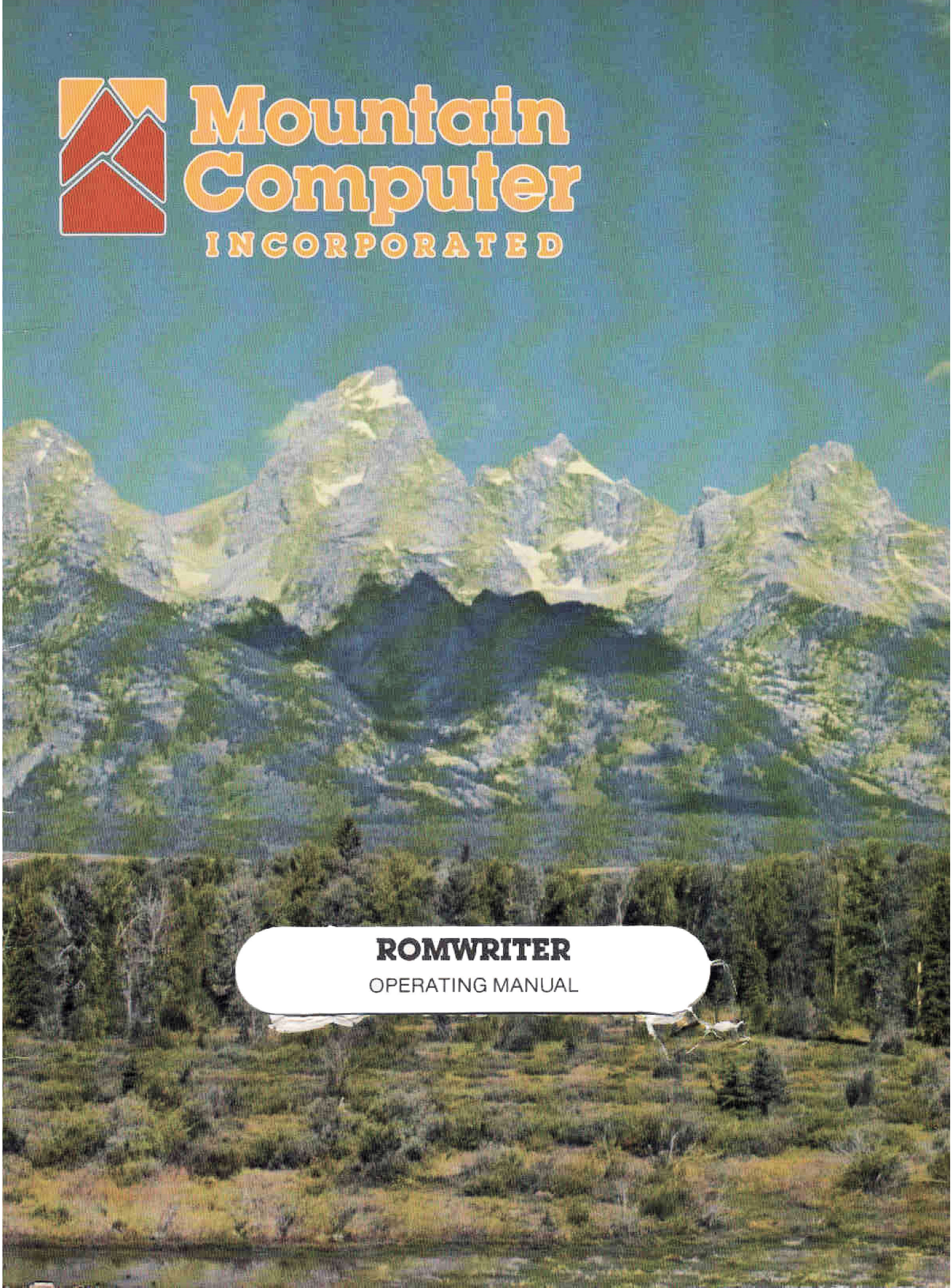

ROMWRITER

OPERATING MANUAL

ROMWRITER

OPERATING MANUAL

Manual Number 11-00236-01A

Table of Contents

INTRODUCTION

Mountain Computer's ROMWRITER™ allows you to program the 2K byte 2716* EPROM with your Apple II** Computer. The EPROMs must be programmed with machine language instructions. BASIC programs will not execute from an EPROM. The programmed EPROMs may be used directly in Mountain Computer's ROMPLUS+ or in other applications.

