Z-Ram™II
User's Manual

Æ APPLIED ENGINEERING

Z-Ram™ II
User's Manual

v1.0

Any comments or suggestions for improving this manual would be appreciated. Please forward your remarks to:

Applied Engineering
P.O. Box 798
Carrollton, Texas 75006

Attention: Manager. Technical Publications

Applied Engineering
P.O. Box 798
Carrollton, Texas 75006

Sales: (214) 241-6060 9AM - 11 PM 7 days
Technical Support: (214) 241-6069 9AM - 5 PM Monday - Friday

Contents

1

2

3

4

Contents

Preface

About This Manual

This manual is included with the Z-Ram II accessory card for the Apple //c personal computer. It tells you all about the installation and use of your Z-Ram II card. Don't worry! You don't have to be an engineer to install the Z-Ram II and you certainly don't have to be a programmer to take advantage of its many features. This manual assumes you have read your Apple //c Owner's Manual and are familiar with the operation of your computer. For additional sources of information, please refer to Appendix A.

Here's a summary of the material covered in this manual:

Chapter 1   gives a description of Z-Ram II. explains what it can do for you, and provides step by step instructions for the installation of Z-Ram II in your Apple //c.

Chapter 2   explains how the Z-Ram II can be used to enhance the performance of AppleWorks.

Chapter 3   contains the instructions on how to use the RamDrive software, included with the Z-Ram II, to utilize the extended memory of Z-Ram II to emulate solid state disk storage.

Chapter 4   is the technical reference portion of this manual. The explanation of Z-Ram II extended auxiliary memory is intended primarily for heavy-duty programmers. Instructions for installing additional memory chips to Z-Ram II are also provided.

Chapter 5   is the CP/M Operating System reference chapter. The CP/AM 4.08 System Master disk and its utilities are explained. The directions for accessing the public domain software included on the CP/AM 4.08 System Master disk are also included in this chapter.

Appendix A   is a listing of reference manuals and textbooks on topics related to the Apple //c and Z-Ram II.

Appendix B   contains information about CP/M patches to Wordstar.

Appendix C   tells you how to run the Z-Ram Memory Test.

Preface

Applied Engineering
Technical Support

To assure the utmost in customer satisfaction, Applied Engineering has a staff of technicians dedicated to answering specific technical questions about Applied Engineering products and software. Technical support representatives are available between the hours of 9 AM and 5 PM Central time, Monday through Friday. The technical support telephone number is (214) 241-6069.

If you call for technical support, please try to provide as much information as possible about your problem. The support representative will need the following information:

The specific Applied Engineering product related to your question.

The serial number of your Apple //c computer.

The configuration of your computer. (What peripherals are attached?)

The name, version, and publisher of the software package related to your problem.

The results of any testing or troubleshooting done by you or your dealer.

Remember: "When all else fails, *read the instructions!* (...then call technical support.)"

Getting Started

About Z-Ram II Z-Ram II is a combined memory expansion card and co-processor card for the Apple //c personal computer. It installs completely within the I/c. just under the keyboard, and is totally powered by the //c's power supply. Depending on the amount of memory installed on the Z-Ram II card, the 128K of memory available on the//c can be expanded from 384K to over 1 Megabyte. Additional memory, up to 1 Megabyte, can be installed by the user at any time. A high speed Z-80 microprocessor on the Z-Ram filets you run thousands of CP/M programs, including Wordstar, dBASE II, Turbo Pascal, and Microsoft BASIC.

Z-Ram II is completely compatible with all Apple I/c hardware and software. With the Z-Ram II installed, your //c functions exactly the same except that it now has the capability to do more, faster! The expanded memory can be used to enhance application software such as AppleWorks. Magic Office System, Supercalc 3a, Flashcalc, and many others. Additional utility programs from Applied Engineering, included with Z-Ram II, allow the expanded auxiliary memory to be used to emulate high speed solid-state disk drives.

Installation is easy. Two screwdrivers are the only tools required. You do not have to make any permanent modifications to the //c and there is nothing to cut or solder. By following the instructions in this chapter, the entire installation can be done in fifteen to thirty minutes.

System Requirements Simple. All you need is an Apple IIc and a video monitor capable of displaying 80 column text.

> **Important!** Make sure your Apple //c is working properly before you begin the Z-Ram II installation! It would be a good idea to run a couple of your favorite programs, just to verify the Operation of your disk drive, keyboard, and computer.

Before you begin... This chapter tells you everything you need to know about installing Z-Ram II in your Apple //c. Z-Ram II was designed to be installed easily even by someone with limited technical ability, but please consider this before you begin. If you open your Apple IIc case during the Apple 90-Day Warranty period, you may void that warranty. In the unlikely event that you physically damage your //c by improperly installing Z-Ram II, that damage would probably not be covered under warranty or an extended service program, such as "Applecare". This is not intended to shatter your confidence, but if you doubt your ability to install Z-Ram II, we recommend you have it done by a qualified technician. Many dealers will install and test Z-Ram II for a nominal charge and warrant their work. Applied Engineering cannot accept responsibility for damage due to an improper installation.

Chapter 1

What You Need...   Here's all you will need to install your Z-Ram II in your Apple//c:

- A backup copy of the Super AppleWorks Desktop Expander Disk. See your Apple //c Owner's Manual for information on how to make a backup copy.

- The CP/AM 4.0B disk included with your Z-Ram II.

- A small Phillips screwdriver. A #1 Phillips point (3/16" tip) is recommended.

- A flatblade screwdriver. (The tip should be 1/8" to 1/4" wide.)

- A flat, clear workspace with adequate lighting.

Figure 1: Z-Ram II Installation Tool
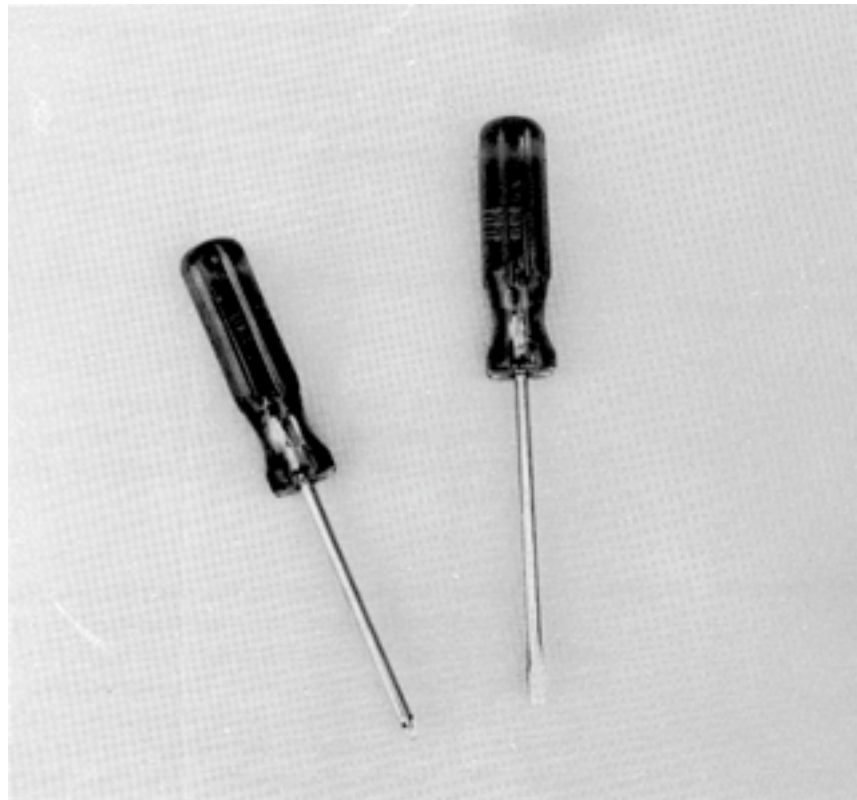
Installation   In this section you will be instructed on how to remove the top cover of the //c (probably the hardest part), how to unplug and lift out the keyboard and disk drive, unplug two "chips" from their sockets on the //c main logic board, and move them to the Z-Ram II board sockets. Then you will be given the procedure for plugging the Z-Ram II into the proper sockets on the //c main logic board, connecting one wire, and replacing the disk drive, keyboard, and the top cover. By carefully following the following step by step instructions and using simple hand tools (2 screwdrivers), you should be able to install your Z-Ram II in about IS to 30 minutes.

> **IMPORTANT:**   Before performing any one of the steps in the installation of Z-Ram II, read through the step thoroughly and make sure that you understand it completely. Be sure to read every word carefully and pay close attention to details. Several illustrations are provided to supplement the instructions.

Step 1   Unpacking Z-Ram II

Your Z-Ram II has been thoroughly tested prior to packaging, but you should carefully remove the Z-Ram II from its anti-static bag and check for any physical damage that may have occurred during shipment. Carefully detach the black "grabber" wire from its shipping position. It is unhooked by holding the flanges of the grabber between your index finger and forefinger and depressing the grabber cap with your thumb. As Figure 2 illustrates, you would hold it just like you would a hypodermic syringe. This will extend the tiny metal hook on the end of the grabber, allowing the grabber to be removed from its shipping position.

Figure 2: Z Ram II "grabber" connector and styrofoam pin protection blocks.
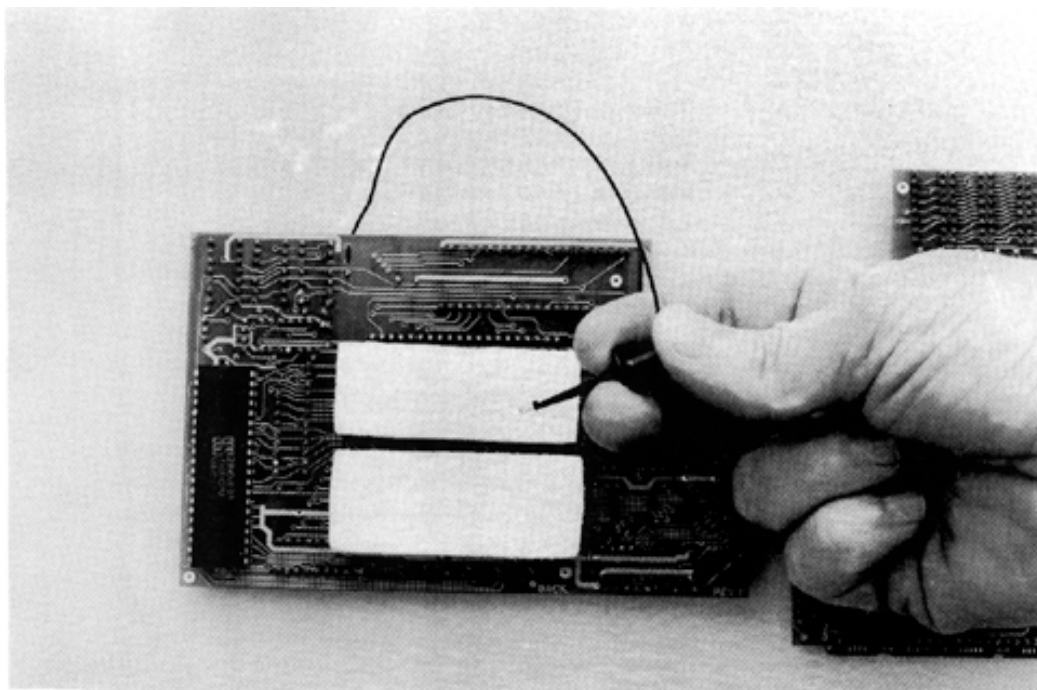
Remove the white styrofoam blocks from the pins protruding from the bottom side of the Z-Ram II. Carefully inspect the pins, making sure that they are all straight. If necessary, gently bend them back into alignment. Straighten the black wire on the Z-Ram II, removing any kinks. Replace the styrofoam pin-protector blocks.

> **Important!**: Leave the styrofoam pin-protection blocks in place until you are ready to plug the Z-Ram II into the I/c. They will prevent accidental damage to the pins, which are delicate and easily bent.

Step 2 Removing the top cover of the //c

Switch the //c power OFF and remove all cables connected to the computer. Turn the computer upside-down and, using the Phillips screwdriver, remove all the screws from the bottom of the case. (There should be 10 screws.) There are three different types of screws, shown as A, B, and C in Figure 3. As you remove these screws mark them as to type and keep them separated.

Figure 3: Apple //c case screws.
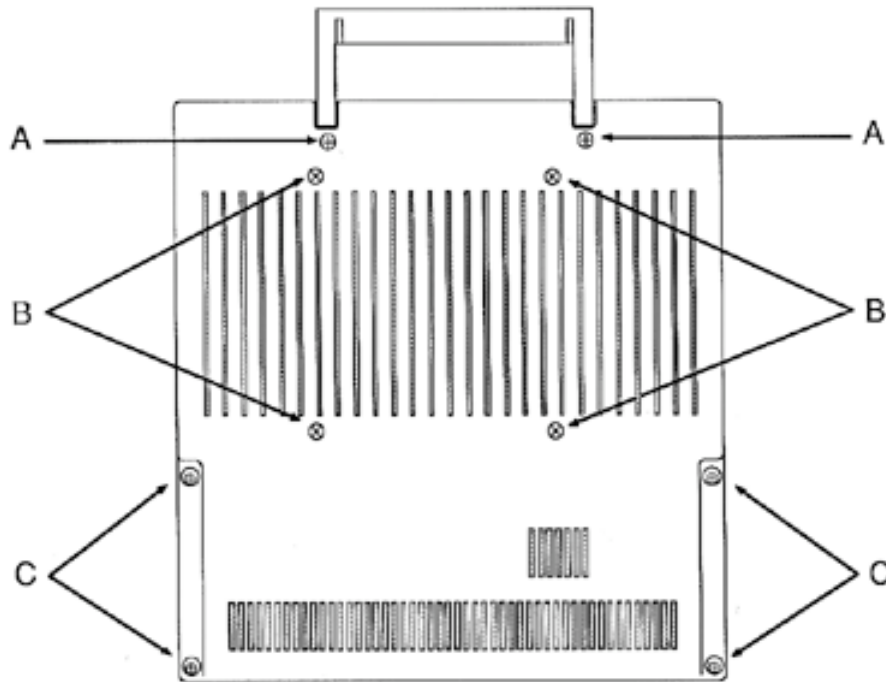
Turn the computer right-side-up. Lift the <u>left</u> front (keyboard) edge of the top cover about 1/2 inch, insert the flatblade screwdriver between the top and bottom covers, to a depth of approximately 1/4", as shown in Figure 4. (The insertion point for the screwdriver should be directly in line with the gap between the "V" and B" keys on the keyboard.) While lifting on the left front edge of the top cover, gently pry (lever) the screwdriver handle up towards the keyboard. This will release the snap connector which holds the front of the //c case together.

Figure 4: Releasing the //c keyboard bezel snap.
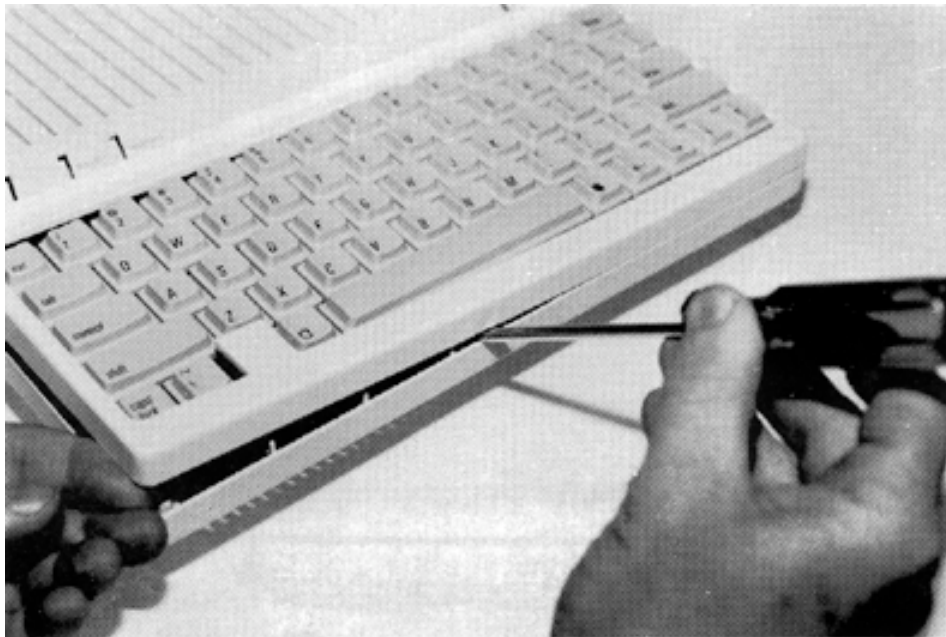
Note: There are several snap connectors holding the //c case together. If some of them break off, don't worry, the case is still good. These snaps are only required during the original assembly of the //c to align and hold the case together before the screws are installed. Just make sure that the broken plastic tabs don't rattle around inside the computer and get into the disk drive!

The next two snaps are a little tougher. These are located on either side of the disk drive door and are unsnapped by lifting the keyboard end of the top cover while holding down on the drive door handle. It may help to gently wiggle the top cover from side to side while lifting. The cover is very flexible but it's tough! The tabs will break before the case does.

After the cover is free from the disk drive door, continue lifting the top cover rearwards until it can be unhooked from the rear of the //c. Check the top cover to see that there are two metal tabs (Tinnerman nuts) attached to the back panel. See Figure 5. If they are not there, find them! They can do serious damage to the computer if left inside. Set the top cover aside and proceed to the next step.

Figure 5: Tinnerman fasteners inside the *//c* top cover.

Tinnerman
Fasteners
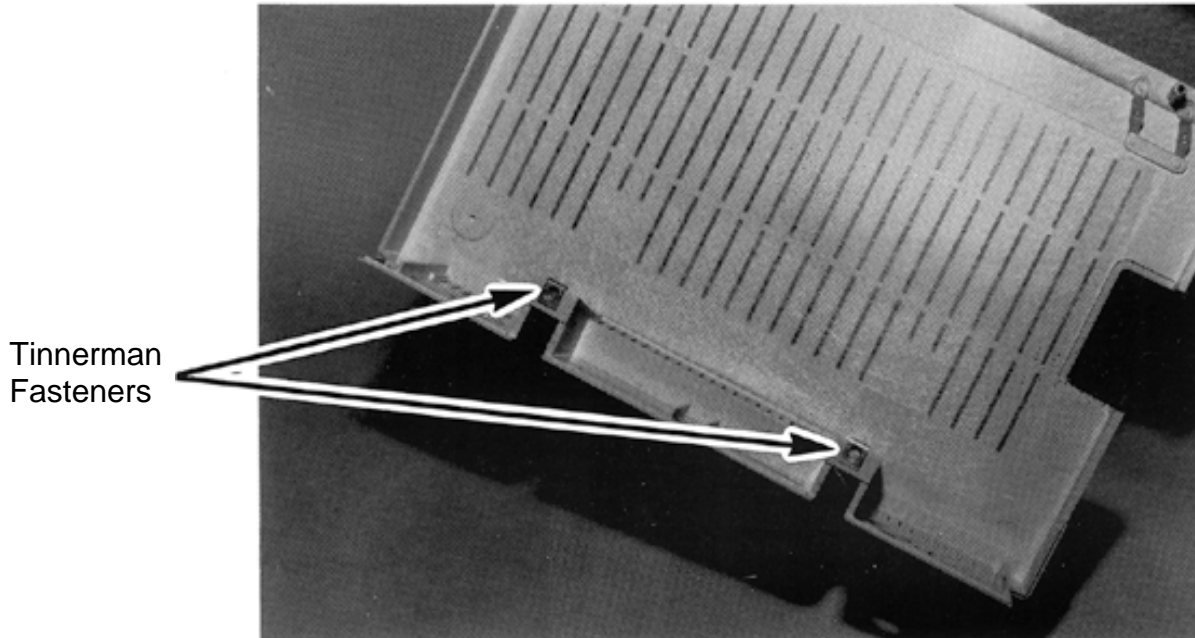
Step 3   Removing the Keyboard and Disk Drive

Lift the front (spacebar) edge of the keyboard, gently moving it away from the disk drive, exposing the ribbon cable connecting the keyboard to the //c main logic board. Disconnect this cable from the main logic board receptacle by lifting it upward with a gentle rocking motion. Set the keyboard aside.

