MODULAR CIRCUIT TECHNOLOGY



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

MOD-EMUP-A Universal Programmer

Owner's Reference Guide



Copyright

© 1994 MODULAR CIRCUIT TECHNOLOGY

This publication is protected by copyright, and all rights are reserved. No part of this publication may be reproduced or transmitted by any means or in any form without prior consent, in writing, from 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.

Introduction

What is the MOD-EMUP-A Universal Programmer?	1-1
What's Included	1-2
Conventions Used.	1-3
System Requirements	1-4

Installation

Device

Default	3-1
EPROM	3-2
Programming Main Menu	3-3
DOS Shell	3-3
Load BIN or HEX File to Buffer	3-4

	Save Buffer to Disk
	Edit Buffer
	Change I/O Base Address of Adaptor Card
	Display Loaded File History
	Display Buffer
	Modify Buffer Structure
	Mfr. Select
	Type Select
	Target Zone
	Program Speed, Algo
	Blank Check
	Read
	Program
	Verify
	Auto (B&P&V)
	Compare & Display
	Display
	Quit
EEF	PROM Software
	Erase
	Data Protection
SEF	RIAL PROM Software
BPF	ROM Software
	Swap Hi-Low Nibbles in Buffer
	Program Initial Byte
MP	U Software
	Program Lock Bit
	Program Encryption Code3-26
PLI	O Software
	PLD Option
	Load JEDEC File to Buffer
	Security Fuse Blow
	Other Options for PLD Software

Gang-Adaptor

Multiple Programmer	4-1
Using the Gang-Adaptor.	4-2

Tester

Tester
Logic IC Software 5-2
IC Type Select
Function Test
IC Number Specify 5-4
Search Unknown IC Number 5-5
Loop Test
User Defined Test Vector 5-7
DOS Shell 5-8
Load Test Vector into User Buffer 5-8
Save Test Vector from User Buffer
DRAM Test Program 5-14
SRAM Test Program
SIP/SIMM
PLD Vector Test
View/Edit Test Vectors 5-19
Display or Print the Result of the Function Test
User Defined PLD Vectors

Setup, File, and Utility

Setup	1
File	-2
DOS	-2
Cross Reference	2
Device List	.3
Version List	4

Utiliti	ies	6-4
U	Jtilities in the MOD-EMUP-A Access Software	6-5
U	Jsing Utilities from DOS Prompt	6-9

Technical Reference

Device List	1
Optional Adaptors	2
Trouble-Shooting	5
10 Things to do Before Calling Your Dealer	6



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 standalone 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:



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 č: 🕤

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> 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.



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 staticshielded 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.

Installation

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.

Installation

2-5

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	11:0 3 a
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	9:0 2 a
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	
			127 0 49 1. 4	

437,248 bytes free

Disk2

EXE	38,080	11-17-94	11:20a
EXE	30,960	9-14-94	1:44p
LIB	11,342	7-29-91	10:02a
LIB	3,657	2-06-90	12:16p
LIB	29,152	11-03-93	3:55p
VEC	224	11-07-91	9: 3 4a
VEC	64	11-07-91	9: 3 4a
EXE	29,952	9-15-94	2:53p
EXE	30,240	9-15-94	3:16p
EXE	30,432	9-15-94	3:36p
EXE	27,968	9-16-94	3:10p
EXE	33,680	9-14-94	5:31p
EXE	25,648	9-14-94	11:49a
EXE	36,688	9-16-94	4:44p
EXE	39,024	11-02-94	10:15a
EXE	34,016	9-16-94	9:4 3 a
EXE	32,208	9-14-94	2:41p
EXE	31,680	9-15-94	4:04p
EXE	34,688	10-18-94	9:15a
EXE	32,112	9-14-94	2:02p
EXE	32,688	9-16-94	5:48p
EXE	32,752	9-14-94	9:59a
EXE	32,416	9-15-94	5:19p
DOC	16,508	1-22-94	9:57p
file(s)		646,179 bytes	
		805,376 bytes f	free
	EXE EXE LIB LIB VEC EXE EXE EXE EXE EXE EXE EXE EXE EXE E	EXE 38,080 EXE 30,960 LIB 11,342 LIB 3,657 LIB 29,152 VEC 224 VEC 64 EXE 30,240 EXE 30,242 EXE 30,242 EXE 30,242 EXE 27,968 EXE 25,648 EXE 36,680 EXE 25,648 EXE 39,024 EXE 34,016 EXE 32,208 EXE 31,680 EXE 31,680 EXE 32,208 EXE 34,016 EXE 32,208 EXE 31,680 EXE 32,208 EXE 32,208 EXE 32,688 EXE 32,688 EXE 32,688 EXE 32,616 DOC 16,508 file(s)	EXE 38,080 11-17-94 EXE 30,960 9-14-94 LIB 11,342 7-29-91 LIB 3,657 2-06-90 LIB 29,152 11-03-93 VEC 224 11-07-91 VEC 64 11-07-91 EXE 30,240 9-15-94 EXE 30,432 9-15-94 EXE 30,432 9-15-94 EXE 30,432 9-15-94 EXE 27,968 9-16-94 EXE 25,648 9-14-94 EXE 36,688 9-16-94 EXE 36,688 9-16-94 EXE 36,688 9-16-94 EXE 34,016 9-16-94 EXE 34,016 9-16-94 EXE 32,208 9-14-94 EXE 31,680 9-15-94 EXE 32,068 9-14-94 EXE 32,688 10-18-94 EXE 32,688 9-14-94

MOD-EMUP-A Owner's Reference Guide

Disk3

2-6

PROG	DEV	139,350	7-25-95	5:17p
DIAG03B	EXE	25,920	12-05-95	9:4 2 a
BIN2HEX	EXE	12,945	9-06-93	9:23a
CUT2	EXE	9,263	5-03-94	1:54p
CUT4	EXE	9,48 3	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	4:0 2 p
SERPC	EXE	24,716	12-08-93	5:0 3 p
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, 43 1	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 copyprotected. 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.

