MODULAR CIRCUIT TECHNOLOGY™

MODULAR CIRCUIT TECHNOLOGY™

USER'S GUIDE

MCT-EPROM

EPROM PROGRAMMER

Your MCT-EPROM Programmer has all the features you need to program the popular 27 series EPROM (2716-27512). It can copy ROM programs from disk files to EPROMs or from one EPROM to another, it can compare the copy with the original to verify accuracy, and it can "blank-check" EPROMs before copying onto them. In addition, the MCT-EPROM Programmer has extensive memory manipulation commands that allow you to modify ROM programs once they are loaded into the memory buffer.

To help you use these many features, the MCT-EPROM Programmer's software combines all the programming commands in one easy-to-use menu. Simply locate the desired function (copy, save, etc.) on the menu and press the appropriate key on the keyboard: you need not memorize any commands.

With its many programming features and ease of use, the MCT-EPROM Programmer will prove itself to be a very effective programming tool for you. Thank you for purchasing the MCT-EPROM Programmer.

TECHNICAL SUPPORT

The MCT-EPROM Programmer should perform well for you. If it does not, please contact your dealer, or call the Technical Support Department at JDR Microdevices. The toll free number is **800-538-5000**. Please include a daytime phone number when writing.

PARTS LIST

Your MCT-EPROM Programmer package includes the following items:

one programmer adapter card
one ZIF socket & cable
one software diskette (EPROM Writer)
one user's manual

MANUAL OVERVIEW

The following chapter summaries will help you identify the chapters you need to read. However, for best results, you should read the entire manual before installing and using the MCT-EPROM Programmer.

1 INTRODUCTION

Summarizes manual content and arrangement. Lists items required for installation and use.

2 INSTALLATION

Provides detailed installation instructions.

3 SOFTWARE CONTENTS

Describes the software programs included with the software diskette.

4 USING THE EPROM PROGRAMMER

Provides detailed instructions for performing common programming tasks.

5 QUICK REFERENCE GUIDE

6 PROGRAMMING & VOLTAGE TABLES

TABLE OF CONTENTS

1 Introduction 7
Manual Contents 7
Manual Arrangements 7
Items Required for Installation 7
Tools 7
Computer System 8
DOS 8
EPROM Chips 8

2 Installation 9
Step 1: Remove Computer Case Cover 9
Step 2: Insert Programmer Card 10
Step 3: Secure Programmer Card 12
Step 4: Replace Case Cover 12
Step 5: Connect the Cable 13

3 Software Contents 14
HEXOBJ 14
UPP512 15
Memory Buffer 15
Default Settings 15
Note to Programmers 16

4 Using the EPROM Programmer 17
Copy from Disk to EPROM 17
Step 1: Select EPROM Type 17
Step 2: Place EPROM in Socket 19
Step 3: Load Disk File 20
Step 4: Copy Buffer 21
Copy from EPROM to Disk 23
Step 1: Read EPROM 23
Step 2: Copy Buffer 23

EPROM to EPROM Copying 25
Step 1: Read Source EPROM 25
Step 2: Insert Target EPROM 26
Step 3: Copy Buffer 26
Step 4: Compare Source & Target 26
Combining Two EPROMs 26
Step 1: Read 1st EPROM 27
Step 2: Read 2nd EPROM 27
Step 3: Copy Buffer 28
Dividing One EPROM into Two 29
Step 1: Read Source EPROM 29
Step 2: Copy Half of Buffer 29
Step 3: Copy Remaining Buffer 29

5 Reference Guide 31
<E> EPROM Type 31
<Q> Quit 32
<L> Load 32
<S> Save 32
<D> Buffer Commands 33
Display & Print 34
Substitute 34
Checksum 35
Fill 36
Move 36
<B> Blank Check 38
<V> Verify 39
<C> Copy 39
<1> Read 39
<2> Verify (A) 39
<3> Copy (A) 40
<4> Copy (B) 40
<5> Verify (B) 40

EPROM Type & Programming Tables 41

1 INTRODUCTION

MANUAL CONTENTS

This manual contains important instructions on the proper installation and use of your MCT-EPROM Programmer; please read it carefully. In addition to the detailed instructions on using the Programmer, there is a quick-reference guide to the EPROM Writer software as well.

MANUAL ARRANGEMENT

The manual is divided into seven parts:

(1) Introduction
(2) Installation
(3) Software Contents
(4) Using the EPROM Programmer
(5) Quick-Reference
(6) Appendix: EPROM Type & Voltage
(7) Index

ITEMS REQUIRED FOR INSTALLATION

TOOLS

You will need a phillips screwdriver in order to remove your computer's case cover and expansion slot cover.