Disconnect the disk drive ribbon cable from the main logic board in the same manner. It is located between the disk drive and the row of connectors on the rear of the computer. Lift the disk drive clear of the computer and set it aside also.

Step 4   Preparing the //c Main Logic Board

Referring to figure 6, locate the CPU chip and the MMU chip near the front right corner of the //c main logic board. The MMU and CPU chip sockets are labeled on the circuit board, in white lettering, just above (to the rear) of the sockets. (CPU stands for Central Processing Unit; MMU for Memory Management Unit.)

Figure 6: Removing the CPU and MMU chips.

MMU

CPU
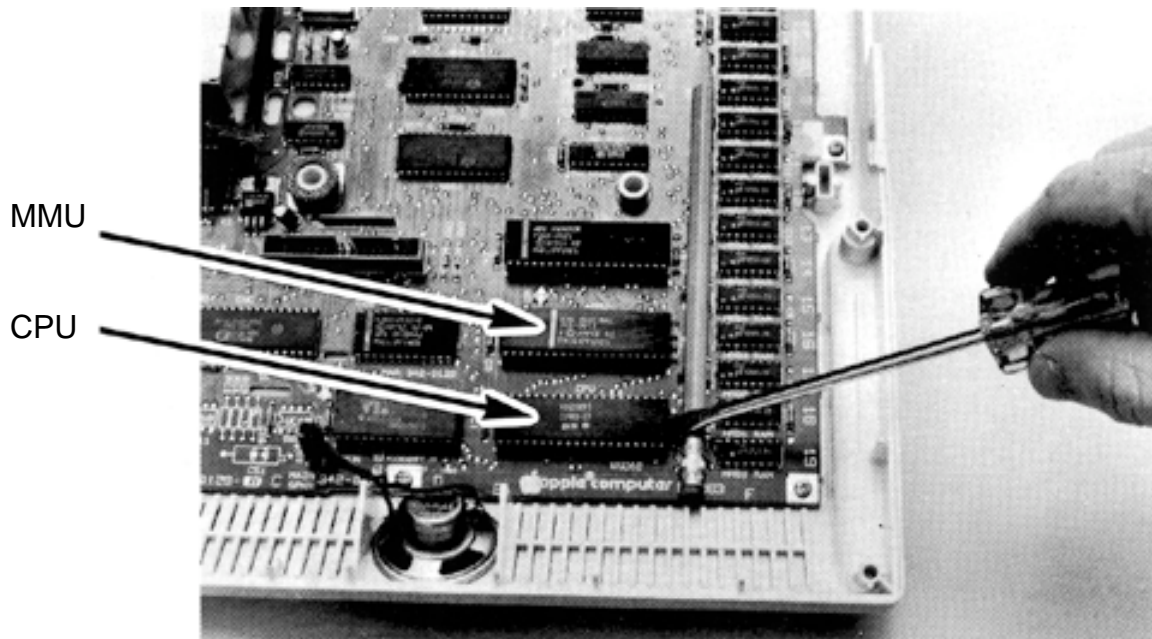
Important!: Some (very few) Apple *//c's* were manufactured with the MMU chip soldered in. It your//c does not have a socket for the MMU, the MMU will have to be desoldered and a socket installed. This is very tricky and should only done by a professional with the proper tools. Please contact Applied Engineering Technical Support for details.

Figure 7: Soldered and Socketed MMU chips

MMU
Main Logic Board

End View of SOLDERED MMU

PRY HERE

NOT HERE

MMU
Socket
Main Logic Board

End View of SOCKETED MMU

7
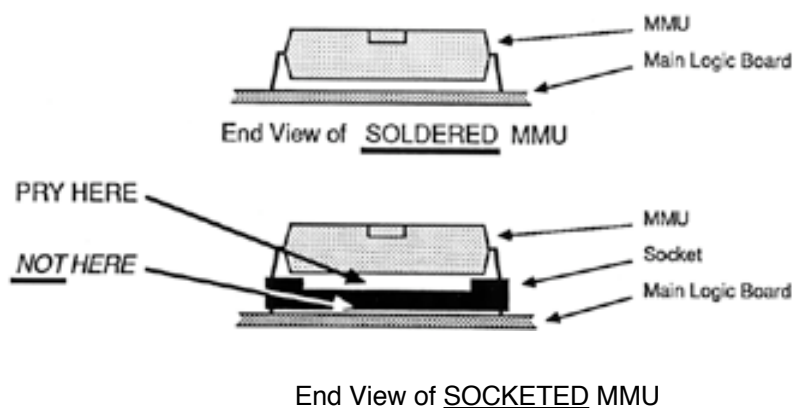
Using the flatblade screwdriver, gently pry these chips out of their sockets. Be careful not bend the pins! Lift alternate ends of the chips just a tiny bit at a time until the chip is free of the socket. As shown in Figure 6, a pencil can be used as a fulcrum. Make sure you pry just the chips and not the sockets! (See Figure 7!) Any bent pins can now be straightened with needle nosed pliers or tweezers.

After removing the CPU and MMU chips, check for and discard the small plastic strips that may be sandwiched between the chips and their sockets. These strips,called "chip insertion guides", are used by the chip insertion machine during the assembly of the //c. Not all //c's were manufactured using these guide inserts. The presence of these guides raise the chips too high in their sockets and may not allow the Z-Ram II to be fully seated into the CPU and MMU sockets. Figure 8A illustrates a chip socket with the chip insertion guide installed. Figure 8B is an example of a socket without the insertion guide.

Visually check the MON and IOU chips and remove their chip guides if necessary. The MON Monitor) chip is a 28 pin chip immediately to the left of the CPU chip. The IOU (Input/Output Unit) chip is a 40 pin chip directly to the rear of the MMU chip.

Figure 8:   Chip Insertion Guides

Integrated Circuit "Chip"
Chip Insertion Guide
//c Main Logic Board

A   B
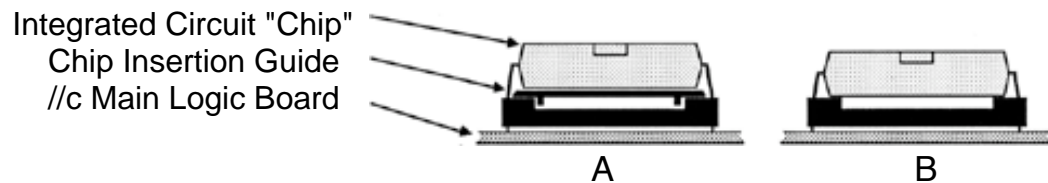
Step 5   Preparing the Z-Ram II Board

Position the Z-Ram II board as shown in Figure 9 and locate the empty CPU socket. Verify that the styrofoam pin-protection blocks are still in place on the bottom of the Z-Ram II card. Install the CPU (65C02) chip in the socket with the NOTCH of the CPU chip oriented as illustrated in Figure 9. (Hint: The CPU chip has "65C02" written somewhere on it.) Make sure all pins are fully seated in the socket.

Figure 9: Z-Ram II CPU and MMU sockets
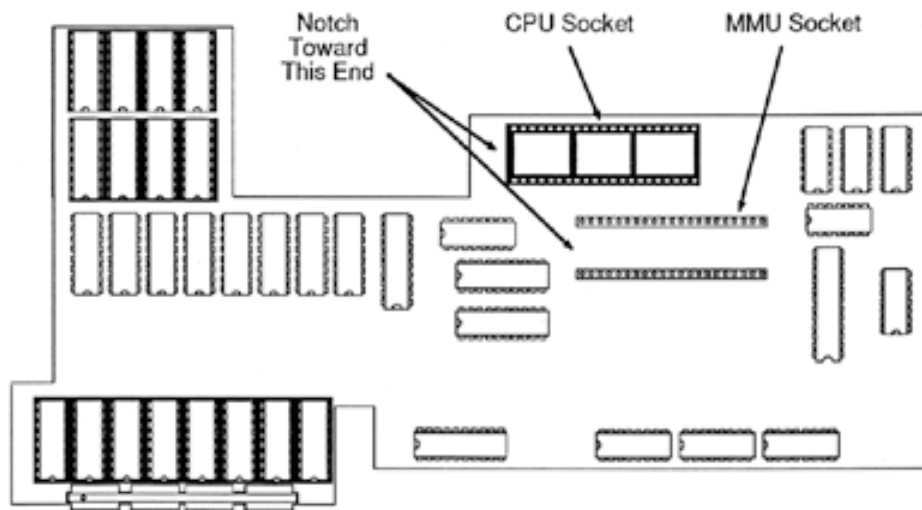
Figure 10: Seating the MMU chip
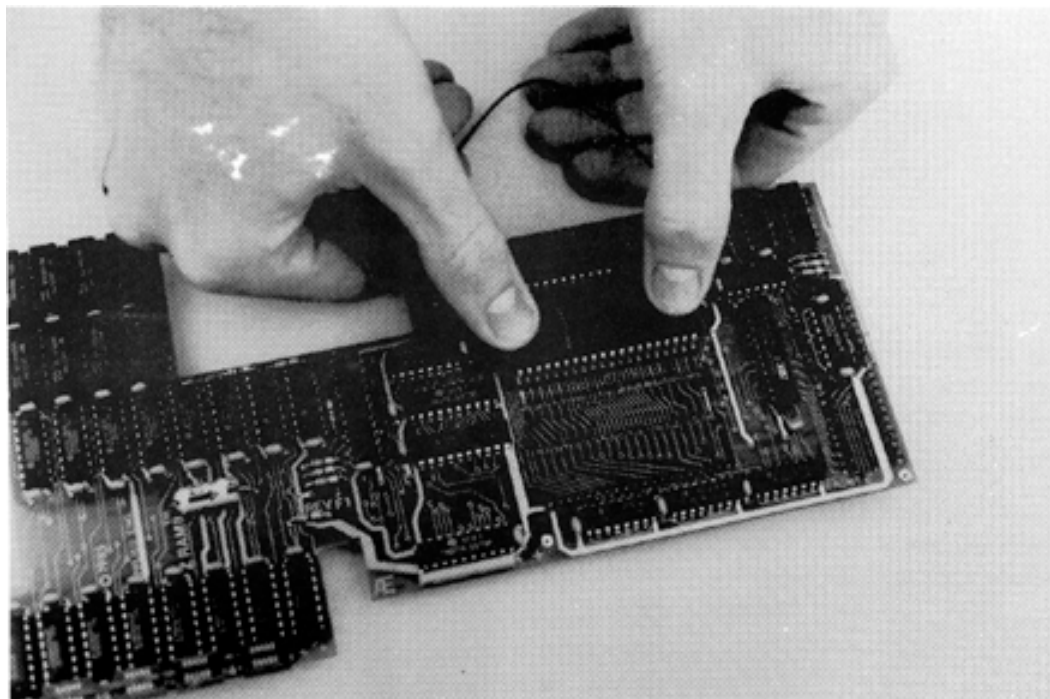
Referring again to Figure 9, locate the MMU socket. It is a "super low profile" socket and does not really look like a socket at all. It is flush with the circuit board and resembles two rows of brown plastic with little white "buttons" instead of pin holes. Position the MMU chip over this socket, with the MMU notch oriented toward the middle of the Z-Ram II circuit board, as indicated in Figure 9. Make sure each pin of the MMU is centered over its respective socket "button" and with firm, even pressure slowly press the chip into the socket. If you see a pin start to bend, stop immediately and straighten it! The pins on the MMU socket will puncture the buttons, which help hold the chip firmly in the socket and prevent pin corrosion. The Z-Ram II is now ready to be installed in the//c,

Step 6 Installing the Z-Ram II in the//c

Remove the foam pin-protection blocks from the underside of the Z-Ram II board. Position the Z-Ram lover the //c main logic board as illustrated in Figure Ii. Peeking under the Z-Ram las you lower it over the CPU and MMU sockets on the main logic board, carefully align the Z-Ram II pins with the sockets. Make sure the black wire attached to the Z-Ram II is not wrapped underneath the Z-Ram II board. Position your fingers exactly in the location shown by Figure 11 and press down gently but firmly until the board is seated firmly in the sockets. The connectors should slide smoothly and evenly into the sockets, if uneven or excessive resistance is felt, stop and check the pins for straightness.

Figure 11: Positioning and seating the Z-Ram II
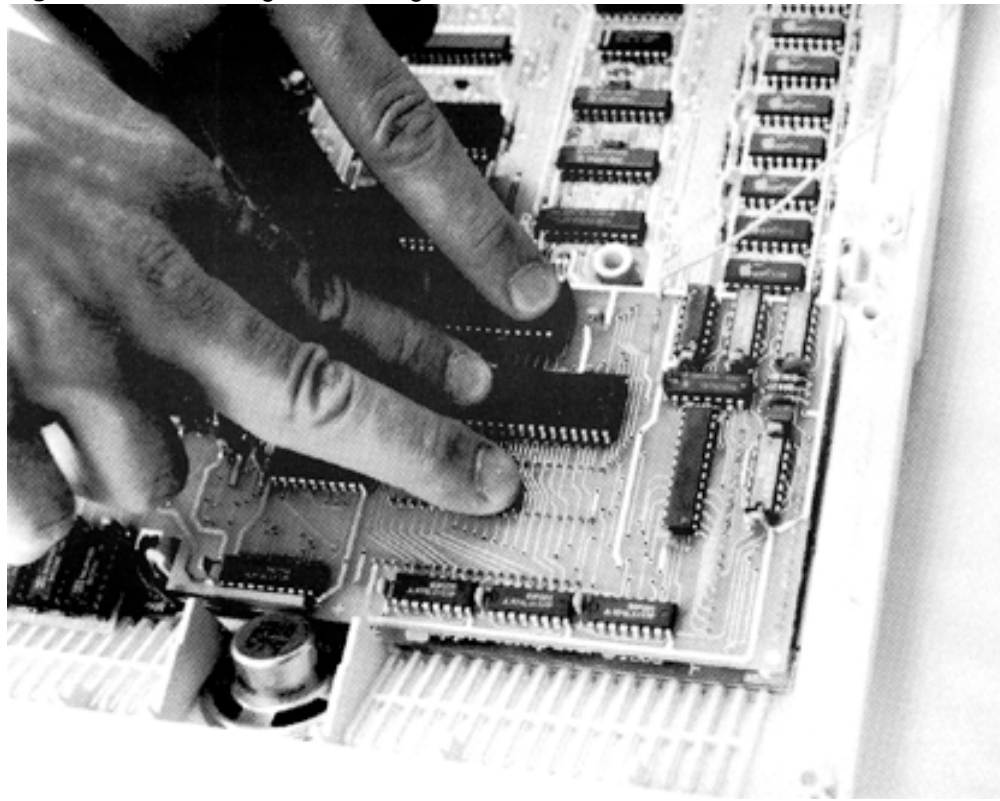
If you notice the Z-Ram II card does not seem to seat properly or lay down flat, it may be caused by the IOU or MON chip on the//c main logic board being too high in their sockets. See page 8 for details and instructions for removing the chip insertion guides.

Step 7 Connecting the "grabber" wire.

Refer to Figure 12 and locate the TMG chip on the//c main logic board. You will find "TMG" in white letters printed on the circuit board just to the rear of this chip. Latch the "grabber" connector to PIN 2 of this chip. Stick one piece of double sided tape under the connector to hold it in place and use the other piece to hold the wire down against the //c main logic board. Double check to make sure the grabber" is securely connected to **Pin 2** of the TMG chip.

Figure 12: Fastening the Grabber Wire to the TMG chip
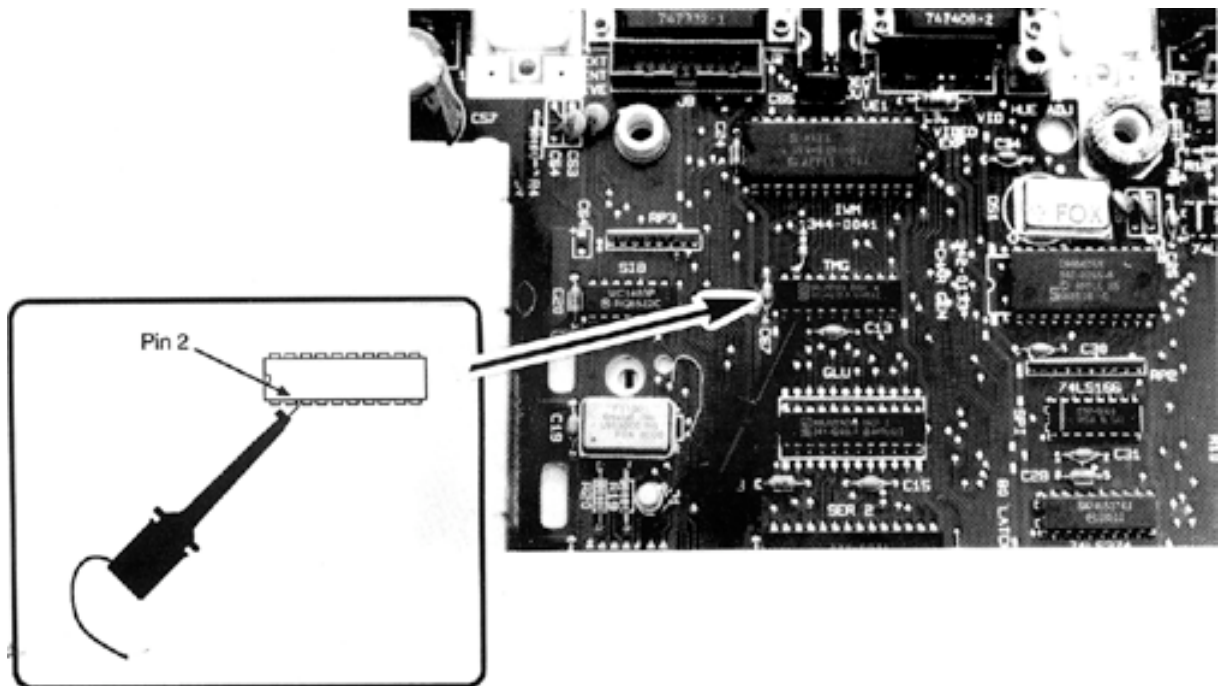
Step 8 Replacing the Disk Drive

Place the disk drive back in position. Plug the disk drive ribbon cable connector into its socket on the main logic board. This connector is keyed and cannot (easily) be plugged in backwards. Verify that the other end of this ribbon cable is securely seated in the disk drive connector.

