software in the background. To return to the Programming Main Menu, type:

exit <Enter>

Load BIN or HEX File to Buffer

Choosing the "Load BIN or HEX File to Buffer" option from the Programming Main Menu will call the following screen:



Select Drive **Enter File Name**

There are two ways to load a BIN or HEX file into the buffer.

1. When the initial BIN/HEX screen appears, a flashing cursor appears following the prompt:

   `File name:`

   Type in the complete file name, including the drive letter and path and press <Enter>.

   or

2. Press the <Tab> key to activate the scrolling file list. Using the <Arrow>, <Page up>, <Page down>, and <Enter> keys, you are able to move freely within your disk's directory to locate the file to be loaded. Once found, highlight the file name and press <Enter>.

   To view files on an alternate drive, press the <Tab> key to activate the scrolling file list and then press the letter of the drive where the file is loaded.

The file name that you have selected will appear in the LOAD window, followed by the choices:

<B>in, <I>ntel HEX, <M>otorola S HEX

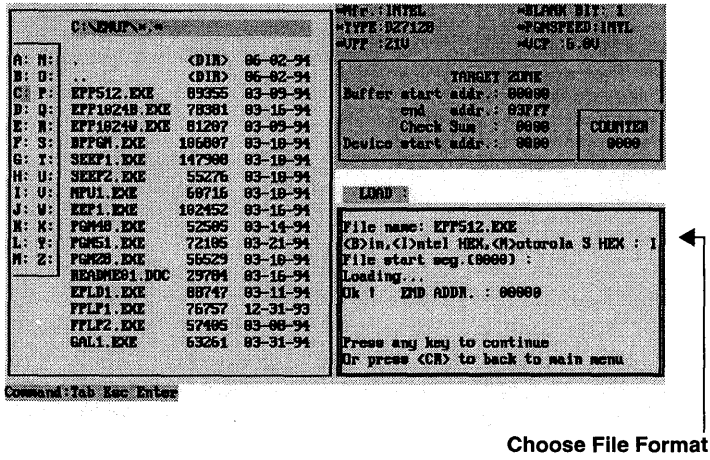
- <B>in

  A BIN file contains the exact data in the format that it will be programmed into the chip. This is the default format used by the MOD-EMUP-A for saving the buffer data.

- <I>ntel HEX and <M>otorola S HEX

  A HEX file is an ASCII file generated by either an Intel or Motorola compiler. The file format complies with the specifications developed by Intel and Motorola.

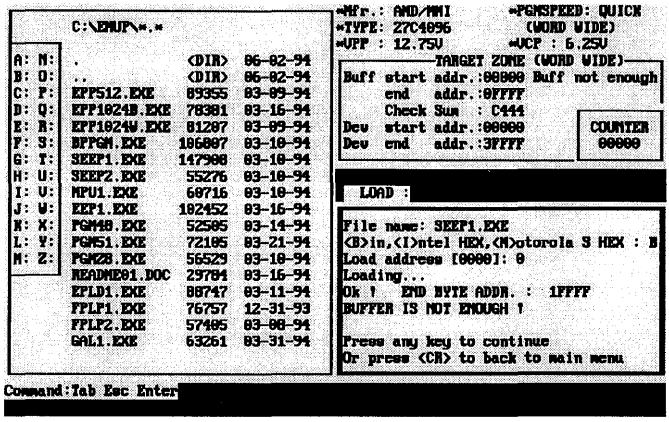
Choose the proper format, then type in the appropriate address and press <Enter>.



Choose File Format

If you would like to load another file into the buffer, press any key, otherwise press <Enter> to return to the Programming Main Menu.

If you are working with a large IC, or you do not have enough memory, you may see the following error:



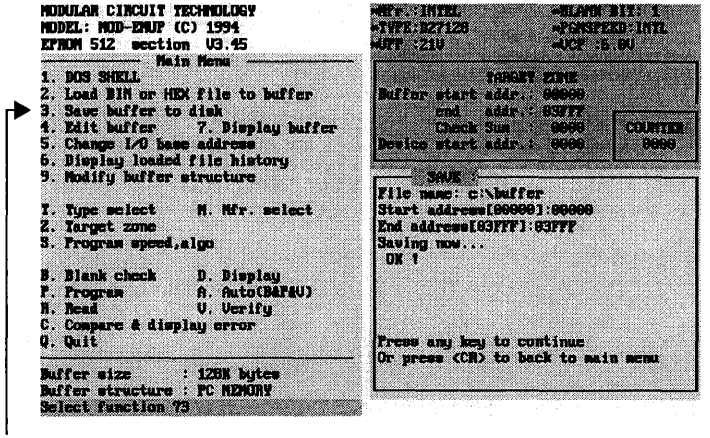
Low Memory Error

If this occurs, you will need to modify your buffer structure, which is discussed in "Modify Buffer Structure" on page 3-11.

Save Buffer to Disk

The buffer stores the data to be programmed into the EPROM. Once the appropriate data is loaded into the buffer, you may choose the option "Save Buffer to Disk" from the Programming Main Menu. A flashing prompt will appear in the SAVE window. Type in the complete file name, including the drive letter and path, and press <Enter>. A second prompt will appear requesting the starting address of the portion of the buffer to be saved. Type in the starting address and press <Enter>. The last prompt will appear, asking you to enter the ending address of the buffer to be saved. Type in the ending address and press <Enter>. When the file has been successfully saved, you may press any key to continue saving

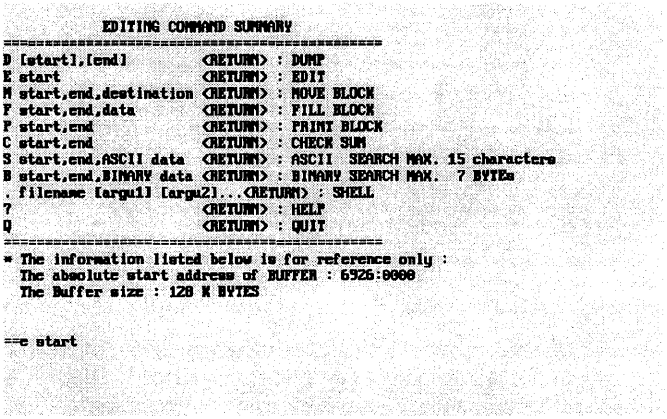
additional files, or press <Enter> to return to the Programming Main Menu.



Save Buffer to Disk Option

Edit Buffer

You may edit the information stored in the buffer by choosing the option "Edit Buffer" from the Programming Main Menu. The following editing command summary screen will appear:



This screen lists the command summary for editing the information that is currently loaded in the buffer. Each command line within the window shows the proper syntax for entering a command.

A flashing cursor will appear at the bottom of the command window next to the "==" prompt. Enter your command here.

For example, to display (or "dump") the buffer contents between addresses 0000 and 574, type:

```
==D 0000,574 <Enter>
```

In this example, "D" is the command for dump, "0000" is the starting address and "574" is the ending address.

To edit the buffer, type the command "E", followed by the starting address of the section you want to edit. For example:

```
==E 0170 <Enter>
```

The default starting address is 0000. Use the arrow keys to select the information you want to edit.

To exit the "Edit Buffer" mode and return to the Programming Main Menu, press <Esc>, then type:

```
==Q <Enter>
```

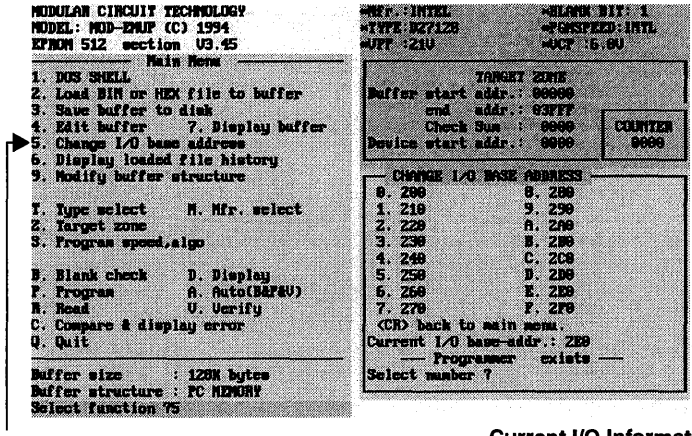
The following is a sample buffer dump from starting address 0000 to ending address 170:

```
00000  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00010  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00020  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00030  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00040  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00050  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00060  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00070  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00080  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00090  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
000A0  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
000B0  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
000C0  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
000D0  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
000E0  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
000F0  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00100  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00110  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00120  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00130  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00140  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00150  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00160  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
00170  00 00 00 00 00 00 00 00 -00 00 00 00 00 00 00 00 ................
 [ESC]: Back to command prompt, [Ctrl X]: Toggle between BINARY and ASCII
```

To edit the buffer contents, use the arrow keys, page up and page down to position the cursor and enter new information. When you have finished entering your changes, press <Esc>, then <Enter> to return to the "==" prompt.

Change I/O Base Address of Adaptor Card

The "Change I/O Base Address of Adaptor Card" option in the Programming Main Menu allows the I/O base address of the adaptor card to be changed when setting up the MOD-EMUP-A address. The screen will appear as follows:



I/O Base Address Option

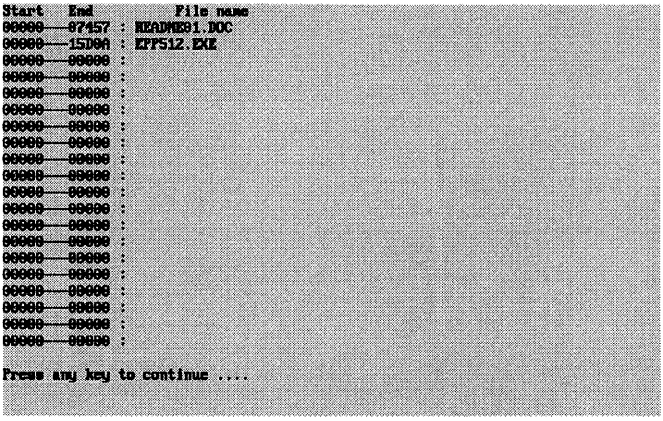
Current I/O Information

The bottom right-hand window of this screen displays the current I / O address. To select a new address, type in the appropriate number or letter at the "Select number?" prompt. It is important that the jumper settings on the adaptor card are set properly for the address you wish to use. If the adaptor card is not set properly, a message "Programmer does not exist" will display. If this occurs, check your adaptor card jumper settings. Refer to page "Setting the Jumper for the MOD-EMUP-A I/O Address" on page 2-1 of this manual for more information on jumper settings.

Note: The I/O address can also be changed from the pull-down menu "Setup" by choosing "I/O Base Addr." (see page 6-1).

Display Loaded File History

Choosing the "Display Loaded File History" option will display the file history and size of what has previously been loaded into the memory buffer. It will also show the file that is currently in the buffer. The following is an example of this screen:



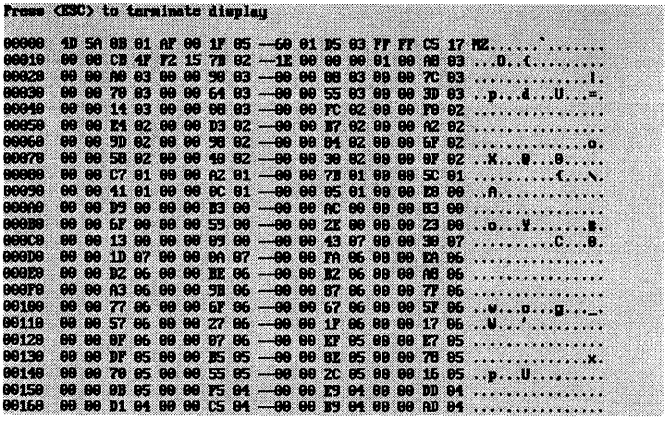
This screen indicates that README01.DOC was previously loaded into memory and started at address 0000 and ended at address 07457. It also shows that EPP512.EXE is currently loaded into the memory buffer and starts at address 05D0A and ends at address 05D0A.

To return to the Programming Main Menu, press any key.

Display Buffer

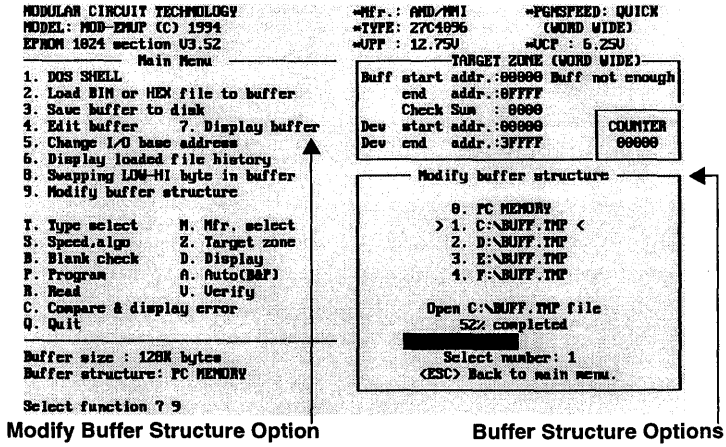
Choosing the "Display Buffer" option from the Programming Main Menu will display the current contents from the starting address to the ending address of the memory buffer. The screen will scroll through the addresses, by depressing <Pause> the scroll can be stopped. Press <Enter> to continue the scroll, and <Esc> to terminate the display.

An example of a memory buffer display might appear as follows:



Modify Buffer Structure

If you are working with a large IC, or you do not have enough memory, you may receive an error message asking you to modify the buffer structure. The settings for the current buffer structure are located in the lower left corner of the Programming Main Menu. The default setting is for PC MEMORY, with a size of 128 Kbytes. To change this setting, choose "Modify Buffer Structure" from the Programming Main Menu. A list of possible buffer structures will appear in the lower right corner of the screen. Select the number that corresponds with the drive where you would like to create a new buffer directory. The following screen shows a buffer being created on the C: drive.