2-9



MOD-EMUP-A Owner's Reference Guide

User Notes

3

Device

Choosing "Device" from the Access Main Menu will call a submenu 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.

9. DEFAULT	and the second state of th		
.EEPROM		MOD-ENUP	
Serial PROM	Universal	Programmer & Tester	
.NPU/NCU .PLD	MODULAR C	IRCUIT TECHNOLOGY.	
	A11 M	ights Reserved	
		DI	spault
		TYPE : Am2716 ALGO. : INTL	UPP: 25.00U
		Device Driver Fil Adaptor Name :	le : EPP512.EXE

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.

Device

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.

	NODULAR CIRCUIT ! NODEL: NOD-ENUP EPRON 512 section	TECHNOLOGY (C) 1994)m U3.45	-1977: 11(178) -1977: 927128 -077: 210	HEADY BITS 1 HOUSE BITS AUGT 15.800
	1. DOS SHELL 2. Load BIN or HI 3. Save buffer to 4. Edit buffer 5. Change 1/0 bas 6. Display loaded 9. Modify buffer	X file to buffer disk 7. Display buffer e address I file history structure	TABLE Buffer start alds end addr Check San Douice start aldr.	7 2018 9 000 9 00177 9 000 8 000 COLMTER 9000
	I. Type select Z. Target zone 3. Program speed.	N. Mfr. select		
	B. Blank check P. Frogram R. Read C. Compare & disj Q. Quit	D. Dieplay A. Auto(B āřā V) V. Verify Jay error		
Pro	Buffer size Buffer structure Select function	: 128K bytes : FC HENDRY in 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:



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>

3-4

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:

	C:NERUPN#.#		WPF :210 WCF :	5.8U:INTL 5.8U
A: N: B: 0: C: P:	CDIR) CDIR) EPP512.EKE 89355	06-02-94 06-02-94 03-09-94	TANGET ZOME Buffer start addr.: 00000	
D: Q: E: R: F: S:	EPF10248.ERE 76301 EPP10249.ERE 81207 BPPGM.ERE 106007	03-16-94 03-09-94 03-10-94	end addr.: 63FFF Check Sum : 6000 Device start addr.: 6000	COUNTER 0000
: I: : U: : V:	SEEP1.EXE 147300 SEEP2.EXE 55276 MPU1.EXE 60716	03-10-94 03-10-94 03-10-94	LOAD :	-
J: V: K: X: L: Y:	EEP1.EXE 102452 PGM48.EXE 52505 PGM51.EXE 72105	03-16-94 03-14-94 03-21-94	File name:	
M: Z:	PGN28.EXE 56529 README91.DOC 29784	83-18-94 83-16-94		
	PPLP1.EXE 00737 PPLP1.EXE 76757 PPLP2.EXE 57405	12-31-93 03-00-94		
	GAL1.EXE 63261	03-31-94		

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.

3-5

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

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

C 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.

3-6 MOD-EMUP-A Owner's Reference Guide

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

	C:\ENUP*.*			*TYPE: 27C4096 (WORD *UPP : 12.75U *UCP : 1	WIDĖ) 5.250
A: N:		(DIR)	86-82-94	TARGET ZONE (VOR	D WIDE)
B: 0:		(DIR)	06-02- 91	Buff start addr. :00000 Buff	not enough
C: P:	EPP512.EXE	89355	83-89-94	end addr.:9FFFF	
D: Ö:	EPP1024B.EKE	78381	03-16-94	Check Sun : C444	
E: W:	EPP1024W.EXE	61297	03-09-94	Dev start addr.:00000	COUNTER
P: 3:	BPPGH.EXE	106807	03-10-94	Dev end addr.: 3FFFF	00000
G: I:	SEEP1.EXE	147908	03-10-94		
H: U:	SEEPZ.EXE	55276	03-10-94		
1: 0:	RPU1.EXE	68716	03-10-94		999 00 (24 million - 24 cm - 2
1: 0:	EEP1.EXE	102452	03-16-94		
R: K:	FGIME . EXE	52585	03-14-94	File name: SEEFI.EKE	
L: 2:	PGH51.EKE	72165	83-21-99	(B)in, (I)ntel HEX, (N)otoroli	I S MEX : B
M; Z:	FGRZS.EXE	56529	03-10-94	road address [0000]: a	
85 398	NENDIESI.DUC	47(01	03-10-71	Loading	
	LFLUI.LAL	00/2/	03-11-71	TOK T END BYLE HUUN IFI	
	TELEL BAL	(0(3)	12-31-33	IDOLLEN 12 UNI FUNDAU I	
	CALL PUR	57761	03-00-71	I manage and the second second second	
	OULLI . EAD	03701	03-31-34	Tress any key to continue	
				I FUR PRESS (UN) TO DACK TO NO.	N 800

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.

	NDDULAR CINCUIT TECHNOLOgy NDDEL: NOD-ENUP (C) 1994 EPNON 512 section V3.45	HTT: INTAL HEADY BIT: 1 TYPE: 127128 HONSPERD: INTL HTT: 210 HOCF: 5: 50
•	Fin firm firm firm firm firm firm firm firm	Tenbert sign: Buffur start addr. 00000 End addr. 00077 Chock Sum 0000 Desics start addr. 0000 Desics start addr. 0000
	T. Type select H. Hfr. select 2. Target zone 3. Frogram speed, algo	File name: c:\baffer Start addrews[00000]:00000 End addrews[03FFF]:03FFF Saving nov
	B. Blank check D. Display F. Frogram A. Auto(B&FAU) N. Read V. Verify C. Compare & display error Q. Quit	OK 1 Press any key to continue
	Buffer size : 128K bytes Buffer structure : PC NIZHORY Select function 73	Or press (CR) to back to main senu

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:

[start],[end] start start,end,destination	CRETURN> : DUMP CRETURN> : EDIT			
' start,end,data start,end start,end start,end,ASCII data start,end,BINAN data filenase largull larg	CRETURNS : MOVE BLD (RETURNS : FILL BLD (RETURNS : FILL BLD (RETURNS : CHECK SU (RETURNS : ASCII S (RETURNS : SHEARY S W21(RETURNS : SHE (RETURNS : QUIT	CN CN DCN M Earch Max. 15 Earch Max. 7 LL	characters BYIES	
• The information list The absolute start and The Buffer size : 121 	d below is for refer dreaw of BUFFER : 69 K BYTES	ence only : 25:0000		

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.