COMPUTER SYSTEM

The MCT-EPROM programmer is designed for the IBM PC, PC/XT, PC/AT and compatibles. In addition, we recommend a microprocessor speed of 8Mhz or less; you may encounter difficulties if you attempt to use the EPROM programmer in a computer running at 10Mhz or 12Mhz.

DOS

The EPROM Writer software is designed to work with MS-DOS or PC-DOS (versions 2.0 or greater).

EPROM CHIPS

The MCT-EPROM Programmer can be used to program any of the following EPROMs:

2716
2732
2732A
2764
2764A
27128
27128A
27256
27512

Note: the MCT-EPROM programmer is not recommended for CMOS-type EPROMs (such as the 27C128).

2 INSTALLATION

To install the EPROM Programmer, you need only complete the following five steps:

(1) remove the computer cover
(2) insert the programmer card
(3) secure the card
(4) replace the computer cover
(5) connect the ZIF socket

Caution: turn off the computer before you begin installation! If you insert the programmer card into the expansion slot while the computer is turned on, you may damage the computer and the programmer card.

STEP 1: REMOVING THE CASE COVER

Before removing the computer cover, turn off the power and unplug the power cord from the wall outlet. Then, remove the phillips screws at the rear of the case (refer to figure 1).

REMOVE THE SCREWS ON THE BACK OF THE CASE

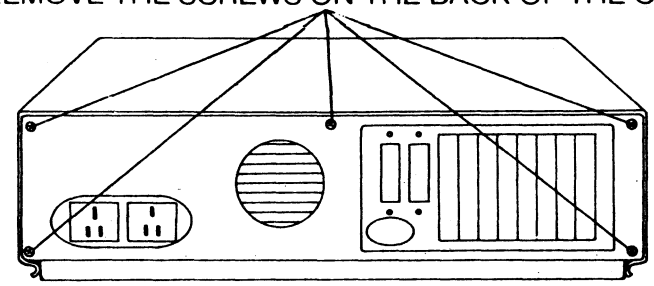


FIGURE 1

Finally, remove the top cover by sliding it forward (refer to figure 2).

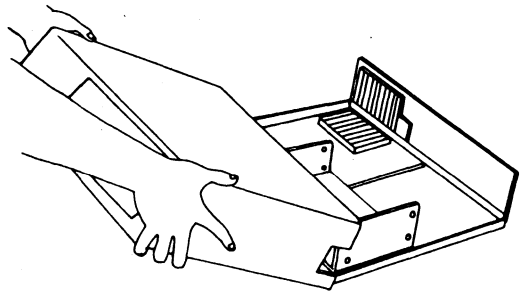


FIGURE 2

Note: if you have a flip-top computer case, you will not have to remove the top cover; simply press the two buttons near the top of the case, and raise the flip-top.

STEP 2: INSERTING THE PROGRAMMER CARD

After removing the computer cover, choose an expansion slot (any slot will work) for the programmer card and remove the slot cover (refer to figure 3).

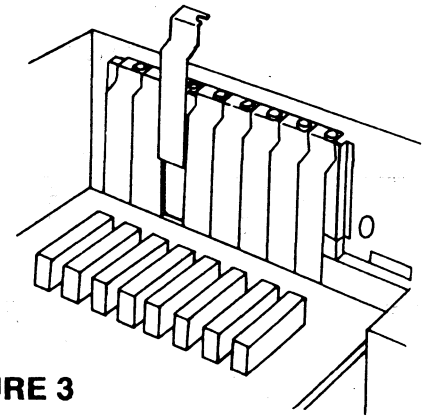
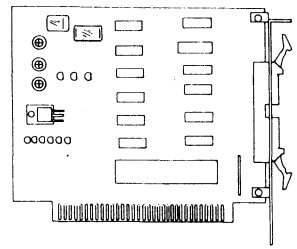


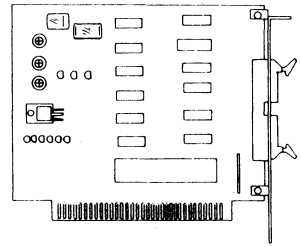
FIGURE 3

Then, before inserting the programmer card into an expansion slot, close the card's two connector latches (refer to figures 4). This will allow you to slide the card's external connector through the rear of the computer.

FIGURE 4



OPEN



CLOSE

Next, tilt the card slightly and lower it into the computer case so that the external connector slips through one of the slot cut-outs in the rear panel of the case.

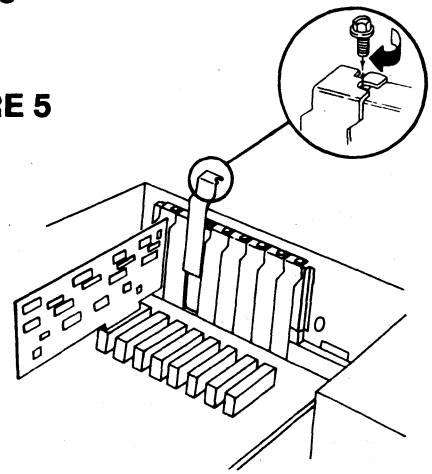
Finally, after the external connector is correctly positioned in the rear panel slot cut-out, lower the programmer card straight down into the slot and press it firmly into place.