ROMWRITER was designed to give the user flexibility in the programming and testing of an EPROM. The user may program information into all 2Ø48 bytes of a 2716 EPROM just as easily as programming only 20 bytes. ROMWRITER will also allow you to read the contents of the EPROM, so the EPROM does not need to be moved elsewhere for testing. The EPROM resides in the common peripheral address space, $CØØ to $CFFF. It is also possible to read the contents of a 2316 ROM which is pin for pin compatible with the 2716 EPROM.

ROMWRITER comes complete with software supplied on diskette for easy EPROM programming. The user simply specifies starting and ending addresses in the EPROM and a disk file or location in memory of the data to be burned into the EPROM. All data burned into the EPROM is automatically verified by the software.

The ROMWRITER board itself features a zero-insertion force socket for the EPROM. An on-board switch will turn all power to the EPROM off which allows you to insert or remove the EPROM without turning the Apple off. A Write Protect switch prevents accidental programming of the EPROM. Another switch disables the Apple peripheral convention and allows programming the location $CFFF.

This manual is written for the user who is familiar with the Apple and Applesoft. The manual covers the operation of the switches and EPROM socket. A tutorial illustrates the use of the PROMBURNER software.

---

*Intel compatible single power supply (5 volt) 2716 EPROM

**Apple II is a trademark of Apple Computer Inc., Cupertino, CA.

ROMWRITER INSTALLATION

ROMWRITER may be used in any peripheral slot of the Apple, except Slot #Ø. You might want to position ROMWRITER so that you can reach the EPROM while the board is mounted in the slot. Follow these steps to install ROMWRITER in your Apple.

1. Turn your Apple OFF. Never insert or remove any board while the power is on.

2. Remove the cover from the Apple.

3. Carefully insert ROMWRITER into any peripheral slot except slot number zero. Make sure the board is evenly and firmly seated.

4. When you are programming with ROMWRITER, you will leave the cover off so that you may access the switches and the EPROM chip.

EPROM INSTALLATION

Now, assuming that your ROMWRITER is installed in the Apple, there is a certain procedure for inserting and removing the 2716 EPROM's on the ROMWRITER. Observe the zero-insertion force socket on the lower left-hand corner of the ROMWRITER board. Notice there is a lever on the lower left corner of the ZIF socket. This lever actuates a cam in the socket which opens or closes the contacts for each of the 24 pins. (See Figure 1). It is almost impossible to bend a pin when using this type of socket.

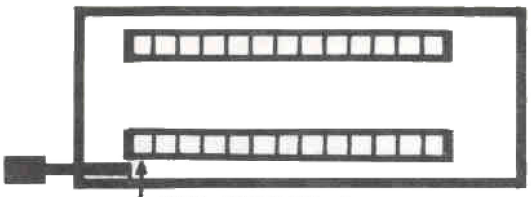

FIGURE 1

When the lever is up (perpendicular to the ROMWRITER board), the EPROM may be easily inserted or removed. When the lever is down (parallel with the board) the EPROM is firmly gripped in the socket. Never remove or insert an EPROM when the lever is down, as you will damage the EPROM. At all times it is important to observe the orientation of the EPROM and pin 1 of the socket. See Figure 1 for the location of pin 1 on the socket. You can destroy an EPROM very quickly by inserting it backwards, so be careful.

Assuming that ROMWRITER is installed and the Apple is on, follow these steps to install an EPROM:

1. The PROM PWR switch (in the upper left hand corner of the board) should be turned OFF by flipping the switch upward.

2. The ZIF socket should be opened, that is, the lever should be up and perpendicular with the board.

3. Observe the location of pin 1 on the 2716 EPROM, and match it with pin 1 of the ZIF socket. Carefully insert the EPROM. If you experience any resistance while inserting the 2716, check that the socket is open, and check for bent pins on the 2716.

4. Now hold the 2716 in place with one hand while closing the ZIF socket. Move the lever to the down position (parallel to the board).

5. The PROM PWR switch is now flipped downward, which turns on the power to the EPROM.

Removing the 2716 is almost the same. First, PROM PWR must be turned OFF. You must hold the 2716 while moving the ZIF socket lever up. Otherwise the 2716 will fall out of the socket and onto the mother board. This may cause damage to the 2716 or the computer, so be careful.