3-8 MOD-EMUP-A Owner's Reference Guide

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	88	89	00	99	88	89	80	90	90	88	00	98	00	88	99	00	
66618	88	89	88	90	60	88	60	66		89	88	88	88	88	88	08	
00020	88	88	88	88	88	89	88	98		88	88	68	88	88	88	88	
00030	88	89	80	88	88	88	88	68		89	88	00	88	88	88	68	
66648	88	88	88	88	88	89	60	88	00	88	88	68	88	88	89	88	
00050	88	88	80	88	60	88	60	68		88	88	60	88	88	88	08	
66668	88	88	88	88	68	88	89	88	00	88	88	68	80	88	88	88	
00070	68	89	88	88	88	89	88	68		88	80	60	88	89	88	88	
00000	69	88	88	88	88	89	88	66	09	80	88	88	88	88	88	88	
666636	68	88	88	88	68	88	80	88		88	60	88	B 0	80	88	88	
00000	88	88	88	88	88	89	60	00	00	88	88	60	90	60	88	88	
000B9	60	88	68	88	88	89	89	88	89	68	88	68	98	68	68	88	
66660	88	88	88	88	88	89	88	88		88	88	88	88	88	88	88	
ecede	60	88	88	88	88	88	60	00		60	88	99	90	60	88	88	
6666E9	88	88	88	88	88	88	60	60	88	88	88	00	88	60	88	88	
000F9	88	88	88	88	88	89	80	90	80	60	88	98	88	88	69	88	
00100	88	88	88	88	68	89	60	00		88	68	80	90	88	99	88	
60119	88	88	88	88	88	89	88	90	00	88	60	88	90	88	88	88	
69129	88	00	88	88	68	88	68	90	88	88	88	80	90	60	88	88	
00130	80	88	88	88	68	89	88	88	88	88	88	60	80	68	88	88	
66146	88	88	88	88	68	88	60	98	80	88	88	90	90	88	00	60	
00150	88	88	60	88	88	88	60	88	88	88	88	88	90	88	00	88	
00160	00	88	88	86	88	88	60	88	60	88	88	88	88	88	60	88	
00170	00	88	88	88	88	88	89	60		88	88	88	88	88	68	88	
(ESC)	: B	ack	to	C0		ndi	pro	not	, [Ct	rl	K]:	To	ool	e bi	etu	ccn	BIMARY 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.

3-9

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).

3-10

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:

Start	End	File name		
66666		READNES1.DOC		
66666		EPPS12.EXE		
80008				
DISIONO D	00000			
00000	00000			
0.00000	00000			
00000				
00000				
00000				
00000-				
88888				
00000-				
00000	00000			
00000	00000			
0.0000				
000000	- 00000 2			
00000				
00009-				
00000-				
86668				
66660	- 00000 :			
00000	00000			
	00000			
CLERK (ану жеу т	o continue		

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.

Device

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

Press	(ESC)	to 1		inste	die	pl	IJ.								
86666	10 58	88	81 (17	65	68	81	35	83	FF	FT	cs	17	HZ
86919	69 66	CE	47	72 15	73	82	1E	88	60	60	81	88	66	83	
00020	69 66		83 I	H9 86	98	83		68	68	83	89	88	70	83	
86939	69 66	78	83 6	HA 88	64	83	89	88	55	83	88	88	30	83	. n. d. II =
80848	69 66	14	83 6		88	83		88	FC	82	88	88	PB.	82	and the second sec
00050	69 66	24	82.6		DE	82		88	87	82	88	88	82	82	
00068	69 68	90	82 6		-	62		88	64	82	88	88	6P	67	
60676	68 66	58	82.6	10 GE	48	82		68	-	82	68	88	-	82	XBA
86869	68 68	C7	81 6	18 88	82	61		88	78	81	88	88	50	81	
86998	69 68	41	81 E	H9 66	80	81		88	85	81	88	88	28	88	
88868	68 66	89	88 6		89	88		88	SC.	88	88	66	83	64	
88988	88 88	6.	66 B	19 AR	59	RR.		88	77	66	88	88	23	88	
	88 88	13	-	-	69	-		88	43	87	88	00	-	87	C 9
Reade	88 88	10	87 6		80	97		88	10	-	88	-	-		
AGAEA	88 88	82	86.6	10 80	RE	86		88	12	86	88	88	98	86	
RAAPA	68 60	03	86.6		48	66		80	87	-	88	00	78	06	
00100	00 00	77	-	10 00	69	~		00	23	-	00	-	57	-	
89119	00 00	57	-		777	26		20	19		88	200	17	0.6	N
89129	00 00	BP.			87	-	00	90	WP.	2	86	80		95	
00120	00 00	202	00 0		EC.	22		80	97	00	80	00	700	00	
00140	00 00	20	00 0		- F O	20		00	20		00	00	10	00	····X.
00450	00 00	20	22 5			23		00	24	00	00	00	10	23	··· p···· u···· ····
00120	00 00	80	83 6		13	22			23		88	86	00	94	*******
00108	00 00		172 8	2. March	63	2.8		98			망난	66	110	67	*************

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.