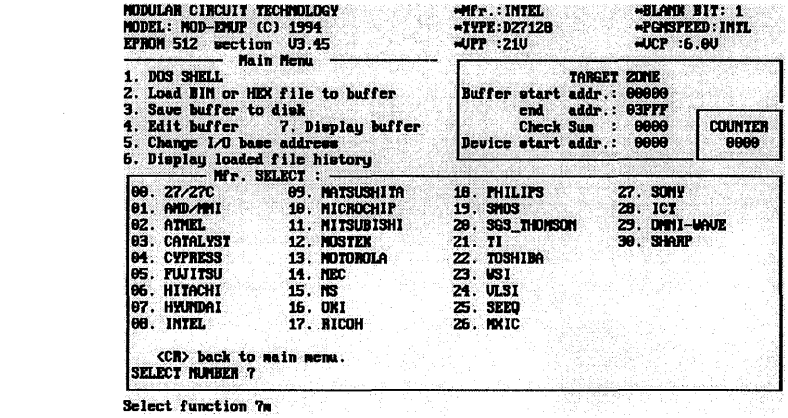


Modify Buffer Structure Option

Buffer Structure Options

Mfr. Select

Choosing "Mfr. Select" from the Programming Main Menu returns you to the list of manufacturers for the type of chip you selected. If you have already selected a manufacturer and need to change your selection, use this option. The following is a sample manufacturer selection screen:

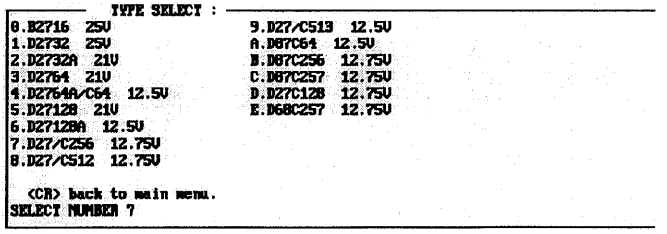


Note: If you are running individual files from diskette (see the note in "Installation" on page-2-7), you will use this screen to choose a manufacturer. For a listing of manufacturers, device types, algorithms, and the corresponding device driver program names, see the section "Cross Reference" on page 6-2.

Type Select

Choosing "Type Select" from the Programming Main Menu returns you to the list of the chip types for the manufacturer you selected. If

you need to change the chip type during your programming, use this option. The following is a sample "Type Select" screen:



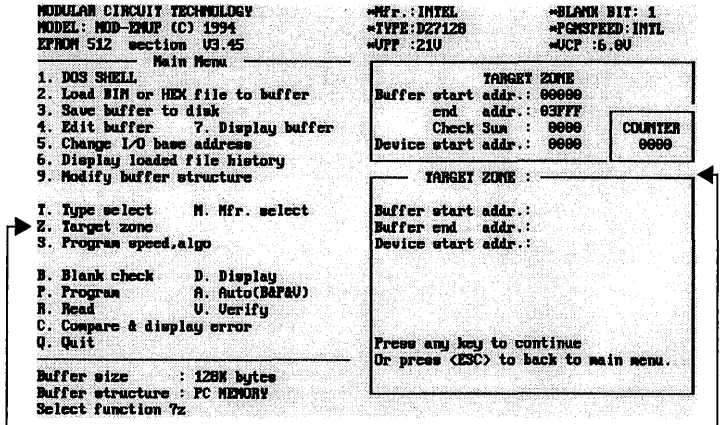
If a change of chip type is needed, but the manufacturer is the same, use this screen to change the chip type. "Type Select" and "Mfg. Select" can be done in any order.

Caution: EPROM chips are not always clearly labeled. If you cannot determine a correct voltage, use the lower voltage for that chip type first. If the chip does not program correctly, erase the chip and use a higher voltage.

Target Zone

This option allows the starting address, ending address, and the device starting address to be modified in the buffer (target zone). Any modification will be updated in the Target Zone status field.

The following is an example of the "Target Zone" screen:

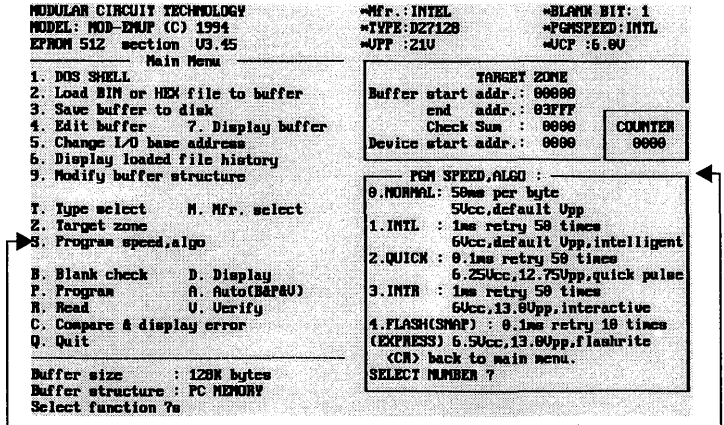


Target Zone Option

Target Zone Screen

Program Speed, Algo

The "Program Speed, Algo" screen allows you to change the programming algorithm for the chip you have chosen. An algorithm is automatically selected when you choose the manufacturer and chip type. The available options for your chip type will be displayed in this window. The screen for the INTEL NSD27128 will appear as follows:



Program Speed, Algorithm Option

Program Speed, Algorithm Scree

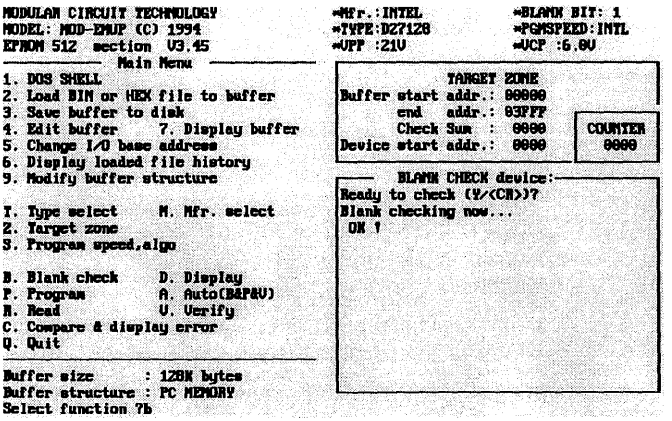
Note: Do not change the default programming algorithm unless you have detailed information regarding your chip.

Blank Check

The "Blank Check" option checks the chip in the ZIF socket for any data that may be present on the chip. If data is present on the chip, programming can still be done in the unprogrammed area.

Note: The "Blank Check" operation is very sensitive to electronic noise, and can sometimes incorrectly report that a chip has data. To avoid incorrect reporting, do not operate the MOD-EMUP-A in areas of high electronic emissions, such as near power trunk lines, heavy appliances, etc.

If the chip is blank, the following screen will appear:



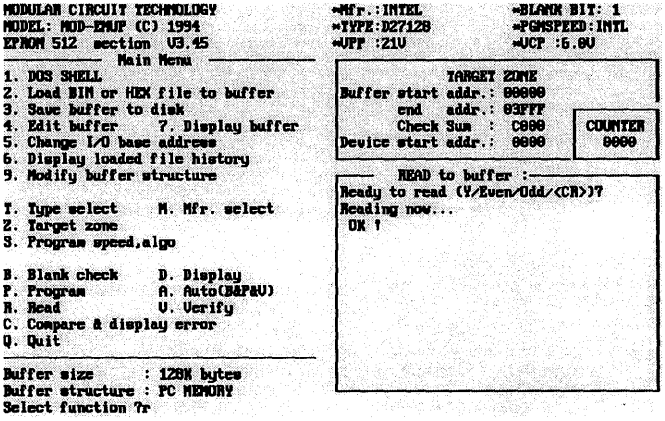
Tech Tip: If a chip fails to blank check, perform a "Read" operation on the chip, and choose "Display Buffer." If the buffer shows a FF in every location, then the chip is blank.

If the chip is not blank, erase the chip with an Ultraviolet EPROM eraser.

Read

The "Read" option will read the contents of a chip and transfer the data to the memory buffer. When option R is selected, the screen will display the message "Reading now...". When the chip has been read into the buffer, an "OK" message will display and the check sum in the status field will display.

If the chip reads properly into the buffer, the screen will appear as follows:

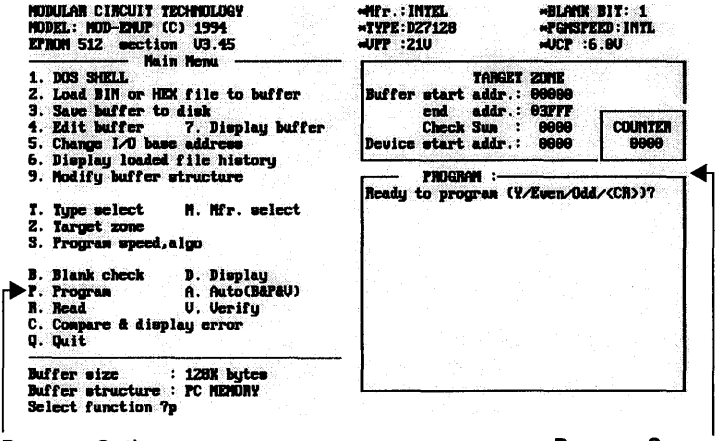


Select the type of read you want preformed:

- "Y" instructs the MOD-EMUP-A programmer to perform a normal read
- "Even" instructs the MOD-EMUP-A programmer to read only the data that is stored in the even bytes of the buffer
- "Odd" instructs the MOD-EMUP-A programmer to read only the data that is stored in the odd bytes of the buffer

Program

To program the contents of the memory buffer into the chip, choose the "Program" option from the Programming Main Menu, and press "Y". The following screen will appear:



Program Option

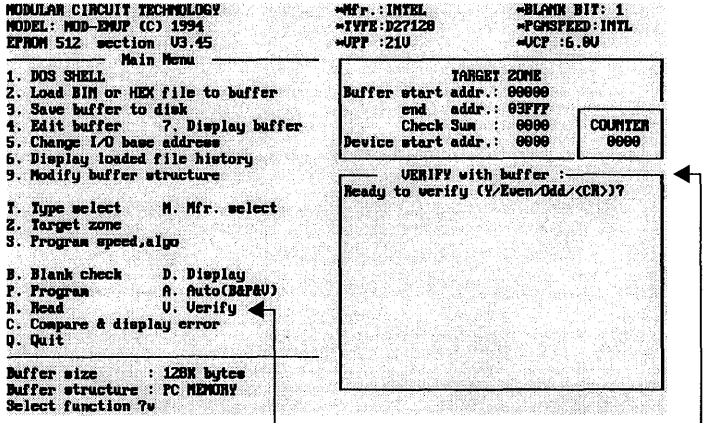
Program Screen

Select a type of programming:

- "Y" instructs the MOD-EMUP-A to program all bytes of the buffer into the chip. This is "normal" program mode
- "Even" instructs the MOD-EMUP-A to program all even bytes of the buffer into the chip
- "Odd" instructs the MOD-EMUP-A to program all odd bytes of the buffer into the chip

Verify

The "Verify" option lets you compare a portion of chip with a portion of the contents in the buffer. This is usually done after a programming procedure to ensure that the chip was programmed correctly. When this option is chosen, a "Verify with Buffer" window will appear on the screen. A screen similar to the following will appear.



Select a type of verify you want to preform:

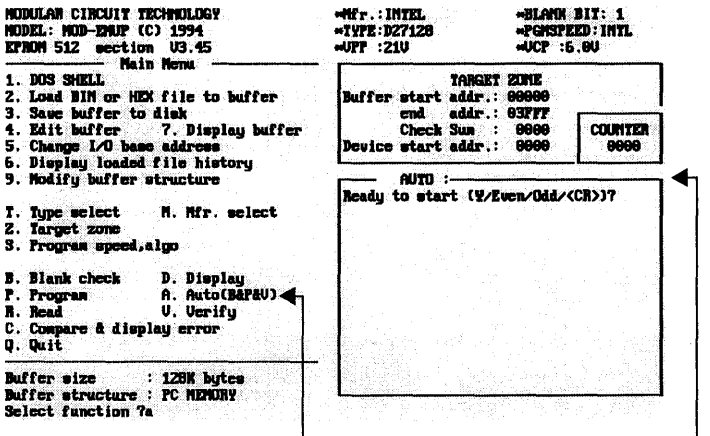
- "Y" compares the whole chip with the entire contents of the buffer
- "Even" compares the whole chip with the even bytes of the chip
- "Odd" compares the whole chip with the even bytes of the chip

When "OK!" appears in the "Verify" window, the chip has verified properly.

Auto (B&P&V)

The "Auto (B&P&V)" option preforms three steps. It first blank checks the chip currently in the ZIF socket, then attempts to program the contents of the buffer into the chip, and finally verifies that the chip has been programmed correctly. When the "Auto

(B&P&V)" screen appears, select "Y" from the following screen to begin the auto function:



Select auto function you want to preform:

- "Y" is a normal program and verify mode. The MOD-EMUP-A programmer programs the entire buffer contents and verifies the contents of the entire chip against the entire contents of the memory buffer
- "Even" programs the even buffer bytes and verifies the entire chip against the even bytes of the buffer
- "Odd" programs the even buffer bytes and verifies the entire chip against the odd bytes of the buffer

When "OK!" appears in the "Verify" window, the chip has verified properly.

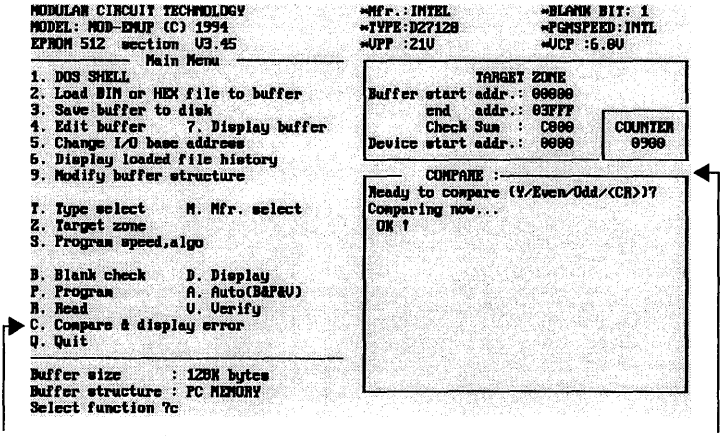
If an error occurs during the auto function, the software will terminate and that error will be displayed in the "Auto (B&P&V)" window. If an error does occur, attempt to execute each function of the auto process separately, using the "Blank Check", "Program", and "Verify" routines listed in the Programming Main Menu.

If the auto operation is successfully completed, then a message "OK" will be displayed in the "Auto (B&P&V)" window.

Compare & Display

Choosing the "Compare & Display" option from the Programming Main Menu will compare the contents of a chip to the contents of

the memory buffer. When this option is selected, the screen will display the chip starting and ending address, the buffer starting address and the buffer ending checksum. If the chip compares correctly with the buffer, the following screen will appear:

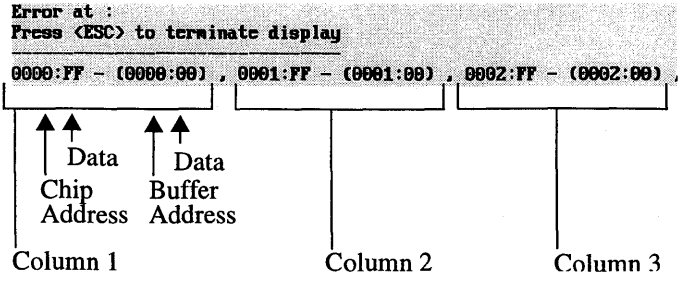


Compare and Display Option **Compare and Display Screen**

Select auto function you want to preform:

- "Y" is a normal program and verify mode. The MOD-EMUP-A programmer verifies the contents of the entire chip against the entire contents of the memory buffer
- "Even" programs the even buffer bytes and verifies the entire chip against the even bytes of the buffer
- "Odd" programs the even buffer bytes and verifies the entire chip against the odd bytes of the buffer

If differences occur between the chip and the memory buffer, those differences will be displayed. The following screen displays a difference between the chip and the memory buffer.