EPROM ERASURE

The 2716 EPROM may be used over and over again. After the EPROM has been programmed, it must be erased before it may be programmed again. The 2716 EPROM will begin to erase when exposed to light of 4000 Angstroms or less. If a 2716 is left in the sunlight, it will take about 1 week to erase the chip. Normal fluorescent lights will erase the 2716 in about three years. This is why you should place a paper sticker over the window if you don't want the EPROM erased.

We recommend a high intensity ultra-violet lamp for erasing the 2716. There are several lamps on the market that work fine. The lamp should have an emission around 2537 Angstroms. Ultra-Violet Products, Inc. manufacture lamps which may be purchased at a computer store or a rock and mineral store. When using one of these lamps, place the 2716 one inch from the bulb. (If there is a sticker on the window, remove it first.) It takes about 20 minutes to erase the 2716. An erased 2716 will have $FF at every byte.

ROMWRITER SWITCHES

The ROMWRITER BOARD has three switches which you will need to set in order to select options on the ROMWRITER. The switches are located in the upper left-hand corner of the board. Two of the switches are on a four position DIP switch. The switches on the left and right edges of the DIP switch are used. The two center switches on the DIP switches are not used and perform no function.

PROM POWER SWITCH

A two position toggle switch (PROM PWR) is in the upper left-hand corner of the board. This switch controls the power applied to the EPROM socket. When this switch is in the "OFF" position, the 25 volt and 5 volt supplies are removed from the ZIF (zero insertion force) socket, making it impossible to program or read an EPROM. You should turn the PROM PWR SWITCH "OFF" before you insert or remove an EPROM from the ZIF socket. After you have inserted an EPROM into the socket, you should turn this switch "ON" so that you may read or write to the EPROM.

CAUTION - Make sure the PROM PWR Switch is OFF before removing or inserting an EPROM. If the switch is not OFF, you could destroy the EPROM.

WRITE PROTECT SWITCH

This switch (WR PRT) is located on the left edge of the four position DIP switch. When this switch is in the "up" position, the write protect feature is on. When the write protect feature is on, it will not be possible to write to the EPROM. When the switch is "down", it will be possible to write to the EPROM. In normal operation, the write protect switch should always be "up", except when you wish to burn information into the EPROM. In this way, you will not accidentally program your EPROM.

CFFF OFF SWITCH

This switch (CFFF) is located on the right edge of the four position DIP switch. To understand the use of this switch, you need to understand Apple's peripheral conventions. Apple has set aside the address space from $C8ØØ to $CFFF as an address space common to all peripherals. Apple's convention states that whenever the address $CFFF is referenced, all peripherals that might be active at that time must deactivate themselves.

Now suppose that you wish to program all 2Ø48 bytes of your EPROM. The EPROM resides in the address space $C8ØØ to $CFFF (which is 2Ø48 bytes). As the software references the addresses and presents the data to ROMWRITER, the information is burned in the EPROM. But what happens when the software tries to program the last byte? The software references address $CFFF and the ROMWRITER board obeys the Apple peripheral convention and turns itself off. This is where the CFFF OFF switch comes in. When this switch is in the up position (CFFF OFF), the ROMWRITER board will not follow the peripheral convention. When this switch is up, you may read from or write to address $CFFF without deactivating ROMWRITER. When this switch is down, the ROMWRITER will respond to the $CFFF reference in the normal fashion. Regardless of what position the CFFF OFF switch is in, a system RESET will deactivate the ROMWRITER.

There is one main program supplied with ROMWRITER. This program, named PROMBURNER on the diskette, controls the ROMWRITER board and provides a very easy way to burn data into an EPROM. The program features self-explaining prompts for the various parameters, field editing abilities, the saving or loading of data to or from the disk, and EPROM burn verification. The program will even warn you if the EPROM has not been erased.

FIELD EDITING COMMANDS

All data input to PROMBURNER is either a single character or a field of characters. There are a few prompts in the program which require only a single character as a response. These prompts will specify the legal response characters. If the character entered has an illegal value, the bell will sound. Please note that pressing the CTRL-C character at any time will terminate the program.

Other prompts may require more than one character to be input. These are called fields. A field will accept characters up to a maximum number determined by that field. For example, a hexadecimal address field will not accept more than four characters. The maximum associated with a field for a filename is 30 characters.

There are a few commands you may use while entering fields. The first of these commands is the CTRL-A command. Pressing the CTRL-A key will cause the entire field as it appears on the screen to be entered. It does not matter where the cursor is positioned when CTRL-A is pressed; the whole field is entered as is.