Caution: make sure that the card is inserted all the way into the slot! The card may be damaged if it is not fully seated in the slot when the computer is turned on.

STEP 3: SECURING THE CARD

Once the card is inserted into the expansion slot, secure the card using the slot cover screw as shown in figure 5.

FIGURE 5



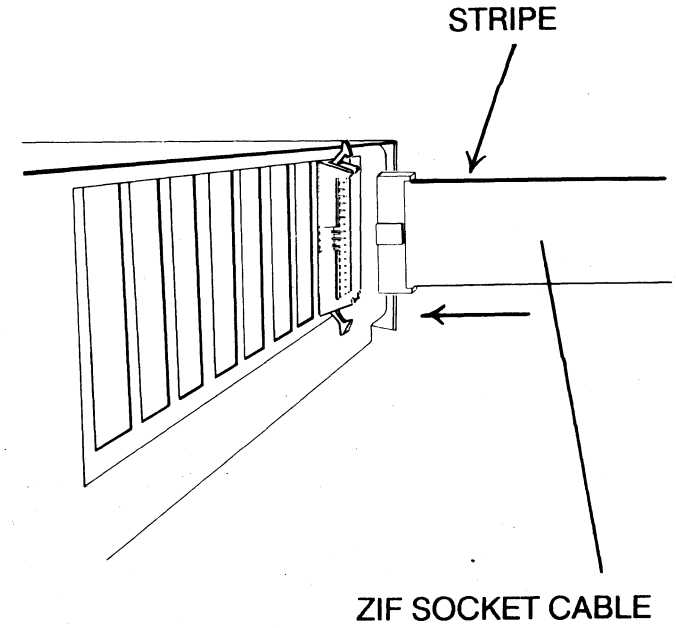
STEP 4: REPLACING THE CASE COVER

Now that the programmer card has been installed, slide the top cover into place and fasten it with phillips screws.

STEP 5: CONNECTING THE CABLE

Now, for the final step, connect the ZIF socket's cable to the programmer card. You will notice that there is a notch in the cable's end plug. Insert the cable plug into the programmer card's external connector so that the notch in the plug lines up with the slot in the card's connector (refer to figure 6).

FIGURE 6



3 SOFTWARE CONTENTS

The EPROM Writer diskette contains a conversion program called HEXOBJ and an operating program called UPP512.

HEXOBJ

HEXOBJ converts ASCII formatted files to the hex format required by the EPROM Programmer. This is a useful feature because some assemblers (Intel's, for example) generate ROM programs in ASCII format. If you have such an assembler, you must use the HEXOBJ conversion program.

To use the conversion program, insert the EPROM Writer diskette into your computer, and enter HEXOBJ. The screen will display the following:

```
input file:
output file:
```

Next, enter the name (include the drive letter or pathname) of the file to be converted (input file), then enter the name that you wish to give the converted file (output file). After you enter the output file, a wait message will be displayed while HEXOBJ converts the input file and records the converted file onto the destination disk. When the HEXOBJ program completes its work, the screen will display:

```
convert complete
```


Upon completion of the file conversion, HEXOBJ will return you to DOS.

Note: if you wish to exit HEXOBJ before completing a file conversion, press the escape key (ESC).

UPP512

The UPP512 program is the central operating program of the EPROM Writer software. It features a menu listing all of the programming functions.

NOTES ON USING UPP512

Memory buffer. The UPP512 program creates a 64K memory buffer. You may transfer disk files or the contents of an EPROM into this buffer. You may also transfer the contents of the buffer to a disk (the "save" function) or to an EPROM (the "copy" function).

Note: the memory buffer is cleared (erased) each time the read key <R> is pressed.

Default settings. The UPP512 program has a default setting for a 2764 EPROM requiring a programming voltage of 21 volts. If you intend to program a chip other than a 2764 / 21Vpp EPROM, you must change the setting by selecting <**E**> on the menu.

NOTE TO PROGRAMMERS

If you have never written a ROM program, you should be aware of the particular quirks of ROM programming. For example, if you use an assembler, you will not be able to use some of the assembler's instructions (such as those concerned with relocatable memory). For further details, consult the manual provided with your assembler.

4 USING THE EPROM PROGRAMMER

Caution: always select the EPROM type and voltage before inserting the EPROM into the ZIF socket! By taking this simple precaution, you will avoid damaging the EPROM by programming it at the wrong voltage.