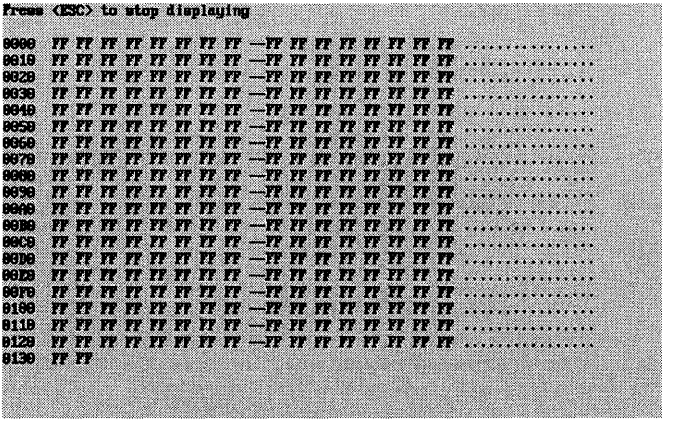
At the chip address 0000, the data FF exists. But at the memory buffer address 0000, the data 00 exists. Similarly, at chip address 0001, the data FF exists. But in the memory buffer at address 0001, the data 00 exists.

If no differences are found, the message "OK!" 23

ill be displayed.

Display

To display the addresses and current contents of the chip in the ZIF socket, choose the option "Display" from the Programming Main Menu. If no chip is in the socket, then all "FF"s will be displayed. The following is an example "Display."

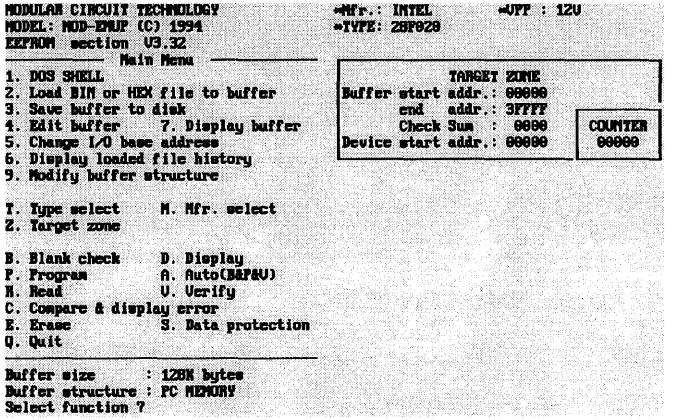


Quit

This option lets you quit the MOD-EMUP-A Programming Main Menu. Press "Q" <Enter>, and then any key to return to the Access Main Menu.

EEPROM Software

Choosing selection "EEPROM" from the Access Main Menu will call a list of manufacturers and chip types. Selecting the desired manufacturer and chip type will produce the following screen:



The EEPROM Programming Main Menu above is very similar to the EPROM Programming Main Menu discussed previously on page 3-3. The three menu item which differ are the additional "Erase" and "Data Protection" functions and the elimination of the "Program Speed, Algo" function.

Erase

EEPROMs are nonvolatile storage devices that retain their memory for up to ten years until erased. Choose the "Erase" function from the Programming Main Menu if the chip you are planning to program can be erased.

Data Protection

Some chips have the capability of write protection. If your chip has this ability, use the "Data Protection" function.

SERIAL PROM Software

Choosing item "Serial PROM" from the "Device" pull-down menu will call a list of manufacturers and chip types for serial PROMs.

Choosing the desired manufacturer and chip type will produce the following screen:

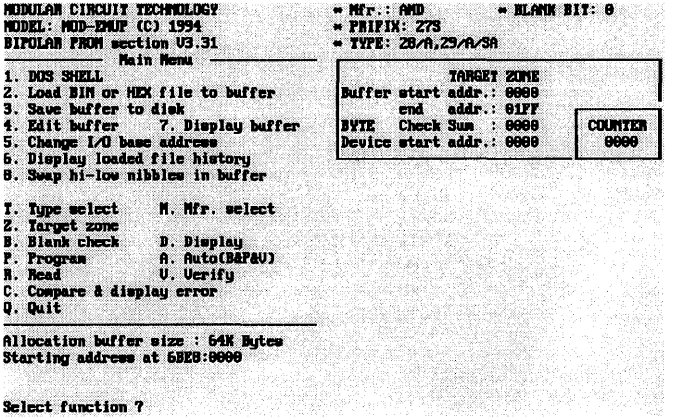
```
MODULAR CIRCUIT TECHNOLOGY              *Mfr.: EXEL          *BLANK BIT: 1
MODEL: MOD-EMUP (C) 1994                *TYPE: XL93C46/P
SERIAL (EE)PROM section V3.37           *UPP :
          Main Menu                     ┌──────────TARGET ZONE──────────┐
1. DOS SHELL                            │Buffer start addr.: 0000       │
2. Load BIN or HEX file to buffer       │      end   addr.: 007F        │
3. Save buffer to disk                  │      Check Sum  : 0000        │ COUNTER
4. Edit buffer      7. Display buffer   │Device start addr.: 0000       │  0000
5. Change I/O base address              └───────────────────────────────┘
6. Display loaded file history

T. Type select      M. Mfr. select
Z. Target zone
B. Blank check      D. Display
P. Program          A. Auto(B&P)
R. Read             V. Verify
C. Compare & display error
Q. Quit
──────────────────────────────────────
Buffer size       : 32K bytes
Starting address at 7660:0000

Select function ?
```

Note: The Programming Main Menu may vary, depending on the chip used.

This menu is very similar to the EPROM Programming Main Menu discussed on page 3-3. The two menu item which differ are the elimination of the "Modify Buffer Structure" and the "Program Speed, Algo" functions.

BPROM Software

Choosing "BPROM" from the Access Main Menu will call a list of manufacturers and chip types. Selecting the desired manufacturer and chip type will produce the following screen:



Note: The Programming Main Menu may vary, depending on the chip used.

The menu items that differ from the EPROM Programming Main Menu page 3-3 are the elimination of the "Modify Buffer Structure" option, and the addition of the "Swap Hi-Low Nibbles in Buffer", and "Program Initial Byte" options.

Swap Hi-Low Nibbles in Buffer

Choosing the "Swap Hi-Low Nibbles in Buffer" option from the Programming Main Menu will allow the exchange of the 4 high order bits to the position of the 4 low order bits and vice versa. This option is useful if you are reading two 4 bit PROMs with the intention of combining them into one 8 bit PROM. Or, separating the contents of one 8 bit PROM and burning in two 4 bit PROM's.

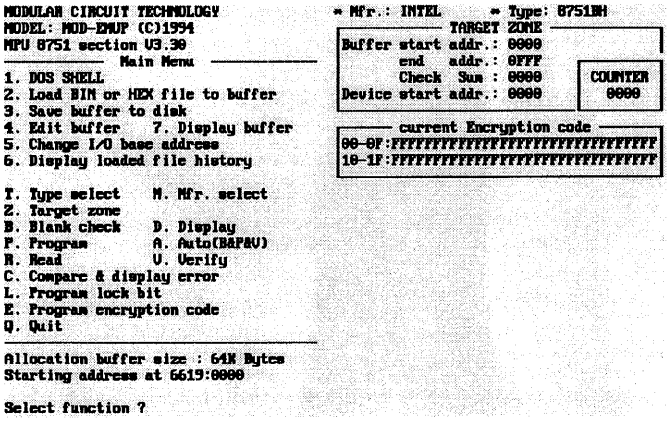
Program Initial Byte

Note: This feature is programmable only on the Cypress CY7C235/245/268/269 chips and will not operate on other chips.

The "Program Initial Byte" option allows the programming of an initialization byte available on Cypress chips CY7C235/245/268/269. The data programmed into the initialization byte is output from the data pin during the initial running of the PROM. The function of this byte is to allow the PROM to output the pre-programmed data during the system initialization. The default value of this byte is "0". To change the initial value, it must be programmed.

MPU Software

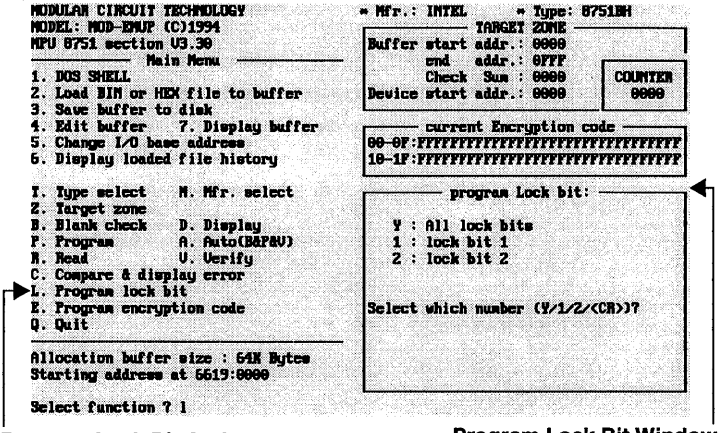
Choosing item "MPU" from the Access Main Menu will call a list of manufacturers and chip types for microprocessors. Selecting the desired manufacturer and chip type will produce the following screen:



Again, this MPU Programming Main Menu is very similar to other programming main menus discussed up to this point. The items that differ from the EPROM Programming Main Menu on page 3-3 are the elimination of the "Modify Buffer Structure" and "Program Speed, Algo" options and the addition of the "Program Lock Bit" and "Program Encryption Code" options.

Program Lock Bit

Choosing "Program Lock Bit" from the Programming Main Menu allows you to select from the following sub-menu:



Program Lock Bit Option

Program Lock Bit Window

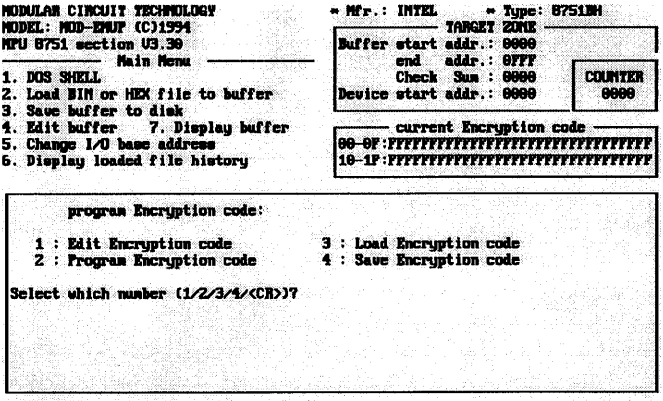
The program lock bit, once programmed, denies electrical access to the program memory. This lock bit keeps the memory from being read out, further programmed, or from executing external program memory. Erasing the EPROM array deactivates the lock bit and will restore the device's full functionality.

Program Encryption Code

Some MPU devices implement a 32-byte EPROM array that can be programmed and used to encrypt (encode) the program code bytes during EPROM verification. Choosing the "Program Encryption Code" option from the Programming Main Menu will display this encryption array.

Unprogrammed bytes have the value FF(H). Therefore, an unprogrammed encryption array will have 32 bytes all containing the value FF (H).

There are four choices within the "Program Encryption Code" window. Chips that allow encryption code editing will display a screen similar to the following when this option is chosen:



Note that the current encryption code window shows that no encryption programming has been done. All addresses contain "F".

Edit Encryption Code

Choosing "Edit Encryption Code" will display the current code and allow its modification. Once modified, pressing <Enter> will return you back to the "Program Encryption Code" menu and the modified code will display in the current encryption code window.

Program Encryption code

Choosing the "Program Encryption Code" option from the Programming Main Menu will program the currently loaded code into the chip.

Load Encryption code

To load the encryption code from a file previously saved, choose "Load Encryption Code" from the Programming Main Menu.

Save Encryption code

Choosing "Save Encryption Code" will save the code to a file for future loading or safe keeping.

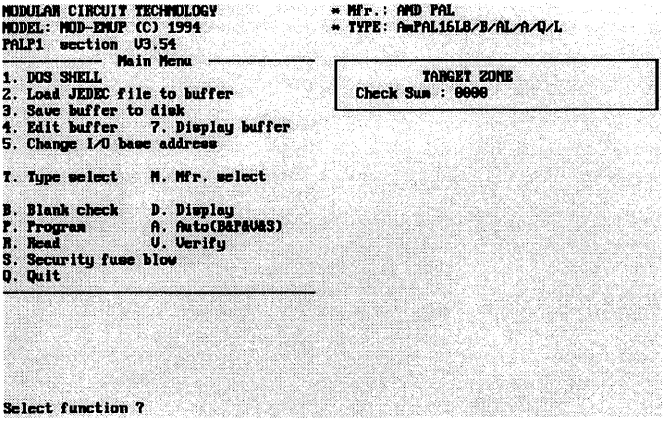
Note: It is recommended to program a Lock Bit when using an Encryption Array.

PLD Software

Generally, a device that can be programmed to perform logic operations is labeled a PLD.

PLD Option

Choosing the "PLD" option from the Access Main Menu will call a list of manufacturers and chip types. Selecting the desired manufacturer and chip type will produce the following screen:

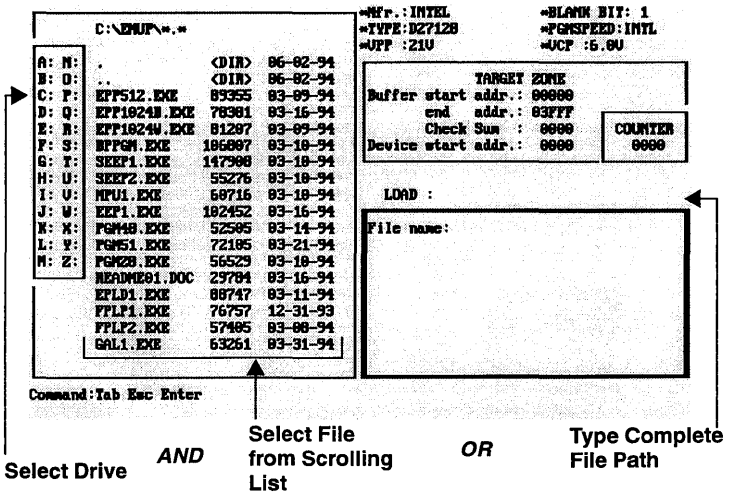


The PLD Programming Main Menu is very similar to those discussed under the EPROM Programming Main Menu option on page 3-3. The menu items that differ are the elimination of the "Load BIN or HEX File to Buffer", "Display Loaded File History", "Modify Buffer Structure", "Target Zone", and "Compare & Display" options, and the addition of the "Load JEDEC File to Buffer" and "Security Fuse Blow" option.