The next command is the CTRL-B command. Pressing the CTRL-B key causes the cursor to be positioned at the beginning of the previous field. If you find you entered something incorrectly at a previous field, you may press CTRL-B (more than once, if necessary) to return to the field in error.

The CTRL-D command will delete the character that the cursor is on, and move every character to the right of the cursor left one position. The cursor does not move, and one character is deleted each time the CTRL-D key is pressed.

The CTRL-N command will insert a space at the cursor location, and move the cursor and any characters to the right of the cursor right one location. Every time CTRL-N is pressed, a space is inserted.

The CTRL-X will delete the entire line and will place the cursor back at the beginning of the field. You would use this to correct gross mistakes. Pressing the left-arrow key will backspace the cursor, allowing you to correct a few characters at a time.

Suppose you are correcting a previously entered field, and half way through the correction, you decide that you would really like the previous field displayed on the screen again. Use the left-arrow key to move the cursor to the beginning of the field. Now press the RETURN key and observe the previous field return to the screen.

TUTORIAL ONE

In these tutorials, you will start by burning a few bytes into a clean EPROM, and work up to loading a file from disk and burning it into the EPROM. The examples are easy and you should follow along on your own computer.

Start by inserting the diskette and booting the system. Let's assume that ROMWRITER is installed in slot #3. Also assume that a fresh, empty EPROM has been inserted into the zero insertion force socket. See the instructions for use of the zero insertion force socket in the EPROM Installation section. After you have booted the diskette, type the command RUN PROMBURNER. The program will print a heading, and then print this prompt on the screen:

ENTER EPROM PROGRAMMER SLOT #(1-7)?

This prompt requires a single character; a number between 1 and 7. Try typing a letter or two and observe the result. Then type the character "3", which is the slot number of ROMWRITER. The next prompt that appears is:

ENTER EPROM STARTING ADDR. IN HEX?

This prompt requires a field with four hexadecimal characters. Legal characters are Ø through 9 and A through F. Illegal characters will cause the bell to sound. Enter a few illegal characters and observe this. The hexadecimal address entered in this field is the address of the first byte to be programmed. The hex address must be greater than or equal to C8ØØ and less than or equal to CFFF. Illegal values will generate an error message on the screen. After entering the value C8ØØ, the next prompt appears on the screen:

ENTER EPROM ENDING ADDR. IN HEX?

Again, you are asked to supply a four character hexadecimal address. This address is the address of the last byte to be burned on the EPROM. The value supplied here must be equal to or greater than the starting address and less than or equal to CFFF. Since in this first example you wish to burn only the first sixteen bytes, enter the hex address C8ØF. The next prompt on the screen will appear:

DO YOU WANT TO LOAD A PROGRAM (Y OR N)?

The correct character would be either "Y" or "N". Since the data to be burned is not coming from the disk in this example, enter an "N".

Let's regress for a moment and talk about the disk file and how it is used with the PROMBURNER program. One of the features of PROMBURNER is the ability to burn an EPROM with data from a disk file. Suppose that your assembler outputs a file which contains machine code. At this point in the PROMBURNER, you could indicate that you want to load that file. You could also create your own files by using the BSAVE command. Any binary file may be loaded for EPROM burning. You will probably use the disk files for most of your work with ROMWRITER.

Now, back to our example. The next prompt will appear:

ENTER PROG. STARTING ADDR. IN HEX?

This prompt expects a four character hexadecimal address. This address corresponds to the first byte of an area in memory which contains the data you wish to burn into the EPROM. The data must already be in that area of memory. The data should be located in an area in memory that does not conflict with the program or DOS. In a 48K Apple, we recommend the area from $6000 to $8000. Data to be burned should be arranged so that the first data byte is at STARTING ADDR., and the next data byte is at location STARTING ADDR+1, and so on. The number of bytes read from this data area is (EPROM ENDING ADDR) - (EPROM STARTING ADDR) + 1. If you were loading the data from a disk file, the number of bytes read from the file is determined in the same way. With the disk file, the length of the binary file may be of any size. Notice that the PROMBURNER program will always start at the beginning of the binary file. If, for example, the file was 4096 bytes long, PROMBURNER would never read the last half of the file. Again, for example, assume that the file is 256 bytes in length and you are programming 32 bytes on your EPROM. The PROMBURNER program will use the first 32 bytes in the disk file, regardless of where the 32 bytes are in the EPROM. So back to our example, let's enter the hexadecimal address 6000. If there is any error in the address you enter, an error message will appear. Since you type in 6000, the first sixteen bytes starting at address 6000 will be burned into the EPROM. The next prompt to appear on the screen is:

ARE THE ADDRESSES CORRECT (Y OR N)?