COPYING FROM DISK TO EPROM

STEP 1: SELECT EPROM TYPE

To transfer the contents of a disk file to an EPROM, insert the EPROM Writer diskette into a floppy drive and enter UPP512. After you have entered UPP512, you will see the following menu:

```
Select from the following:
<E>: EPROM type / Vpp---(2764 / 21V)20
<Q>: Quit.
<L>: Load disk object file in buffer.
<S>: Save buffer on disk.
<D>: Display, modify, checksum or print buffer.
<B>: Blank check.
<R>: Read----------in buffer address 0000.
<V>: Verify--------with buffer address 0000.
<C>: Copy----------from buffer address 0000.
<1>: Read (A)------in buffer any address.
<2>: Verify (A)----with buffer any address.
<3>: Copy (A)------from buffer any address.
<4>: Copy (B)------Blank check and Copy.
<5>: Verify (B)----Verify & Display error.
```

Your first step should always be to select the EPROM type and programming voltage matching that of the EPROM you wish to program. To select the EPROM type and programming voltage that you need, press **<E>**. By doing so, you will enter the following EPROM selection submenu:

```
Select EPROM type & programming voltage (Vpp):
<1>: 2716----Vpp = 25V
<2>: 2732----Vpp = 25V
<3>: 2732A---Vpp = 21V
<4>: 2764----Vpp = 21V
<5>: 2764A---Vpp = 12.5V
<6>: 27128---Vpp = 21V
<7>: 27128A--Vpp = 12.5V
<8>: 27256---Vpp = 21V
<9>: 27256---Vpp = 12.5V
<A>: 27512---Vpp = 21V
<B>: 27512---Vpp = 12.5

Which one?
```

To select an EPROM type and voltage, simply enter the appropriate number (1-9) or letter (A,B). As soon as you enter your choice, you will be returned to the main menu. To exit the EPROM selection submenu, you must select an EPROM type (1-9, or A-B).

EPROM chips are not always clearly labelled; you may find it difficult to determine whether the EPROM you wish to program is a 2732 requiring 25V, or whether it is a 2732A, which requires 21V. If you encounter this problem, try to program the EPROM at the lower voltage first; if this does not work, then select the higher EPROM voltage setting.

Note: the programming voltages in the menu follow Intel's specifications; other manufacturers may have different specifications. The programming voltage tables in the appendix list the specifications of some other manufacturers.

STEP 2: PLACE EPROM IN ZIF SOCKET

After you have selected the correct EPROM type and voltage, insert the EPROM into the ZIF socket. The EPROM must be inserted all the way down at the ZIF socket end labelled "GND" (refer to figure 7).

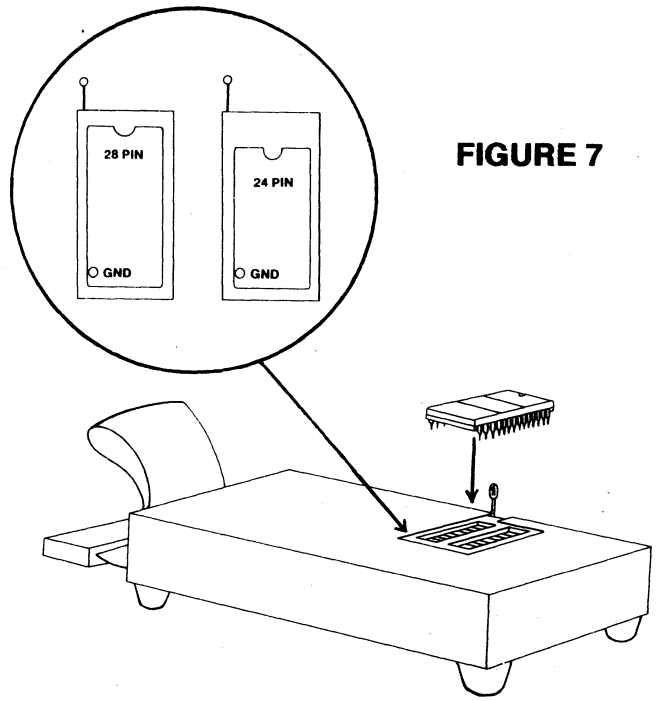


FIGURE 7

Caution: the notch-end of the EPROM must not be inserted at the end of the ZIF socket labelled GROUND! If you do so, you will damage the EPROM.

STEP 3: LOAD DISK FILE INTO BUFFER

After you have inserted the target EPROM into the ZIF socket, transfer your disk file to the memory buffer (the file must be in hex format). To do this, press <**L**>. The main menu will remain onscreen, but the following small submenu will appear under it:

Result: Buffer starting address (4 digits) = HEX

You must now specify the memory buffer location (address) to which the disk file will be sent. Enter 0000 if you wish the disk file to be transferred to the very beginning of the memory buffer. After you enter the address, the screen will display:

Key in file name?