Plug the keyboard ribbon cable connector into its socket on the //c main logic board. This connector is also keyed. Notice there are tabs on the keyboard which lit into the disk drive. (See Figure 13.) Insert these tabs into the drive and lower the front (spacebar) edge of the keyboard into the plastic mounting guides on the bottom case.

Figure 13: Disk drive slots - keyboard tabs

Keyboard
Tabs
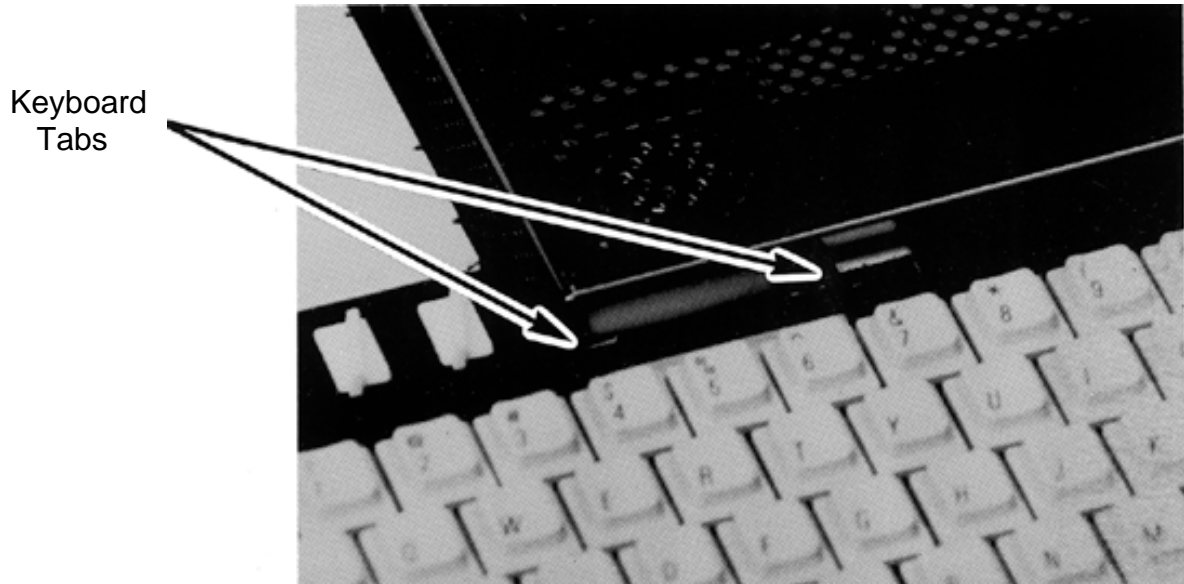
Step 9 Preliminary Test

Before you put the cover back on the computer, you may wish to test your installation. just in case...

Make sure the power switch is in the OFF position, connect the floor mount power supply cable and the cable from the monitor. Insert the backup copy of the Super AppleWorks Desktop Expander disk and turn the computer on. When the main menu appears, select the "Z-Ram Memory Test" option. See Appendix D for more details on this test.

If you wish to test the Z-80 portion of Z-Ram II, boot the CP/AM 4.0B System Master disk. You should see the following message on the screen:

APPLE // CP/AM
60K VER 4.0B
(C) APPLIED ENGINEERING

A>

If this does not appear on your screen, the "grabber" wire may not be correctly installed.

When testing is complete, turn the power OFF and disconnect all cables from the //c.

Step 10   Replacing the Top Cover

> **Note:**   Some of the earlier //c top covers are made in two pieces, a top panel and a back panel. If yours is a two piece design, snap the two together and install them as one piece.

Verify that the Tinnerman nuts are securely attached to the top cover (See Installation Step 2). Snap the //c carrying handle into the notches on the rear panel of the bottom cover. Hook the bottom edge of the back panel (top cover) under the power switch and back panel connectors and lower the keyboard end forward, over the front of the computer. Pulling the cover toward the keyboard while lowering it over the disk drive, as shown in Figure 14, will make it easier to snap the cover in place. Holding the top and bottom covers together turn the computer over and replace the 10 case screws. To prevent binding, don't tighten the screws completely until all screws are threaded in their holes.

Figure 14: Installing the Top Cover
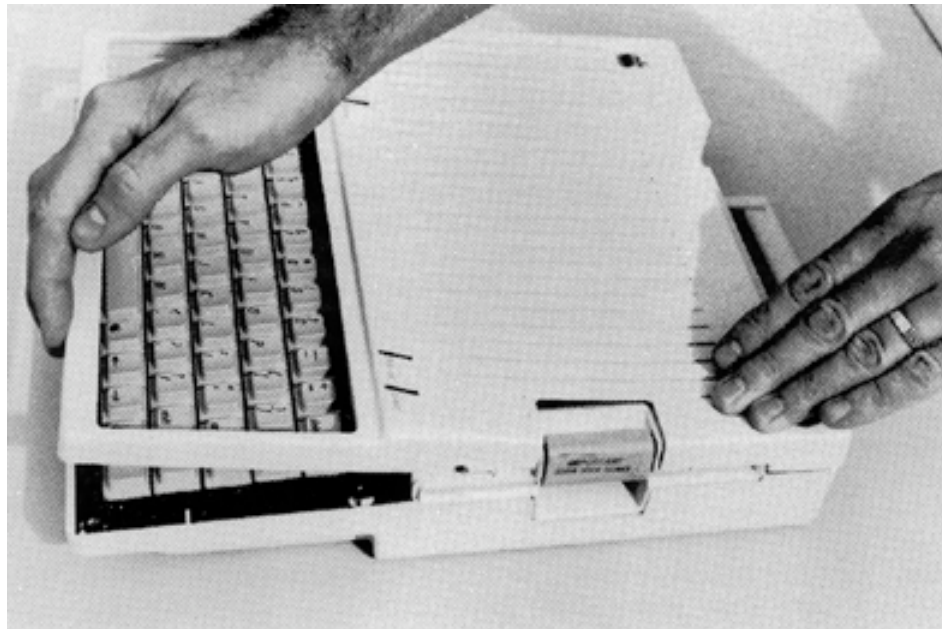
If you inadvertently got them mixed up, the 4 fine-threaded screws hold the disk drive in place (Figure 2, Item B). The 4 coarse-threaded screws with the smaller heads position the keyboard and secure the front end of the case (Item C). The remaining 2 coarse-threaded screws install right next to each hinge of the carrying handle (Item A).

Installation is complete! Congratulations!

Chapter 2

Super AppleWorks!

AppleWorks Enhanced

AppleWorks is the best selling program from Apple Computer that combines a word processor program, data base program, and a spreadsheet program into one "integrated software package". Evaluated separately, each of these programs would be an outstanding "stand-alone" program, but when combined, AppleWorks becomes an extremely versatile and useful productivity tool. But when used with a Z-Ram II equipped Apple //c, AppleWorks can be extraordinary!

Here's how. AppleWorks uses a portion of the computer's extended memory to store active AppleWorks documents (files). This portion of memory is referred to as the AppleWorks Desktop and its size is expressed in Kilobytes or "K'. (A Kilobyte is 1,024 bytes.) With the Apple //c 64K of auxiliary memory, the Desktop size is limited to only 55K. But, depending on how much memory is installed on your Z-Ram II, you could have a Desktop size of 727K! Since the AppleWorks program only expects to "see" 64K of auxiliary memory, the Super AppleWorks Desktop Expander Utility disk, included with your Z-Ram II, will enable you to easily modify AppleWorks disks to utilize the extra memory available with Z-Ram II.

AppleWorks Enhancement

Here is a list of the AppleWorks enhancements available with the Super AppleWorks Desktop Expander Utility:

    ◊   Improved AppleWorks Speed...Fewer disk drive accesses.

    ◊   15,200 lines in the Word Processor (versus 2,250 lines).

    ◊   15,200 records in the Data Base (versus 1,350 records).

    ◊   Greater Spreadsheet capacity.

    ◊   Expanded Desktop...To accommodate larger Desktop files.

    ◊   Multiple disk file saving capability...For those larger Desktop files.

    ◊   Built-in printer buffer (option).

    ◊   Time display and Data Base time-stamp capability (option).

Super AppleWorks Desktop Expander Utility

This program will allow you to modify your AppleWorks STARTUP and PROGRAM disks to take advantage of the Z-Ram II's capabilities, it is a one-time only modification. Even if you add or reconfigure memory on your Z-Ram II you won't need to run this utility again.

This is what you will need to modify your AppleWorks disks:

◊ An Apple //c. (Or any Apple II with at least one floppy disk drive.)

◊ A BACKUP COPY of the Super AppleWorks Desktop Expander Utility disk.

◊ A BACKUP COPY of your ORIGINAL AppleWorks STARTUP and PROGRAM disks. This utility will also modify the AppleWorks program files on an Apple UniDisk 3.5.

IMPORTANT!

Do not modify your ORIGINAL AppleWorks disks! See your AppleWorks Reference Manual or Apple//c Owner's Manual for information on making BACKUP disks.

Use this utility only on <u>unmodified</u> copies of AppleWorks STARTUP and PROGRAM disks. Any patches to AppleWorks required by other programs, such as Pinpoint, should be made <u>after</u> the Super AppleWorks Desktop Expansion.

The Super AppleWorks Desktop Expander will modify all revisions of AppleWorks, but Applied Engineering recommends that you use the most recent revision available. Please see your authorized Apple dealer for information about AppleWorks updates.

Special Note to owners of Applied Engineering's //c System Clock:

The Super AppleWorks Desktop Expander will install the CLOCKWORKS™ on-screen time display patch to the AppleWorks STARTUP and PROGRAM disks. Do not apply the Super AppleWorks Desktop Expander utility to an AppleWorks disk that has been patched using the CLOCKWORKS installation option on the //c System Clock Utilities disk. The Super AppleWorks Desktop Expander utility will not be able to modify a previously modified disk.

To allow time and date stamping for ProDOS files and AppleWorks Data Base time/date categories, you must install the CLOCK.SYSTEM file from the //c System Clock Utilities disk. The CLOCK.SYSTEM file can be installed to the AppleWorks STARTUP disk either before or after using the Desktop Expander. Please refer to your //c System Clock Users Manual for details on how to use the //c System Clock Utilities disk to install CLOCK.SYSTEM.

Modifying AppleWorks

This is how to use the Super AppleWorks Desktop Expander utility to modify the AppleWorks STARTUP and PROGRAM disks:

1. Boot the Super AppleWorks Desktop Expander Utility in the//c internal drive.

2. When the Main Menu appears, select the "Execute AppleWorks Desktop Enhancements" option. The Applied Engineering Super AppleWorks Desktop Expander copyright screen will then appear. Pressing the space bar will display the next screen.

3. The features and options available are described briefly on the display screens, and in greater detail in the next section of this chapter, About the New Features. Read each screen carefully.

4. When prompted for a yes or no input, enter a **Y** for yes or an *N* for No. The prompted options are:

> Install Auto Load Option.
> Install the Printer Buffer Option.
> Install the Applied Engineering Time Display Option.

5. After all options have been selected, you will be prompted to remove the Super AppleWorks Desktop Expander disk and insert a COPY of the AppleWorks STARTUP disk in the internal drive. (Drive one.) Be sure the WRITE PROTECT tab has been removed from the disk. As prompted at the bottom of the display screen, press the space bar to continue or **Q** to quit and return to BASIC.

The Super AppleWorks Desktop Expander program expects to find the AppleWorks STARTUP and PROGRAM volume directories both named /APPLEWORKS. If, at this point, the program displays the following message:

> **Cannot find correct file**
> **Please enter the prefix of the AppleWorks program**
> **?**

enter the complete ProDOS pathname of the AppleWorks STARTUP disk. If the AppleWorks program to be modified is on a UniDisk 3.5 subdirectory, enter the complete ProDOS pathname of the subdirectory containing the AppleWorks files. Be sure to include the (/) slashes. An example ProDOS pathname of an AppleWorks subdirectory on a UniDisk 3.5 (with a volume directory name of /UNIDISK) might be:

> **/UNIDISK/APPLEWORKS**

6. After the STARTUP disk has been successfully modified, you will be prompted to remove it from the disk drive and insert a (WRITE ENABLED) COPY of the AppleWorks PROGRAM disk. The volume directory name of this disk is also expected to be /APPLEWORKS. If the AppleWorks program files being modified are on a UniDisk, press the space bar when prompted to insert the PROGRAM disk.

7. When the modification to the AppleWorks PROGRAM disk has been completed, the program will beep the //c speaker and exit to Applesoft BASIC.

8. Any modifications required by other AppleWorks accessory programs should be done at this time. (You may wish to make an archive copy of the modified "Super AppleWorks" disk prior to any other modifications.) If you have the Applied Engineering //c System Clock, you may now use the //c System Clock Utilities disk to install the CLOCK.SYSTEM file on the STARTUP disk.

9. If you will be using the modified AppleWorks from a hard-disk drive, copy the AppleWorks files to the hard-disk according to the instructions provided with your hard-disk file management program.

About The New Features

The following paragraphs describe the new features of the Enhanced AppleWorks and the instructions for using them. Please read this section very thoroughly.

Desktop Expansion This feature expands the AppleWorks Desktop to accommodate larger files. It increases the maximum number of records in the data base and lines in the word processor from 1,350 and 2,250. respectively, to 15,200 for bath! It also enlarges the Desktop from the standard 55K up to 727K, depending on the amount of memory on the Z-Ram II card. The Desktop Expansion feature is invoked automatically when the modified AppleWorks program is booted.

In order to achieve the extraordinary increase in the number of lines and records, a small portion of the AppleWorks Desktop space was sacrificed. Compared to the additional memory available on Z-Ram II this sacrifice is negligible.

Multiple Floppy-Disk Files Data storage problems arise when the size of a file on the Desktop exceeds the amount of storage space available on the data disk. A standard floppy disk can store only 135K of AppleWorks data. Using the modified AppleWorks, Desktop files larger than the available disk space, will be segmented, or "split", and saved in sections, to two or more disks. The following paragraphs explain this new feature.

Saving Desktop Flies to Disk

When saving a Desktop file which is larger than the data disk space available, AppleWorks will prompt the user with the following message:

Segmented Desktop FIle...
    Please Insert NEXT disk with SAME VOLUME NAME

Remove the data disk from the current disk drive and replace it with the next data disk. All segments of a given segmented Desktop file MUST be saved on data disks having the same volume (disk) name as the primary (first) data disk.

If you discover that you don't have enough formatted blank disks with the same volume name, don't panic! Just use the AppleWorks Disk Formatter utility under the "Other Activities" menu.

In order to enable the multiple disk save capability, one of AppleWorks safety features had to be overridden. Ordinarily, AppleWorks would save a changed file to the disk before deleting the original file. Now it will replace the original file by writing over it. Pressing the Escape key at any time during a save operation will cancel the save and DELETE the current disk file or file segment on disk. (The Desktop file does remain in memory until removed in the normal manner or power is turned off.) BE CAREFUL!

This seems to be a good time to remind you of Murphy's Law:

Whatever **can** go wrong will go wrong!

Back up your Desktop files! A power failure during a disk save operation could conceivably wipe out a file that had taken months to create! It would be a very good idea to have backup copies of your working data disks, especially if you have large amounts of data stored on them.

Adding multiple-disk files to Desktop

When adding multiple-disk, or segmented, files to the Desktop, the file MUST be loaded from the data disks in the same sequence as it was saved. AppleWorks will prompt the user to insert the NEXT data disk(s) until the file is completely loaded.

If you list the files on disks that contain segmented files you will notice that the first segment has the filename that you gave it. The next segment (on the NEXT disk) will have a ".2" appended to the end of your filename. The next disk in the sequence would have a ".3" appended to the filename and so on. This is the way AppleWorks keeps track of the segmented Desktop file sequence. You will probably want to label your data disks in such a way that you too can keep track of the proper sequence.

AppleWorks
Auto-Load feature

AppleWorks was designed for use with only 64K of additional memory, so not all of the PROGRAM disk information is loaded into memory during program startup. Some special 'functions', like displaying help screens, adding files, or using the clipboard, are left on the disk and retrieved only when they are needed. With a Z-Ram II, you can load these special functions into extended memory, minimizing the time consuming disk accesses and allowing you to move between Desktop files and AppleWorks' special functions very quickly.

The Auto-Load feature loads all of the AppleWorks special functions into memory along with the rest of the AppleWorks PROGRAM disk. This extends the AppleWorks startup time by about a minute. The PROGRAM disk startup screen will display a counter indicating the number of functions remaining to be loaded. (During the countdown it may appear that the counter has skipped a function. This is because some functions are relatively small and loaded faster than indicated by the counter.) Pressing the Escape key during this count-down will stop the loading of special functions and immediately display the AppleWorks Main Menu. AppleWorks will access the functions in memory that were loaded before the Escape key was pressed and access the PROGRAM disk for those functions which had not been loaded.

The AppleWorks special functions are loaded into Z-Ram il's memory, but they don't decrease the size of the available Desktop. Magic? Nope! The functions are originally "hidden" in an unused portion of the Desktop memory and as the files on the Desktop grow larger, the special functions are individually "bumped" from memory to make room, If AppleWorks can't find a special function in memory it will go back to the PROGRAM disk.

This brings up an interesting point about using the Auto-Load feature to replace the AppleWorks PROGRAM disk in drive one of a single disk system. If you have all of the

special functions Auto-Loaded in memory, you can first specify drive one as the standard location of your data disk and then replace your PROGRAM disk with your data disk. Keep your PROGRAM disk handy, though. You'll need to insert it whenever you print (Open-Apple-P) or change the standard location of your data disk. (The Open-Apple-H print command does not require the PROGRAM disk.) Please refer to your AppleWorks Reference Manual for more details on the selection of a standard location of your data disk.

If you copy all AppleWorks files to a ProDrive and execute the program from the /RAM emulated disk volume, the AutoLoad feature is not required. This method of executing AppleWorks is tricky and should not be attempted by those not familiar with AppleWorks and Apple ProDOS. (PRofessional Disk Operating System.) This method also requires an AppleWorks file (APLWORKS.SYSTEM) to be modified before being executed from a ProDrive. See the PARTITION instructions in Chapter 3.