This is the last prompt to appear before the program attempts to burn the EPROM. It gives you a chance to verify that the information you have entered is correct, and that it is okay to go ahead and burn the EPROM. If you have made a mistake, type "N" and the cursor will return to the first field. Suppose the mistake was in the third field. After you press "N" and the cursor is at the first field, you press CTRL-A twice so that the first two (correct) fields are re-entered into the program. You will again have to respond to the LOAD PROGRAM prompt by pressing "N", and at last you may correct your error in the third field.

If the EPROM is erased, and everything is set (the switches are correct) then this message appears:

PROGRAMMING EPROM....

If everything goes well, then this message will appear:

EPROM HAS BEEN PROGRAMMED AND VERIFIED.

If you had left the write protect switch ON (up) or if you had left the PROM PWR switch OFF (up), you would see this message:

BAD BURN OF EPROM!

DO YOU WANT TO TRY AGAIN (Y OR N)?

You must correct the condition before you type a "Y". As soon as you type "Y", the program will try to burn the EPROM again. Another problem you might encounter would be an EPROM which has not been erased. If any of the bytes within the range specified by the EPROM starting address and the EPROM ending address are not equal to $FF, then the EPROM is determined to be not erased. If that is the case, you are offered a chance to save the whole EPROM plus the changes. The changes are the data that you were going to burn into the EPROM. The error message and the next prompt will look like this:

EPROM ADDRESSES SELECTED AREN'T ERASED!

SAVE WHOLE EPROM WITH CHANGES (Y-N)?

At this point, you could respond to the prompt with an "N" and the program will then prompt you to remove the EPROM and erase it. Be sure to turn the PROM PWR switch to OFF before removing the EPROM. Once you have erased your EPROM, re-insert it, switch PROM PWR to ON and respond to the prompt on the bottom of the screen with a CTRL-P. You could then continue with the EPROM burning. This procedure is useful if you placed a wrong EPROM into ROMWRITER. It will give you a chance to remove the wrong EPROM and replace it with an erased EPROM.

For this tutorial, respond to the SAVE WHOLE EPROM WITH CHANGES prompt with a "Y". You are then offered a chance to save the same data on a disk file. This has been included as a safety factor or in case you wish to do something else while your EPROM is erasing. The program will prompt you for a filename if you save the data on disk. The file saved will be 2Ø48 bytes long and will be composed of an "image" to be burned into the EPROM. This image is made up of the EPROM data which is not in the range specified by the EPROM starting address and the EPROM ending address plus the data you were going to burn. Suppose you had specified the EPROM starting address as $C9ØØ and the EPROM ending address as $C9FF. You specified your data as coming from $6ØØØ. See Figure 2 for the contents of the data file saved on the disk.