Enter the name of the file you wish to transfer to the memory buffer. You must include the drive letter and any pathnames. After the file name has been entered, the screen will display:

Disk read ok.

The disk file is now in the memory buffer. As soon as the disk file has been successfully read (transferred) into the buffer, the cursor will return to its place at the top of the main menu, allowing you to make another menu selection.

Note: do not be concerned that the last message below the main menu ("Disk read ok") remains onscreen; the submenus under the main menu will persist until another menu selection is made.

STEP 4: COPY BUFFER CONTENTS TO EPROM

After the "**Disk read ok**" message appears, you are ready to copy the buffer contents onto the target EPROM. To do this, choose the copy function <**4**>. Although selection <**3**> or <**C**> will also copy the buffer to the EPROM, <**4**> is the better choice because it automatically determines whether the EPROM is blank and free of defects before copying to the EPROM. Press <**4**>. The submenu will read:

Result: READY? (Y/N)

Now, press <**Y**>. The programmer will either begin copying (it will display a "copy" message), or it will display the following message:

Blank # 1 ERROR AT XXXX

This message indicates that the chip has already been programmed. You will have to erase the EPROM or replace it with a blank EPROM.

If the EPROM is blank, but defective, the Programmer will attempt to copy the buffer onto the defective EPROM. Since it verifies as it copies, the Programmer will detect the defective EPROM and display this message:

Verify # 1 ERROR AT XXXX

However, if the EPROM is both blank and free of defects, the Programmer will copy the buffer contents onto the EPROM. The copy message will appear on-screen; scrolling numbers will also appear. When the transfer is complete, this message will be displayed:

Result: VERIFY # 1—ok—XXXX

Your disk file is now copied, or "burned", into the EPROM.

COPYING FROM EPROM TO DISK

STEP 1: READ EPROM INTO THE BUFFER

To copy the contents of an EPROM onto a disk, press <**E**> to enter the EPROM selection submenu and select the correct EPROM type and voltage. Then, insert the source EPROM (the eprom you wish to copy onto a disk) into the ZIF socket. Finally, press <**R**> to read the EPROM contents into the memory buffer. The screen will display:

Result: Read # 1—ok— YYYY
EPROM CHECK SUM = XXXX

Note: the number above the check sum (YYYY) tells you how many bits of data can be stored in the EPROM. The checksum number itself (XXXX) tells you how many bits of data are actually stored in the EPROM. In other word, YYYY is the EPROM capacity and XXXX is the amount of that capacity that is occupied by the ROM program.

STEP 2: COPY BUFFER CONTENTS TO DISK

After the EPROM has been read into the buffer, the cursor returns to the main menu. When the cursor is in the main menu, press <**S**> to copy the contents of the memory buffer onto the disk. The screen will display:

Result: Total byte to save (4 digits) = Hex

The number you must enter here depends on the capacity of the EPROM that you are using: you must enter a number that exceeds the EPROM's capacity by 1. The following table lists the number to be entered for each EPROM capacity:

Type:	Capacity:	(Enter this number) Total Byte To be Saved
27160	07FF	0800
27320	0FFF	1000
27640	1FFF	2000
27128	3FFF	4000
27256	7FFF	8000
27512	FFFF	0000

When you have entered the total bytes to be saved, the screen will read:

Buffer starting address (4 digits) = HEX

If you want to save the entire contents of the memory buffer, enter 0000. (If you wish to save only part of the buffer, you must specify the beginning address of that part.) The screen will now read:

Key in file name?

Next, enter the name you wish to give your new disk file. Be sure to include the destination drive and any pathname. The file will then be saved on the destination disk. Upon completion of the disk write, the screen will read:

Result: Disk write ok.

The cursor will exit the <**S**> submenu and return to the main menu. You now have a copy of the EPROM program filed away on the disk.

COPYING ONE EPROM ONTO ANOTHER

STEP 1: READ SOURCE EPROM INTO BUFFER

Begin copying an EPROM by pressing <**E**>, then select the EPROM type and voltage matching that of your source EPROM (the EPROM you wish to copy from). Insert the source EPROM into the ZIF socket. Then, read the EPROM's contents into the buffer by pressing <**R**>. (Write down the checksum number: you can use this number later to verify the target EPROM.)

STEP 2: INSERT TARGET EPROM

Next, remove the source EPROM and insert the target EPROM (the EPROM onto which the source EPROM's contents will be copied). If the target EPROM is a type different from the source EPROM, you must press <**E**> and select the appropriate EPROM setting.

STEP 3: COPY BUFFER CONTENTS

To complete the EPROM-to-EPROM copy process, press <**4**>. The Programmer will then copy the buffer contents onto the target EPROM (see "Copy from Disk to EPROM", Step 4). The source EPROM is now copied onto the target EPROM.