Load JEDEC File to Buffer

The JEDEC (Joint Electron Device Engineering Council) fuse map format is the standard file format used for programming PLDs. It contains information on the fuses (i.e. blown/intact) and function test vectors. Most PAL assemblers or compilers will produce JEDEC fuse map files.

Choosing the "Load JEDEC File to Buffer" option from the Programming Main Menu will produce the following screen:



There are two ways to enter the file to be loaded into the buffer.

1. When the initial screen appears, a flashing cursor appears following the prompt:

   `File name:`

   Type in the complete file name, including the drive letter and path and press <Enter>.

   or

2. Press the <Tab> key to activate the scrolling file list. Using the <Arrow>, <Page up>, <Page down>, and <Enter> keys, you are able to move freely within your disk's directory to locate the file to be loaded. Once found, highlight the file name and press <Enter>.

   To view files on an alternate drive, press the <Tab> key to activate the scrolling file list and then choose the drive letter where the file is loaded.

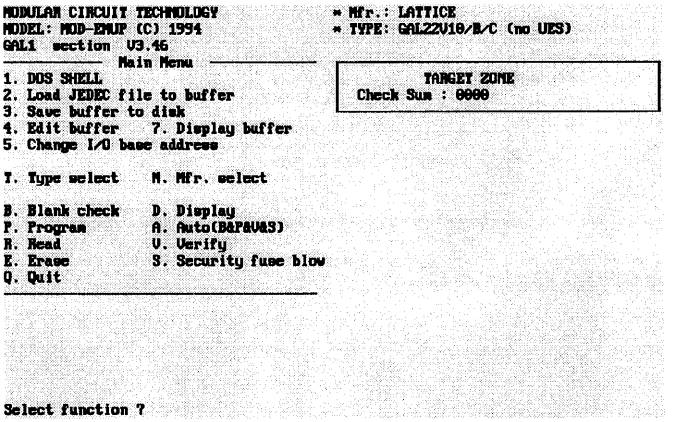
Security Fuse Blow

The "Security Fuse Blow" option allows the security fuse on the selected PLD to be blown. By blowing the security fuse, all further read/write access to the PLD is prevented. Unauthorized copies of the PLD cannot be made.

Other Options for PLD Software

GALs

Choosing a type and manufacturer of a GAL will result in the following screen:



Note: Notice the GAL screen is the same as the PAL screen on page 3-28, except that it offers an "Erase" option. Generally, GALs are erasable and re-programmable, but PAL's are not.

Caution: The security fuse cannot prevent the GAL from being reprogrammed, even if the security fuse has been blown. Erasing the GAL will also erase the security fuse and allow the GAL to be reprogrammed.

PEELs

Choosing a PEEL manufacturer and type will produce a similar screen to those from preceding PLD programming options. The only difference here is the "Auto (B&P&V&S)" selection. Use this option only if you wish to blow the security fuse during the programming process.

Note: You will notice that some of the PLD types for some manufacturers have an asterisk (*) by their respective part number. These part numbers require an adaptor. Refer to "Technical Reference" on page-7-1 for more information on adaptors.

EPLDs

The screen and programming options for EPLD's (Erasable Programmable Logic Device) appear the same as those for the PALs. The operation and description of each menu item is also the same. However, unlike PALs, the EPLD's are UV erasable and can be reprogrammed.

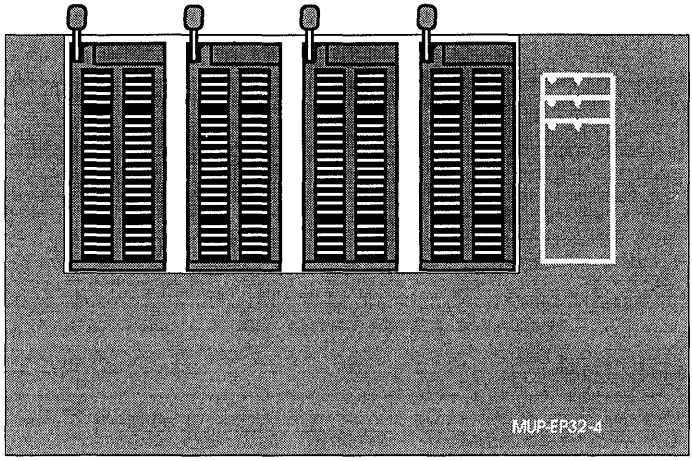
User Notes

4

Gang-Adaptor

Multiple Programmer

The MOD-EMUP-A can be used for programming with gang-adaptors. These devices allow you to program multiple ICs during a single programming session and can usually hold four ICs at a time.



Note: The gang-adaptor software is an optional item that is sold separately from the MOD-EMUP-A programmer. Some gang-adaptors many include programming software. Check your gang-adaptor package for specific details on package contents.

Selecting "Gang-Adaptor" from the Access Main Menu will call a sub-menu with the choices of: "Default", "E/EPROM", "MPU/ MCU", and "PLD". These devices are programmed by selecting a manufacturer and device type.

Using the Gang-Adaptor

Note: The MOD-EMUP-A Access programming software will prompt you to enter the "path" where the gang-adaptor software is located. To avoid having to enter the path each time you use your gang-adaptor software, copy the gang-adaptor *.EXE into the C:\emup directory.

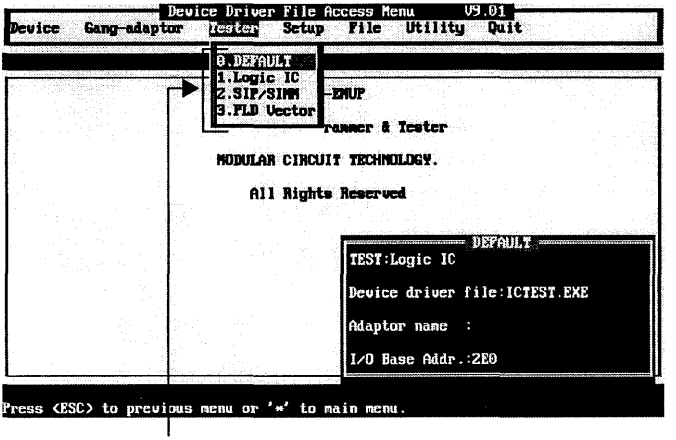
Once you have selected the manufacturer and device type, the Gang-adaptor Programming Main Menu will appear. The series of screens for each device are similar, varying only in information that is specific to the IC selected for programming. Refer to the "EPROM" Programming Main Menu option discussed on page 3-3 for more information on the sub-menu choices available.

5

Tester

Tester

The MOD-EMUP-A software provides a four testing options: "Default", "Logic IC", "SIP/SIMM", and "PLD VECTOR". All testing features are found in the "Tester" pull-down menu.

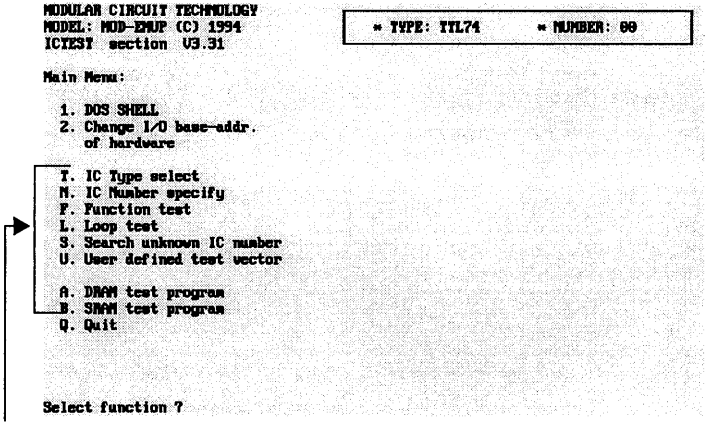


Tester Pull-down Menu

The series of screens for each option are similar, varying only by providing specific information for the test that you choose to run.

Logic IC Software

Choosing the option "Logic IC" will produce the Tester Main Menu:



Unique Tester Main Menu Options

Once you have selected the test type, the Tester Main Menu will appear. The series of screens for each option are similar, varying only by providing information specific to the device that you select for testing. The first two options, "DOS Shell" and "Change I/O Base Address of Hardware" are discussed in "DOS Shell" on page 3-3 and "Change I/O Base Address of Adaptor Card" on page 3-9.

Note: Logic IC and RAM only perform a functionality test. They cannot test the timing or speed of the chips.

IC Type Select

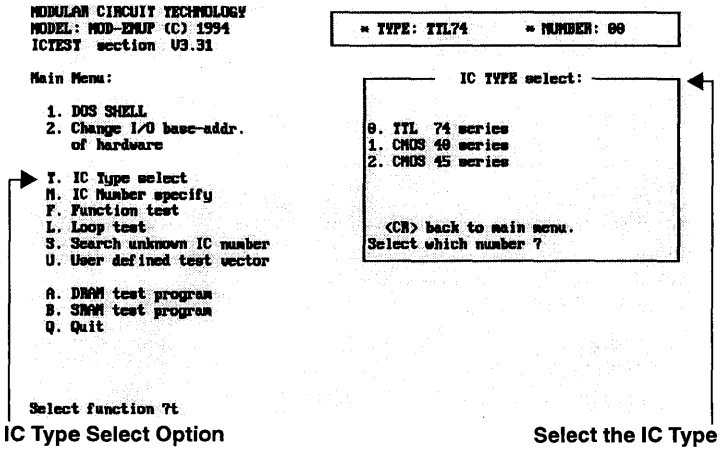
This option allows you to choose a chip from a list of three IC series types. The Logic IC included with the MOD-EMUP-A will test the following chip types:

- 74/54 TTL, HC, HCT or equivalent CMOS
- 40/140 CMOS series
- 45/145 CMOS series
- SRAM 6116, 6264, 62256 series
- DRAM 4164, 41256, 411000, 4416, 4464, 44256

It will also test the following SIP and SIMM modules with the applicable adaptor:

- 41256 x 8, 41256 x 9, 411000 x 8, 411000 x 9

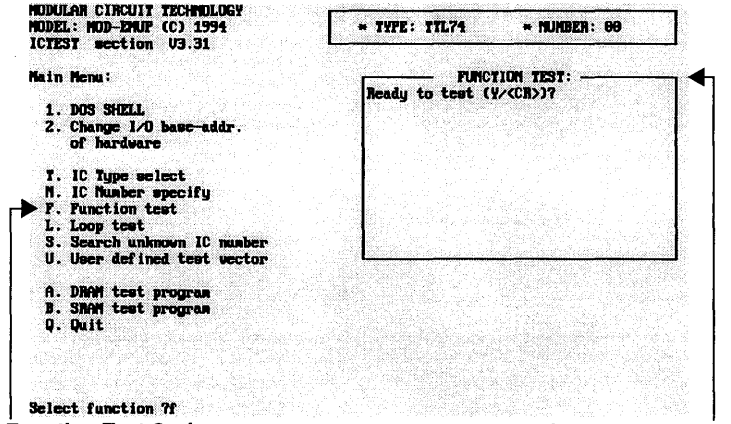
After placing a chip in the ZIF socket, choose "IC Type Select" from the Tester Main Menu, and the following sub-menu will appear:



Choose the IC series type that you wish to test, and press <Enter>. This will return you to the Tester Main Menu selections. The next step is to specify a chip number using either "IC Number Specify" on page 5-4, or "Search Unknown IC Number" on page 5-5.

Function Test

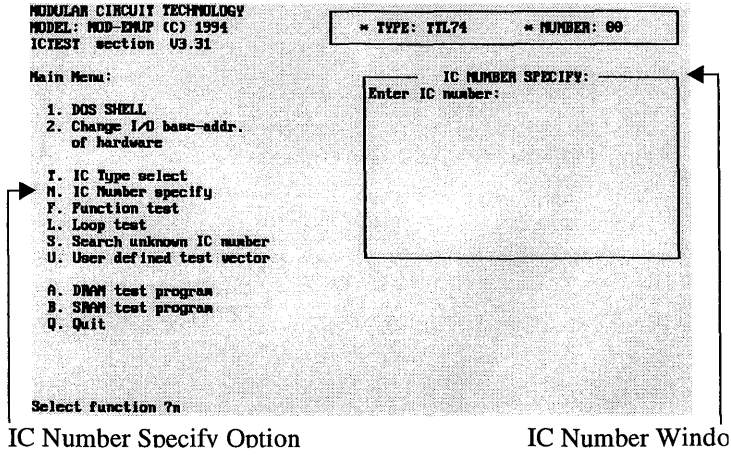
Choosing this option will call the following "Function Test" window:



At the prompt, press "Y" and a single loop function test will be performed. This will test all the possible logic functions once.

IC Number Specify

The first method used to specify a chip number is the "IC Number Specify" option found in the Tester Main Menu.

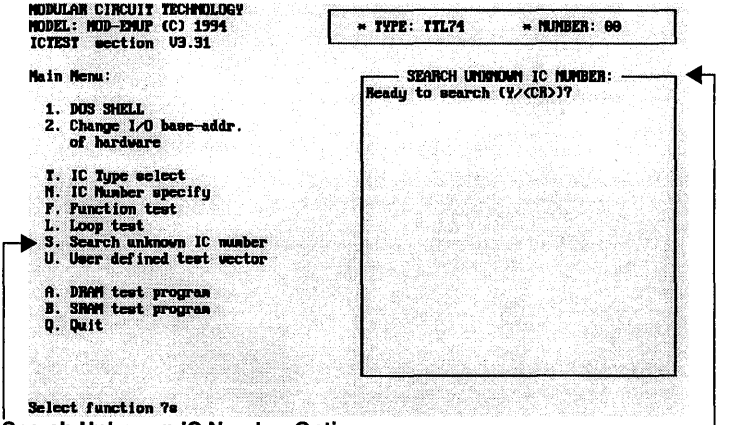


Input the chip number at the prompt without using the series identification numbers. For example, for a 74LS245 chip, you would enter "245". The software will confirm the chip number and respond with "OK!". The total number of vectors will also be displayed.

Note: If the chip you wish to test is not in the software's library of available chips, then you will need to write or load your own test vectors. This will be discussed in "Load Test Vector into User Buffer" on page 5-8.

Search Unknown IC Number

The second way to specify a chip number is by choosing "Search Unknown IC Number" from the Tester Main Menu.