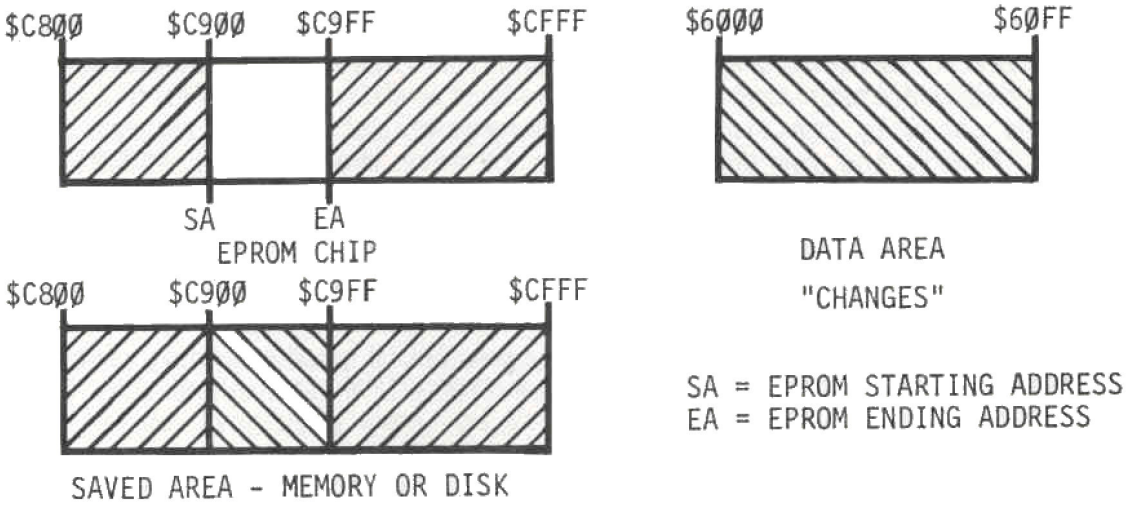

FIGURE 2

The shaded area of the EPROM area are combined with the changes to produce the saved area, which is located in memory or on disk. The blank area of the EPROM in Figure 2 is replaced with the data you specified.

After you have saved the whole EPROM with changes, you will see this prompt on the screen:

REMOVE EPROM-ERASE IT AND REINSERT IT!

HIT (CTRL) P TO CONTINUE.

At this point, switch the PROM PWR switch to OFF, remove the EPROM and erase it. When the EPROM has been erased and reinserted, press the CTRL-P key and continue.

TUTORIAL TWO

ROMWRITER may be used for testing software which will eventually reside in a 256 byte ROM. To facilitate this, you may access the first 256 bytes of the EPROM by referencing addresses $CNØØ to $CNFF, where N is the slot which contains ROMWRITER. The ROMWRITER board will translate any memory references in the $CNØØ to $CNFF range into the first 256 bytes of the EPROM. Therefore a read from $CNØØ-$CNFF will produce the exact same information as a read from $C8ØØ-C8FF.

Start with an erased EPROM in the ROMWRITER. Prepare an area in memory, say $6ØØØ to $6ØFF with some data that you will recognize. Boot the diskette and RUN PROMBURNER. Respond to the prompts, and set the EPROM STARTING address to C8ØØ, the EPROM ENDING address to C8FF and the PROG. STARTING address to 6ØØØ. If all of the addresses are correct, go ahead and burn the EPROM. If this memory area actually contained a program, you could run the program from BASIC by typing "PR#3", and you would not have to move the EPROM or ROMWRITER. Be aware, however, that if that program requires a piece of hardware which is not present, running the program will have unpredictable results. As you can see from this example, an access to addresses $C3ØØ-$C3FF will be the same as accessing the addresses $C8ØØ-$C8FF.

Note that if ROMWRITER is in slot three, you could produce the exact same results as above if you had specified the EPROM STARTING address as C3ØØ and the EPROM ENDING address as C3FF.

TUTORIAL THREE

In this final tutorial, you will learn how to patch a few bytes in an existing EPROM. Suppose that the EPROM contains some data , and you wish to fix an error in the EPROM. The program fix will involve "patching" a few bytes in the EPROM. In this example, suppose the patch is twenty bytes long and is to start at address $C933 in the EPROM. You want to replace those twenty bytes on the EPROM with twenty bytes located in memory. If those twenty bytes in the EPROM were equal to $FF (i.e., that area is erased), you would simply specify the starting and ending addresses and burn away. However, for this example, we will assume that the EPROM is not erased.

First, find a free area in memory and place your patch (the new code) into that area. (We recommend $6ØØØ to $8ØØØ in a 48K Apple system with DOS). Now boot the diskette and RUN PROMBURNER. Respond to the prompts, and set the EPROM STARTING ADDR to C933 and the EPROM ENDING ADDR to C946. This is an area 2Ø bytes long, and represents where the patch is going. Since the data is in memory, respond to the LOAD PROGRAM prompt with a "N". Set the PROG STARTING ADDR to 6ØØØ. If the addresses are correct, go ahead and burn the EPROM. Immediately you will see a message saying that the EPROM is not erased, and you are offered a chance to save the EPROM and changes. Respond with a "Y" and now you will see this prompt:

ALSO SAVE ONTO DISK (Y OR N)?

Type in a "Y" and observe this prompt:

ENTER PROG NAME?

You now enter a filename no greater than thirty (3Ø) characters long. After the file has been written onto the diskette, you may turn the computer off and later load the same disk file into memory. So save the program, turn off the computer and erase your EPROM. After the EPROM is erased, reinsert it into ROMWRITER, turn on the computer and boot the diskette. Now RUN PROMBURNER and enter the starting and ending addresses.