STEP 4: COMPARE SOURCE WITH TARGET

To determine whether you have successfully copied the source EPROM onto the target EPROM, press <**R**> and note the target's checksum. It should be exactly the same as the checksum of the source EPROM.

COMBINING TWO EPROMS INTO ONE

To illustrate this procedure, we will use a concrete example: combining two 2764 EPROMs into one 27128 EPROM.

STEP 1: READ FIRST SOURCE EPROM

Begin as usual by pressing <**E**> and selecting the correct EPROM type. Then, insert the first source EPROM (one of the 2764's) into the ZIF socket and press <**R**> to read the EPROM into the memory buffer.

Note: when an EPROM is read into the memory buffer, the number of buffer addresses occupied by that EPROM is equal to the capacity of the EPROM. Thus, since the capacity of the 2764 is 1FFF, it will occupy all the buffer addresses from 0000 to 1FFF. (The capacities for each 27 series EPROM are listed in the appendix.)

STEP 2: READ SECOND SOURCE EPROM

For the next step, remove the first source EPROM from the ZIF socket and insert the second source EPROM. **DO NOT PRESS <R>**. If you press <**R**> you will erase the memory buffer. Instead, press <**1**>, "Read (A)----in buffer any address". The screen will read:

Result: Buffer starting address (4 digits) = HEX

Since data from the first source EPROM occupies every memory buffer address from 0000 to 1FFF, you cannot use any of those addresses as a starting address without erasing data. Therefore, enter the number 2000, the next number after 1FFF in hexadecimal. The screen will now read:

Buffer last address (4 digits) = HEX

You must now enter the number 3FFF. (This number is derived from simple hex arithmetic: since you began at address 2000, and added the 1FFF addresses of the second EPROM, the total number of occupied buffer addresses is now 2000 + 1FFF = 3FFF.) After you enter 3FFF, the screen should read:

EPROM start address (4 digits) = HEX

Enter 0000. The screen should then display a "Read ok" message.

STEP 3: COPY BUFFER ONTO TARGET EPROM

The final step is to copy the data from the two source EPROMs onto the target EPROM. To do this, remove the second 2764 EPROM from the ZIF socket. Then, press <**E**> to select the correct setting for the 27128 EPROM. Finally, insert the 27128 into the socket and press <**4**>. The buffer, which contains the data from the two 2764 EPROMs, will be copied onto the 27128.

COPY ONE EPROM ONTO TWO EPROMS

Using a second concrete example, we will divide the contents of a 27256 EPROM and copy the divided segments onto two 27128 EPROMs.

STEP 1: READ SOURCE EPROM INTO BUFFER

First, press <**E**> to select the setting for a 27256 EPROM and insert the 27256 into the ZIF socket. Then, press <**R**> to read the EPROM into the buffer. When you have done so, every memory buffer address from 0000 to 7FFF will be occupied.

STEP 2: COPY 1st HALF OF BUFFER

Now, remove the 27256 from the socket and press <**E**> to select the 27128 setting. Insert the first target EPROM (27128). Then, press <**1**> to copy onto the target EPROM. The target EPROM will then be filled to its capacity of 3FFF.

STEP 3: COPY 2nd HALF OF BUFFER

Remove the first target EPROM from the socket, and insert the second of the 27128's. Press <**3**>. The screen will read:

Buffer starting address (4 digits) = HEX

Enter the number 4000, which is the next hex number after 3FFF. The screen will then request the last address of the buffer to be copied. Enter the number 7FFF, which corresponds to the last address of the source EPROM.

After the screen displays a request for the starting address of the target EPROM, enter the number 0000. The second half of of the source EPROM will now be copied onto the second target EPROM.

MANIPULATING THE MEMORY BUFFER

A complete description of the memory buffer manipulation commands is given in the next chapter (refer to <**D**> in chapter 5).

5 REFERENCE GUIDE

This chapter contains brief summaries of each menu commands. Each command is listed in the same order in which it is displayed on the menu.

<E>: EPROM type / Vpp---(2764 / 21V)

Press <**E**> to select the EPROM type and programming voltage that you need. By doing so, you will enter the following eprom selection sub menu:

```
Select EPROM type & programming voltage (Vpp):
<1>: 2716----Vpp = 25V
<2>: 2732----Vpp = 25V
<3>: 2732A---Vpp = 21V
<4>: 2764----Vpp = 21V
<5>: 2764A---Vpp = 12.5V
<6>: 27128---Vpp = 21V
<7>: 27128A--Vpp = 12.5V
<8>: 27256---Vpp = 21V
<9>: 27256---Vpp = 12.5V
<A>: 27512---Vpp = 21V
<B>: 27512---Vpp = 12.5

Which one?
```

To select an EPROM type and voltage, enter **1-9**, **A** or **B**. As soon as you enter your choice, you will be returned to the main menu. To exit the EPROM selection submenu, you must select an EPROM type (**1-9**, or **A-B**).