Printer Buffer Option	With the Z-Ram II installed, the Printer Buffer option will automatically enable you to continue working with files on the AppleWorks Desktop even while your printer is printing. The advisory message "Printing to Print Buffer" will be displayed and the file(s) to be printed are actually "spooled" to unused portions of the Z-Ram II memory banks. The file is then output to the printer from the print buffer. Multiple files can be printed to the print buffer using either one or both of the //c's serial ports. This option does not affect or decrease the size of the available Desktop.
AppleWorks Time Utilities Option	This option was designed to read the output of the Applied Engineering //c System Clock and replace the "Open-Apple-? for Help" message in the bottom line of the AppleWorks screen with a current date and time display. This option also allows the AppleWorks user to enter the current date and time into a Data Base category by typing the @ character as the only entry in that category. Note: The category name must contain either the word DATE or TIME, but should not contain both. For example you should not use this feature in a category named "Date and Time".

The "CLOCK.SYSTEM" file, required by the Applied Engineering //c System Clock, can be patched to the AppleWorks STARTUP disk before or after using the Super AppleWorks Desktop Expander Utility. Instructions are in the //c System Clock User's Manual.

Z-Ram II RamDrive

Introduction

RamDrive is nothing more than a program which allocates all or a specified portion of Z-Ram II extended memory (RAM) to be used as one or more solid-state" disk drives. This program loads itself into a special reserved area of the //c's memory and uses a portion of the Z-Ram II memory to store desired programs and files in "disk format. The computer is thereby "tricked" into thinking that it is accessing a disk drive when it is actually retrieving or storing data in memory.

Increased data access speed and potentially greater disk storage capacity make the RamDrive a convenient, low cost alternative to a hard-disk drive. Loading from and saving to a RamDrive can be as much as 20 times faster than a conventional disk drive due to the elimination of the mechanical activity required of a conventional drive. The RamDrive data storage capacity is dependent on the amount of memory available on your Z-Ram II.

RamDrive is compatible with almost all DOS 3.3 and ProDOS software available for the Apple //c. Any programs or data files which <u>are not</u> copy-protected can be loaded into a RamDrive and run or accessed just like they can on any other disk drive. A disk which cannot be duplicated with a standard copy program like COPYA or FILER (ProDOS volume copy) is probably copy protected. Although, some copy-protection schemes will allow the program files to be copied from the floppy disk to a hard disk (or RamDrive!) but not to another floppy disk. Due to the rising popularity of hard disk drives and RamDrive software, most software publishers are offering copyable upgrades to their previously copy-protected programs. Check with the software publisher or dealer from whom you purchased the software for details.

RamDrive is a volatile storage medium. Since information stored in a RamDrive is actually stored in Z-Ram II memory, the information in the RamDrive disappears when the computer is turned off. This means you will have to save the data you want to keep to a regular disk drive before turning the computer off.

> Proceed with Caution! Be careful how you use RamDrives. Although the use of emulated disk drives can appreciably speed up those programs that require frequent disk access. RamDrives can be tricky to use and potentially hazardous to your data. Keep Murphy's Law in mind when you weigh the advantages and disadvantages of using RamDrive software.

The DOS 3.3 version of RamDrive and its utilities are on side two of the Super AppleWorks Desktop Expander disk. In addition to the AppleWorks utilities, side one contains ProDrive, the ProDOS version of RamDrive, and its special utilities. The installation, configuration, and use of these programs and utilities is explained in this chapter.

Pascal RamDrive instructions are on side two of the CP/AM 4.0B System Master disk. You <u>must</u> have Apple Pascal 1.1 or 1.2 to access the Pascal files or to use the Pascal RamDrive! The instructions for the CP/AM RamDrive are in Chapter 5 of this manual.

RamDrive for DOS 3.3

Important Note: You don't have to be an experienced programmer to use the RamDrive software but you do have to be familiar with the commands and operation of DOS 3.3. Some knowledge of the Applesoft BASIC programming language is also required. For additional information the following reference manuals are suggested:

> **The DOS Users Manual** (Apple Computer, Inc.)
> **The DOS programmer's Manual** (Apple Computer, Inc.)
> **Beneath Apple DOS** (Quality Software)
> **Applesoft Programmers Reference** Manual (Addison-Wesley Publishing)
> **Apple //c Reference Manual** (Apple Computer, Inc.)

This section explains what the DOS 3.3 version of RamDrive is and how to install and use the emulated disk drive(s). The instructions for the additional programs, SPEEDOS, RAMCOPY, and a specially modified version of FID, included on the DOS 3.3 side of the Super AppleWorks Desktop Expander disk, are also provided in this section. Similar documentation is also available on the disk.

Description of RamDrive Depending on the amount of memory (or memory you allocate) on your Z-Ram II, the RamDrive software will automatically configure Z-Ram II memory to emulate from one to six solid-state disk drives. The storage capacity of the individual RamDrives is dependent on the amount of Z-Ram II memory allocated for RamDrive emulation. These disk drives support all DOS 3.3 commands and can be accessed just like conventional DOS 3.3 disk drive(s) The RamDrive phantom disk interface card, emulated by software, is "installed" in slot 3. For each additional 192K of Z-Ram II memory, DOS 3.3 will recognize another emulated disk drive. For example, a 512I< Z-Ram II, with all banks allocated to RamDrive, would be able to emulate one drive (Drive 1) with 744 free sectors, one (Drive 2) with 749, and another (Drive 3) with 239 free sectors. (An initialized DOS 3.3 floppy disk has 496 free sectors; one sector contains 256 bytes of data.) The maximum number of DOS 3.3 files is still limited to 105 per volume (disk). Table 3-1 illustrates RamDrive storage available for various memory configurations.

Installing RamDrive RamDrive(s) can be installed in one of two ways. The first way is to boot side two of the Super AppleWorks Desktop Expander disk and select the "Install RamDrive" option from the main menu. The second way would be to load DOS 3.3 into memory and run (BRUN) the RAMDRIVE binary program file. This could be accomplished from within an Applesoft program by inserting the following program line:

> **PRINT CHR$(4); "BRUN RAMDRIVE,D1"**

This example assumes that the disk containing the RAMDRIVE program is in the internal drive, although the RamDrive program may be copied to and run from any DOS 3.3 disk used in your system.

Table 3-1 **RamDrive Free Sector Chart**

Z-Ram II Memory	D1	D2	Free Sectors 03	D4	DS	D6
64K	234					
128K	489					
192K	744					
256K	744	239				
320K	744	494				
384K	744	749				
448K	744	749	239			
512K	744	749	494			
576K	744	749	749			
640K	744	749	749	239		
704K	744	749	749	494		
768K	744	749	749	749		
832K	744	749	749	749	239	
896K	744	749	749	749	494	
960K	744	749	749	749	749	
1024K	744	749	749	749	749	239

Using RamDrive

If you have only one emulated disk drive (192K or less), then you may access the RamDrive through Slot 3, Drive 1 or 2. You still have only one emulated disk but you need not specify the drive number.

> **Note:** Your //c does not actually have expansion slots. If you do not fully understand the references to "slots" in this section, please consult the DOS Users Manual.

To see that RamDrive is enabled, type CATALOG,S3,D1 in all capitals and press the Return key. The catalog header 'DISK VOLUME 001 should appear on your monitor.

Copying Files To RamDrive

There are three methods to transfer files to a RamDrive: loading and saving program files; using the Apple File Developer program (FID); and EXECuting a text file in combination with the FID program.

- LOAD and SAVE method: Individually LOAD each BASIC program file from disk and SAVE it to RamDrive.

- FID method: BRUN the FID program. The FID program is especially useful for transferring binary and text files in addition to BASIC program files. The Apple version of FID, provided on the Apple DOS 3.3 System Master disk, recognizes disk drives 1 and 2 but not drives 3 through 6. A modified FID, which allows drives 1 through 6 to be accessed (any slot), is included on the DOS 3.3 side of the Super AppleWorks Desktop Expander disk.

- FID and EXEC file method: This is a quick way to transfer a selected list of files to a RamDrive. (See the DOS Users Manual for more information on EXECuting text files.)

An example EXEC file, COPY ALL FILES, is included on side two of the Super AppleWorks Desktop Expander disk. This sample file instructs the FID program to copy certain files on the disk to RamDrive. To use this sample EXEC (text) file, boot the disk and select the "Copy Files to RamDrive" option. This menu option will automatically run SPEEDOS, install RamDrive, and copy the specified files to RamDrive. Or, this program can be executed by typing EXEC COPY ALL FILES,S6,D1' from the Applesoft BASIC prompt. (Remember to install RamDrive first!) RamDrive space permitting, all specified files on the disk (in this example) will automatically be copied to RamDrive in Slot 3, Drive I.

You must customize COPY ALL FILES for every desired combination of source and destination slots and drives. Customization is easy with any word processor program which saves files to disk in text file format (like DOS 3.3 AppleWriter). To further understand how COPY ALL FILES works, use FID to manually transfer some files from 56,D1 to 53,D1; carefully record each keystroke you make, then load the COPY ALL FILES into a word processor and compare it with your keystrokes.

> Note: When creating an EXECutable text file for FID, be sure that none of your filenames contain any DOS 3.3 commands. (e.g. READING.1 ; CATALOG LIST)

After customizing your own COPY ALL FILES text file, making a turnkey startup disk is easy with the following 'HELLO' program:

```
10 REM Turnkey HELLO program
20 REM FID, RAMDRIVE, COPY ALL FILES, and all files to be copied must be on this disk.
30 PRINT CHR$(4);"BRUN RAMDRIVE"
40 PRINT CHR$(4);"EXEC COPY ALL FILES"
50 END
```

Executing DOS 3.3 Programs from RamDrive

Once copied to a RamDrive, most programs can be executed normally. Some programs, however, may have certain routines which expect the program disk to be in the //c internal drive. (Slot 6, Drive 1.) The assistance of an experienced programmer may be required if you need to modify such a program to access the RamDrive.

Remember that data stored in RamDrives are lost when power to the computer is turned off or interrupted. Unless your //c is equipped with an uninterruptable power supply (UPS), it would probably not be a good practice to store data files in a RamDrive without keeping current back-up files on a non-volatile storage medium. (Disk storage.)

Activating and Deactivating Ram Drives

The DOS 3.3 'INIT' command does not initialize a RamDrive, it merely deactivates it. It does not install DOS 3.3 or specify a greeting program on the RamDrive. For example, 'INIT HELLO,S3,D1 will disable RamDrive 1, but will not erase any information stored on that RamDrive volume. To reactivate a RamDrive, run the RamDrive program again. All files previously saved on that RamDrive can once again be accessed. Ram Drives will also be deactivated by a warm-boot (Open-Apple-Control-Reset or PR#6), but can be reactivated by running the RAMDRIVE program.

To erase previous RamDrives and create empty ones, hold the Solid-Apple (⬤) key down during RamDrive program execution. For example, type 'BRUN RAMDRIVE,S6,D1', hold the Solid-Apple key down and press RETURN. Continue holding the Solid-Apple key down until the "disk use" indicator light goes out.

23

Chapter 3

**RamDrive
Audio-Visual
Access Indicators**

Just as the red "disk use" indicator light indicates a disk access in progress, RamDrive provides a visual access indicator. Whenever DOS 3.3 accesses a RamDrive, an inverse 'R' or 'W' will appear in the lower right corner of the display. The 'R' indicates a disk read operation; the 'W' signifies a disk write operation.

In addition to the visual indicator, an optional audio access indicator is also available. The audio indicator is a high pitched click for a 'read from disk" access and a lower pitched click for a "write to disk" operation. To activate the audio indicator option, hold the Open-Apple key down while the RamDrive is being enabled. For example, type **BRUN RAMDRIVE,S6,D1**, hold the Open-Apple key down, and press RETURN. Continue to hold the Open-Apple key down until the "disk use" indicator goes out and you hear an audio click. (If you don't hear the click you may need to increase the //c speaker volume.) Each audible click indicates a sector read or write operation. You may use this procedure to activate the audio indicator even if the RamDrive is already enabled.

**Changing RamDrive
Parameters**

Several modifications can be made to RamDrive to suit the user's needs. The procedure for implementing these modifications is to first BLOAD RAMDRIVE into memory, POKE the new values into specific memory locations, and then CALL 24576 to run the RAMDRIVE program. Table 3-2 provides the memory locations, parameters, and initial values for the RamDrive options. The options are explained immediately following the table.

Table 3-2 **RamDrive Parameters**

Location (decimal)	Initial Value (decimal)	Description
24578		Entry point into RAMDRIVE program
24579	1	80 column option (1 =enable; 0=disable)
24580	0	Double Hi-Res option (enable=1; disable = 0)
24581	3	Emulated slot number (1 to 7)
24582	1	Emulated Volume number
24583	24	Duration of read audio indicator (0 to 255)
24584	18	Duration of write audio indicator (0 to 255)
24585	6	Frequency of read audio indicator (0 to 255)
24586	18	Frequency of write audio indicator (0 to 255)
24587	18	Read visual indicator character (decimal ASCII Value)
24588	23	Write visual indicator character (decimal ASCII Value)
24589	0	Lowest emulated drive number (0 for default or 1 to 3)
24591	15	Directory entries ÷7 for emulated drive 1(1 to 15)
24592	15	Directory entries ÷7 for emulated drive 2 (1 to 15)
24593	15	Directory entries ÷7 for emulated drive 3 (1 to 15)
24594	15	Directory entries ÷7 for emulated drive 4 (1 to 15)
24595	15	Directory entries ÷7 for emulated drive 5 (1 to 15)
24596	15	Directory entries ÷7 for emulated drive 6 (1 to 15)
24597	0	Bank lock out bit map for banks 8 through 15
24598	1	Bank lock out bit map for banks 0 through 7

Note: If a RamDrive is currently active, you must re-install the RamDrive before any modifications will have any effect. This can be accomplished by the CALL 24576 statement.

80 Column and Double High-Resolution Graphics

The 80 column and double high resolution screen displays reside in Bank 0 (64K) of the //c auxiliary memory. As it is provided from Applied Engineering, the RAMDRIVE program is configured to "lock out" this bank from being used by RamDrive. This allows the 80 column and double hi-res display memory area to be used without conflict, regardless of the parameter values in locations 24579 or 24580.

Otherwise, the default values for these parameters allow the use of the 80 column screen and disallow the the use of the double high resolution screen feature. If bank 0 is used for RamDrive emulation and a disallowed display is used anyway, parts of the data on the first RamDrive will be "clobbered". If you will not require the 80 column display and want to gain an additional 4 sectors of RamDrive space, enable RamDrive with Applesoft statements like these:

]100   PRINT CHR$(4);"BLOAD RAMDRIVE,S6,D1
]110 POKE 24579,0:CALL 24576:REM This will disallow 80 columns

(The bracket at the beginning of each line represents the Applesoft BASIC prompt. It is not part of the statement!)

To use the double high resolution graphics display, which displaces 8K (32 sectors) of RamDrive space, change line 110 to:

]11O POKE 24580,1:CALL 24576:REM Allow use of Double HiRes

Emulated Slot

This option allows you to specify the slot in which the "phantom" RamDrive controller card is installed. Slot 3 is the default value, as shown in Table 3-2, but any slot between 1 and 7 could be selected by changing the value at location 24581. If the emulated slot is also the actual disk controller slot (slot 6), then S6,D1 would be the only physical drive and drives 2,3,..,6 would be the emulated drives. The physical drive 2 in slot 8 (the //c external drive)would be ignored in this case.

Lowest Drive Number

The value in location 24589 is used to set the lowest emulated drive number. With reference to the emulated slot 6 example above, if the lowest drive value was changed to 3, the physical slot 6 drives would be drives I and 2. The emulated drives would then be drives 3, 4, 5, & 6. (assuming sufficient Z-Ram II memory)

Emulated Volume Number

The contents of this memory location, 24582, specifies the volume number of the first emulated drive. Subsequent emulated drives will have a volume number one greater than the preceding volume.

Read and Write Visual Access Indicators

The default indicators can be changed from 'R' and 'W' to any other characters by POKE-ing the appropriate decimal ASCII value into the corresponding memory location. The visual Indicators can be effectively disabled by changing the values at locations 24587 and 24588 to 160. (The "high ASCII' SPACE character.)

Directory Entries

You may specify the maximum number of files (directory entries) allowed on each emulated RamDrive. Divide the number of desired directory entries by seven and enter the integer value in the memory location corresponding to the emulated drive number. (see Table 3-2) The default value provides for 105 directory entries ($105 \div 7 = 15$). The minimum allowed is 7 files; the maximum is 105. Obviously, fewer directory entries specified yields more emulated disk data storage space. Up to 14 additional sectors can be be gained per drive.

Audio Access Indicators

The duration and frequency of the optional audible read and write access tones can be adjusted by varying the values in locations 24583 through 24586. Of course, the //c speaker volume control still adjusts the loudness.

Bank Lockout

Z-Ram II memory is organized into memory "banks", each containing 64K bytes of data. Up to 16 banks (1 megabyte) of //c and 7-Ram II auxiliary memory can be used for RamDrive emulation. When the RamDrive is initially created by the RAMDRIVE program, the //c's bank 0 is locked out. That is, the RamDrive cannot access that particular bank for use as part of an emulated disk drive. The bank 0 lock out default was used to allow other application programs which use that bank to be run without conflicting with the RamDrive. With RamDrive installed and bank 0 locked out, DOS 3.3 based programs can run, using bank 0 as they normally would, and also access the RamDrive(s).

Important note about bank numbers: The numbering scheme for Z-Ram II memory banks is determined by the type of RAM chip (64K or 256K), which memory chip sockets are used, and the amount of RAM installed. See Chapter 4, Z-Ram II Technical Reference, for more details.

If you wish to lock out additional banks of Z-Ram II memory or use all of the available banks, you must make one or two simple modifications to the RAMDRIVE program. This involves calculating the values used to lock out the desired banks, loading the RAMDRIVE program into memory, inserting those values into specific memory locations, and then executing the modified RAMDRIVE program.

There are two memory locations associated with modifying the number of banks locked out. These are (decimal) 24598 for banks 0 through 7 and 24597 for banks 8 through 15. To lock out an individual bank in the bank 0 to bank 7 range, first load (BLOAD) the RAMDRIVE program into memory. Raise 2 to the power of the bank number to be locked out, POKE the result in location 24598, and issue a CALL to location 24576 to activate the RamDrive(s). The following example illustrates the statements required to lock out bank 1.

```
]BLOAD RAMDRIVE
]POKE 24598,(2ˆ1)
]CALL 24576
```

To lock out more than one bank in the bank 0 to 7 range, POKE the sum of the individual bank results. For example, to lock out banks 0, 1, 2, and 3:

$(2ˆ0 + 2ˆ1 + 2ˆ2 + 2ˆ3) = (1 + 2 + 4 + 8) = 15$; therefore POKE 24598,15

To lock out a bank or series of banks in the bank 8 to 15 range, apply the same formula except divide the sum of the results by 256 and POKE the final result in location 24597. For example, to lock out banks 8 and 9:

$(2ˆ8 + 2ˆ9) + 256 = (256 + 512) + 256 = 3$; therefore POKE 24597,3

Some application programs which are capable of accessing the extended memory available on Z-Ram II may use the //c's auxiliary memory bank 0 and only the first few banks of the Z-Ram II's memory. For this reason, it is advisable to lock out the lowest numbered banks first, reserving them for program use.