When you are burning the EPROM with data saved in a file, you must specify the EPROM STARTING address as C8ØØ and the EPROM ENDING address as CFFF. (The CFFF OFF switch must be UP). The reason for this is because the file saved on the disk is 2Ø48 bytes long and the file contains the old EPROM contents with the new patch. Since this file is an "image" of the corrected EPROM, and the EPROM is now erased, you must burn all 2Ø48 bytes of the EPROM. Type a "Y" in response to the LOAD PROGRAM prompt. The program will now prompt you for a filename. Enter the filename you used when you saved the data. The program will load it into memory and proceed to ask you if the addresses are correct. The program knows where to find the data that was loaded from the disk and therefore will not ask for the PROG STARTING ADDR. Now that the addresses are correct, go ahead and burn your EPROM.

TUTORIAL FOUR

This tutorial will allow you to duplicate a ROM or EPROM. To perform the duplication process execute the following:

1. Place the ROMWRITER PROM PWR switch to the OFF (up) position.

2. Install the master ROM/EPROM in the U12 socket.

CAUTION

The master IC must be installed correctly (refer to Figure 1 for proper orientation) or the device will be damaged when power is applied.

3. Lock the master ROM/EPROM into the socket (position the lever in the down position).

4. Apply power to the socket by placing the PROM PWR switch to the ON position.

5. If the last byte of memory on the master is to be copied, place the CFFF switch (end dip switch away from keyboard) to the OFF (up) position.

6. Boot the ROMWRITER diskette and when the screen prompt appears, type the appropriate Peek Command (refer to Table 1.) This will transfer the master device data to the C800 address block in the Apple II memory.

Table 1 PEEK COMMAND

ROMWRITER I/O Slot Position	Hex Address	Integer BASIC	Applesoft BASIC
1	C100	-16128	49408
2	C200	-15872	49664
3	C300	-15616	49920
4	C400	-15360	50176
5	C500	-15104	50432
6	C600	-14848	50688
7	C700	-14592	50944

7. Transfer the master device data from Apple Main memory (location C800 to CFFF HEX) to a diskette file by typing:

BSAVE <FILENAME>, A$C800, L$800

8. Turn the PROM PWR switch to the OFF position and remove the master ROM/EPROM.

9. Install and lock the blank ROM/EPROM into the socket.

10. Place the WR PRT switch (dip switch closest to keyboard) to the OFF (down) position.

11. Run the PROMBURNER program (tutorial 1) and use the filename of the file you saved in step 7.

NOTE

The program burn will take about 3-minutes to complete for 2048 bytes.

DIRECT INTERFACE

There are only four commands for the ROMWRITER board. These commands are used to activate or deactivate the ROMWRITER board and to place the ROMWRITER into the program or read mode.

Whenever the RESET key is pressed, the ROMWRITER is deactivated and placed into the read mode. After the computer is turned on, the ROMWRITER will be in the read mode. To activate the board, you must reference the address $CNØØ, where N is equal to the slot number that ROMWRITER occupies. For example, if ROMWRITER is in slot number three, and you are in the monitor (* prompt), you may simply type "C3ØØ" and a return. The reference to that address will activate the board. From machine language, the instruction "LDA $C3ØØ" will activate ROMWRITER. From Applesoft, the command "POKE 4992Ø,Ø" will activate ROMWRITER. The table below gives a slot number, the hex address and the addresses used in BASIC commands.

SLOT NUMBER	HEX ADDRESS	INTEGER	APPLESOFT
1	C1ØØ	-16128	49408
2	C2ØØ	-15872	49664
3	C3ØØ	-15616	4992Ø
4	C4ØØ	-1536Ø	5Ø176
5	C5ØØ	-151Ø4	5Ø432
6	C6ØØ	-14848	5Ø688
7	C7ØØ	-14592	5Ø944

To deactivate ROMWRITER, press the RESET key, or if the CFFF OFF switch is down, reference the $CFFF address. From the monitor, type "CFFF" and return. From machine language, the instruction "LDA $CFFF" will deactivate ROMWRITER. From Applesoft, use the command "POKE 53247,Ø". Notice that if the CFFF OFF switch is up, the only way to deactivate ROMWRITER is by pressing the RESET key.