<Q>: QUIT

<Q> is the exit command. You may exit EPROM Writer from the main menu only. To exit some of the submenus, you may have to press a key other than <Q>.

<L>: LOAD DISK OBJECT FILE IN BUFFER

<L> is the command key to press when you wish to transfer a file from a disk to the memory buffer. As soon as the disk file has been successfully read into the buffer, the cursor will return to its place at the top of the main menu, allowing you to make another menu selection. (The disk file must be in hex format.)

If you wish to exit the <L> submenu without reading the disk file into the buffer, press the ENTER, or RETURN, key. The submenu will persist onscreen, but the cursor will return to the main menu.

<S>: SAVE BUFFER ON DISK.

Press <S> when you wish to save the contents of memory buffer to disk. It requires you to know the byte size of the buffer contents, in hexadecimal.

<D>: BUFFER COMMANDS

Press <D> to enter into the memory buffer manipulation submenu. This submenu will allow you to examine, print, or manipulate the contents of the memory buffer. The <D> submenu appears onscreen as follows:

Display command:
D<start address>.<end address>
ex: D0100.02ff
press <space bar> to stop display command.
Print command:
P<start address>.<end address>
press <space bar> to stop print command.
Substitute command:
S<start address>
press <CR> to advance to the next byte.
press <Q> <CR> to stop substitute command.
Check sum command:
C<start address>.<end address>
Fill command:
F<start address>.<end address>
Move command:
M<start address>.<end address>
Quit command:
press <Q> <CR> to return to main menu.

DISPLAY AND PRINT COMMANDS

The display command and the print commands are fairly straightforward: to display or print a segment of buffer memory, simply enter **D** or **P** (for display or print, respectively) along with the beginning address of the desired buffer segment and the end address of that segment. Separate the beginning and end addresses with a period. For example, if you want to print the contents of memory from address 0000 to address 0100, enter p0000.0100.

SUBSTITUTE COMMAND

To use the Substitute Command, enter **S** along with the address of the buffer at which you wish to start. For example, enter s0000 if you wish to begin at the very beginning of the memory buffer. After you have entered the address, the screen will display:

```
-s0000
0000:XX-
```


XX is the current content of the buffer at the location 0000. To substitute for XX, enter the new number, for example, 55. The screen will now read:

```
-s0000
0000:XX-55
0001:XX-
```

Notice that you advance to the next byte after each substitution. When you have finished substituting, press <**Q**>, then press ENTER.

```
-s0000
0000:XX-55
0001:XX-q
```

This will return you to the <**D**> submenu.

CHECK SUM COMMAND

To check the sum of the buffer memory, enter **C** along with the starting and ending addresses of that segment. For example, if you entered a command to check the sum of the buffer from 0000 to 0100, the check sum would then appear onscreen underneath the command line:

```
c0000.0100
--check sum = XXXX
```

Upon completing the check sum, the program returns to the <**D**> submenu.

FILL COMMAND

The fill command is used to program the vacant addresses of the memory buffer. To use this command, simply enter **F**, along with the beginning and ending buffer addresses (separate these with a period). For example, enter:

f0000.0100

The screen will then ask you for the number to be inserted into those memory addresses. Enter the number. The number will then be inserted in every memory address between the beginning and ending addresses. When the fill has been completed, the program returns to the <**D**> submenu.

MOVE COMMAND

The "move" command is somewhat similar to the cut and paste commands of word processing programs. It allows you to move the contents in one area of the memory buffer to another area of the buffer, leaving the former addresses empty.

To move a block of buffer contents to a new address, enter **M** along with the beginning and ending addresses of the block to be moved. For example:

m0000.0100

After you enter the block of memory to be moved, the screen will read:

—destination address =

Now, enter the address to which you wish to move the block. The block will then be moved to the new address. At the conclusion of the move, the program returns to the <**D**> submenu.

To exit the <**D**> submenu, press <**Q**>. You will be returned to the main menu. If you are already in one of the memory manipulation commands (display, substitute, etc.), you must first exit them and then exit the <**D**> submenu.

<B>: Blank check.

To determine whether a chip is blank or defective, select <**B**>. The programmer will then automatically check the chip. If the chip is blank, the screen displays this message:

```
Blank #1——0k——1XXXX
```

If the chip is defective or is already programmed, this message will be displayed:

```
Blank # 1——ERROR AT XXXX
```

<R>: Read----------in buffer address 0000.

Press <**R**> if you wish to read the entire content of the source EPROM into the buffer, or if you want to know the amount (sum) of data contained in the EPROM. After you press R, the screen will display the following.

```
Result: Read # 1——ok—— 0XXX
EPROM CHECK SUM = XXXX
```


<V>: Verify--------with buffer address 0000.