Remember that RamDrive is initially configured to lock out bank 0 of auxiliary memory to prevent possible memory conflicts with programs which may use that bank, If your software does not require bank 0, it can be unlocked by entering the following statements:

```
]BLOAD RAMDRIVE
]POKE 24598,0
]CALL 24576
```

Or, in the deferred (program) mode, as part of a greeting program:

```
]10 PRINT CHR$(4);"BLOAD RAMDRIVE"
]20 POKE 24598,0
]30 CALL 24576
```

RAMCOPY Instructions

RAMCOPY is a disk duplication utility for the Z-Ram II equipped Apple //c. It decreases the number of disk swaps required during disk duplication by using the extra memory available on Z-Ram Ito store more of the source disk data in one pass. The RamCopy user interface is similar to Apple's COPYA disk copy utility program.

RAMCOPY is a binary file and can be executed with the BRUN command or selected from the main menu of the DOS 3.3 side of the Super AppleWorks Desktop Expander disk. Slot 3, or an emulated RamDrive, cannot be specified as either the original or duplicate slot. (Use the FID program to copy files to or from emulated drives.) RAMCOPY will copy standard DOS 3.3, ProDOS, Apple Pascal 1.1, and CP/M disks.

> **WARNING! RAMCOPY uses the** first two 64K banks it finds on Z-Ram II. It will wipe out any RamDrive information that is stored in these banks. To prevent this from happening, lock out these banks from RamDrive. RAMCOPY will also disable RamDrive, requiring you to reactivate the RamDrive.

SPEEDOS Documentation

This DOS 3.3 speed-up utility, included on the Super AppleWorks Desktop Expander Disk can make DOS 3.3 run up to 40 times faster using it in combination with a RamDrive! It is a public domain program written by Lee DeRaud and published in Call A.P.P.L.E.

Use FID to copy SPEEDOS onto your startup disks. SPEEDOS can be enabled by typing 'BRUN SPEEDOS' at the Applesoft prompt or from within a greeting (HELLO) program.

SPEEDOS modifies only the "image of DOS 3.3 in memory; it does not modify DOS 3.3 on the disk. When SPEEDOS is enabled, the INIT command is disabled. RamDrive will also be unable to load or save files outside the memory range from $200 to $BFFF when SPEEDOS is installed.

RamDrive Technical Information

The following information is intended for hard core programmers only. Don't blame us if your mind turns to mush after reading this section.

- After enabling the RamDrive, a PEEK(24590) statement will indicate the highest emulated drive number (0 - 5) available. A value of 0 in this location indicates the drive specification from the lOB to be ignored (1 drive). Together with the lowest emulated drive number found in location 2458g. you can find out how many RamDrives you have available.

- A PEEK to location 24599 will return the number of 64K banks (0 to 15) used by RamDrive. This number may be less than the amount of //c and Z-Ram II auxiliary memory if some banks have been locked out.

- A list of specific banks used by RamDrive can be found in memory locations 24600 to 24615. A value of 255 signifies no bank.

- Some DOS 3.3 enhancement programs or utilities may not work with RamDrive installed. Utilities which relocate DOS 3.3 into high memory will not work with RamDrive.

- Track 1 and track 17 map to the same memory. Unused sectors on the directory, track 17, are used in track 1.

- The RamDrive program loads into an area of memory starting at $6000, which is freed after RamDrive is enabled. RamDrive uses memory from $A581 to $A59D, from $AAB8 to $AAC0, and overwrites the second through fifth bytes of the RWTS routine at $B7B5. If the highest emulated drive number is greater than 2, DOS 3.3 is patched at $A9B5. When the iNIT command is issued, all patches to DOS 3.3 are removed, disabling the Ram Drive.

- The RESET vector from page 3 points to a patch area above the RWTS in DOS 3.3. The purpose is to enable 64K bank 0, if it is not already, whenever the RESET key is pressed. Use care in changing the RESET vector after RamDrive is enabled.

- After each disk access to RamDrive, the bank switched memory ($D000 to $DFFF in main memory) will be left in the following conditions:

  - RAM bank 2 will always be write enabled.
  - ROM will be read enabled if Applesoft was in use prior to the disk access.
  - RAM bank 2 will be read enabled if Applesoft was not in use.

  These rules insure that Applesoft and Integer BASIC and other programs using bank switched memory will work properly.

- The RWTS routine may be called at $3D9 as described in the DOS Programmer's Manual noting the following exceptions:

  - The lOB and buffer must reside in the memory range from $200 to $BFFF.
  - Track numbers may range from I to 49 only. If an I/O error results, you may have too little memory for the requested track on that drive. All non-existent or reserved tracks and sectors are flagged in the VTOC.
  - Sector numbers may range from 0 to 15

ProDrive™ for ProDOS

Important Note: ProDrive is the ProDOS version of RamDrive. It is not difficult to use, but you must be familiar with the ProDOS commands and file structure, and the Applesoft BASIC programming language. The following documents are strongly recommended for additional information:

        The ProDOS User's Manual (Apple Computer, Inc.)
        BASIC Programming With ProDOS (Addison-Wesley Publishing)
        Beneath Apple ProDOS (Quality Software)
        Applesoft programmer's Reference Manual (Addison-Wesley Publishing)
        The Apple I/O Reference Manual (Apple Computer, Inc.)

    This section provides a description of the ProDrive emulated disk drive and explains how to install, configure, and use it. Instructions for creating a turnkey file copy utility are also included in this section. An abbreviated version of this documentation is also available on the ProDOS side of the Super AppleWorks Desktop Expander Disk.

Description of ProDrive   When ProDOS is loaded into the Apple //c's memory it automatically creates an emulated ProDOS volume, with a volume directory name of /RAM, using the 64K bank 0 of the //c's auxiliary memory. The PRODRIVE program on side one of the Super AppleWorks Desktop Expander disk can utilize all or a combination of the additional Z-Ram II memory banks for use as a larger emulated ProDOS volume. The size of this emulated volume is dependent on the amount of available Z-Ram II memory and the number of banks locked out. The ProDrive fully emulates a ProDOS volume (initially named '/RAM') with a phantom disk interface assigned to the //c's firmware as slot 3, drive 2.

Installing ProDrive   The ProDrive can be created in one of two ways. The first way is to boot side one of the Super AppleWorks Desktop Expander disk and select the "Install ProDrive" option from the main menu. The second way would be to load ProDOS and BASIC.SYSTEM into memory and run (BRUN) the PRODRIVE binary program file. (The PRODRIVE program may be copied to and run from any ProDOS disk used in your system.) This could be accomplished from within an Applesoft startup program by inserting the following program line:

        ]10 PRINT CHR$(4);"BRUN PRODRIVE"

    or using the ProDOS DASH (—) command:

        ]10 PRINT CHR$(4);"—PRODRIVE"

    The ProDrive will remain active until the computer is powered-down, the computer is warm-booted (Open-Apple-Control-Reset or PR#6), or its data or volume directory is overwritten by another operating system or application program. Obviously, if the computer's power is removed, all information stored in RAM (the computer's memory) will be lost and irrecoverable. If the computer is warm-booted into an operating system or application program that does not corrupt the data contained in the ProDrive banks, it is possible to recover the ProDrive data.

Executing the PRODRIVE program creates an initialized (blank) /RAM volume directory. Holding the Closed-Apple (⚫) Key down while the PRODRIVE program is running (about 5 seconds) will activate the ProDrive but will not initialize the previous /RAM volume directory. All information previously stored in that ProDrive, if not overwritten by another program, can again be accessed.

Accessing ProDrive The ProDrive emulated disk drive supports all ProDOS commands and protocol, except that it cannot be formatted. Files can be renamed, saved to, loaded from, and deleted from the ProDrive just like a conventional ProDOS volume.

ProDOS files can be copied to and from the ProDrive using Apple's FILER program or the COPYFILES program. Both programs are included on the ProDOS side of the Super AppleWorks Desktop Expander disk. The instructions for COPY.FILES are provided later in this chapter. See the ProDOS User's Manual for FILER instructions or use the FILER's TUTOR menu options. When using the FILER do not use the "FORMAT A VOLUME" or "COPY A VOLUME" on a ProDrive.

Executing Programs Once the ProDrive has been installed and the program files have been copied to /RAM,
from ProDrive you must execute the program using software commands only. Do not perform a (Open Apple-Control-Reset or PR#6.) warm-boot; it will wipe out the /RAM volume directory! To start a program with software commands, you should first be familiar with the ProDOS startup routine and be able to determine the filename of the program's boot or startup program.

When a ProDOS floppy disk is booted in the //c*'s* internal drive, it first loads the operating system file, PRODOS, into memory. Then, ProDOS executes the first system file it finds in the currently prefixed directory with a filetype of SYS and filename suffix of .SYSTEM. if the first system file is BASIC.SYSTEM, like the Super AppleWorks Desktop Expander disk, the BASIC.SYSTEM program looks for and, if found, executes a BASIC program file (file type BAS) with the filename STARTUP. Other SYS type system files usually either contain the application program or load and execute other program files in the directory.

Assuming you have installed ProDrive and copied your program files to /RAM, you must first set the prefix to /RAM. (Enter PREFIX /RAM<return> at the BASIC prompt.) Then type CAT<return>. Check the directory for the first SYS type file or a file named STARTUP.

If a STARTUP file is found, type: —**STARTUP**<return>. If a SYS file is found, use the dash command to execute the file. (i.e. —**APLWORKS.SYSTEM**<return> for AppleWorks.)

Although most ProDOS based software can be run from a ProDrive, not all programs will work as expected. Some ProDOS programs may expect a certain volume directory name other than /RAM. Disk copy-protection may prevent the successful execution of programs from /RAM, although copy-protection schemes, such as the one used by Sensible Speller, will allow the original program to be copied to and executed from a non-floppy disk. (e.g. /RAM volume or a hard disk.) Some programs, such as the ProDOS version of AppleWriter *II,* purposely disable the /RAM volume directory. (See *Apple Writer*

Patch at the end of this chapter.) If your software does not run as expected you should contact the publisher of your software or seek the assistance of an experienced programmer to instruct you on the modification of your software or the execution of your particular program from a /RAM volume directory.

Bank Lock Out Feature

When the ProDrive is initially created by the PRODRIVE program, the 64K bank 0 is locked out. That is, the ProDrive cannot access that particular bank for use as part of the emulated disk drive. The bank 0 lock out default was used to allow other application programs which use that bank to run without conflicting with the ProDrive memory area. With ProDrive installed and bank 0 locked out, ProDOS based programs can run, use bank 0, and access the ProDrive.

> **Caution**: The bank lock out procedure described in this section is more complicated than it sounds. Don't attempt it unless you are very familiar with Applesoft BASIC and ProDOS. Instead, you may wish to use the PARTITION program described later in this chapter.

If you wish to lock out additional banks of Z-Ram II memory or use all of the available banks, you must make one or two simple modifications to the PRODRIVE program. This involves calculating the values used to lock out the desired banks, loading the PRODRIVE program into memory, inserting those values into specific memory locations, and executing the modified PRODRIVE program.

> **Note**: The numbering scheme for Z-Ram II memory banks is determined by the type of RAM chip (64K or 256K), which memory chip sockets are used, and the amount of RAM installed. See Chapter 4, Z-Ram II Technical Reference, for more details.

There are two memory locations associated with modifying the number of banks locked out. These are (decimal) 8195 for banks 0 through *7* and 8196 for banks 8 through 15. To lock out an individual bank from bank 0 to bank *7,* raise 2 to the power of the bank number to be locked out and POKE the result in location 8195. For example, to lock out bank 1:

$2^1 = 1$ therefore POKE 8195,2

To lock out more than one bank in the bank 0 to 7 range, POKE the sum of the individual bank results. For example, to lock Out banks 0, 1, and 2:

$(2^0 + 2^1 + 2\sim2) = (1 + 2 + 4) = 7;$ therefore POKE 8195,7

To lock out a bank or series of banks in the bank 8 to 15 range, apply the same formula except divide the sum of the results by 256 and POKE the final result in location 8196. For example, to lock out banks 8 and 9:

$(2^8 + 2^9) \div 256 = (256 \div 512) \div 256 = 3$; therefore POKE 8196,3

To execute the now modified PRODRIVE program issue a CALL to location 8192.

The following is a summary of Applesoft BASIC statements to lock out banks 0, 1 2, 8, and 9 as in the examples above. (The bracket at the beginning of each line represents the Applesoft BASIC prompt. It is not part of the statement!)

```
]BLOAD PRODRIVE
]POKE 8195, (2ˆ 0 + 2ˆ1 + 2ˆ2)
]POKE 819B,((2ˆ8 + 2ˆ9)/256)
]CALL 8192
```

Or, as part of a BASIC startup program, this example could appear:

```
]10 PRINT CHR$ (4);"BLOAD PRODRIVE"
]20 POKE 8195,7
]30 POKE 8196,3
]40 CALL 8192
```

Using PARTITION   There is an easier way to lock out banks! The PARTITION program on side one of the Super AppleWorks Desktop Expander Disk is a menu-driven utility that will reconfigure the bank lock out parameters within the PRODRIVE program. PARTITION causes the specified *lower numbered* 84K banks of available memory to be locked out from use by the ProDrive. PARTITION will also modify a COPY of your AppleWorks STARTUP disk, enabling AppleWorks to recognize the /RAM volume as well as use the locked out memory banks (free memory) for expanding the AppleWorks Desktop.

PARTITION is not difficult to use. Just follow these step by step instructions.

1.   Make BACKUP copies of your AppleWorks disks (modified or unmodified) and the Super AppleWorks Desktop Expander disk. Do not write protect the copies. Put your originals in a safe place!

2.   Boot side one of the copy of the Super AppleWorks Desktop Expander in your //c.

3.   Select the "Exit to BASIC" option.

4.   At the Applesoft BASIC prompt (**]**),type:

   RUN PARTITION<return>

5.   Read the brief paragraph on the screen.

   If you wish to modify a PRODRIVE program file only, select option O.

   If you wish to modify a PRODRIVE program file and an AppleWorks STARTUP disk, select option A.

The bar graph that appears represents, from left to right, the amount of free memory available for program use, the amount of memory allocated for ProDrive use, and the amount of uninstalled memory, up to 3 megabytes. (This program is also used to configure ProDrive for use on the 3 megabyte RamWorks II for the Apple IIe.)

6. Use the left and right arrow keys to adjust the desired free memory and ProDrive sizes.. Press the return key when completed.

7. Insert the Write enabled Super AppleWorks Desktop Expander disk containing the PRODRIVE file to be modified into the internal drive. At the ? prompt type:

> **/AW.EXP/PRODRIVE**<return>

> Or, if you have changed the volume directory or filename of either of the above, enter the appropriate ProDOS pathname.

Note: If you encounter an error that results in **a PATH NOT FOUND, SYNTAX ERROR, OR WRITE PROTECTED** message, type RUN<return> at the BASIC prompt —or re-boot and begin again at step 5 of this procedure.

8. After the disk-use light goes out, remove the Super AppleWorks Desktop Expander disk and insert the write enabled AppleWorks STARTUP disk into the internal drive. At the ? prompt type:

> **/APPLEWORKS/APLWORKS.SYSTEM**

> Or, if you have changed the volume directory or filename of either of the above, enter the appropriate ProDOS pathname.

When executed, the modified PRODRIVE program will now install the /RAM volume in the banks above the ones used by the patched AppleWorks. The modified AppleWorks STARTUP disk is now patched to use only up to the amount of memory you specified in the PARTITION program.

COPY.FILES Instructions

Included on side one of the Super AppleWorks Desktop Expander Disk is an Applesoft BASIC utility program, COPY.FILES, which will allow you to list the specific files you wish to automatically load into your ProDrive. This program can also load files from several different ProDOS volumes. Some knowledge of BASIC programming is required to enter the prefixes and filenames of the files to be copied.

Selecting the "Copy Files to ProDrive" option from the "ProDrive options menu" of the Super AppleWorks Desktop Expander will install and initialize a ProDrive, RUN the COPY.FILES program, and copy the files specified in the program's DATA statements (program lines 10010 to 10390) to /RAM.

The COPY.FILES program can also be RUN from Applesoft BASIC, but only after the ProDrive has been created. (It does not automatically install a Prodrive.) If the ProDrive has not been installed, COPYFILES will attempt to copy the specified files into the 64K ProDOS /RAM volume. This program also assumes the ProDrive volume directory name is /HAM.

You may select the volumes and files to be copied by changing the DATA statements in the COPY.FILES program. To illustrate the proper format of the DATA statements let us use the sample statements currently in the program. Boot the Super AppleWorks Desktop Expander disk, choose the "Exit to BASIC" option, and,

at the BASIC prompt ( ]), type: **LOAD COPY.FILES.**   Then type:
LIST 10000,10100. Line 10010 contains the number of files to be copied from
the volume indicated after the comma in the same line. This line establishes the
ProDOS prefix for the filenames listed in the next 4 DATA statements. The volume
name must include the reading and ending "/" pathname delimiters as in the example
below.

```
]10010        DATA  4,/AW.EXP/
]10020        DATA  BASIC.SYSTEM
]10030        DATA  FILER
]10040        DATA  PRODRIVE. DOC
]10050        DATA  COPY. FILES. DOC
]10060        DATA  0,/END!
]10070        DATA  110390 DATA
```

Line number 10060 in this example informs the COPYFILES program that there are
no more files to copy. This statement is required to properly end the COPY.FILES
program. If you were to change this line to read **10060 DATA 5,/XYZ/** the
program would prompt you to insert the volume named "/XYZ" and proceed to copy
from it the 5 files you had listed in the next four DATA statements.

Change the DATA statements to contain the prefixes and filenames of the files you
wish to copy to /RAM and save the program back to the disk. DATA statements
following the program line containing **DATA 0,/END/** are superfluous and can be
deleted. The program may be saved to another disk, but the GET.LEN file, required
by COPYFILES, must also be copied from the Super AppleWorks Desktop Expander
disk to any disk containing COPY.FILES.

AppleWriter *II* Patch

The following Applesoft BASIC programs can be used to modify a **BACKUP** copy of the ProDOS versions of AppleWriter *II*. (The AppleWriter version number is displayed on the AppleWriter copyright screen.) Boot side one (ProDOS side) of the Super AppleWorks Desktop Expander and select the "Exit to BASIC" option. Remove the Super AppleWorks Desktop Expander disk from the internal drive and replace **it** with the AppleWriter BACKUP disk. Enter the appropriate command sequence below. (The bracket at the beginning of each line represents the Applesoft BASIC prompt. It is not part of the statement!)

AppleWriter // version 2.0 (See your Apple dealer for details on update to 2.1.)