NODULAR CIRCUIT TECHNOLOGY NDDEL: NOD-EXUP (C) 1934 EPNON 1024 wection U3.52 Main Nenu 1. DOS SHELL 2. Load Bin or HEX file to buffer	-Mfr.: AMD/MMI -FGMSFEED: QUICX -TYFF: Z7C4995 (WORD WIDE) -VFF: 12.75U -VCF: 5.25U -TARGET ZONE (WORD WIDE) 						
3. Save buffer to diak 4. Edit buffer 7. Display buffer 5. Change I/O base address	Check Sum : 8000 Dew start addr.:80000 Dew end addr.:3FFFF 80000						
 B. Swapping LOW-HI byte in buffer Modify buffer structure 	Modify buffer structure						
T. Type select N. Hfr. select S. Speed, algo Z. Target zone B. Blank check D. Display P. Program A. Auto (B&P) B. Bead U Userio	0. 1: C: BUTF : THP < 2. D: BUTF : THP < 3. E: SUTF : THP 4. F: SBUTF : THP						
C. Compare & display error Q. Quit	Open C:\NUFF.THP file 52% completed						
Duffer size : 128K bytes Buffer structure: PC MEMDRY	Select number: 1 (ESC) Back to main menu.						
Select function 7 9							

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:

MODULAN CIRCUIT TI MODEL: NOD-ENUP (C EPROM 512 wection	CHNDLOGY) 1994 1 03.45	#NFr.: Intel #Type: D27128 #UFF : 21U	+BLANK BIT: 1 +PGMSPEED:INTL +UCP :6.0U			
1. DOS SHELL 2. Load BIN or HD 3. Save buffer to 4. Edit buffer 5. Change I/O base 6. Display loaded	file to buffer disk 7. Display buffer address file history	TAR Buffer start addr end addr Check Sun Device start addr	ET ZUNE .: 00000 .: 03FFF : 0000 .: 0000	COUNTER 9909		
00. 27/27C 01. AND/MAI 02. ATMEL 03. CATALYST 04. CYPRESS 05. FUITSU 06. HITACHI 07. HYUNDAI 08. INTEL CCR> back to a SELECI NUMBER 7	69. MATSUSHITA 19. MICROCHIP 11. MITSUBISHI 12. MOSTEK 13. MOTOROLA 14. MEC 15. NS 16. OKI 17. RICOH	18. PHILIPS 19. SHU3 29. SG3_THOMSON 21. TI 22. TUSHIBA 23. USI 24. ULSI 25. SEEQ 26. MMIC	27. SONY 28. ICT 23. DHN1- 30. SHARP	WAUE		

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

Device

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

TYPE SELECT : -			 	
0.82716 Z5U	9.027/C513	12.50		
1.D2732 25V	A.D87C64	12.50		
2.02732A 21V	B.D87C256	12.75U		
3.D2764 21V	C.067C257	12.750		
4.02764A/C64 12.5V	D.D27C128	12.750		
5.D27128 21V	E. D68C257	12.75U		
6.D27128A 12.5U				
7.D27/C256 12.75U				
8.027/C512 12.75V				
(CR) back to main menu.				
SELECT NUMBER 7				

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:

NODULAN CIRCUIT TECHNOLOGY NODEL: NOD-ENUP (C) 1994 EPROM 512 section V3.45	•Mfr.:Intel •Bland Bit: 1 *Type:D27129 •PGRSPEED:Intl. *Upp:210 •Ucp::6.00
Ann Menu 1. DOS SHELL 2. Load BIM or HEX file to buffer 3. Save buffer to disk 4. Edit buffer 7. Display buffer 5. Change L/D base address 6. Display loaded file history	TARGET ZONE Buffer start addr.: 00000 end addr.: 03777 Check Sun : 0000 Device start addr.: 0000 0000
7. Type select M. Hfr. select 2. Target zone 3. Frogram speed, algo	Buffer start addr.: Buffer end addr.: Device start addr.:
B. Blank check D. Display P. Frogram A. Auto(B&P&V) R. Read V. Verify C. Compare & display error Q. Quit	Press any key to continue
Buffer size : 128K bytes Buffer structure : PC NENORY Select function 7z	Or press (ESC) to back to main menu.

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:

NDDULAR CIRCUIT TECHNOLOGY NDDEL: NOD-ERUP (C) 1994 EPROM 512 Bection U3.45	 MFr.: INTEL *IVPE: D27128 *FGNSPEED: INTL *UPP: 21U *UCP: 5.8U
1. DOS SHELL 2. Load BIN or HEX file to buffer 3. Save buffer to disk 4. Edit buffer 7. Display buffer 5. Change L/O base address 6. Display loaded file bictory.	TARGET ZONE Buffer start addr.: 00000 end addr.: 03FFF Check Sum : 0000 Device start addr.: 0000 0000
5. Rodfy buffer size : 1288 bates	PGM SPEED, ALGO : 0.NORMAL: Some per byte SUcc.default Upp 1.INIL : Ime retry 50 times 6.Ucc.default Upp, intelligent 2.QUICM : 0.Ime retry 50 times 6.ZSUcc.12.75Upp, quick pulse 3.INTR : Ime retry 50 times 6.Ucc.13.0Upp, interactive 4.FLASH(SNAP) : 0.Ime retry 10 times (EXPRESS) 6.SUcc.13.0Upp, labrite (CR) back to main menu. SELECT NUMBER 7

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:

NODULAN CIRCUIT TECHNOLOGY NODEL: NOD-ENUP (C) 1994 EPNON 512 section V3.45	-HFF.:INTEL -BLAND +TYPE:D27128 -PGMSP +UPF:21U -UCF:	BIT: 1 EED:INTL 6.00
1. DOS SHELL 2. Load BIN or HEX file to buffer 3. Save buffer to disk 4. Edit buffer 7. Display buffer 5. Church 10 base address	TARGET ZURE Buffer start addr.: 90999 end addr.: 93PFP Check Sum : 6099 Duble start addr.: 8999	COUNTER
5. Display loaded file history 9. Modify buffer structure I. Type select H. Mfr. select 2. Target zone 3. Frogram speed, algo	BLANN CHECK dewice:- Ready to check (V/CCD>)? Blank checking nov DK 1	0000
B. Blank check D. Display P. Program A. Auto(B&P&U) R. Read U. Verify C. Compare & display error Q. Quit		
Duffer size : 128K bytes Buffer structure : PC MEMORY Select function 7b		



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:

HODULAN CIRCUIT TECHNOLOGY HODEL: HOD-ZHUP (C) 1994 ETRON 512 section 03.45	-Mfr.:Intel ==BLANN BIT: 1 =TYPE:D27128 ==PGHSPEED:INTL =UPF:210 ==UCF:5.80
1. DOS SHELL 2. Load BIM or HEX file to buffer 3. Save buffer to disk 4. Edit buffer 7. Display buffer 5. Change 1/0 base address	TARGET ZUNE Buffer start addr.: 90909 end addr.: 93977 Check Sum : 0090 Device start addr.: 9099 9099
5. Jospiej Joanne Tite History 9. Modify buffer structure 1. Type select M. Mfr. select 2. Target zone 5. Program speed, algo	READ to buffer :- Ready to read (Y/Even/Odd/(CR>)? Reading now OK 1
B. Blank check D. Display P. Frogram A. Auto(J&PA U) R. Bead U. Verify C. Compare & display error Q. Quit	
Buffer size : 128% bytes Buffer structure : PC MEMORY Select function 7r	

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:

NODULAR CIRCUIT TECHNOLOGY	-Hfr.:Intel -Blank Bit: 1
NODEL: NOD-ERUP (C) 1994	-Type:D27128 -Grspeed:Intl
EFRON 512 mection U3.45	-Upp:210 -Ucp:6.80
1. DOS SHELL	TARGET ZIME
2. Load BIN or HEX file to buffer	Buffer start addr.: 00000
4. Edit buffer 7. Display buffer	Check Sun : 0000 COUNTER
5. Change 1/0 base address	Device start addr.: 0000 0000
5. Display loader file history 9. Modify buffer structure 1. Type select M. Mfr. select 2. Target zone	PHOGRAM : Ready to program (V/Ewen/Odd/(CR))?
B. Blank check D. Display F. Frogram A. Auto(B&F&U) R. Red U. Verify C. Compare & display error O. Duti	
Buffer size : 128N bytes Buffer structure : PC NEMORY Select function 7p	

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

3-17



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.

NODULAN CINCUIT TECHNOLOGY NODEL: NOD-ENUP (C) 1994 EPNDN 512 section U3.45	NFr.:INTEL -BLAN NYPE:D27128 -PGNSI UPP:210 -UCP	EBIT: 1 YEED: INTL 6.00
1. DOS SHELL 2. Load BIN or HEN file to buffer	TARGET ZONE Buffer start addr.: 00000	
3. Save buffer to disk 4. Edit buffer 7. Display buffer 5. Change 1/0 base address	Check Sun : 0000 Device start addr.: 0000	COUNTER 0000
5. Proping Loads file mistory 9. Modify buffer structure 7. Type select M. Mfr. select 2. Target zone 3. Frogram speed,algo	UERIFY with buffer Ready to werify (Y/Even/Odd	((CR))?
B. Blank check D. Display F. Frogram A. Auto(32780) R. Head U. Verify G. Compare & display error Q. Quit		
Buffer size : 128K bytes Buffer structure : PC NENURY Select function 7v		

Verify Option

Verify Screen

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:

NODULAN CINCUIT TECHNOLOGY NODEL: NOD-ENUP (C) 1994 EPROM 512 meetion U3:45	≈Nfr.:INTEL ≈Type:D27128 ≈Upp::210	-Blank -Pghsp -UCP ::	BIT: 1 EED:INTL 5.00
1. DOS SHELL 2. Load BIN or HEX file to buffer 3. San buffer to disk	TAI Buffer start add		
4. Edit buffer 7. Display buffer 5. Change 1/0 base address 6. Display Londed file bistory	Check Sur Device start add	: 9999 r.: 9999	COUNTER 8088
9. Modify buffer structure	Ready to start (Y/Even/Odd/	(CR>)7
T. Type select H. Mfr. select Z. Target zone 3. Frogram speed, algo			
B. Blank check D. Display P. Program A. Auto(B&P&U) R. Read U. Verify C. Compare & display error Q. Quit			
Buffer size : 128K bytes Buffer structure : PC MEMORY Select function ?a	L <u></u>		
Auto (B&P&V) Option		Auto (B&	kP&V) Scree

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

3-20 MOD-EMUP-A Owner's Reference Guide

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:

NDDULAN CIRCUIT TECHNOLDGY NDDEL: NOD-ENUF (C) 1994 EPROM 512 mection U3.45	+HFr.:INTEL +BLANK +TYPE:D27128 +FGNSP +UPP:21U +UCP:	BIT: 1 EED:INTL 6.8V
1. DOS SHELL 2. Load DIN or HEX file to buffer	TARGET ZUME Buffer start addr.: 00000	
5. Save suffer to also 4. Edit buffer 7. Display buffer 5. Change 1/0 base address	Check Sun : C000 Device start addr.: 0000	COUNTER 0900
9. Modify buffer structure	COMPARE :	
T. Type select M. Mfr. select Z. Target zone S. Frogram speed,algo	Comparing now ON 1	EXCHOLO
B. Blank check D. Display P. Program A. Auto(B&P&V) H. Read U. Uerify C. Compare & display error Q. Quit		
Buffer size : 128K bytes Buffer structure : FC MEMORY Select function 7c		

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."

Frees	(ESC) to	stop displaying
8888		n n n n n -n
9919	FF FF FF	FF FF FF FFFF FF FF FF FF FF FF FF
9029	FF FF FF	PF FF FF FF
8G38	TT TT TT	TT T
8848	77 77 77	TT FT FT FT FT
8658	FT FT FT	TT TT TT TT
8868	FF FF FF	TT TT TT TT TT
9979	TT TT TT	TT T
99999	FF FF FF	TT TT TT TT TT -TT TT TT TT TT TT TT
68'96	FF FF FF	TT TT TT TT TT
8888	FF FF FF	FF FF FF FF
ARBA		PP 79 79 77 77
eerse	** ** **	
aana		FF
		** ** ** ** **
COND		
0100		VP VV VP VV VV VP VP VV VV VV VV VV
DATE		
0110	VE PR PR	
0120		
or and	22 E 2	
	1	

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:

MOD-EMUP-A Owner's Reference Guide

NODULAN CINCUIT TECHNOLOGY NODEL: NOD-ENUP (C) 1994 EEPROM section V3.32		-Hfr.: Intel: -Upp : 120 -Type: 289929				
1. DOS SHELL 2. Load BIN or HE 3. Save buffer to 4. Edit buffer 5. Change 1/0 bas 6. Display loaded	X fils to buffer disk 7. Display buffer e address file history	Buffer Device	start end Check start	TARGET addr.: addr.: Sun : addr.:	ZUNE 00000 3FFFF 0000 00000	COUNTER 99999
T. Type select Z. Target zone	M. Mfr. select					
B. Blank check P. Program N. Read C. Compare & disj E. Erase Q. Quit	D. Display A. Auto(BEPSU) U. Verify Ilay error S. Data protection					
Buffer size Buffer structure Select function 7	: 128% bytes : PC MEMORY					

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.