Search Unknown IC Number Option

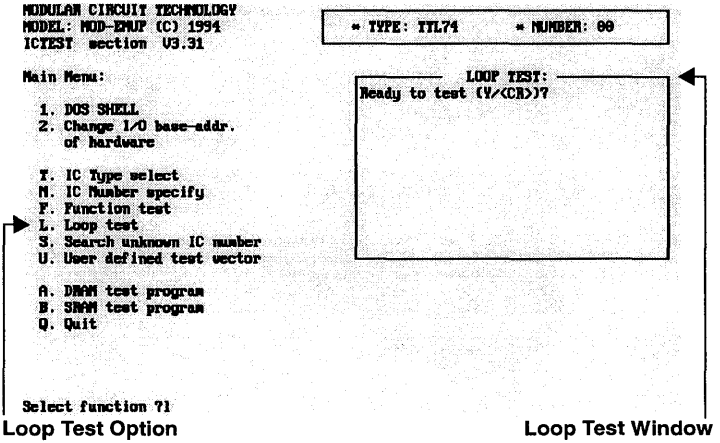
Search Unknown IC Number Window

The "Search Unknown IC Number" window will appear prompting you to press "Y" to begin the search. After pressing "Y", a "Searching now..." message will be displayed followed by the chip number the software has determined to be correct.

Note: The software will only find the proper IC number if the IC is good.

Loop Test

Choosing this option from the Tester Main Menu will call the "Loop Test" window. This function is useful for testing intermittent ICs.



At the prompt, choose "Y" and an endless loop test of the chip's logic combinations will be performed. In the event an error is incurred, testing will stop and an error message will be displayed.

To end the testing loop, press any key.

User Defined Test Vector

The User Defined Test Vector Menu provides several options for dealing with test vectors. Choosing this option from the Tester Main Menu will produce the following screen:

```
User defined test vector menu :
=======================================
  1. DOS SHELL
  2. Load test vector into user buffer
  3. Save test vector from user buffer
  4. Edit test vector in user buffer

  F. Function test
  L. Loop test
  D. Display result during testing

  Q. Back to last menu.
```

A test vector is the definition of input and output logic states applicable to the various pins of an IC. There can be up to 1000 vectors specified for a single IC thus allowing a sequence of logic events to be tested. There are 24 pins on the ZIF socket which can be defined. These are shown in Figure 5-1.

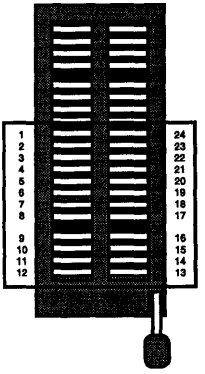


Figure 5-1. Definable Pins on the ZIF Socket

The first step in defining your own test vectors is to obtain the specifications and pin-outs of the chip you intend to test. This information will be necessary to determine the proper states of each pin both as input and output.

DOS Shell

Choosing the "DOS Shell" option from the Programming Main Menu will cause the software to search for COMMAND.COM on the boot disk drive. The following screen will appear:

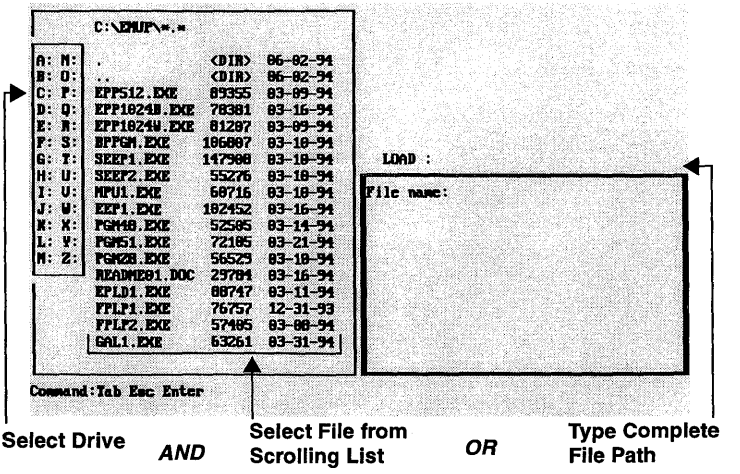
```
Type EXIT to return to Programmer menu

Microsoft(R) MS-DOS(R) Version 6.20
            (C)Copyright Microsoft Corp 1981-1993.

C:\EMUP>>
```

The software will execute the COMMAND.COM command and pass control back to DOS, allowing you to perform DOS operations while running the MOD-EMUP-A Access Software in the background. To return to the User Defined Test Vector Menu, type:

exit <Enter>

Load Test Vector into User Buffer

Choosing the "Load Test Vector into User Buffer" option from the Tester Main Menu will produce this screen:



There are two ways to load a file into the buffer.

1. When the initial test vector screen appears, a flashing cursor appears following the File name: prompt. Type in the complete file name, including the drive letter and path and press <Enter>.

   or

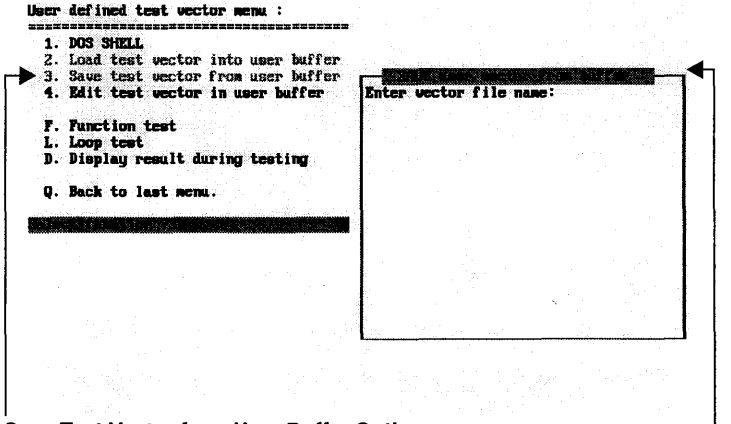
2. Press the <Tab> key to activate the scrolling file list. Using the <Arrow>, <Page up>, <Page down>, and the <Enter> keys, you are able to move freely within your disk's directory to locate the file to be loaded. Once found, highlight the file name and press <Enter>.

   To view files on an alternate drive, press the drive letter <Enter>.

   The file name that you have selected will appear in the "Load" window.

Save Test Vector from User Buffer

Use the "Save Test Vector from User Buffer" option to save any edited or newly created vector information. Place all necessary filename information here.

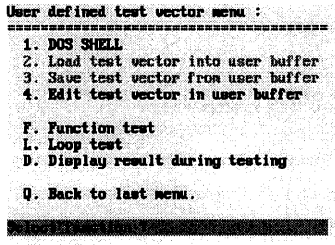


Save Test Vector from User Buffer Option

Save Test Vector from User Buffer Window

Edit Test Vector in User Buffer

Choosing the "Edit Test Vector in User Buffer" option from the User Defined Test Vector Menu will produce the following screen:



The "Edit Test Vector in User Buffer" screen displays the information necessary to write test vectors using the proper syntax. The "Vector Symbol Definition(s):" provides the proper abbreviations to use in your vector lines. The vector lines are labeled V0001- V1000 allowing one-thousand lines of vector test definition.

Example Test Vector

The chip used in this example test vector definition is a National Semiconductor DM74LS393N. This is a 14 pin chip. The DM74LS393N is placed in the ZIF socket with pin 1 of the chip

corresponding to Pin 6 of the 24 pin testing area on the ZIF socket. See Figure 5-2.

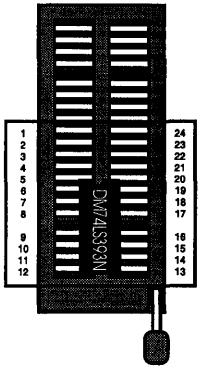


Figure 5-2. 14-pin chip in the ZIF Socket

The following screen shows a six line test vector definition for the DM74LS393. Examine the information carefully. Notice how the symbols listed in the "Vector Symbol Definition:" correspond to the actual symbols used in V0001-V0006.

```
Edit test vector in buffer                          1111111111222222
                                                    1234567890123456789012345
Vector symbol definition:
  0 : apply LOW to IC.                        V0001 XXXXXX1LLLLGLLLL10EXXXXX
  1 : apply HI  to IC                         V0002 XXXXX90LLLLGLLLL01EXXXXX
  L : expected output result from IC is LOW   V0003 XXXXX10LLLLGLLLL01EXXXXX
  H : expected output result from IC is HI    V0004 XXXXX90HLLLGLLLL01EXXXXX
  M : same symbol as last vector              V0005 XXXXX10HLLLGLLLL01EXXXXX
  X : same symbol as last vector, but don't   V0006 XXXXX90HLLLGLLLL01EXXXXX
      care output result from IC              V0007
  E : apply 5 V to IC, only TP 19,20,22,24    V0008
      can be specified                        V0009
  G : apply GND to IC, only TP 12             V0010
      can be specified                        V0011
Cursor movement key assignment:               V0012
  UP    ARROW : cursor up                     V0013
  DOWN  ARROW : cursor down                   V0014
  LEFT  ARROW : cursor left                   V0015
  RIGHT ARROW : cursor right                  V0016
  PgUp        : last page                     V0017
  PgDn        : next page                     V0018
  Ins         : insert 1 line                 V0019
  Del         : delete 1 line                 V0020
  <CR>        : next line 1st position
  <ESC>       : end editing
```

Vector Symbol Definitions

Test Vector Definition

Vector Lines

V0001

Since Pin 1 of the DM74LS393 corresponds to Pin 6 of the ZIF socket, pins 1 through 5 in the test vector are symbolized with an "X". This indicates a "don't care output result".

Pin 1 of the chip is an active low clock input. At location 6 of the test vector, an "X" will be used since our first vector is going to clear the chip. A clock cycle will not be necessary.

Pin 2 of the chip (Pin 7 of the ZIF) is an active high CLEAR input, so a "1" is placed at this location to clear pins 3-6 of the chip. So, on pins 8-11 of the test vector (and ZIF socket) a LOW would be expected. This is why a "L" symbol is placed in these locations.

Pin 7 of the chip is GROUND. At location 12 of the test vector, a "G" symbol is placed to signify applying ground to the chip.

Pin 12 of the chip (Pin 17 of the ZIF) is another active high CLEAR input which affects the output of pins 8-11 of the chip (Pins 13-16 of the ZIF and vector). Applying a "1" to Pin 12 of the chip will produce LOW outputs from pins 8-11 of the ZIF. Therefore, "Ls" are placed in positions 8-11 of the test vector.

Pin 13 of the chip is another active low clock input. So, as with Pin 1 discussed earlier, an "X" is applied at this location.

Pin 14 of the chip is Vcc. So, at location 19 of the test vector, the symbol "E" is used to apply 5 volts to the IC.

V0002

Pin 1 of the chip (location 6 of the test vector) has a LOW applied to it to prepare the chip for a clock cycle in the subsequent vector lines.

Pin 2 of the chip (location 7 of the test vector) has a LOW applied to it remove the chip from its previous CLEAR state.

Again, pins 3-6 will have an expected LOW result since no clock cycle has yet occurred to cause the chip to count. "Ls" at test vector locations 8-11 reflect this result.

Pin 7 is always GROUND (location 12 of the vector).

Pins 8-11of the chip (locations 13-16 of the vector) will expect a LOW as before. This is due to pin 12 (location 17 of the vector) being HIGH.

Pin 14 of the chip is always Vcc. An "E" at location 19 of the vector.

V0003

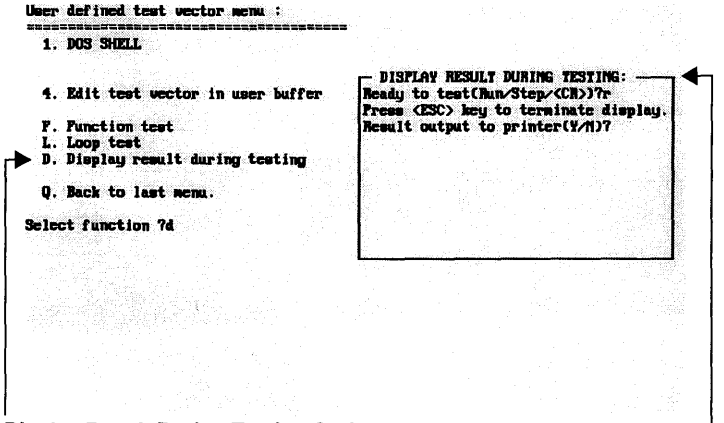
Pin 1 (location 6 of the vector) now goes high.

This is to prepare the chip for a high to low transition at this vector location. This transition is the clock cycle that will cause the chip to operate correctly.

Examine the remaining vector lines and follow vector locations 6-8 closely. You will notice that the chip begins to count as expected in lines V0004-V0006.

Display Result During Testing

After the test vector has been written it can be run to test its validity. Choosing the "Display Results During Testing" option will result in the following screen:



Display Result During Testing Option

Display Result During Testing Window

The "Display Result During Testing" menu provides three choices: "Run", "Step", and "<CR>" (<Enter>). "Run" will execute the test line by line without hesitation. "Step" will allow you to step through the test a line at a time. Pressing <Enter> after each test line advances to the next line.

Pressing <Enter> will return you to the vector menu.

If you choose "Run" or "Step" you will be prompted whether or not you wish to print the result.

