Checkmate Technology, Inc.

™

MultiRam™ C

The Memory Card with a Future

Users Manual

Checkmate Technology, Inc.

509 South Rockford Dr.
Tempe, Arizona 85281
(602) 966-5802

Port #MR8005—1

AceCalc, MagiCalc and Magic Office are the trademarks of Artsci. Inc.
Apple is the registered trademark of Apple Computer, lnc.
Apple Pascal, AppleWorks, FID, DOS 3.3, aixl ProDOS are trademrks of Apple Computer, Inc.
David DOS is the tradmark of David Data, Inc.
Diversa-DOS is the trademark of DSR. Inc.
FlashCalc is the trademark of VisiCorp/Paladin
IACalc is the trademark of Internaticmal Apple Core
ProntoDos is the trademark of Beagle Brothers
Quark is the trademark of Quark, Inc.
The Spreadsheet is the trademark of A.P.P.L.E.

APPLE COMPUTER INC MAKES NO WARRANYIES, EITHER EXPRESS OR IMPLLIED REGUARDING THE ENCLOSED SOFTWARE PACKAGE, ITSMERCHANTIBILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE FXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED IN SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIAL LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

TABLE OF CONTENTS

GETTING STARTED

USING THE MULTIRAM C CARD

PROGRAMING THE MULTIRAM C CARD

MULTIRAM C SERVICE GUIDE

Preface

HOW TO USE THIS MANUAL

This manual is divided into four parts.

Part 1

GETTING STARTED, should be read first by everyone. Chapter 1 summarizes MultiRam C’s features and minimum hardware requirements. Chapter 2 gives step—by—step instructions for installing MultiRam C and adding memory to the card. Chapter 2 is required reading for those who intend to install their own RAM. Chapter 3 covers testing and troubleshooting procedures.

Part 2

USING THE MULTIRAM C CARD, shows how to use the software included with your MultiRam C Card. Chapter 4 explains how to use MultiRam C’s large memory with AppleWorks. Chapter 5 shows how to use MultiRam C as a fast RAM disk drive for DOS 3.3 and ProDOS. Chapter 6 covers software and hardware compatibility questions.

Part 3

PROGRAMING THE MULTIRAM C CARD, should be read if you intend to develop software using MultiRam C’s memory. If you don’t intend to program MultiRam C yourself, you say skip this section. Chapter 7 details programming techniques required to use the standard //C 64K auxiliary memory bank. Chapter 8 shows how to use all of MultiRam C’s 64K auxiliary memory banks.

Part 4

ThE MULTIRAM C SERVICE GUIDE, should be read by everyone. Chapter 9 describes the MultiRam C warranty. Chapter 10 tells how you can get service if you have a problem and explains RAM replacement and software update policies.

This manual assumes you know how to operate your Apple //c. If you are a new Apple //c owner or if any terms or procedures used in this manual are unfamiliar to you, please refer to your Apple //c owner’s manual for help.

Part 1

GETTING STARTED

Getting Started describes the MultiRam C Card's features and instructs you in installing and testing the card.

Chapter 1

INTRODUCTION

The MultiRam (tm) C Card is an expandable memory card for the Apple //c that brings great power and versatility to your Apple.

MultiRam C’s memory is expandable from 256K to 512K. When MultiRam C is installed inside an Apple //c, the card’s memory is added to the Apple //c’s 128K built-in memory to give the //c up to 640K of total memory.

MultiRam C fits inside the Apple //c, underneath the keyboard. The card simply plugs into the //c motherboard and requires no special tools, jumpers or soldering to Install. MultiRam C’s low power design means the card will cause no heat buildup inside the case and will allow a battery pack to be used with the //c for lengthy sessions.

Memory can be easily added to MultiRam C in either 256K or 64K increments. MultiRam C is shipped with a minimum of 256K memory installed. Instructions for installing and testing your own RAM on the card are covered later in this manual.

The MultiRam C Card is 1001 compatible with all existing //c software. Some programs currently will find and use most or all of MultiRam C’s memory. In the near future, several new and powerful business programs from third party software publishers will become available fur the MultiRam C that will use all of MultiRam C’s memory.

The best selling program for the Apple. AppleWorks, can now use all of MultiRam C’s memory. MultiRam C and its AppleWorks expansion software greatly expand the Desktop that is normally 55K for AppleWorks with an ordinary 128K Apple //c. A 512K MultiRam C Card, for example, offers usable AppleWorks Desktop space of 413K! By expanding AppleWorks memory limitations, MultiRam C truly turns AppleWorks into one of the best integrated packages available for any microcomputer. Use the MultiRam C Card, AppleWorks, and a hard disk drive together and the Apple //c becomes a serious business machine capable of the moat demanding tasks.