    ]PREFIX /AW2MASTER <return>
    ]UNLOCK AWD.SYS <return>
    ]BLOAD AWD.SYS,TSYS,A$2000 <return>
    ]POKE 8484,243 <return>
    ]POKE 8486,179 <return>
    ]BSAVE AWD.SYS,TSYS,A$2000 <return>
    ]LOCK AWD.SYS <return>
    ]PR#6 <return>

AppleWriter// version 2.1

    ]PREFIX /AW2MASTER <return>
    ]BLOAD AWD.SYS,T$0C,A$2000 <return>
    ]POKE 8484,243 <return>
    ]POKE 8486,179 <return>
    ]UNLOCK AWD.SYS <return>
    ]BSAVE AWD.SYS,T$0C,A$2000 <return>
    ]PR#6 <return>

Z-Ram II Technical Reference

Introduction

This chapter provides the information required to add and configure memory on your Z-Ram II. It contains the specifications for the RAM (Random Access Memory) chips, tells where you can get them, and then, how to install them. Special information for assembly language programmers, concerning the use of Z-Ram II extended auxiliary memory, is also provided in this chapter.

Adding Z-Ram II Memory

To install additional memory chips on the Z-Ram II card, it will be necessary to remove the card from the computer and replace the styrofoam pin-protector blocks onto the gold pins protruding from the bottom of the Z-Ram II card. These blocks will help prevent accidental damage to the pins when the new memory chips are being pressed into their sockets.

> **Warning!** Don't ZAP your chips! There are many static sensitive components in the //c and on the Z-Ram II. Be very careful to discharge any static electricity on your body by touching the //c power supply case inside The //c before touching any of the components!

Memory Chips

Selection of the proper RAM chips for Z-Ram II can be tricky. There are several chip manufacturers and many different types and specifications of chips available. Z-Ram II can use either 64K or 256K RAM Dual In-Line Package (DIP) chips. (256K chips are recommended.) One block of the Z-Ram II memory uses a very special, limited production 24 pin dual-sided 256K x 8 Single In-Line Package (SIP) RAM chip. All memory chips must be DYNAMIC RAM chips and have a speed specification of 200 nS (nanoseconds) or faster. Applied Engineering recommends a rating of 150 nS.

The 64K and 256K DIP memory chips with the correct specifications can be obtained at some computer or electronics parts stores. Or you can get them through Applied Engineering, probably at lower cost. The 256K x 8 memory chip is available commercially but it is extremely expensive when purchased individually. Since Applied Engineering gets bulk-quantity discounts on its purchases of memory chips, they can offer them at lower prices. Applied Engineering uses only the highest quality memory chips and will warrant them for 5 years. (Most electronics parts vendors sell their chips "as is".) With Applied Engineering memory chips you are also assured of getting the right chips.

Chapter 4

Memory Chip Installation

You may return your Z-Ram II to Applied Engineering for a memory upgrade with no additional charge for the installation or testing. Call Applied Engineering Technical Support for the latest memory chip prices and get a Return Material Authorization (RMA) number. The Technical Support telephone number is (214) 241-6069.

The Z-Ram II card can have from 256K to 960K of memory installed using 64K chips, 256K chips, or a combination of both. The memory on Z-Ram II is organized into 4 "blocks", designated block A, B, C, and D. Illustration 4-1 shows which memory chip sockets are allocated to each block. Block A contains 8 soldered-in 256K memory chips. Blocks B or C must be either completely empty or completely populated with 8 chips of the same memory size. NEVER mix 64K chips and 256K chips in the same block! Block D consists of the single 256K x 8 SIP memory chip.

Illustration 4-1 **Z-Ram II Memory Blocks**

The DOT indicates Pin 1 end
of SIP memory chip.
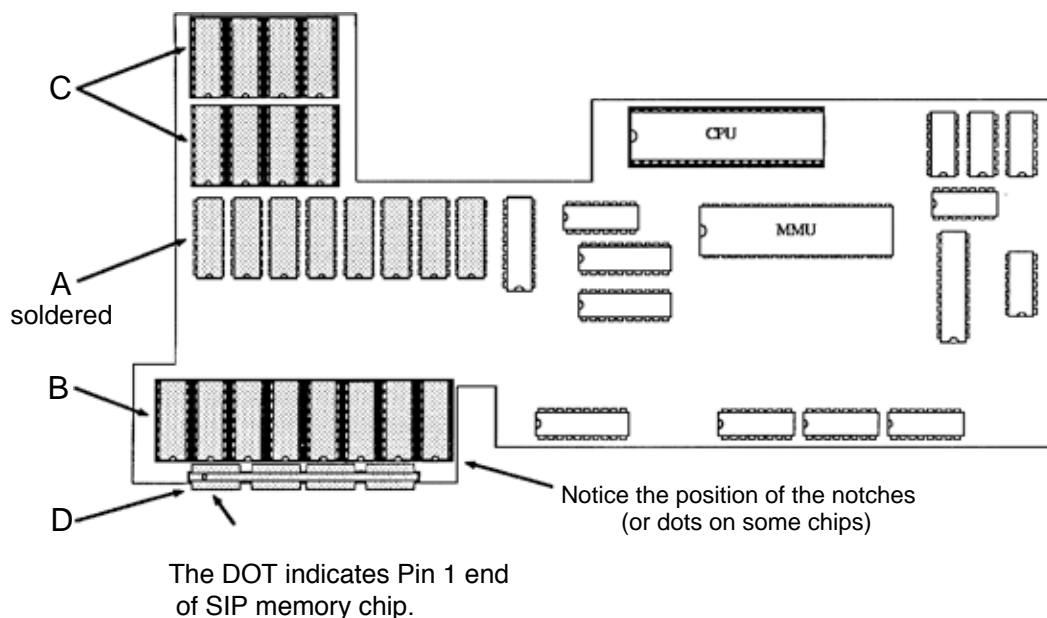
Illustration 4-I shows the correct orientation of the RAM chips on the Z-Ram II card. Note the position of the notch (or dot on some chips). Make sure all of the RAM chip pins are centered over their respective holes in the chip socket and press the chip gently but firmly into its socket.

It is extremely important that all RAM chips are inserted properly. If power is applied to a chip that is plugged in backwards, Z-Ram II will probably not be damaged, but the chip will be ruined immediately. Verify that each RAM chip is in the correct position and has all its legs fully seated in the socket. Z-Ram II is now ready for installation into your Apple //c. Installation instructions are provided in Chapter 1.

Memory Bank Configurations

Table 4-1 shows the relationships between memory blocks, banks, and memory chip sizes for the Z-Ram II. From this table you can determine the addressable 64K banks for any possible configuration of memory chips. Z-Ram II is restricted to 16 addressable auxiliary memory banks. Bank 0 is the 64K bank of the //cs auxiliary memory, leaving 15 remaining bank locations available for Z-Ram II extended auxiliary memory. For this reason, the last bank in block D is not used.

Table **4-1 Z-Ram II Bank Configuration Chart**

Block	RAM Chip Size	Bank Numbers (decimal notation)			
A	256K	03	07	11	15
B	64K	01		.	
	256K	01	05	09	13
C	64K	02	.	.	
	256K	02	06	10	14
0	192K	04	08	12	

Another way to determine what banks are available on your Z-Ram II, is to run the Z-Ram Memory Test on the Super AppleWorks Desktop Expander Disk. The test will display the valid auxiliary memory banks found on both the //c and Z-Ram II. (Please note that the bank numbers are referenced in hexadecimal notation.)

For Programmers Only...

Since Z-Ram II is completely compatible with all software for the //c, the purpose of this chapter is to describe only the feature which is unique to Z-Ram II, extended auxiliary memory. This material is definitely intended for assembly language programmers. Additional information concerning the access and control of auxiliary memory can be found in the Apple //c Reference Manual.

Warning: Don't attempt to use the auxiliary memory directly from an interpreter such as BASIC or Pascal. These interpreters use certain portions of main memory that, when switched to auxiliary memory can cause your program and the interpreter to 'bomb.

Bank Switching

The 65C02 microprocessor has the memory address capability of only 64 kilobytes. In order to address more memory, a feature built in to the Apple //c's firmware called "bank switching" is used. Soft switches control whether the 65C02 is addressing the 64K bank

of main memory or the single 64K bank (bank 0) of auxiliary memory available on the //c main logic board. In order to use more than just 64K of auxiliary memory, Z-Ram II combines this bank switching technique with its own firmware feature, the bank select register. The bank select register determines which of 16 possible 64K banks of auxiliary memory the 65002 is addressing.

The bank select register is mapped into the //c's memory space at location $0073 (49267). The programmer can select one of 16 valid banks by writing the bank number into this location. (Table 4-1 provides the bank numbers for various memory configurations.) Once the bank selection has been made, soft switches built into the //c firmware function as they normally would. This allows data to be transferred from main memory to auxiliary memory and vice versa.

Data transfers between banks of auxiliary memory must be done in two stages. The first step is an intermediate transfer to main memory, then from main memory to the desired bank(s). Transfer can also be accomplished, one byte at a time, using the Accumulator.

Bank 0 in Z-Ram II always contains the text information for the 80 column display and the graphics information for the double high resolution display. This bank must be active whenever the program updates the display screen.

The bank select register is initialized to zero on a power-up, but not after a reset. Please refer to the programming suggestion on "Reset Vector".

The location of the bank select register is also shared with the system. Writing to the bank select register will also trigger the paddle strobe, which is used to read the paddle inputs. To insure the paddles are read correctly, wait at least 3 milliseconds after changing banks before starting the paddle read routine.

The contents of the bank select register cannot be read. The program must keep track of the current bank number in a reserved location within each bank.

Programming Suggestions

Interrupts

Programmers should be aware that Z-Ram II has multiple interrupt vectors. Since the interrupt vector is located at $FFFE, each auxiliary bank of memory contains an interrupt vector. Auxiliary memory may be switched in when an interrupt occurs; therefore, routines that use auxiliary memory should be prepared to disable or process interrupts. It is recommended that interrupts be disabled in programs not requiring the their use. Please refer to the Apple//c Reference Manual for details on handling interrupts.

Reset Vector

All programs should start by initializing the bank register to 0 (video bank). Since Z-Ram II cannot detect a hardware reset, the software should be able to handle a reset by storing a 0 in the bank register. The following assembly language subroutine intercepts the reset soft vector at $3F2-$3F3 and changes it to point to the new reset routine.

```
*Store off old reset soft vector
        LDA         $3F2
        STA         $300          ;IN $300.301
        LDA         $3F3
        STA         $301
*Set reset vector to point to $302
        LDA         #$02
        STA         $3F2          ;($3F2 -> $302)
        LDA         #$03
        STA         $3F3
        EOR         #$A5          ;Inits new validity check byte
        STA         $3F4
*Reset routine
        LDA         #$A9          ;LDA #0
        STA         $302
        LDA         #$0
        STA         $303
        LDA         #$8D          ;STA $C073
        STA         $304
        LDA         #$73
        STA         $305
        LDA         #$C0
        STA         $306
        LDA         #$60          ;JMP ($300)
        STA         $307
        LDA         #$00
        STA         $308
        LDA         #$30
        STA         $309
```

Available Banks   The following subroutine will find all valid auxiliary memory banks, determine the size of Z-Ram II memory, and save the results into BankTbl. BankTbl will have the number of 64K banks found, followed by the table of valid bank numbers. Keep in mind that the bank numbers are not linear.

For example: BankTbl:09 00 01 03 05 07 09 0B 00 0F' indicates a 512K Z-Ram II containing banks 1,3,5,7,9,8,0, and F. Bank 0 is the //c auxiliary memory bank.

```
            BankSel      EQU          $C073

       *Write bank number to each bank
                         STA          $C009        ;Store in alternate zero page
                         LDY          #$7F         ;Valid banks range $00 to $7F
            FindBanks    STY          BankSel      ;Go through each bank
                         STY          $00          ;Store the bank number
                         TYA
                         EOR          #$FF
                         STA          $01          ;Second self-check
                         DEY
                         BPL          FindBanks
       Read them back to find valid banks and save in table
                         LDA          $00
                         TAY
                         TAX
            FindThem     STY          BankSel      ;Search through all banks
                         STA          BankSel+3
                         CPY          $00
                         BNE          NotOne       ;Check bank number
                         TYA
                         EOR          #$FF
                         CMP          $01          ;Check second double-check
                         BNE          NotOne
                         INX
                         TYA                       ;Found valid bank-save in table
                         STA          B an kTbl, X
            NotOne       INY                       ;Go through all valid bank ranges
                         BPL          FindThem
                         LDA          #$00         ;Reset to video bank
                         STA          BankSel
                         STA          $C008
                         STX          BankTbl      ;Size of Ram card
                         LDA          #$FF
                         STA          BankTbl+1 ,X ;Mark end of table
                         JMP          Continue
            BankTbl      DS           50           ;Enough banks for three meg.
```

CP/AM 4.0B

CP/M for Your
Apple //c

The Z-80 microprocessor on the Z-Ram II now enables the Apple //c to run thousands of CP/M programs. In fact, with Apple and CP/M programs combined, you now have access to the largest body of software available. You can also take advantage of the large base of professional programs written for the CP/M operating system.

This chapter is not a tutorial for learning CP/M. It is intended to provide users, already familiar with the CP/M operating system, with the resources to run CP/M software on the Z-Ram II equipped Apple //c. The user is given a brief summary of CP/M commands and an explanation of the utilities and programs provided on the CP/AM 4.0B System Master disk. The CP/AM Technical Reference section of this chapter is intended for hard-core CP/M hackers only. All others, please stay awayl If you need to know more about CP/M, please refer to the list of recommended CP/M reference material in Appendix A.

Applied Engineering's CP/AM, included on the CP/AM 4.02 System Master disk, is the operating system that enables the Z-Ram II equipped Apple //c to run programs written for the CP/M operating system. Disks containing CP/M programs must be in Apple disk format! Programs written for Digital Research CP/M version **2.2,** as well as virtually all older CP/M programs, will run under CP/AM 4.08. Since the primary purpose of the CP/AM 4.0B operating system is to allow the user to run CP/M software on the //c, no developer tools, like an assembler or debugger, are provided, although there are such tools available in the public domain.

Before You BegIn...

Whatever can go wrong, will go wrong I Make a BACKUP copy of the CP/AM 4.08 system master disk. This is not only just a good idea, it is required. Some CP/AM commands require the floppy disk be write-enabled. (The CP/AM 4.08 System Master has no write-enable notch.) Use the following procedure to make the copy:

Boot the CP/AM 4.02 System Master disk. This message should appear on the screen:

>**APPLE][CP/AM**
>**60K VER 4.0B**
>**(C) 1985APPLIED ENGINEERING**
>
>**A>**

At this prompt (A>) type the following command if you have a two-drive system:

>**COPY B:=A:**

If you have a one-drive system type:

>**COPY A:=A:**

The directions, displayed on the screen, will prompt you to insert a SOURCE disk, which is the original CP/AM 4.08 disk, and a DESTINATION disk, which is any blank disk with the write-protect tab removed. For a one-drive copy it will be necessary to alternately insert the SOURCE and DESTINATION disks until the disk is completely copied. Store the original CP/AM 4.02 System Master disk in a safe place.

CP/AM   4.0B
Conventions

The following is a review of the standard CP/M command and file conventions as well as some special notes about CP/AM command syntax used in this chapter. In the command examples shown in this text a carriage return is shown as **<cr>** Control characters are indicated by preceding the command character with a ∼ (circumflex). Spaces inserted in a CP/AM command line are critical to the execution of the command sequence.

> Warning: Don't press CONTROL-RESET while your CP/M program is running! Some programs can be clobbered this way!

File Names

CP/AM file names follow the follow the standard CP/M format. A file name can have from one to eight printable ASCII characters, except for the ones shown below.

<div align="center">

< > . , ; : ? [ ] * =

</div>

File Extensions

CP/AM file name extensions also follow the CP/M standard. An extension can have from one to three printable ASCII characters and cannot contain any of the "disallowed" characters. The extension is separated from the end of the file name by a period. For example:

FILENAME.EXT

Since CP/M programs use the file extension to identify the file type, it is recommended that the following extensions be used only for their intended purpose.

.ASM	Assembly language source file.
.MAC	Assembly language macro file.
.HEX	Hex format source code file.
.REL	Relocatable machine language file.
.COM	Transient command program file.
.PRN	Print file. (Assembly language listing.)
.LIB	Library file.
.SUB	Command list for SUBMIT execution.
.C	C language source code.
.BAS	BASIC source code file.
.PAS	Pascal source code file.
.COB	COBOL source code file.
.FTN	FORTRAN source code file.
.TXT	Text file.
.DOC	Documentation file.
.$$$	Temporary file.

Filename "Wildcards"   A Wildcard character is a symbol which represents a character or a string of characters in either a filename or extension. CP/M uses two such wildcards, the (?) question mark and the (*) asterisk. The asterisk is used to represent either the entire filename , extension, or a trailing character string of a filename or extension. The question mark represents only a single character in either string. The following examples illustrate the uses of wildcards.

DIR A:TED.*

This command displays a directory listing of all files on Drive A that have a filename of TED, regardless of the extension.

DIR *.COM

This will return a listing of all Drive A files which have an extension of ".COM". The drive in this example is the default drive.

DIR B:S.COM

This example will return a listing of all Drive B files with filenames beginning "S" and an extension of ".COM".

DIR A:????.DOC

Since the question mark represents only one character in the string, this command will return a listing of files having an extension of ".DOC" and filenames containing any four (or fewer) characters.

Disk Drive Specifier   When a command is used without a drive specifier, the default drive will be assumed for the operation. The default drive is indicated at the CP/AM prompt by the letter immediately to the left of the caret (>). When the CP/AM system is first booted, the "A>" prompt is displayed. This informs the user that drive A (the //c internal drive) is the default drive and the operating system is ready for a command.

To change the default drive, type the desired drive drive letter followed by a colon and a carriage return. For example, to change the default drive from A to B (the //c external drive), you would use the following command:

A> B:<cr>

Note: Unlike DOS 3.3, you cannot change disks without informing the operating system. CP/AM stores information in memory about the current disk. To let CP/AM know that you have replaced the original, type a control-C at the CP/AM prompt.

Line Editing
Commands

These commands provide some control over the screen display and allow you to correct typing errors. Some commands can be entered by pressing a single key on the keyboard: others require the control key and another to be pressed at the same time. The line editing commands are summarized below.

ˆM <cr> **ˆJ**	Terminate the command line
ˆX	Cancel and erase the command line.
ˆU	Cancel the command line. (Does not erase the command line.)
ˆH **Delete** **left arrow**	Cancel and erase last character of the command line. (backspace function)
ˆE	Continue typing the command line on the next line of the screen.
ˆR	Repeat the current command line.
ˆS	Pause (and resume) the display scrolling.
ˆp	Send output (or cancel output) to printer. (serial port 1)
ˆC	Interrupt the current transient program and perform "warm start" and log in a new disk. Do not confuse warm start with warm boot (Open Apple-Control-Reset.)