Once the test has been run, the vector lines and their results will be displayed, followed by the successful completion message:

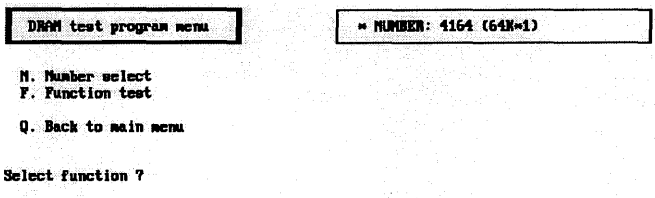
```
Function test complete
Press any key to continue
```

If an error does occur, the following message will display:

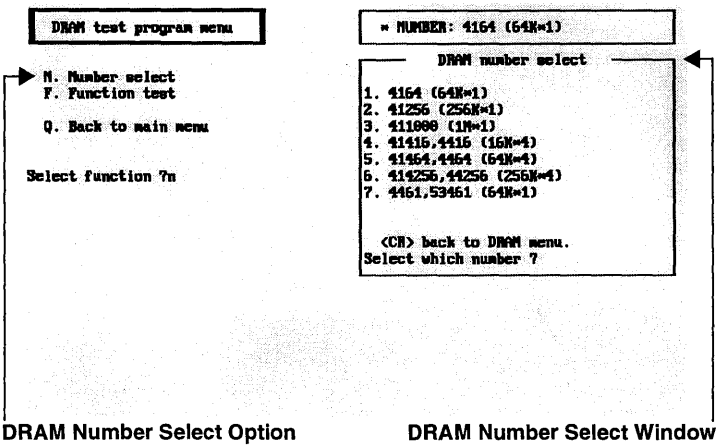
```
Function error
```

DRAM Test Program

The Tester Main Menu provides an option for testing DRAM. The DRAM Test Program Menu allows you to choose the IC number and then to run a function test from the following menu:

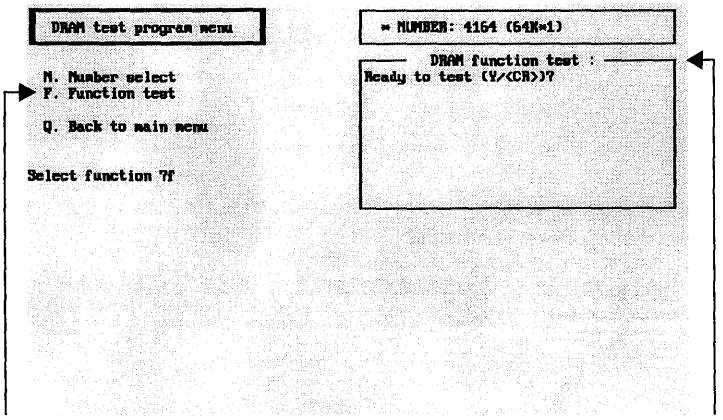


To run the DRAM Test Program, choose the option "Number Select":



DRAM Number Select Option DRAM Number Select Window

Select the desired DRAM number and press <Enter>. Once you have entered a DRAM number, select "Function Test".

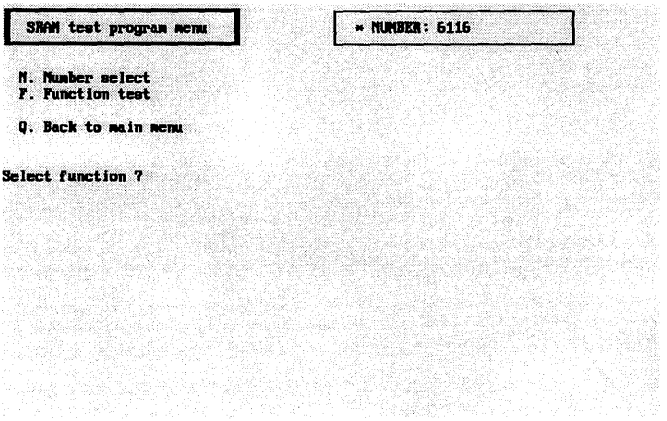


DRAM Function Test Option DRAM Function Test Wind

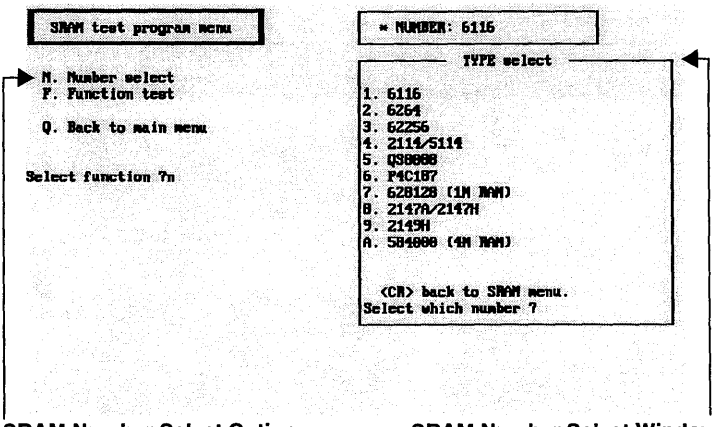
To run the DRAM function test press "Y". To skip the test, press <Enter>.

SRAM Test Program

The Tester Main Menu provides an option for testing SRAM. The SRAM Test Program Menu allows you to choose the IC number and then run a function test from the following menu:



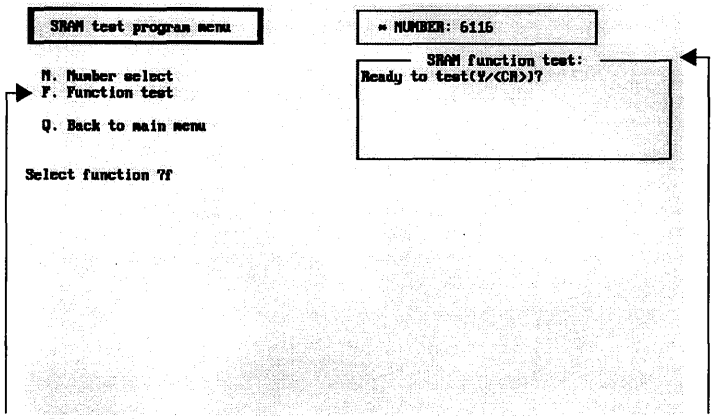
To run the SRAM Test Program, choose the option "Number Select":



SRAM Number Select Option **SRAM Number Select Window**

Select the desired SRAM number and press <Enter>.

Once you have entered a SRAM number, select "Function Test".

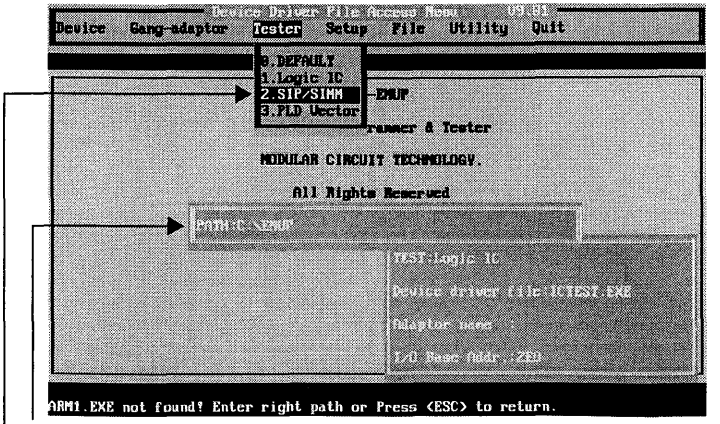


SRAM Function Test Option **SRAM Function Test Window**

To run the SRAM function test press "Y". To skip the test, press <Enter>.

SIP/SIMM

The Tester Main Menu provides an option for testing SIPs and SIMMs. To run the SIP/SIMM test program, choose "SIP/SIMM" from the Tester pull-down menu. The following screen will appear:



Path to SIP/SIMM Software

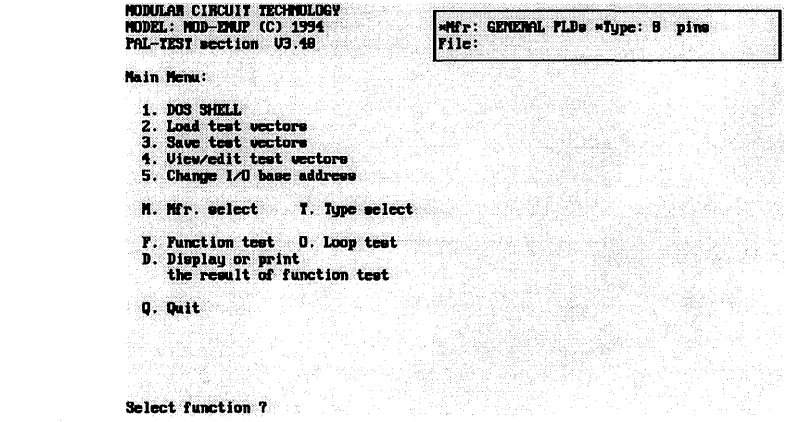
SIP/SIMM Option

Enter the path to load the SIP/SIMM test software and press <Enter>. To run the software, refer to "Tester" on page 5-1 or the SIP/SIMM manufacturer's software manual.

Note: The SIP/SIMM adaptor is an optional item that is sold separately from the MOD-EMUP-A programmer. Some programming adaptors may include SIP/SIMM software. Check your software package for specific details.

PLD Vector Test

Choosing this item from the Tester pull-down menu will result with the following screen:

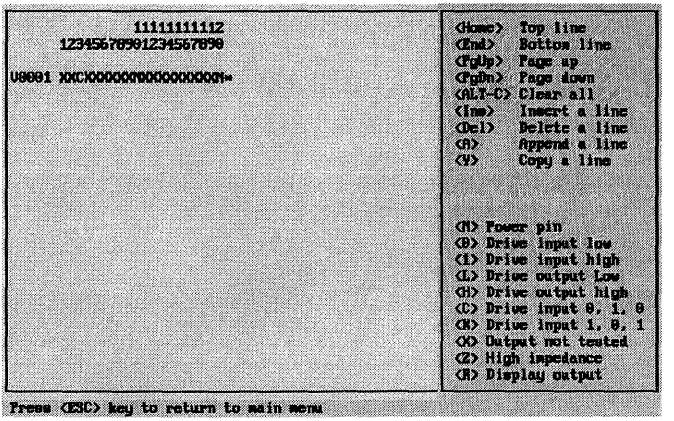


The PLD Vector Test Menu is very similar to those discussed under "Logic IC Software" on page 5-2. The menu items that differ are the elimination of "Search Unknown IC Number", "User Defined Test Vector", "DRAM Test Program", and "SRAM Test Program", the addition of "Display" or "Print the Result of Function Test" option, and a change in "View/Edit Test Features".

Defining your own test vectors for PLDs requires utilizing a text editor. The test which you create must be saved in ASCII format. Refer to page 5-7 for more information on test vectors.

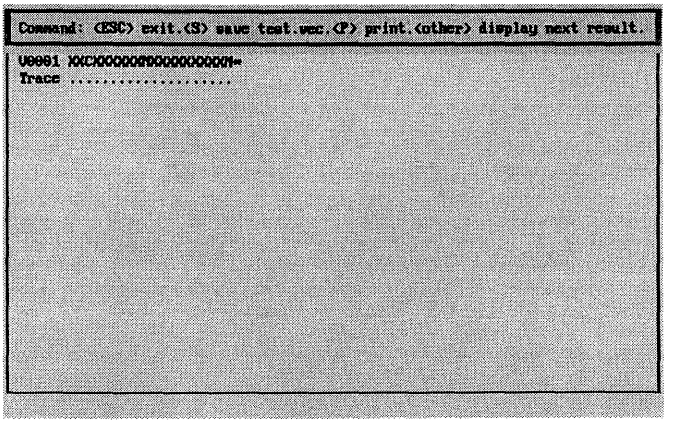
View/Edit Test Vectors

Choosing this option will display the vector which you have previously loaded. A screen similar to the following will appear:



Display or Print the Result of the Function Test

Choosing this option will produce the following screen:



Press any key to execute the vector test. The results of the test will display on the screen. If you wish to print the results to a printer, press <P>.

User Defined PLD Vectors

Using the "PLD Test Vector" feature of the MOD-EMUP-A Access Software requires using a text editor such as DOS Edit to create an ASCII file. Unlike the "IC Test" feature, the facility to create specific test vectors for PLDs using only the MOD-EMUP-A Access Software is not available.

Example PLD Source Code and Decoding

The following example shows you how a text file is used to write a test vector. The first section of the example shows the sample source code that has been programmed into a PLD. The second half of the example walks you through a test for decoding the first block "F" at Address 300H. The PLD used in this example is a Texas Instruments 16L8-25CN.

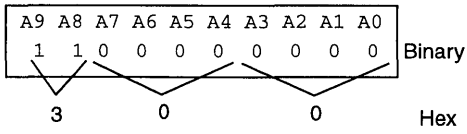
Sample PLD Source Code

```
INPUTS
Pin  1=SA0    ;Address  A0
Pin  2=SA1    ;Address  A1
Pin  3=SA2    ;Address  A2
Pin  4=SA3    ;Address  A3
Pin  5=SA4    ;Address  A4
Pin  6=SA5    ;Address  A5
Pin  7=SA6    ;Address  A6
Pin  8=SA7    ;Address  A7
Pin  9=SA8    ;Address  A8
Pin  11=SA9   ;Address  A9
Pin  17=!HI   ;HI from U7
Pin  18=!Block_DCD     ;Card is selected + Also A Out
OUTPUTS
Pin  12=!G    ;Output   G
Pin  13=!F    ;Output   F
Pin  14=!E    ;Output   E
Pin  15=!D    ;Output   D
Pin  16=!C    ;Output   C
Pin  19=!PortEqual;Comparator Output
Field Address=[SA9..1]     ;Notice "0" not included
LO=!SA0
F=Address:(300..303) & Block_DCD    ;1st    Block
E=Address:(304..307) & Block_DCD    ;2nd    Block
D=Address:(308..30B) & Block_DCD    ;3rd    Block
C=Address:(30C..30F) & Block_DCD    ;4th    Block
PortEqual=Address:(300..30F)
```

Decoding the First Block of the PLD Sample Source Code

This example uses the "Sample PLD Source Code" on page 5-20 to test for decoding of the first block "F" at Address 300H.

First, determine the state of the address lines to reflect the address of 300H (see the following illustration).



Once the decoding of the address has been set, you can determine the output states of the PAL.

According to the source code, the program decoding for address 300H (Block F) will drive Pin 13 LOW, while the remaining Pins 12, 14, 15, and 16 (Blocks G, E, D, and C) will be driven HIGH.

Input Pin 17 is not used in the chip's current configuration and will be specified as "X" (don't care) in the test vector.

Input Pin 18 indicates that the card is being selected and will be input in the LOW state.

Output Pin 19 indicates that an address within the addressable range of this PAL is being decoded. The test here meets that requirement and therefore will be driven LOW.

The information below shows the test vector you created using DOS "Edit".

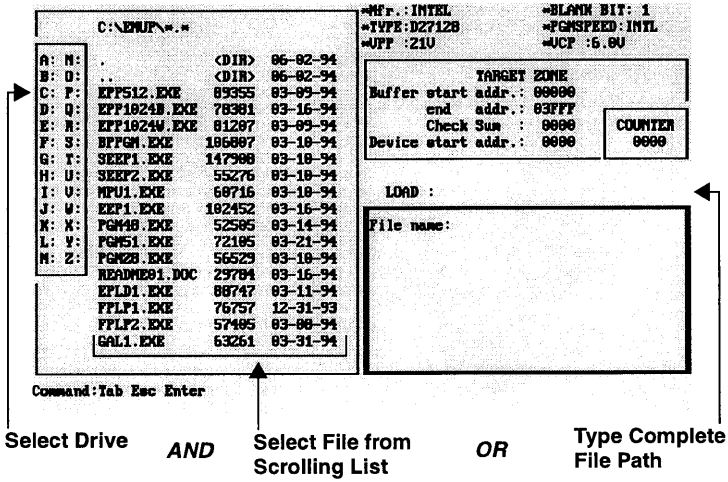
```
                    11111111112222222223333333334
          12345678901234567890123456789012345678901
V0001
 XXXXXXXXXX000000001N1HLHHHXXLNXXXXXXXXXX
```

Notice that the first ten pins and the last ten pin locations are labeled "X". This is because the 40 pin ZIF socket on the MOD-EMUP-A only allows the testing of 20 or 24 pin PLDs. Therefore, the first and last ten pins (in our 16L8, 20 pin PLD example) are not used by the programmer.