There is a Read/Write status latch on the ROMWRITER board. A Write to this latch will control the mode of the ROMWRITER. Whenever the RESET key is struck, the ROMWRITER goes into the read mode. Once in the read mode, if the ROMWRITER is re-enabled, you may read from the EPROM or run a program which is in the EPROM. By writing a "zero" into the latch, you place ROMWRITER into the read mode. The address of the status latch is $CØ8Ø+NØ, where N is the slot number. Assuming slot number three, from the monitor type "CØBØ:ØØ" and return. From machine language, load the accumulator with zero and write it out to the latch. The instructions "LDA #$ØØ" and "STA $COBO" will do the trick. From Applesoft, use this command to place ROMWRITER into the read mode: "POKE (4928Ø+(SLOT*16)),Ø". SLOT is a variable which is equal to the slot number which contains ROMWRITER.

By writing a "one" into the latch, ROMWRITER is placed into the programming mode. From the monitor, type "CØBØ:Ø1" and return. In machine language, use these instructions: "LDA #$Ø1" and "STA $CØBØ". From Applesoft, the command "POKE(4928Ø+(SLOT*16)),1" will place ROMWRITER in the programming mode.

A read from the status latch will tell you if ROMWRITER is "ready" or "not ready". When ROMWRITER is in the programming mode, a write to any address in the $C8ØØ to $CFFF range (or the $CNØØ to $CNFF range) will cause the status to go from "ready" to "not ready". When this latch indicates "not ready", a byte is being burned into the EPROM. After the byte has been burned, the status will indicate "ready". When ROMWRITER is "ready", the next byte may be burned into the EPROM. The high-order bit (bit 7) is equal to "Ø" for "ready" and equal to "1" for "not ready". In practice, you will be using the PROMBURNER software, and will have no need to check the status of the board.

Limited Warranty for Mountain Computer Inc. Hardware

Your factory-built Mountain Computer Inc. product is warranted against defects in materials and workmanship for a period of one year from the date of delivery. We will repair or replace products that prove to be defective during the warranty period, provided they are returned to Mountain Computer Inc. No other warranty is expressed or implied. We reserve the right to refuse to repair any product that, in our opinion, has been subjected to abnormal electrical or mechanical abuse. Products less than two years out of warranty will be repaired for a nominal flat fee. Before sending your Mountain Computer Inc. unit in for repair, contact our Customer Service Representative for a Return Authorization Number.

Limited Warranty for Mountain Computer Inc. Software

Computer software programs cannot replace your sound business judgment or make decisions for you. You, therefore, assume complete responsibility for any decisions made or actions taken based on information obtained using Mountain Computer Inc. software programs and instructional materials.

Mountain Computer Inc. software and the attached instructional material are sold "AS IS", without warranty as to their performance. The entire risk as to the quality and performance of the computer software is assumed by you.

However, to the original purchaser only, Mountain Computer Inc. warrants the software diskette or cassette to be free from defects in materials and faulty workmanship under normal use and service for a period of one (1) year from the date of purchase. If, during this one year period, a defect in the diskette or cassette should occur, it may be returned to Mountain Computer Inc. or an authorized Mountain Computer Inc. dealer for replacement of the cassette or diskette without charge to you. Your sole and exclusive remedy in the event of a defect is expressly limited to replacement of the diskette or cassette as provided above. To provide proof that you are the original purchaser, please complete and mail the attached Registration/Warranty Card to Mountain Computer, Inc. within thirty (30) days of the date of purchase.

If the failure of the diskette or cassette, in the judgment of Mountain Computer Inc. resulted from accident, abuse or misapplication of the diskette or cassette, then Mountain Computer Inc. shall have no responsibility to replace the diskette or cassette under the terms of this warranty. In such an event, replacement of the diskette or cassette is available to the original purchaser at a nominal charge.

The above warranties for goods are in lieu of all warranties, express, implied or statutory, including, but not limited to, any implied warranties of merchantability and fitness for a particular purpose, and of any other warranty obligation on the part of Mountain Computer Inc. In no event shall Mountain Computer Inc. or anyone else who has been involved in the creation and production of this product be liable for indirect, special or consequential damages, such as, but not limited to, loss of anticipated profits or benefits resulting from the use of this product, or arising out of any breach of this warranty. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation may not apply to you.

Mountain Computer
INCORPORATED

Located in the Santa Cruz Mountains of Northern California, Mountain Computer, Inc. is a computer peripheral manufacturer dedicated to the production of use-oriented high technology products for the microcomputer. On-going research and development projects are geared to the continual supply of unique, innovative products that are easy to use and highly complementary in a broad variety of applications.