Press <**V**> to verify the target EPROM's contents with the contents of the buffer.

<C>: Copy--------from buffer address 0000.

Press <**C**> to copy the entire contents of the memory buffer onto an EPROM.

<1>: Read (A)-------in buffer any address.

Press <**1**> to specify the buffer address to which you desire to transfer data. This command is useful in such operations as combining smaller EPROMs into one large EPROM.

<2>: Verify (A)-----with buffer any address.

Press <**2**> to verify that the content of a target EPROM matches a specified block of data contained in the memory buffer. Use this function after copying a large EPROM onto two smaller ones.

<3>: Copy (A)-----from buffer any address.

Press <3> if you wish to copy a specific block of buffer addresses onto a target EPROM. This function is useful for copying a larger EPROM's contents onto two smaller EPROMs.

<4>: Copy (B)-----blank check and copy.

This function is a copy function. Before copying onto the target EPROM, however, it checks the target EPROM to determine whether the EPROM is blank or is already pro-grammed. If the EPROM is already programmed (or is defective), a "Blank Error" message will be displayed.

<5>: Verify (B)----verify & display.

Press <5> to find error locations

APPENDIX: EPROM TYPE & VOLTAGE

ADVANCED MICRO DEVICES

TYPE	VOLTAGE
2716	Vpp=25V
2732	Vpp=25V
2732A	Vpp=21V
2764	Vpp=21V
2764A	Vpp=12.5V
27128	Vpp=21V
27128A	Vpp=12.5V
27256	Vpp=12.5V
27512	Vpp=12.5V

FUJITSU

TYPE	VOLTAGE
2732A-3	Vpp=21V
2732A	Vpp=21V
MBM2764-20	Vpp=21V
MBM2764-25	Vpp=21V
MBM2764-30	Vpp=21V
MBM27128-20	Vpp=21V
MBM27128-25	Vpp=21V
MBM27128-30	Vpp=21V
MBM27256-25	Vpp=12.5V
MBM27256-30	Vpp=12.5V

HITACHI

TYPE	VOLTAGE
HN462716	Vpp=25V
HN462732	Vpp=25V
HN462732P	Vpp=25V
HN482732A	Vpp=21V
HN482764G-2	Vpp=21V
HN482764G	Vpp=21V
HN482764G-3	Vpp=21V
HN482764P	Vpp=21V
HN482764P-3	Vpp=21V
HN4827128G-25	Vpp=21V
HN4827128G-30	Vpp=21V
HN4827128G-45	Vpp=21V
HN4827128P-30	Vpp=12.5V
HN4827128AG-2	Vpp=12.5V
HN27128AG-30	Vpp=12.5V
HN27256G-25	Vpp=12.5V
HN27256G-30	Vpp=12.5V
HN27512G-25	Vpp=12.5V
HN27512G-30	

INTEL

TYPE	VOLTAGE
2716	Vpp=25V
2732	Vpp=25V
2732A	Vpp=21V
2764A-2	Vpp=21V
2764A	Vpp=21V
2764A-3	Vpp=21V
P2764A	Vpp=21V
P2764A-3	Vpp=12.5V
27128	Vpp=12.5V
27128-3	Vpp=21V
27128-4	Vpp=21V
27128A-2	Vpp=21V
27128A-3	Vpp=12.5V
27256	Vpp=12.5V
27256-3	Vpp=12.5V
27512	Vpp=12.5V
27512-3	Vpp=12.5V
	Vpp=12.5V

MITSUBISHI

TYPE	VOLTAGE
2716	Vpp=25V
2732	Vpp=25V
2732A	Vpp=21V
M5L2764K-2	Vpp=21V
M562764K	Vpp=21V
M5L2764K-3	Vpp=21V
M562764P-3	Vpp=21V
M5L27256K	Vpp=12.5V

NATIONAL SEMICONDUCTOR

TYPE	VOLTAGE
2716	Vpp=25V
2732	Vpp=25V
2764	Vpp=21V
2764H	Vpp=21V

NEC

TYPE	VOLTAGE
2716	Vpp=25V
2732	Vpp=25V
2732A	Vpp=21V
D2764D-2	Vpp=21V
D2764D	Vpp=21V
D2764D-3	Vpp=21V
D27128D	Vpp=21V
D27128D-3	Vpp=21V
D27128D-4	Vpp=21V
D27128C-3	Vpp=21V
D27256D	Vpp=21V
D27256D-3	Vpp=21V

TOSHIBA

TYPE	VOLTAGE
2716	Vpp=25V
2732	Vpp=25V
2732A	Vpp=21V
TMM2764D-2	Vpp=21V
TMM2764D	Vpp=21V
TMM27128D-20	Vpp=21V
TMM27128D-25	Vpp=21V
TC57256D-20	Vpp=21V
TC57256D-25	Vpp=21V

MCT-EPROM