With the RAM disk software included with MultiRam C. programs can be accessed user twenty times faster than ia normal with standard Apple 5 1/4 disk drives. With 256K installed, MultiRam C can take the place of two ordinary Apple disk drives; four if a full 512K is installed. Put your favorite unprotected programs onto a RAM disk and discover how se’ it can be to quickly move between application programs.

FEATURES

MultiRam C offers many standard and optional features. MultiRam C's flexibility insures your Apple //c of a long, productive life.

Standard Features

The MultiRam C package comes with these standard features:

- Two banks of sockets to add up to 512K to the Apple //c.
- A minimum of 256K installed.
- Uses standard 150 nanosecond 256K or 64K RAM chips.
- Easily user upgradeable at any time with three memory configurations possible:

  256K 320K 512K

- Software included to:
  - Configure AppleWorks to use all of MultiRam C's memory.
  - Use MultiRam C as a ProDOS RAM disk emulator.
  - Use MultiRam C as a DOS 3.3 RAM disk emulator.
  - Test MultiRam C memory for defective RAM.
- 100% //c software compatible.
- Software updates available.
- Memory management addressing modifiable through a factory PAL update.
- CMOS low power design.
- Full 5 year warranty on the card and all Checkmate Technology supplied components.

Optional Features

Checkmate Technology offers these options:

- 64K or 256K RAM kits to expand the memory of MultiRam C with tested, 5 year warranted RAM chips.

- Pascal RAN disk emulator software.

HARDWARE REQUIREMENTS

The minimum equipment requirements to use MultiRam C are:

- An Apple //c computer. MultiRam C will not work iN an Apple //e or in the Apple II, II+ or Apple II+ compatible computers.

- A monitor suitable for 80 column display.

Chapter 2

INSTALLATION

This chapter contains all the information you will need to install your MultiRam C Card. Package Contents, card care instructions, a physical description, and instructions for RAM additions and card installation are covered.

UNPACKING THE MULTIRAM C CARD

Check the contents of your MultiRam C package to be sure you have the following

- This owner’s manual
- MultiRam C Card
- Six plastic support posts (2 long and 4 short)
- A floppy disk, containing:

  Side 1 — AppleWorks memory expansion program
  Side 2 — Utilities which include DOS 3.3 and ProDOS RAM disk drive emulator programs, and a RAM test program
- A warranty card

Contact your dealer or Checkmate Technology if anything is missing.

You should fill out and return your warranty registration card as soon as possible. Only registered owners will be notified of updates to the software packaged with the MultiRam C Card.

CARD CARE

Always ground yourself before handling or touching the MultiRam C Card. Touch any metal object to discharge static electricity from your body. Static electricity, which can build up on your body, can damage RAM chips. The 256k RAM chips installed on your card are more sensitive to damage by stat electricity than 64~ chips.

Handle the MultiRam C Card carefully, by the edges. Do not flex your card while handling it as this can lead to weakening of connections.

Avoid touching the connector pins on the bottom of the card. These pins must kept clean and straight as they are inserted into two sockets on the motherboard to connect MultiRam C to the Apple. Do not remove the anti—static foam covering the ends of the pins until you are ready to install the card. you remove the card from the //c for any reason. reinsert the anti—static to over the ends of the pins to protect them from damage.

Never insert your card into or remove it tram the computer while the Apple’s power is on. You may damage the card, the Apple, and other devices attached to the Apple. You will void your MultiRam C warranty.

When installing added RAM, as discussed later, try to minimize the number of times RAM is inserted into and removed from the MultiRam C’s sockets. Even our quality sockets will be degraded by excessive insertion and removal of RAM chips.

PHYSICAL DESCRIPTION

After unpacking the MultiRam C card, carefully set it down on a clean, dry surface, with the sockets and other component facing up. Do not remove the anti–static foam protecting the connector pins at the bottom of the card. Compare your card with Figure 2.1 and locale the following components.

Ram Banks

There are two banks of sockets on the card into which RAM chips can be inserted:
Bank A and Bank B.

Bark A is located across the top left half of the card. Bank A consists of eight sockets, labeled A8 to Al from left to right, and its sockets are marked by a line around them.

Bank B is located across the bottom left half of the card. Bank B consists of eight sockets, labeled B8 to B1 from left to right, and its sockets are marked by a line around them.

Adding to or changing the RAM chips of these two banks is discussed later.