CP/AM
Command Summary

There are two types of CP/AM commands, resident commands and transient commands. The resident commands, consisting of system commands and line editing commands, are built into the operating system. The transient commands are actually utility programs stored on disk. This section provides a review of the resident commands and a description of the transient commands included with CP/AM 4.0B.

Resident
System Commands

There are six system commands: DIR, ERA, REN, TYPE, USER, and SAVE. These commands and their proper syntax are summarized below.

DIR

Display Directory

DIR<cr>	Displays a directory of the files on the default drive.
DIR B:<cr>	Displays of the contents of drive B. Drive A remains the default drive.
B:DIR<cr>	Reassigns the default drive to drive B and displays its directory.

ERA   Erase a File

> **ERA B:ABCD.BAS<cr>**
> Erases the file on drive B with a filename of ABCD and an extension of .BAS. Under CP/AM, the drive specifier is mandatory.
>
> **ERA A:*.BAS<cr>**
> Erases all the files on drive A with and extension of .BAS.
>
> **ERA A:*.* A:*.*<cr>**
> Using the asterisk wildcard in this manner erases all files on drive A. As a safety precaution, to prevent accidental disk erasure, the drive specifier and wildcards must be entered twice, as shown.

REN   Rename a File

> **REN NEWNAME.BAS=OLDNAME.BAS**
> Changes the file named OLDNAME.BAS to NEWNAME.BAS

TYPE   Display an ASCII file on the screen.

> **TYPE ABCD.TXT<cr>**
> Displays the ASCII text file named ABCD.TXT on the monitor. The TYPE command will display one page of text at a time. Pressing any key will advance to the next page. The listing of the display can be paused at any time by entering a Control-S (ˆS), and resumed by pressing any key or another ˆS. By pressing ˆC (or any key other than ˆS) while the file is scrolling will abort the listing.

USER   Specify the current user

> **USER 7 <Cr>**
> Reassigns the user designation from the default of User 0 (or previous user) to User 7. There are fifteen possible user areas on a disk, 0 through 15, which are logically separate from one another. The current user is indicated in the command prompt, by the lower case character to the left of the current drive indicator. User 0 is indicated by no character; user 2 is indicated by an "a"; user 2 by a "b". User 7, as in this example, would display this prompt:
>
> **g A>**
>
> The lower case g is the seventh letter of the alphabet, indicating user seven is the current user.

SAVE   Save memory to disk

The SAVE command causes the contents of system memory (the Transient Program Area) to be saved to a file on disk. This command is useful to assembly language programmers, but not to most commercial CP/M software users. The format for this command is:

> **SAVE n filename.typ** where 'n' is the number of 256 byte pages of memory, starting at address 100Hex, to be saved to disk. The following example will save 2048 bytes of the system memory to the file MEMSAVE.XXX.

<p align="center">A>SAVE 8 MEMSAVE.XXX</p>

Transient Commands The routines for the resident commands discussed in the previous section are always in memory. CP/AM transient commands are not built-into the operating system; they are conveniently stored on disk until they are needed. This allows great versatility, since they can be deleted if not needed or copied to or from another disk as required.

Transient command files on disk are identified with a type extension of '.COM'. Since CP/AM will not recognize the transient command as a valid system command, it automatically searches the disk directory for a filename matching the command and having an extension of '.COM'.

There are also many excellent transient command files available from CP/M user groups, public domain software libraries, and commercial dealers. The transient commands provided on the CP/AM 4.08 System Master disk are explained in this section.

PIP   File copy utility

PIP (an acronym for Peripheral Interchange Program) will allow you to copy CP/M files from one device to another. PIP is executed by typing 'PIP' at the command prompt as shown below.

<p align="center">A. PIP<cr></p>

The asterisk prompt which is then displayed below the command line is the PIP command prompt and indicates the program is ready to accept a PIP command expression. The format for the command expression is:

x:newname.typ=y:oldname.typ

This expression will copy the existing file oldname.typ, on drive y, to a newly created file on drive x, named newnametyp. The filename represented by 'newname' is not required

to have the same name as its original.

Example 1:

> *** B:STARTUP.DOC=A:STARTUP.Doc<cr>**
> *****

Here, PIP would copy 'STARTUP.DOC from drive A to the new file named 'STARTUP.DOC on drive B. After PIP has completed the copy, it displays the asterisk, prompting the next command expression. If only a carriage return is pressed at the asterisk prompt. PIP will stop, returning to the CP/AM command prompt.

Example 2:

> **A> PIP B:-A:STAT.COM<cr>**
> **A>**

This is a shortcut. PIP is executed and followed by a PIP command expression in the same command line. Since there is no newname.typ' specified, the oldname.typ' of 'STAT.COM' is used as the default name for the new tile on drive B. Notice that PIP returns to the CP/AM system prompt.

Example 3:

> **A>PIP B:=A:*.<cr>**
> A>

Using the asterisk wildcard, all files will be copied from drive A to drive B. This will copy only files; not the CP/AM system.

If you specify, as shown in Example 4, CP/AM will verify the copied file against the original after the copy has been completed. (The brackets around the V are part of the command.)

Example 4:

> **A>PIP B:=A:STAT.COM[V]<cr>**
> **A>**

STAT   Status

STAT, which stands for status, provides the user with information about one or more files on one or more disks. The information returned by STAT depends on the various arguments which can follow the STAT command.

Example 1:

> **A>STAT<cr>**
> **A: R/W, Space: 12K**
>
> **A>**

49

In this example, only the STAT command was issued, returning the read/write status and the amount of useable space left on drive A. If drive B, or any other drive, had been accessed since the last warm start or cold boot, STAT would also return the status of that drive. R/W indicates that the drive is both read and write enabled. If STAT had returned a status of R/O, this would indicate that the disk has been changed and not logged-in with a ˆC.

Example 2:

       A,STAT B:<cr>
       B: R/W, Space: 112K

       A>

The drive specifier following the STAT command will cause STAT to return the status of only the drive specified.

Example 3:

       A>STAT B :STAT.COM<cr>
       Recs Bytes Pex Acc
        35   5k   1 R/W B: STAT   .COM
       B: R/W   Space: 112K

By specifying a specific (unambiguous) filename. STAT will provide information about that file. The drive specification is optional. In this case, STAT has returned information about the STAT.COM file on drive B. The Recs (records) column shows the number of 128k byte records in the file and the Bytes column indicates the number of bytes in the file. If Bytes divided by Recs doesn't equal eight, don't be alarmed. Close counts. The number of 16K extents (file pieces) in the specified tile is indicated in the Pex column. The Acc column shows the read/write status of the file, itself.

Example 4:

       A>STAT B:*.*<cr>
       Recs Bytes Pex Ace
       28   4k   1 R/W B: COPY .COM
       26   4k   1 R/W B: FORMAT .COM
       18   3k   1 R/W B: RAMDRIVE.COM
       35   5k   1 R/W B: STAT .COM
       B:R/W Space: 122K

This example illustrates a variation of Example 3, using the asterisk (ambiguous) wildcard to select all files on drive B.

FORMAT   Erase and format a blank CP/AM disk.

Before information can be stored on a disk, the disk must be formatted, or prepared to receive that information. This utility allows you to erase all previous data on the disk and format it using the CP/AM format. There are no command expressions for this command. You need only type the command, FORMAT, and specify the the proper disk drive when prompted. A typical FORMAT session, using a two drive system, is shown below.

Example 1:

A> FORMAT<cr>

CP/AM FORMAT VER. 1.0
COPYRIGHT (C) 1985 BY APPLIED ENGINEERING

FORMAT DISK IN WHICH DISK DRIVE --> B

INSERT DISK TO BE FORMATTED AND HIT ANY KEY.
<space barb.

FORMATTING...

DO YOU WISH TO REPEAT THIS OPERATION? N

If you had selected the drive A as your formatting drive, you should have removed the disk containing the CP/AM system and replaced it with the disk to be formatted. When FORMAT terminates, it returns to the CP/AM operating system. Before you type N at the final prompt, reinsert the disk containing the CP/AM operating system into the current drive.

A non-system disk formatted under CP/AM can store 138K of data. A system disk, one that has the CP/AM 4.0B operating system on disk tracks 0 and 1, has a data storage capacity of 126K. FORMAT does not install the operating system on the formatted disk.

COPY   Copy an entire disk.

This utility is similar to PIP except COPY will duplicate an entire disk including the CP/AM operating system. COPY can also be used to copy only the CP/AM 4.0B operating system to a disk already containing CP/M programs. This will eliminate the need for a CP/AM preboot disk.

Example 1:

A>COPY<cr>

CP/AM COPY VER. 1.0
(C) 1985 APPLIED ENGINEERING

*B:=A:<cr>

INSERT SOURCE   DISK IN A:

INSERT DESTINATION DISK IN B:

PRESS RETURN TO CONTINUE
<Cr>
COPYING...

DO YOU WISH TO REPEAT THIS OPERATION? N<cr>

*ˆC
A>

In this example COPY session, notice the COPY command, entered without an argument, returns the asterisk prompt. At this point, the Destination drive = Source drive argument is entered, which is followed by the "insert source/destination disk" dialog. You could then replace the disk in the source drive with the one to be copied and place a blank disk in the destination drive. (A formatted disk is not required.) To exit the COPY utility and return to the CP/AM prompt, type Control-C (ˆC) at the asterisk prompt.

Example 2:

A>COPY A:=A:<cr>

CP/AM COPY VER. 1.0
(C)   1985 APPLIED ENGINEERING

INSERT SOURCE   DISK IN A: AND PRESS RETURN<cr>
INSERT DESTINATION DISK IN A: AND PRESS RETURN<cr>
INSERT SOURCE   DISK IN A: AND PRESS RETURN<cr>
INSERT DESTINATION DISK IN A: AND PRESS RETURN<cr>
INSERT SOURCE   DISK IN A: AND PRESS RETURN<cr>
INSERT DESTINATION DISK IN A: AND PRESS RETURN<cr>
INSERT SOURCE   DISK IN A: AND PRESS RETURN<cr>
INSERT DESTINATION DISK IN A: AND PRESS RETURN<cr>
INSERT SOURCE   DISK IN A: AND PRESS RETURN<cr>
INSERT DESTINATION DISK IN A: AND PRESS RETURN<cr>

DO YOU WISH TO REPEAT THIS OPERATION? N<cr>
A>

This example session illustrates the use an argument following the COPY command. If the source and destination drives had not been the same, COPY would have displayed the dialog screen shown in Example 1. Entering N' at the "...REPEAT THIS OPERATION?" prompt would end COPY and return to the CP/AM system in drive A.

The following example shows how to copy only the CP/AM 4.02 operating system from a source disk to a destination disk.

Example 3:

A>COPY B:=A:/S

This will copy only the operating system from drive B to tracks 0 and 1 of drive A. Install the CP/AM 4.0B operating system only on a backup copy of your original CP/M program disk.

RESTOR   Restore a previously ERAsed file.

If a file or files have been accidentally erased and nothing has been subsequently written to the disk, this utility can be used to recover the file(s). The format for this command is:

A.RESTOR d:oopsname.typ<cr>

The drive specifier (d:) is required only if the erased file does not reside on the current drive. The filename represented by oopsname.typ" is the name of the eased file. After the file is restored, it should be checked for possible damage and repaired.

RAMDRIVE   Install CP/AM RamDrive

With this utility program, the memory available on Z-Ram II, combined with the //c's 64K bank of auxiliary memory, can be used to emulate a CP/AM RamDrive, or emulated disk drive. For more information on RamDrives please refer to the introduction to Chapter 4. The example, below, illustrates a RamDrive emulation on a //c with a 512K Z-Ram II installed.

Example 1:

A. RAMDRIVE<cr>

RAMDRIVE is emulating C: with 576K available

A>STAT C: <Cr>
C: R/W, Space: 546K

A>

In Example 1, RamDrive is installed by entering the RAMDRIVE command. "RAMDRIVE is emulating C:" indicates the RamDrive, designated as drive C, is now installed into 576K (512K + 64K) of available memory. Checking the STATus of drive 0, the emulated disk disk storage available on drive C is 546K.

RAMDRIVE emulates drive C as the default RamDrive. If you have a single drive //c system and wish to assign the RamDrive as drive B, use the command line shown in Example 2 to <u>initially</u> install a RamDrive. If a RamDrive is already installed, the current RamDrive specifier will remain in effect.

Example 2:

A.RAMDRIVE B: <cr>

To copy files to the CP/AM RamDrive, use the PIP utility. COPY will not copy to or from the RamDrive, nor will it copy the CP/AM 4.08 operating system to the RamDrive.

The RamDrive remains enabled until the //c power is turned off or the system is cold booted. The Control-C warm start does not affect the RamDrive. When the system is cold booted (Control-Open Apple-RESET), the RamDrive must be re-enabled by issuing the RAMDRIVE command. Entering the RAMDRIVE command while a RamDrive is already installed has no effect on the files stored in the RamDrive, unless you hold the CLOSED-APPLE key down while the program is executing. This will install an empty RamDrive, erasing any data that was previously stored there. Holding the OPEN-APPLE key down during RAMDRIVE execution will enable the audible disk access indicator. Make sure the speaker volume control is not at full volume! The "BLEAT" of the first RamDrive access may startle you.

Just as the red "disk use" indicator light indicates a disk access in progress, RamDrive provides a visual access indicator. Whenever the emulated drive is accessed, an inverse "R" or 'W' will appear in the lower right corner of the display. The "R" indicates a read operation; the "W" signifies a write operation.

RAMDRIVE also optionally supports the Double High Resolution Graphics feature of the Apple //c. If your CP/M program requires the use of double high resolution graphics, enter the RAMDRIVE command as shown in Example 3. This will install the RamDrive and create a file named "DBLHIRES" on the emulated drive. This file occupies and "protects" the double high resolution graphics display area in the //cs auxiliary memory and can be easily disabled by using the ERA command to erase the file.

Example 3:

A>RAMDRIVE G <cr>

RAMDRIVE is emulating C: with 1024K available

A,.DIR C: <cr>

C: DBLHIRES 32
A>

54

AUTORUN   Set startup.

AUTORUN allows you to Customize a CP/AM boot disk. Whenever an AUTORUN modified system disk is cold booted, the first command line is automatically entered, followed by a carriage return. The example, below, shows how AUTORUN can be used to install a RamDrive whenever the system disk is cold booted.

Example 1:

A>AUTORUN RAMDRIVE<cr>

CP/AM AUTORUN VER 1.0

A>

SUBMIT   Automatic sequential command execution.

Usually, CP/M commands are entered and executed one at a time from the keyboard. By taking its commands from a special file instead of the keyboard, SUBMIT allows you to group several commands together in a "batch" for sequential processing without your intervention.

The commands to be executed must be valid CP/M commands and contained in a CP/M text file with a filetype extension of ".SUB." A SUBMIT command file can be easily created and edited with a CP/M text or line editor. The PIP utility can also be used to create a ".SUB" file. Each line of the **text** file must contain a valid CP/M expression, followed by a carriage return. Example 1 uses the TYPE command to display the contents of a sample submit file on the CP/AM 4.02 System Master disk.

Example 1:

A>TYPE SAMPLE.SUB<cr>
dir
ramdrive
pip c:=A:plp.com
stat C:

A>

When SUBMIT is executed using the SAMPLE.SUB file, as shown in Example 2, the commands in the ".SUB" file are executed automatically, one line at a time. SUBMIT creates a temporary command file on the current disk from your ".SUB" file with the filename of '$@$.SUB" which it then uses to execute the command sequence. This temporary file is erased upon completion of the command sequence.

Example 2:

A>SUBMIT SAMPLE.SUB<cr>

This example uses the SUBMIT file shown in Example ito first display a directory of drive A, the Current drive. Then the following line installs a RamDrive as drive C. In the next line, PIP copies itself into the newly created RamDrive and the last line returns the status of the Ram Drive.

To create a SUBMIT command file using PIP, follow the procedure outlined in Example 3.

Example 3:

A>PIP A:STARTUP.SUB=CON:

-ramdrive<cr>
-pip c:=a:*.com<cr>
-dir c:<cr>
-ˆZ<cr>

A>

The command line in Example 3 creates a file on drive A with the filename of "STARTUP.SUB" and assigns the keyboard or 'CON: as the input device. The commands to be executed are then entered following the PIP's hyphen prompts. Each line is ended with a carriage return. The last line, or "end-of-file." is indicated by a control-Z, followed by a carriage return.

SUBMIT and AUTORUN can be used in combination to create a startup disk which will automatically execute a SUBMIT file when the disk is cold-booted. Example 4 illustrates the AUTORUN command line using SUBMIT and the file created in Example 3.

Example 4:

A>AUTORUN SUBMIT STARTUP.SUB<cr>

CPAM60   Install 60K CP/AM Operating System

CPAM60 will install the CP/AM 4.0B Operating System on a formatted CP/AM disk with 12K or more disk space available. In the event that the CP/AM boot tracks on a system disk are damaged, CPAM60 will reinstall the operating system.

This command is similar to the "/S" feature of the COPY command except that CPAM60 installs a totally unmodified operating system to the disk in drive A instead of copying the current operating system to a specified disk. The CP/AM 4.0B operating system is contained in this program, but CPAM60 is n~ the operating system file loaded into memory when the system disk is booted.

Example 1:

A,CPAM60<cr>

 APPLE][OP/AM 4.0B
60K DISK UPDATE PROGRAM
(C) 1985 APPLIED ENGINEERING

INSERT DISK INTO DRIVE A:
AND PRESS RETURN TO BEGIN
<cr>

Upon successful completion of the installation, the following message is displayed:

 PRESS RETURN TO RE-BOOT SYSTEM

Non-Standard Printer Port Configuration The CP/AM 4.0B operating system uses the default Product Identification Number (PIN) of the //c serial port I for output to the printer. Each time the //c is turned on, the following port 1 PIN is in effect:

  (Default) Product Identification Number: 166/1124

  (1) Printer mode
  (6) 8 Data bits/2 stop bits
  (6) 9600 baud (bits per second)
  (1) No parity
  (1) No echo of output to screen
  (2) Insert line feed after carriage return
  (4) Insert carriage return after 80 characters

If your printer does not work automatically with the default PIN, use the Apple /Ic System Utilities disk to reconfigure the PIN. The instructions for changing the PIN are in the Apple tic Owner's Manual or Apple //c System Utilities manual.

If you are required to use a non-default PIN setting, you must boot the properly configured Apple //c System Utilities disk prior to booting into the CP/AM 4.0B operating system. Remember, a warm boot (Control-Open Apple-Reset or PR#6) will not erase the PIN currently in effect. A cold boot (power-off/power-on) will reset the PIN to the default setting.

Public Domain Programs on CP/AM 4.0B System Master disk:

For your convenience, Applied Engineering has included some useful utility programs on the CP/AM 4.0B System Master disk. These programs are public domain software and are not supported by Applied Engineering.

The documentation for these public domain programs is also included on the disk. The documentation can be displayed on the screen by using the TYPE command and echoed to the printer by toggling the Control-P command. The PIP command can also be used to output all of the documentation files to your printer by using the following command line.