3-23

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

MODULAR CIRCUIT TECHNOLOGY MODEL: MOD-EMUP (C) 1994 SERIAL (EE)FROM mection V3.37		«Mfr.: »Type: «UPP :	«Mfr.: EXEL «TYPE: XL93C46/P «UPP :		-BLANK BIT: 1	
1. DOS SHELL 2. Load BIN or H 3. Save buffer to 4. Edit buffer 5. Change L/O bas	n nemu 2K file to buffer 5 disk 7. Display buffer 5 ddress	Buffer Device	start end Check start	TARGET addr.: addr.: Sun : addr.:	20ME 0000 0077 0000 0000 0000	COUNTER
T. Type select 2. Target zone 8. Blank check P. Frogram R. Read C. Compare & dim Q. Quit	N. Hfr. select D. Display A. Auto(D&P) U. Verify play error					
Buffer wize Starting address Select function '	: 32N bytes at 7669:0000	_				



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.



MOD-EMUP-A Owner's Reference Guide

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:

HODULAN CINCUIT TECHNOLOGY HODEL: HOD-ENUP (C) 1994 BIPOLAN PROM section V3.31	- Mfr.: AND - Blank Bit: 0 - Prifix: 273 - Type: 28/A,23/A/SA		
1. DOS SHELL 2. Load BIN or HEX file to buffer	TARGET ZUME Buffer start addr.: 0000		
 Save buffer to disk Edit buffer 7. Display buffer Change I/O base address Display loaded file history Swap hi-low nibbles in buffer 	end addr.: 01FF BYTE Check Sum : 0000 Device start addr.: 0000	CUUMTER 8988	
T. Type welect N. Mfr. select Z. Target zone B. Blank check D. Display F. Program A. Auto(B&FAU) R. Read U. Verify C. Compare & display error Q. Quit			
Allocation buffer size : 64N Bytes Starting address at 68E8:0000 Select function ?			



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 preprogrammed 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. 3-26

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:

NODULAR CINCUIT TECHNOLOgy NODEL: NOD-EMUP (C)1994 NPU 8751 section U3.30	* Mfr.: INTEL * Type: 8751BH TARGET ZURE Buffer start addr.: 0000			
Anin Menu 1. DOS SHELL 2. Load BIN or HEX file to buffer	end addr.: 0777 Check Sus : 0000 Device start addr.: 0000	COUNTER 0000		
 Save suffer to also A. Edit buffer 7. Display buffer Change 1/0 base address Display loaded file history 	Current Encryption c 00 OF :FFFFFFFFFFFFFFFFFFFFFFFF 10-1F :FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	ode TETETETETE TETETETETE		
program Encryption code:				
1 : Edit Encryption code 2 : Program Encryption code	3 : Load Encryption code 4 : Save Encryption code			
Select which number (1/2/3/4/ <cr>)?</cr>				

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:

NDDULAR CIRCUIT TECHNOLOGY NODEL: NOD-ENUP (C) 1994 PALP1 section U3.54	* Hfr.: And Pal * Type: Ampal1618/8/Al/A/Q/L
1. DOS SHELL 2. Load JEDEC file to buffer 3. Save buffer to disk 4. Edit buffer 7. Display buffer 5. Change 1/0 base address	TARGET 20HE Check Sun : 0000
T. Type select H. Hfr. select	
B. Blank check D. Display P. Frogram A. Auto(BAFAUA3) R. Read U. Verify S. Security fuse blow Q. Quit	
Select function ?	

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.

3-29



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:

HODULAN CIRCUIT TECHNOLOGY HODEL: HOD-EMUP (C) 1994 GAL1 section U3.45 		* Hfr.: LATTICE * TYPE: GAL22V18/B/C (mo UES)
		TANGET ZONE Check Sun : 9999
T. Type select	N. Mfr. select	
B. Blank check P. Program R. Head E. Erase Q. Quit	D. Display A. Auto(BAPAVAS) U. Varify S. Security fuse blow	
Select function		



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.

MOD-EMUP-A Owner's Reference Guide

User Notes

4

Gang-Adaptor

Multiple Programmer

The MOD-EMUP-A can be used for programming with gangadaptors. 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.



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:

MODE ICTE	L: NOD-2NUP (C) 1994 ST section V3.31	+ TYPE: TTL74	- NUMBER : 00
Main	Nemu:		
1. 2.	DOS SHELL Change 1/O base-addr. of hardware		
T. N. F. S.	IC Type select IC Number specify Function test Loop test Search unknown IC number User defined test vector		
A. B. Q.	DRAM test program SRAM test program Quit		
9æ1æ	t function ?		

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:

MODEL: MOD-EMUP (C) 1994 ICTEST section V3.31	* TYPE: ITL74 * NUMBER: 00
Main Menu: 1. DOS SHELL 2. Change 1/0 base-addr. of hardware	PURCTION TEST: Ready to test (Y/CR>)?
 IC Type select IC Number specify F. Function test L. Loop test S. Search unknown IC number U. User defined test vector 	
A. DHAM test program B. SHAM test program Q. Quit	
Select function ?f	
Inction lest Option	Function Test Win

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.