It is important that you use the correct spacing when using the text editor. You should leave two spaces between the vector line number and the beginning of the data and no spaces between the data bits.

Notice that pin 20 of this example is labeled "N". On the "IC Test" example a "G" was used for GROUND. When programming PLDs, use the "N" at Pin 20 or Pin 22, depending on the device.

To execute the test vector from the MOD-EMUP-A software. Choose "Load test vector of JEDEC file" option. This will produce the following screen:



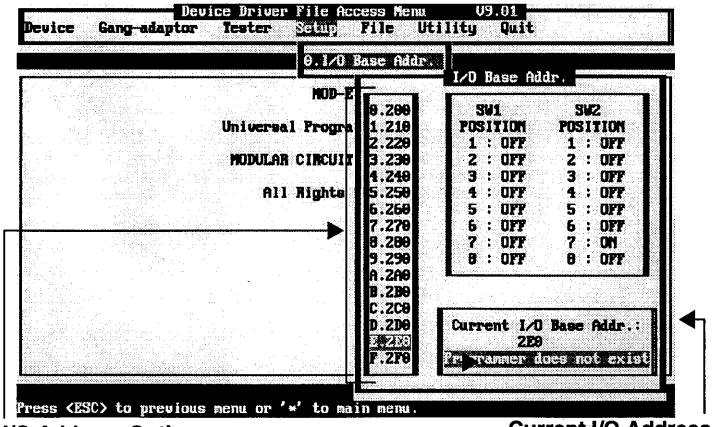
As before when loading a file, be sure to include the drive letter and complete path of the file you wish to load. Once the file is loaded, the screen will revert back to the original PLD menu with the file that has been loaded appearing in the "Work File" window.

6

Setup, File, and Utility

Setup

The "Setup" option allows you to change the I/O base address of the adaptor card. The following screen will appear when you choose "Setup" from the Access Main Menu:



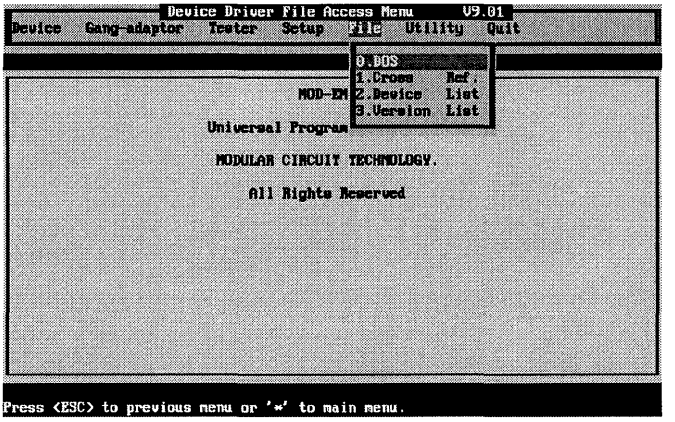
I/O Address Options **Current I/O Address**

The lower right-hand window of this screen displays the current I/O address. To select a new address, type in the appropriate number or letter at the "Select number?" prompt. It is important that the jumper settings on the adaptor card are set properly for the address you wish to use. If the adaptor card is not set properly, a message "Programmer does not exist" will display. If this occurs, check your adaptor card jumper settings. Refer to "Setting the Jumper for the MOD-EMUP-A I/O Address" on page 2-1 of this manual for more information on jumper settings.

Note: The I/O address can also be changed from any device Programming Main Menu by selecting the "Change I/O Base Address of Adaptor Card" option.

File

The "File" menu allows you to access files and information by providing access to DOS and information lists for the MOD-EMUP-A programming software. The pull-down "File" menu appears as follows:



DOS

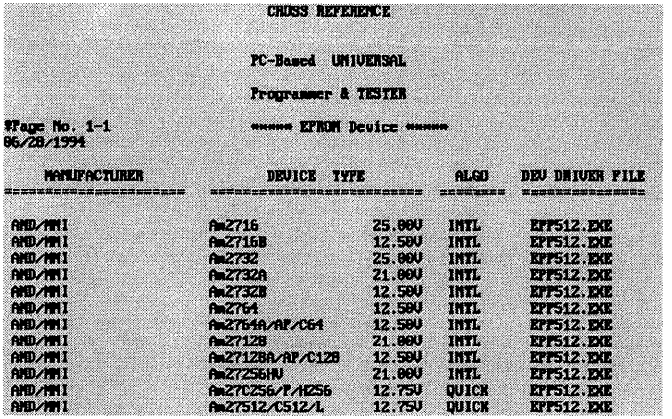
The DOS option takes you to the DOS prompt, while running the MOD-EMUP-A Access software in the background. You may run DOS commands from this prompt, such as loading MOD-EMUP-A device programming files. When you are finished working in DOS, type:

exit <Enter>

Cross Reference

The "Cross Reference" option provides information for each device type, listing manufacturer, device type, algorithm, and device driver type. You have the option to print the information to screen, printer, or a file. If you choose to print the list to screen, be prepared to halt the scroll by depressing the <Pause> key. To continue the scroll

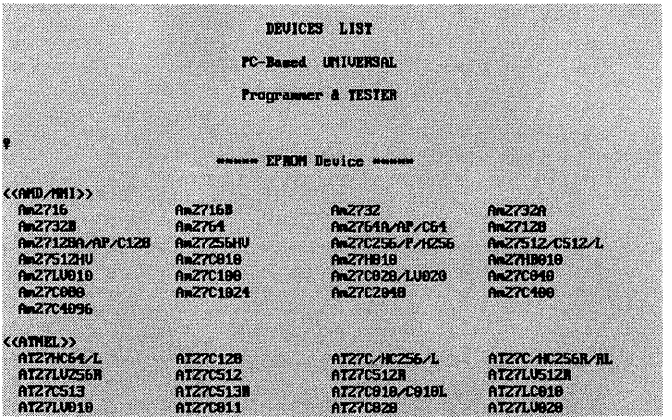
after pressing <Pause>, press <Esc>. The beginning portion of the screen will appear as follows:



```
                        CROSS REFERENCE

                      PC-Based  UNIVERSAL

                      Programmer & TESTER

Page No. 1-1            ***** EPROM Device *****
06/28/1994

     MANUFACTURER          DEVICE  TYPE          ALGO    DBU DRIVER FILE
=====================  ========================  ======  ===============

AMD/MMI                Am2716                25.00V  INTL    EPT512.EXE
AMD/MMI                Am2716B               12.50V  INTL    EPT512.EXE
AMD/MMI                Am2732                25.00V  INTL    EPT512.EXE
AMD/MMI                Am2732A               21.00V  INTL    EPT512.EXE
AMD/MMI                Am2732B               12.50V  INTL    EPT512.EXE
AMD/MMI                Am2764                12.50V  INTL    EPT512.EXE
AMD/MMI                Am2764A/AP/C64        12.50V  INTL    EPT512.EXE
AMD/MMI                Am27128               21.00V  INTL    EPT512.EXE
AMD/MMI                Am27128A/AP/C128      12.50V  INTL    EPT512.EXE
AMD/MMI                Am27256HV             21.00V  INTL    EPT512.EXE
AMD/MMI                Am27C256/P/M256       12.75V  QUICK   EPT512.EXE
AMD/MMI                Am27512/C512/L        12.75V  QUICK   EPT512.EXE
```

Device List

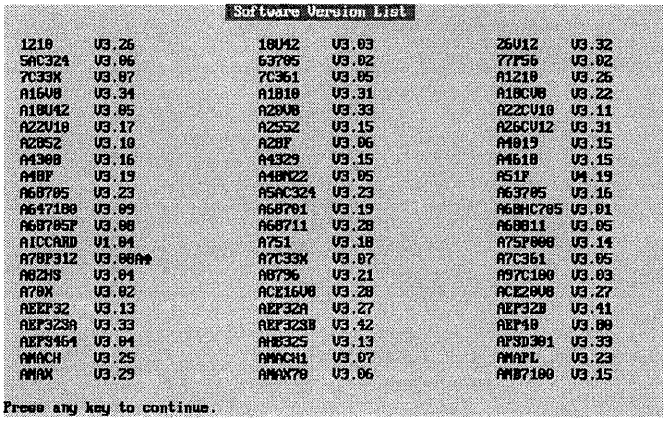
The "Device List" option provides a list of all programs available for each device. You have the option to print the information to screen, printer, or a file. If you choose to print the list to screen, be prepared to halt the scroll by depressing the <Pause> key. To continue the scroll after pressing <Pause>, press <Esc>. The beginning portion of the screen will appear as follows:



```
                        DEVICES  LIST

                      PC-Based  UNIVERSAL

                      Programmer & TESTER

Page

                    ***** EPROM Device *****

<<AMD/MMI>>
 Am2716          Am2716B         Am2732          Am2732A
 Am2732B         Am2764          Am2764A/AP/C64  Am27128
 Am27128A/AP/C128 Am27256HV      Am27C256/P/M256 Am27512/C512/L
 Am27512HV       Am27C010        Am27H010        Am27H010
 Am27LV010       Am27C100        Am27C020/LV020  Am27C040
 Am27C080        Am27C1024       Am27C2048       Am27C400
 Am27C4096

<<ATMEL>>
 AT27HC64/L      AT27C128        AT27C/HC256/L   AT27C/HC256R/RL
 AT27LV256R      AT27C512        AT27C512R       AT27LV512R
 AT27C513        AT27C513R       AT27C010/C010L  AT27LC010
 AT27LV010       AT27C011        AT27C020        AT27LV020
```

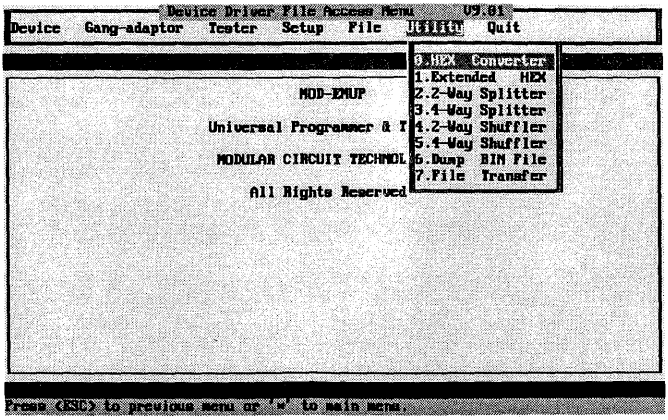
Version List

The "Version List" provides a list of device programs available, followed by their version number. You have the option to print the information to screen, printer, or a file. If you choose to print the list to screen, the information will be displayed one screen at a time. To advance to the next screen, press any key. The beginning portion of the screen will appear as follows:



Utilities

A number of utility programs used to manipulate files can be accessed from the "Utilities" menu or entered at the DOS command line. The following screen shows the options available from the "Utilities" pull-down menu.



Note: Not all utility functions are accessible from within the MOD-EMUP-A access software program. If the operation you need is not available from within the "Utility" pull-down menu, access the utility from the DOS prompt.

Utilities in the MOD-EMUP-A Access Software

The "Utility" pull-down menu allows you to modify files from within the MOD-EMUP-A Access Software. The eight options available are, "HEX Converter", "Extended HEX", "2-Way Splitter", "4-Way Splitter", 2-Way Shuffler", "4-Way Shuffler", "Dump BIN File", and "File Transfer".

HEX, Splitter, Shuffler, and Dump

Choosing an option other than "File Transfer" will call one of the following dialog boxes. Enter the requested information at the prompt and press <Enter>. For more information on each utility, refer to pages 6-9–6-12.

HEX Converter and Extended HEX

Choosing "HEX Converter" or "Extended HEX" will call the following dialog box. "HEX Converter" uses the file HEXBIN.EXE, while "Extended HEX" runs HEXBIN2.EXE.

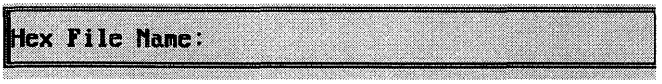


Figure 6-1. Dialog Box for "HEX Converter" and "Extended HEX"

2-Way and 4-Way Splitter

Choosing "2-Way Splitter" or "4-Way Splitter" will call the following dialog box. "2-Way Splitter" uses the file SPLIT2.EXE, while "4-Way Splitter" runs the file SPLIT4.EXE.

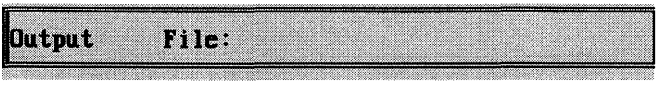


Figure 6-2. Dialog Box for "2-Way Splitter" and "4-Way Splitter"

2-Way and 4-Way Shuffler

Choosing "2-Way Shuffler" or "4-Way Shuffler" will call the following dialog box. "2-Way Shuffler" uses the file SHUFF2.EXE, while "4-Way Splitter" runs the file SHUFF4.EXE.

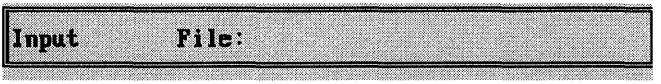


Figure 6-3. Dialog Box for "2-Way Shuffler" and 4-Way Shuffler"

Dump BIN File

Choosing "Dump BIN File" will call the following dialog box. "Dump BIN File" runs the file DUMP.EXE.

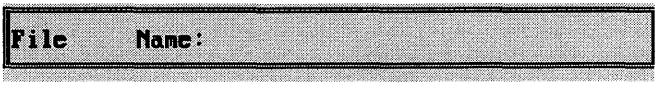


Figure 6-4. Dialog Box for "Dump BIN File"

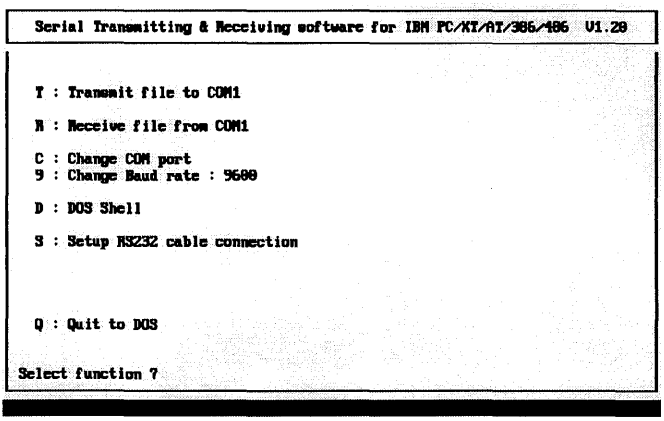
File Transfer

The "File Transfer" feature allows a file to be transmitted between two computers connected together directly through a serial port connector.