A>PIP LST:=A:filename.DOC<cr>

In this command line, "filename.DOC" represents the unambiguous filename of the documentation file you wish to print.

CP/AM 4.0B Technical Reference

Important!   This section is intended for *Z-80 machine language programmers* only! All others should stay away or risk mental meltdown. The technical information about CP/AM 4.0B contained in this section is provided specifically for those programmers using CP/M program development tools. (These tools are not provided with the CP/AM 4.0B System Master disk.)

CONFIGIO   Configure Input/Output Parameters

This command utility allows the advanced CP/M machine language programmer to reconfigure the CP/AM operating system by redefining selected operating system parameters. The screen function definitions can be modified, keyboard characters can be redefined, and user specified system I/O drivers can be installed. All values, displayed and input, are in hexadecimal notation. CONFIGIO menu options 1,2, and 3 modify the parameters of the operating system currently in memory. Option 4 allows the current operating system I/O configuration block to be written to (or read from) a CP/AM system file on disk.

Menu Option 1, Configure CP/AM Terminal Emulation. The Terminal Screen Function Definition screen displays the current software and hardware screen function definitions. The CP/AM 4.02 Operating System is initially configured to emulate the SOROC 10 120/ IQ 140 terminals and, if changed, can be reset to this default by selecting option M.

The software definition column specifies the hexadecimal character string required of the program to accomplish the desired screen function. The hardware definition column specifies the character sequences, issued by the operating system, to be sent to the terminal upon receipt of the software definition sequences. The Lead-In character option allows the lead-in character to be specified for all screen functions requiring a lead-in character. The XY Coordinate Offset option and the XY Transmit Order option are used in combination to establish the operating system cursor coordinate protocol. (Table 5-3 provides the memory locations associated with the Screen Function Definitions.)

Menu Option 2, Redefine Keyboard Characters. This configuration option was originally designed for earlier Apple II computers which did not have all the keys necessary to run CP/M software. While this is of little use to Apple *//c* users, up to six keyboard characters can be redefined. As the initial default, there are no keys redefined. (Table 5-2 contains the memory locations associated with the original keyboard character and the redefined character.

Menu Option 3, Load User I/O Drivers. Using this option, special user-written system I/O driver files can be loaded into the operating system currently in memory. Please refer to the paragraph entitled 'Custom CP/AM Input/Output Drivers," for details on the proper file format.

Menu Option 4, Read/Write I/O Configuration Block. This option allows the current operating system configuration block to be written to the CP/AM system file of a disk. It also allows the configuration block to be read from a disk's system tile.

Table 5-1: **Z-80 / 65002 Memory Address Translation** This table provides the corresponding memory maps for the Z-80 and 65002 processors.

Z-80 ADDRESS	65002 ADDRESS
0000H-0FFFH	$1000-$1FFF
1000H-I FFFH	$2000-$2FFF
2000H-2FFFH	$3000-$3FFF
3000H-3FFFH	$4000-$4FFF
4000H-4FFFH	$5000-$5FFF
5000H-5FFFH	$6000-$6FFF
6000H-6FFFH	$7000-$7FFF
7000H-7FFFH	$8000-$8FFF
8000H-8FFFH	$9000-$9FFF
9000H-9FFFH	$A000-$AFFF
A000H-AFFFH	$B000-$BFFF
B000H-BFFFH	$D000-$DFFF
C000H-CFFFH	$E000-$EFFF
D000H-DFFFH	$F000-$FFFF
E000H-EFFFH	$0000-$0FFF
F000H-FFFFH	$0000-$0FFF

Table 5-2: **Keyboard Redefinitions**

Compared Character Location	Redefined Character Location
F3ACH	F3ADH
F3AEH	F3AFH
F3B0H	F3B1H
F3B2H	F3B3H
F3B4H	F325H
F386H	F387H

Table **5-3: Screen Function Definitions**

Software	Hardware	Description
F396H	F3A1H	The cursor coordinate offset is contained in the least significant 7 bits (0-6) of this byte. The most significant bit (7) is the transmit order of the X and Y coordinates. If bit 7 is zero the Y coordinate is transmitted first. If it is a one, the X coordinate is first.
F397H	F3A2H	Lead-in character. Zero if no lead-in.

Note: In following bytes, the most significant bit (7) specifies a lead-in character is required for the given function. Bits 0-6 contain the character that corresponds to that particular function. A byte value of zero indicates the given function is disabled.

F398H	F3A3H	Clear screen
F399H	F3A4H	Clear to end of page
FS9AH	F3A5H	Clear to end of line
F39B H	F3A6H	Normal text
F39CH	F3A7H	Inverse text
F39DH	F3A8H	Home cursor
F39EH	F3A9H	Address cursor
F39FH	F3AAH	Move up one line
F3A0H	F3ABH	Non-destructively move cursor forward

Custom CP/AM Input/Output Drivers

The following is the 110 driver file format required by CONFIGIO:

Byte:
1	Number of custom drivers contained in this file.
2	LS (least significant) byte of destination address of subroutine code.
3	MS (most significant) byte of destination address of subroutine code.
4	LS byte length of subroutine code.
5	MS byte length of subroutine code.
6	Driver type (1 or 2)
Type 1	(Unchained drivers.)
7	I/O vector number to be modified. (see Table 5-4.)
8	LS byte of address to be placed into I/O vector defined in byte 7.
9	MS byte of address to be placed into I/O vector defined in byte 7.
Type 2	(Chained drivers.)
7	I/O vector number to be modified. (see I/O Vector table.)
8 in	LS byte of address to which the current contents of I/O vector indicated in byte 7 is to be loaded.
9 in	MS byte of address to which the current contents of I/O vector indicated in byte 7 is to be loaded.
A...n	Driver object code, the length of which is specified by bytes 4 & 5.
n+1	Beginning of the next custom driver.

Table 5-4: **Input/Output Vectors**

Vector Number	Vector Address	Vector Name	Description
1	0F380H	Console Status	Returns OFFH in register A if a character is ready to read and OOH if otherwise.
2	0F382H	Console Input #1	Read a character from the console into the A
3	0F384H	Console Input #2	Register
4	0F386H	Console Output #1	Sends a character in register C to the console
5	0F388H	Console Output #2	device. Register B contains zero if a standard character or if sending out a screen function command it contains the screen function number.
6	0F38AH	Reader Input#1	Reads a character from the reader device into
7	0F38CH	Reader Input#2	register A.
8	0F38EH	Punch Output #1	Sends the character in register C to the
9	0F390H	Punch Output #2	punch device.
A	0F392H	List Output #1	Sends the character in register C to the printer
B	0F394H	List Output #2	device.

Logical to Physical I/O Devices

The CP/AM **IOBYTE** at location 0003H can be used to redirect the logical input and output devices to different I/O vector table locations. The IOBYTE has the following structure:

LIST		PUNCH		READER		CONSOLE	
7	6	5	4	3	2	1	0	<- BitPositions

Console field

    0   Console Input #1 and Console Output #1
    1   Console input #1 and Console Output #1
    2   Reader Input #1 and List Output #1
    3   Console Input #2 and Console Output #2

Reader field

    0   Console Input #1
    1   Console Input #1
    2   Reader Input #1
    3   Reader Input#2

Punch field

     0   Console Output #1
     I   Punch Output #1
     2   Reader Input #2 (data is output through this vector)
     3   Punch Output #2

List field

     0   Console Output #1
     1   Console Output #1
     2   List Output #1
     3   List Output #2

The area within the CP/AM operating system, reserved for user installed device drivers, is located from F200H to F37FH. Re aware that the RAMDRIVE program utilizes the I/O block area from F280H to F37FH.

Here is an example of a device driver routine to remove line feed characters:

```
;List device driver.
;
            ORG   100H                 ;Load subroutine at 100H.
ORIGIN   EQU   0F300H               ;Actual origin of driver.
OFFSET  EQU   ORIGIN-START
;
            DB    1                    ;Number of custom drivers.
            DW    0F300H               ;Address to load driver.
            DW    PGMEND-START         ;Length of driver code.
            DR  2                      ;Driver patch type.
            DB    0AH                  :1/0 vector number.
            DW    OUTPUT+OFFSET+1  ;Address to store current vector contents.
            DW    START+OFFSET         ;New I/O vector address.
;
START:   LD    A,C                  ;Move output character to register A.
            CP    0AH                  ;Check for line feed character.
OUTPUT: CALL  NZ,0000H             ;If not line feed, send character to printer.
                                       ;If line feed, not send to printer. Continue.
            RET                         ;Return to calling program.
;
            END
```

The object code for this driver would be:

01 00 F3 07 00 02 0A 04 F3 00 F3 79 FE 0A C4 00 00 09

63

Calling 65C02
Subroutines

Address (65C02) of subroutine must be placed in memory location F300H. Register contents required by the subroutine can be placed in the Register Pass Table. (see Table 5-5.) Then the following two instructions must be executed:

    LD HL.(0F3DEH)
    LD (HL),A

When the above instructions are executed, the Z-80 is temporarily disabled and the 65C02 registers are loaded from the corresponding Z-80 memory locations in the register pass table. Upon completion of the 65C02 subroutine, the contents of the register pass table will be the results, if any, of the subroutine.

Table 5-5: **Register Pass Locations**

Z-80 Address	65C02 Address	Description
F045H	$0045	65C02 A register.
F046H	$0046	65C02 X register.
F047H	$0047	65C02 Y register.
F048H	$0048	65C02 P (status) register.
F049H	$0049	65C02 S (stack) register.
F300H	$03D0	65C02 subroutine address.
F3DEH	$030E	Z-80 Slot address.

This is a sample Z-80 call to a 6502 monitor subroutine to beep the Apple speaker:

```
;Z-80 subroutine to beep the Apple speaker
;
Z80         EQU    0F3DEH      ;Z-80 address.
A$SUB       EQU    0F3D0H      ;65C02 subroutine pointer.
REG$A       EQU    0F04SH      ;65C02 A register.
BEEP        EQU    0FF3AH      ;Address of 65C02 subroutine.

START:      XOR    A           ;Clear the A register.
            LD     (REG$A),A   ;Demonstration of parameter passing.
            LD     HL,BEEP     ;Load HL with address to be called.
            LD     (A$SUB),l-IL ;Store 65002 subroutine pointer.
            LD     H L, (Z80)  ;Get Z-80 address.
            LD     (HL),A      ;Write to Z-80 address.
;
;65C02 subroutine executed and Z-80 reactivated.
;
;
            RET                ;Return from Z-80 subroutine.
;
            END
```

For More Information...

in order to Install and use your Z-Ram II. you should be at least familiar with the features and operation of your Apple //c system. Be sure you have read the **Apple //c Owner's Manual**, included with your Apple //c system. Additional valuable information about your //c hardware and software the following books are recommended:

Apple //c Reference Manual (Apple Computer, Inc.) Contains nuts and bolts information about the bits and bytes of the //c. A must for serious programmers.

Applesoft Tutorial (Addison-Wesley Publishing 1-800-447-2244) is a step by step guide for first-time Applesoft BASIC programmers.

Applesoft BASIC Reference Manual (Addison-Wesley Publishing 1-800-447-2244) gives in depth details on the Applesoft BASIC programming language. It is intended for serious programmers.

AppleWorks Reference Manual (Apple Computer, Inc.) describes all the features of AppleWorks. Is included in AppleWorks program package.

AppleWorks (Microsoft Press), by Charles Rubin. is an excellent presentation of AppleWorks.

AppleWorks: The Program for the Rest of Us (Scott, Foresman, and Company), by Michael L. Sloan, explains not only the features of AppleWorks, but also how to increase its usefulness.

ProDOS User's Manual (Apple Computer, Inc.) provides an overview of Apple's Professional Disk Operating System (ProDOS) and explains how to use the ProDOS User's Disk.

Beneath Apple ProDOS (Quality Software), by Don Worth and Pieter Lechner, provides additional information about ProDOS for both the novice Apple user and the advanced programmer.

Basic Programming wIth ProDOS (Addison-Wesley Publishing 1-800-447-2244) gives a more detailed explanation of ProDOS and its commands.

The DOS User's Manual (Apple Computer, Inc.) provides an overview of Apple's Disk Operating System (DOS 3.3) and explains how to use the DOS 3.3 System Master Disk.

The DOS Programmer's Manual (Apple Computer, Inc.) describes DOS 3.3 and its commands and file structure.

Beneath Apple DOS (Quality Software), by Don Worth and Pieter Lechner, is an in-depth presentation of DOS in an interesting and understandable format.

For those interested in learning more about the CP/M operating system and programs, the following books re recommended:

The CP/M Handbook (Sybex, inc. 1 -800-227-2346), by Rodney Zaks

CP/M and the Personal Computer (Addison-Wesley Publishing 1 -800-447-2244) by Dwyer and Critchfield

How to Get Started With CP/M (Dilithium Press) by Carl Townsend

Introduction to Wordstar (Sybex, Inc. 1-800-227-2346), by Arthur Naiman

dBASE II Users Guide (Green/Prentice-Hall)

Microsoft BASIC(Dilithium Press 1-503-648-2713)

Wordstar 3.3x on the Apple //c

This appendix pertains specifically to Wordstar versions 3.30, 3.31, 3.32, and 3.33.

CWS33X   CWS33X <u>must</u> be executed prior to executing Wordstar on the Apple //*c.* The CWS33X program modifies certain parameters within the CP/AM operating system in memory, making CP/AM 4.08 and Z-Ram II compatible with Wordstar. The CWS33X modifications to the operating system in memory remain in effect until the operating system is rebooted or a new system disk is logged in.

Auto-loading   The SUBMIT command and the STARTWS.SUB file on the CP/AM 4.08 System Master
Wordstar files   disk can be used to automatically execute CWS33X, install a RamDrive, copy (PIP) Wordstar files to RamDrive, and execute Wordstar from the RamDrive. To do this you must first create a Wordstar startup disk consisting of the following files

> CP/AM 4.0B System Master files:
> (The CP/AM 4.08 60K operating system.)
> CWS33X.COM
> PIP.COM
> SUBMIT.COM
> RAMDRIVE.COM
> AUTORUN.COM
> STARTWS.SUB
> Wordstar files:
> WS.COM
> WSOVLY1.OVR
> WSMSGS.OVR

Boot the Wordstar startup disk and enter the following command line at the "A>" prompt:

> A>AUTORUN SUBMIT STARTWS

The STARTWS.SUB file, as provided on the CP/AM 4.08 System Master, contains the following command sequence:
> **CWS33X\<cr\>**
> **RAMDRIVE\<cr\>**
> **PIP A:=C:WS*.*\<cr\>**
> **C: \<cr\>**
> **WS\<cr\>**

Three additional SUBMIT files are included on the CP/AM 4.08 System Master disk:

STARTALL.SUB which installs a RamDrive, executes CWS33X, loads Wordstar and Spellstar to RamDrive, and executes Wordstar.
STARTSS.SUB which installs a RamDrive, copies Spelistar files to RamDrive and starts Spellstar.
STARTMM.SUB which installs a RamDrive, copies Mailmerge files to RamDrive and starts Mailmerge.

Appendix B

Wordstar Video Displays

Some versions of Wordstar may not display the text on the screen properly. The edit screen should be displayed in normal video. (Light letters on a dark background.) The menus and help screens should be displayed in inverse video. (Light background with dark letters.) If your video display is reversed, use the following procedure to modify the CP/AM operating system:

- Use the PIP program to copy the CONFIGIO.COM program to the "Wordstar system disk," the disk containing the CP/AM operating system you will be using with Wordstar. (CONFIGIO is an acronym for **CP/AM** Input/Output Configuration Program.)

- Execute CONFIGIO from the Wordstar system disk. (**A>CONFIGIO<cr>**)

- Select option **1** <cr> (Configure CP/AM Terminal Emulation) from the main menu.

- Select option **E** <cr> (Lo-Lite Text) from the Terminal Screen Function Definitions menu.

- Specify **N** <cr> to the prompt, "LEAD-IN CHARACTER REQUIRED (Y/N?)."

- Enter: **0F** <cr> (<u>zero</u> F) for the CHARACTER CODE.

- Enter: **H** <cr> (hardware)

- Select option **F** <cr> (Hi-Lite Text) from the Terminal Screen Function Definitions menu.

- Specify **N** <cr> to the prompt, "LEAD-IN CHARACTER REQUIRED (Y/N?)."

- Enter: **0E** <cr> (<u>zero</u> E) for the CHARACTER CODE.

- Enter: **H** <cr> (hardware)

- Select option **X** <cr> (EXIT) from the Terminal Screen Function Definitions menu.

- Select option **4** <cr> (READ/WRITE I/O CONFIGURATION BLOCK) from the main menu.

- Specify **W** <cr> to the prompt, "READ-WRITE-EXIT (R/W/X)?

- Press return. (Your Wordstar system disk should already be in the **A**: drive.

- Select option **X** <cr> (EXIT) from the main menu.

The CP/AM 4.0B operating system on your Wordstar system disk is now modified to display normal and inverse Wordstar video screens properly.

The Z-Ram II Memory Test

The purpose of this test program is to verify the basic operation and hardware reliability of the Z-Ram II card. If you encounter problems when running a program that you feel could be Z-Ram II memory related, this test is a good first step in diagnosing the problem.

DO NOT attempt to repair your computer unless you are sure of what you are doing!

To run this test program, turn the computer off, insert the Super AppleWorks Desktop Expander" disk in your internal drive, and turn the computer on. When the main menu appears, select the "Z-Ram Memory Test" option.

During the first part of the test you will see the screen partially liii with characters, followed by several lines of text. As the test continues, the screen will clear, followed by a display of the banks of memory found by the test program. In the upper right hand corner of the screen is a pass indicator" which will show how many times the memory test has been performed. After eight passes the test ends and displays the results on the screen. The message MEMORY TEST PASSED" indicates the Z-Ram II card is functioning correctly.

You may run this program as many times as you wish. The amount of time required to complete this test depends on the amount of memory available on the Z-Ram II.

The best test for the Z-80 (CP/M) feature of the Z-Ram II is the CP/AM 4.0B operating system. Boot a "known good" copy of the CP/AM 4.0B System Master disk, install RAMDRIVE, and use the **"PIP C:=A:.*.*"** command line to copy files to the RamDrive.

If the tests pass and you are still experiencing problems with your program, the next step would be to eliminate the software as a possible source of the problem. Try running different software packages or another copy of the program you are having problems with. Our experience has shown that a large majority of "program crashes" are not necessarily caused by hardware failures but by software problems. Many software publishers offer technical support for software related problems.

If the Z-Ram Memory Test or the CP/M test fails, turn the system off, insure that all of the socketed memory chips and connectors are correctly and securely seated. (Often, just reseating the chips and connectors has a magical healing effect on a sick computer!) Turn the system on and run the test again. If it still fails, contact Applied Engineering Technical Support. Please provide the following information:

> —>   Version number of Super AppleWorks Desktop Expander disk.
> —>   Serial number of your Apple I/c.
> —>   Serial number, original memory configuration, and the current memory configuration of your Z-Ram II card.
> —>   Software package and software revision level.