NUDULAR CIRCUIT TECHNOLOGY MODEL: MOD-EMUP (C) 1994 ICTEST section V3.31 * TYPE: TTL74 * NUMBER: 00 Main Menu: IC NUMBER SPECIFY: Enter IC number: 1. DOS SHELL Change I/O base-addr. of hardware T. IC Type select M. IC Mumber specify F. Function test L. Loop test S. Search unknown IC number U. User defined test vector A. DNAM test program B. SNAM test program Q. Quit Select function 7n IC Number Specify Option IC Number Windo

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 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.

MODULAN CIRCUIT TECHNOLOGY MODEL: MOD-EMUT (C) 1994 ICTEST section U3.31	+ TYPE: TTL74	* NJABER: 00
Main Menu: 1. DOS SHELL 2. Change 1/0 base_addr. of hardware	LO Ready to test C	UP TEST: V/CR))?
 T. IC Type select M. IC Mumber specify F. Function test L. Loop test S. Search unknown IC mumber U. User defined test vector 		
A. DAAM test program B. SNAM test program Q. Quit		
Select function ?)		
oop Test Option		Loop Test Wind

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. DUS SHELL	
Load test vector into user buff	cr
3. Save test sector from user buff	rir-
4. Rdit test westor in user buffer	
P Burning Land Land	
r. runceion cest	
L. LOOP test	
D. Display result during testing	
Q. Back to last news.	

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

5-7

5-8

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:



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:

ect D		Select File from	OR	Type Complet
Comments	l:Tab Esc Enter			
	GALLENE 5326	1 83-31-94		
	FFLF1.EXE 7675	7 12-31-93		
	EPLD1.EXE 8874	7 83-11-94		
	READMED1.DOC 2978	4 83-16-94		
1. T. N: 2:	PENZE RVE 5657	9 83-19-94		
K: X:	PGMUB.EXE 5256	5 83-14-94		
J: U:	EEP1.EXE 18245	2 83-16-94		
1: U:	MPUL.EXE 6071	6 93-18-94 File nas	e:	
H: U:	SEEPZ.EXE 5527	6 83-18-54		
F: 3: C: T:	SFFGR.EAE 10000	9 93-10-31 Inen ·		
E: R:	EPP1024W.EXE 8120	7 83-89-94		
D: Q:	EPP10248.EXE 7838	1 93-16-94		
C: P:	EPP512.DE 8935	5 83-89-94		
A: N: B: D:	, CDIM	2 80-82-39 13 86-82-94		
-				
	C:\EMUP*.*			방법 사람이 많은 것은 것이 없는 것이다.

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 Window



MOD-EMUP-A Owner's Reference Guide

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

Tester

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		111111111122222 123456789012345678901234
Vector symbol definition: 9 : apply LOW to IC. 1 : apply HI to IC L : expected output result from IC is LOW H : expected output result from IC is HI N : same symbol as last vector. K : same symbol as last vector, but don't care output result from IC E : apply S Uto IC, only IP 19.20.22.24 can be specified G : apply GND to IC, only IP 12 can be specified Cursor movement key assignment: UF ANROW : cursor up DOWN ANROW : cursor up	U0001 U0002 U0003 U0004 U0005 U0005 U0005 U0006 U0007 U0008 U0009 U0010 U0011 U0013 U0013 U0014	123456/050123456/0501234 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LEFT ARROW : cursor left NIGHT ARROW : cursor right FgUp : last page FgUn : mext page Ins : insert 1 line Del : delete 1 line CCR> : next line 1st position CCSC> : end editing	00015 00015 00017 00019 00019 00029	

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.

5-12

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:



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.



MOD-EMUP-A Owner's Reference Guide

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:

DNAM test program m	cnu	* NUMBER: 4164 (64K+1)	
N. Number select F. Function test			
Q. Back to main menu			
Select function 7			

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



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

DRAM test program menu	* NUMBER: 4164 (648*1)
N. Number select F. Function test	DRAM function test : Ready to test (V/CR>)?
Q. Back to main menu	
Select function 7f	
DRAM Function Test Option	n 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:

N. Munder select Q. Back to main menu elect function ?	SRAM test program menu	+ NUMBER: 6116
Q. Back to main memi	N. Number select 7. Function test	
lect function ?	Q. Back to main menu	
	lect function 7	
그 동생이 같다. 방법에 지수가 있는 것이 가지 못했는 것은 것이 같이 같다. 방법에 관심하게 가지 않는 것이 같이 가지 않는 것이 같다.		

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



1

Select the desired SRAM number and press <Enter>.

5-1'

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



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:




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:

MODULAR CIRCUIT TECHNOLOGY MODEL: MOD-EMUP (C) 1994 PAL-TEST section U3.48	*Mfr: GENERAL FLDs *Type: 8 pins File:
Main Menu:	
1. DOS SHELL 2. Load test vectors 3. Save test vectors 4. Ujev/edit test vectors 5. Change 1/0 base address	
M. Mfr. select T. Type sele	et
F. Function test D. Loop test D. Display or print the result of function test	
Q. Quit	

Select function ?

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:

1111111112 12345670501234567050	(Home) Top line (Ind) Bottos line (India) Face as
U9991 XXC10000000000000000000000000000000000	Chulm> Page down (ALT-C) Clear all (Im>) Insert a line (Del) Belete a line (A) Append a line (Y) Copy a line
	(1) Fower pin (3) Drive input law (1) Drive input high (1) Drive output Law
	 (i) Drive output high (b) Drive input 0, 1, 0 (c) Drive input 1, 0, 1 (c) Datput not tested (c) Datput not tested (c) High impedance (d) Display output

Display or Print the Result of the Function Test

Choosing this option will produce the following screen:

Command: (ESC) exit.(S) a	save test.vec.d	> print.(oth	ir) display no	xt result.
00001 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	(N=			
17812				

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

INPU	TS .						
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=!Blo	ck_DCD	;Card is	selected +	Also A Out		
OUTF	UTS						
Pin	12=!G	;Output	G				
Pin	13 = !F	;Output	F				
Pin	14 = !E	;Output	Е				
Pin	15=!D	;Output	D				
Pin	16=!C	;Output	С				
Pin	19=!Por	tEqual;Com	parator Ou	itput			
Fiel	d Addres.	s=[SA91]	;Noti	ce "0" not	included		
LO=!	SA0						
F=Ac	ldress:(3	۵۵303) ه	Block_DC	D ;1st	Block		
E=Ad	ldress:(3	04307) &	Block_DC	D ;2nd	Block		
D=Aċ	ldress:(3	0830B) &	Block_DC	D ;3rd	Block		
C=Ac	ldress:(3	0C30F) &	Block_DC	D ;4th	Block		
PortEqual=Address:(30030F)							

1

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".

```
111111111222222223333333334
1234567890123456789012345678901234567890
V0001
XXXXXXXX000000001N1HLHHHXXLNXXXXXXXXX
```

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.