Note: To use the "File Transfer" feature":

- ❑ The two computers must be connected through a direct serial port connection.
- ❑ Both computers must be running the same version of the MOD-EMUP-A Access Software
- ❑ Both computers must be set at the same baud rate
- ❑ One computer must be set to transmit
- ❑ One computer must be set to receive

Choosing "File Transfer" from the "Utility" pull-down menu will produce the following menu:



Transmit Files to COM1

Choosing "Transmit Files to COM1" allows you to select the file you want to send. There are two ways to identify a file to be transmitted.

1. When the initial screen appears, a flashing cursor appears following the prompt:

   `File name:`

   Type in the complete file name, including the drive letter and path and press <Enter>.

   or

2. Press the <Tab> key to activate the scrolling file list. Using the <Arrow>, <Page up>, <Page down>, and <Enter> keys, you are able to move freely within your disk's directory to locate the file to be loaded. Once found, highlight the file name and press <Enter>.

   To view files on an alternate drive, press the drive letter <Enter>.

   The file name that you have selected will appear in the LOAD window.

Receive Files from COM1

If you plan to receive a file, choose "Receive Files from COM1" and enter the name of the incoming file.

Change COM Port

The "Change COM Port" option toggles between the COM1 and COM2 ports. Use this option to send data through a specified port for file transfer.

Change Baud Rate

The baud rate can be set to the following speeds: 1200, 2400, 4800, 9600, 19200, and 38400.

DOS Shell

Choosing the "DOS Shell" option will cause the software to search for COMMAND.COM on the boot disk drive. The following screen will appear:

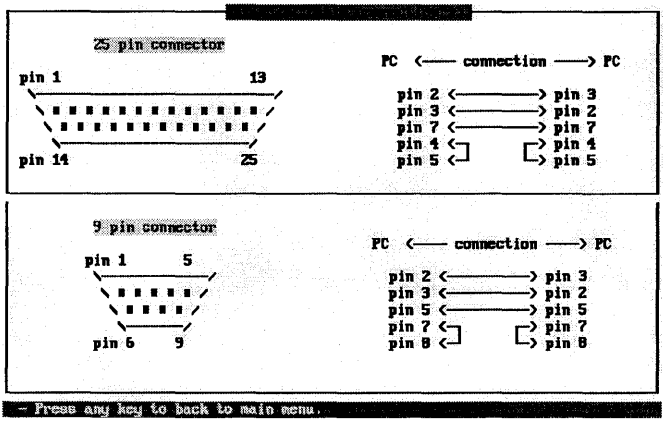
```
Type EXIT to return to Programmer menu

Microsoft(R) MS-DOS(R) Version 6.20
            (C)Copyright Microsoft Corp 1981-1993.

C:\EMUP>>
```

The software will execute the COMMAND.COM command and pass control back to DOS, allowing you to perform DOS operations while running the MOD-EMUP-A Access software in the background. To return to the Programming Main Menu, type:

exit <Enter>

Setup RS232 Cable Connection

The "Setup RS232 Cable Connection" provides the following diagram for wiring a cable with a DB9 on one end and a DB25 on the other.



Using Utilities from DOS Prompt

The utility files that are accessible from the DOS prompt are located on Disk 4 of the MOD-EMUP-A software

BIN2HEX

The "BIN2HEX" utility is used to convert a bin file back to hex.

CUT2

The "CUT2" utility takes a single file and cuts it into two separate files. Each of the new files maintain the same starting address as the original file.

CUT4

The "CUT4" utility takes a single file and cuts it into four separate files. Each of the new files maintain the same starting address as the original file.

DUMP

Binary ROM code cannot be displayed using the "DOS TYPE" command. The "DUMP.EXE" can convert such a binary file to hex and display it on the screen.

The input command is as follows:

```
A:\> DUMP FILENAME [start address] <Enter>
```

EDITBIN

The "EDITBIN" utility allows you to edit the information that is loaded in the buffer. See "Edit Buffer" on page 3-7 for more information.

HEXBIN

The "HEXBIN" utility will convert a file in hex format to binary. Some assemblers or compilers produce files in hex format for use with stand alone programmers. Since the MOD-EMUP-A uses I/O control through the CPU of the computer, the information for it to use must be in binary. The "HEXBIN" utility has a conversion limit of 64K bytes.

The HEXBIN.EXE (and HEXBIN2.EXE) can convert five types of HEX formats. Those formats are:

1. <I>ntel Hex
2. <M>otorola SHex
3. <D>igital Research
4. <T>eketronic
5. <H>Intel Hex-32

The starting address may be specified to eliminate any unwanted information from being converted. This will decrease the size of the BINARY file.

HEXBIN2

The "HEXBIN2" utility works the same as the "HEXBIN", but has a maximum conversion size of 256K bytes.

REVERT

The "REVERT" utility will change the hex value of the file to its inverse. The following table lists the hex numbers and their inverse:

Initial Hex Value	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Inverse Hex Value	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

SEP2MB

The "SEP2MB" utility will separate a source file that is larger than 2 megabytes into 2 separate files. The first file will be will be a full 2 megabytes, and the second file will be the remainder.

SERPC

The "SERPC" utility allows you to connect two systems using a serial cable and transfer information. This utility is the same as "File Transfer" from the "Utility" pull-down menu.

SHUFF2

The "SHUFF2" utility can shuffle two (2) 8 bit source files into a 16 bit file. The first 8 bit file will become the LOW byte and the second file will become the HIGH byte.

SHUFF4

The "SHUFF4" utility can shuffle four (4) 8 bit source files into one 32 bit file. The first 8 bit file will become the first byte of the 32 bit file. The second file will become the second byte, etc.

To execute the "SHUFF" files, type in the filename at the DOS prompt and you will be prompted to enter the information needed to perform the merge.

SPLIT216

This utility splits a 16 bit source file into two (2) 8 bit files. One file contains the data lying on the LOW order bytes of the 16 bit file. The other file contains the data lying on the HIGH order bytes. This utility is useful, for instance, if programming an EVEN and ODD eprom.

To execute these files, type in the file name (Split216, Split3, Split4, or Split 8) at the DOS prompt. You will be prompted to supply information needed to perform the function.

SPLIT2

The "SPLIT2" utility works the same as the "SPLIT16" utility, but is not limited to a 16bit source file.

SPLIT3

The "SPLIT3" utility works the same as the "SPLIT216" utility, except it splits a source file into three(3) 8-bit files.

SPLIT4

This utility splits a 32-bit source file into four(4) 8-bit files. The first file contains data from the first byte of the 32-bit file. The second file contains the data from the second byte of the 32-bit file, etc.

SPLIT8

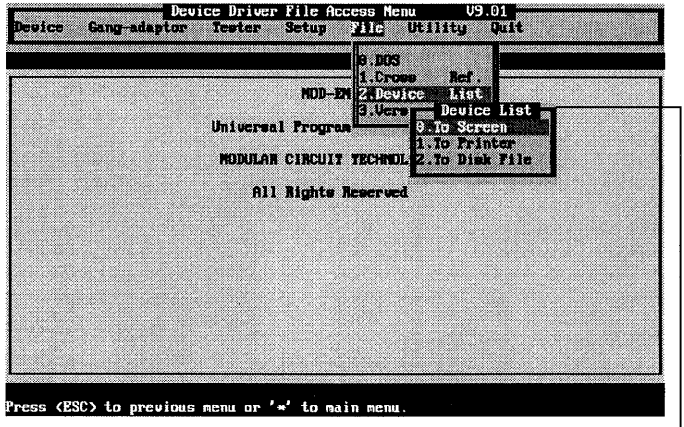
The "SPLIT8" utility works the same as the "SPLIT216" utility, except it splits a source file into eight(8) bit files.

7

Technical Reference

Device List

For a complete device listing, choose "Device List" from the File pull-down menu. This will document each device by manufacturer, device type, the programming file name, and the algorithm in which each device is programmed. You have the option of printing the list to screen, printer, or a disk file.



Device List Pull-down Menu

Optional Adaptors

The following adaptors are currently available for the MOD-EMUP-A:

EPROM Adaptors

MUP-EP32-4	4-Gang 32-pin DIP EPROMs
MUP-EP40-4	4-Gang 40-pin DIP EPROMs
MUP-PLCC-512	2716-27512 PLCC EPROMs
MUP-PLCC-1M32	32-pin PLCC EPROMs
MUP-PLCC-1M44	44-pin PLCC EPROMs

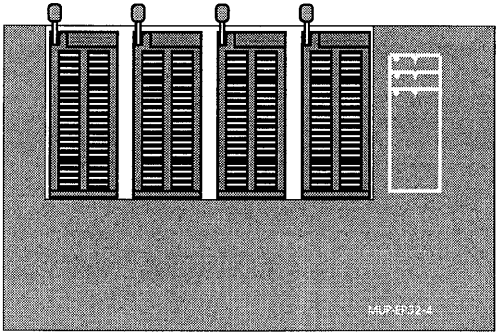


Figure 7-1. EPROM Adapter

EPROM Emulator Adaptors

MUP-RAMROM16	Emulates EPROM 16K
MUP-RAMROM32	Emulates EPROM 32K
MUP-RAMROM256	Emulates EPROM 256K
MUP-RAMROM512K	Emulates EPROM 512K

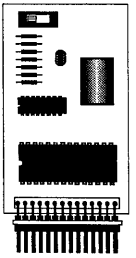


Figure 7-2. EPROM Emulator Adapter

Microprocessor Adaptors

MUP-PLCC-MPU	8751,8748 and 8742 PLCCs
MUP-51-4	4-Gang 8751 family DIPs
MUP-48-4	4-Gang 8748 family DIPs
MUP-C751	87C51 and 87C752 DIPs
MUP-PGA-8796	8751,8748 and 8742 PGAs
MUP-68705	68705 DIP microcomputer

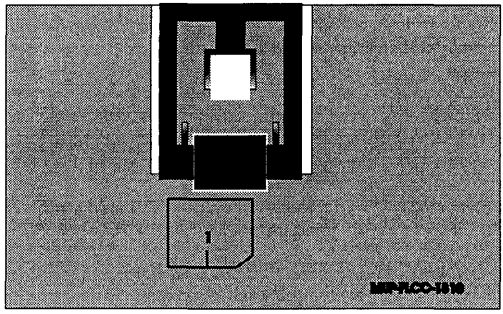


Figure 7-3. Microprocessor Adapter

PLCC/PLD Adaptors

MUP-PLCC-610	PLCC EP600 & EP610s
MUP-PLCC-910	PLCC EP900, EP910 & EP1210
MUP-PLCC-P20	PLCC C,H,N,P,R,V & X Type family PLDS
MUP-PLCC-P20A	PLCC M,P,V,R & X Type 20-22 family PLDs
MUP-PLCC-P20B	PLCC C,L,R,S & X Type 6-20 family PLDs

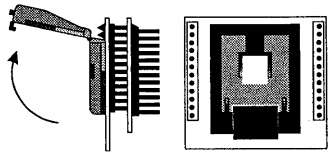


Figure 7-4. PLCC/PLD Adapter

Miscellaneous Adaptors

MUP-SIP-SIMM	DRAM module tester for SIPs SIMMs
MUP-16V8-4	4-Gang for 16V8 family EPLDs
MUP-CE16V8-4	4-Gang for CE16V8 family EPLDs
MUP-18CV8-4	4-Gang for PEEL 18CV8 family of EPLDs
MUP-20V8-4	4-Gang for 20V8 family EPLDs
MUP-CE20V8-4	4-Gang for CE20V8 family EPLDs
MUP-NSDPAL	NS Dtype 16L8, 16R8, 20L8, 20R8 families
MUP-PLS	Programmable Logic Sequencer
MUP-PLCC-1810	PLCC EP1810 family & 5C180 EPLDs

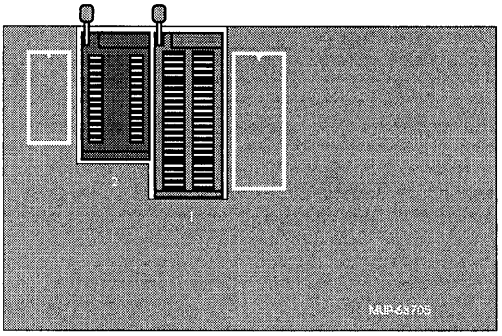


Figure 7-5. Miscellaneous Adapter

Note: Other adaptors will be added to support new devices, and may be available. The most current version of the MOD-EMUP-A Access Software is also available under the part number EMUP-SOFT. Consult your dealer for more information.

Trouble-Shooting

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

1. The power cord may be disconnected from the computer or the wall. Check all power cord and cable connections.
2. Your adaptor card may be grounded improperly. Remove the card, and reinstall. Be sure to secure the adaptor card bracket to the system chassis with mounting screws.
3. Your power supply may not have sufficient power to drive both your system and the adaptor card.

"When I try to use a modular programmer, I get communication error messages!"

1. You may not have the I/O port set correctly for your programmer. Check the I/O port assignment.
2. You may not have a chip properly inserted into the ZIF socket. Remove the chip and reinsert into the ZIF socket.
3. There may not be a good connection between the adaptor and the MOD-EMUP-A. Check the cable connection.
4. Your MOD-EMUP-A programmer may not be able to run at higher bus speeds. Try slowing the speed down. Most motherboards allow you to alternate between a low speed and a high speed by pressing the "Turbo" switch or by using a series of keystrokes (refer to your motherboard/systems manual). You can also change the bus speed by modifying your AT clock in the Advanced Chip Set in the CMOS Setup.

Note: Remember, the bus speed is not the same thing as the CPU speed. The adaptor card will not run with bus speeds greater than 8MHz. If the bus speed is running at a speed greater than 8MHz, you may be able to change the speed. Consult your motherboard manual.

5. A chip is in the ZIF socket when the software is attempting to load. Remove the chip form the ZIF socket and try to load the software again.

"When I install the adaptor card, some of my other peripherals start behaving strangely!"

1. You are probably experiencing an I/O port conflict. Double check the I/O port assignments on all your peripherals, including the adaptor 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 Troubleshooting section.
7. Try it on another system.
8. Compare system requirements with your configuration.
9. Ask your in-house technical support.
10. Ask whoever installed the product for assistance.

MOD-EMUP-A

Revision 2.0 MN

1 July 1996

MMODEMUPA/10

MMAUEMUPA · C4 / 1997