5-22 MOD-EMUP-A Owner's Reference Guide

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:

NOP-EX	01.005 1.Cross 2.Device 3.Version	Bef. List List	
Universal Program			
MODULAR CIRCUIT	TECHNOLOGY.		
All Rights B	eserved		

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

6-3

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

	CRUSS REFER	ENCE		
	PC-Based UNI	VERSAL		
	Programmer &	TESTER		
#Fage No. 1-1 06/28/1994	anana EPRON D	evice ***		
HANLIFACTURER	DEVICE TYP	E	ALGO	DEU DRIVER FILE
	Aw2716	25.000	INTL	EFF512.EKE
and/nn i	Am27168	12.500	INTL	EPF512.EXE
AND/HILL	A#2732	25.000	INTL	KPP512.EXE
RMD/HHI	Am2732A	21.000	INTL	EPF512.DOE
	FIRZ/328	12.500	INTL	EPT51Z.ERE
	0-27540 408 4754	12 500	THE	PRECO PUP
	0=27128	21 000	INTI.	PPEGI2 PVP
AND/INI	9-271288/8F/C128	12.500	INTL	EFF512.EXE
AMD/INI	An27256HU	21.000	INTL	EPP512.EXE
AND/INI	Am270256/1/H256	12.750	DUICH	EPT51Z.EXE
AND/MIL	A=2751275127	12 750	DUICK	REPSI2 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:

	DEV PC-Bas	ices list ed universal	
	Progra	mer å tester	
	nume EPR	DM Device manage	
((AND/HHI))			
Am2716	AnZ7168	An2732	Am2732A
Am27328	An2764	An2764A/AP/C64	Am27128
Am27128A/AP/C128	Am27256HU	An2702567P74256	An27512/C512/L
Am27512HU	Am270919	An27H818	Am27118010
An27LU010	A#27C198	Am27C028/LU020	An27C040
An270888 An2704896	A#2701824	An27C2948	An27C490
(ATNEL>)			
AT27HC64/L	AT27C128	AT27C/HC256/L	AT27C/HC256R/RL
AT27LUZ56R	ATZ7C512	AT2705128	AT27LU5128
AT270513	AT27C513	AT27C918/C918L	AT27LC010
AT27[1019	AT270811	AT270829	AT271 UB29

6-4

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:

218	03.26	19042	E0.EV	26012	U3.32
5AC324	V3,06	63705	V3.02	77 P 56	20.EU
7C33X	V3.07	70361	V3.85	A1219	V3.26
A16V8	V3.34	A1819	V3.31	A18CU8	V3.22
A18U42	V3.85	A2908	V3.33	A22CU10	V3.11
A2ZV18	V3.17	A2552	V3.15	A25CU12	V3.31
A2852	V3.10	A28F	V3.06	A1019	V3.15
A430B	V3.16	A4329	V3.15	A4618	V3.15
A48F	V3.19	A4BN22	V3.05	A51F	VH.19
A68785	V3.23	A5AC324	V3.23	A63785	V3.16
A647180	V3.09	A68701	V3.19	AGBHC705	U3.01
A68785P	V3.08	A68711	V3.28	A69811	U3.05
AICCARD	V1.84	A751	V3.18	A75P000	U3.14
A78P312	V3.08A4	A7C33X	V3.07	A7C361	V3.05
ABZHS	V3.04	A8796	V3.21	A97C180	V3.03
A79X	V3.02	ACE16U8	V3.28	ACE28U8	V3.27
AEEP32	V3.13	AEP32A	V3.27	AEP32B	V3.41
AEP32SA	EE.EU	AEP32SB	V3.42	AEP40	V3.00
AEP3464	V3.84	AHB325	V3.13	AP3D391	U3.33
AMACH	V3.25	AMACHI	V3.07	AMAPL	V3.23
AMAX	V3.29	AMAX79	V3.06	AN37199	V3.15

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.

	9.HEX Converter
HOD-IEMUP	2.2-Way Splitter
	3.4 Way Splitter
Universal frogrammer a 1	5.1-Way Shuffler
 MODULAR CIRCUIT TECHNOL	6.Dump BIN File
All Rights Reserved	





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"

6-6

MOD-EMUP-A Owner's Reference Guide

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
- D 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:

Seri	al Transmitting & Receiving software	for	IBM	PC	/XI/	AT/	386	~186	U1	29
	· .									
T , : '	Transmit file to COM1									
R : 1	Acceive file from COM1									
C : 9 :	Change COM port Change Baud rate : 9600									
D : 1	003 Shell									
s : ;	Setup RS232 cable connection									
_										
Q : -1ect	Quit to DOS									
	- uno			ant i		i				

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:



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>

6-8

6-9

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	Α	В	C	D	Е	F
Inverse Hex Value	F	Е	D	С	В	Α	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.



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



Figure 7-1. EPROM Adapter

EPROM Emulator Adaptors

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





Technical Reference

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 MUP-PLCC-910 MUP-PLCC-P20

MUP-PLCC-P20A

MUP-PLCC-P20B

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





7-3



Miscellaneous Adaptors

MUP-PLS MUP-PLCC-1810

MUP-SIP-SIMM DRAM module tester for SIPs SIMMs MUP-16V8-4 4-Gang for 16V8 family EPLDs 4-Gang for CE16V8 family EPLDs MUP-CE16V8-4 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

> Programmable Logic Sequencer 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.



7-5

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 © 1996 MODULAR CIRCUIT TECHNOLOGY MMODEMUPA/10