CPU & MMU Sockets

Near the middle of the card are two empty sockets.

The top socket is the MMU socket and is labeled “MMU”. To install MULTIRAM C, you must remove the MMU from the //c motherboard and insert the MMU chip into MultiRam C’s MMU socket. The notch on the MMU must point to the left toward the card’s RAM.

The bottom socket is the CPU socket and is labeled “CPU”. To install ‘MultiRam C, you must remove the 65C02 CPU from the //c motherboard and insert the 65C02 into toe MultiRam C’s CPU socket. The notch on the 65C02 must point to the left toward the card’s RAM.

Keyboard Support Posts

On the top of the card are six bores for keyboard support posts and on the bottom of the card are installed support posts.

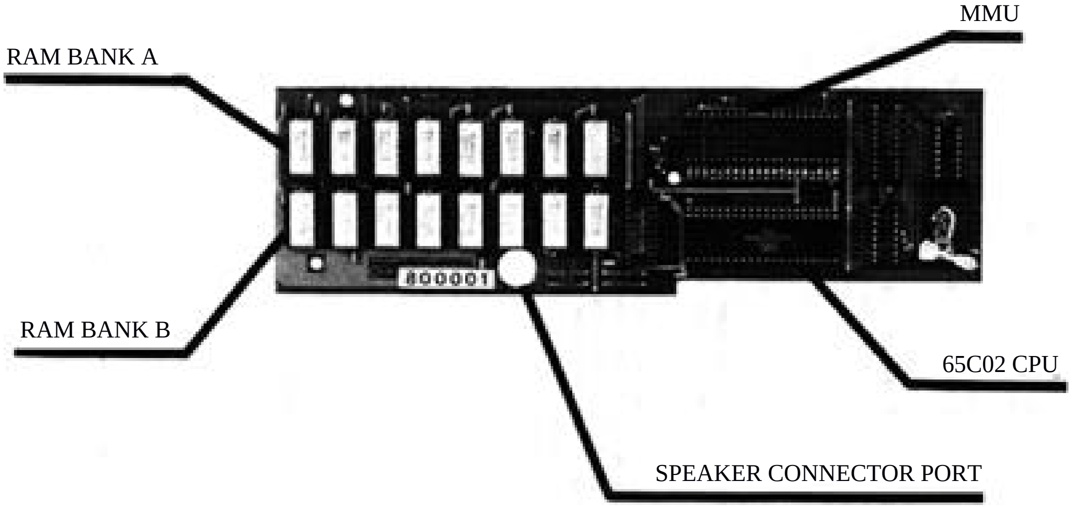

MMU
RAM BANK A
RAM BANK B
800001
65C02 CPU
SPEAKER CONNECTOR PORT

FIGURE 2.1

The posts on the bottom of the card are installed at the factory. These posts help level and support the card.

The holes at the top of the card are to add keyboard support posts. These posts need to be installed by you before inserting the card into the computer. These posts directly support the keyboard.

Speaker Connector Port

On the bottom left aide of the card is a round hole. You will insert the //c’s moles speaker connector through this hole and onto the two pins underneath it after installing the MultiRam C Card into the Apple.

Connector Pins

Turn the card upside down so that you can examine the bottom of the card. Remove the anti–static foam from the sets of pins on the bottom of the card to inspect the pins.

There are four rows of pins, each row containing twenty pins. These pins must be straight and parallel to one another to be correctly inserted into the CPU and MMU sockets on the //c motherboard. They must not be damaged or you will not be able to insert the card into the //c. Always cover these pins with the anti–static foam whenever the card is out of the computer to avoid damaging the pins by accident. Reinsert the foam over the pins.

INSTALLING ADDITIONAL MEMORY

The MultiRam C Card has two banks of eight sockets labeled Bank A and Bank B. Either a set of eight 256K or 64K dynamic RAM chips may be installed in each bank.

With the two banks, there are three memory combinations possible as Table 2.1 shows. A 64K and 128K combination are not shown as each card is shipped with a minimum of 256K in Bank A.

Table 2.1 Ram Banks And Size Combinations

	1	2	3
BANK A	256K	256K	256K
BANK B		64K	256K
Total	256K	320K	512K

If your card has only 256K of RAM installed, you say wish to add RAM to it at a future date to increase Its memory. RAM may be installed by you, the dealer, or Checkmate Technology.

Original Ram

Each MultiRam C Card is shipped with a minimum of 256K in Bank A, which means Bank A should have a full set of eight 256K RAM chips. Your MultiRam C Card may contain RAM in other banks as your dealer may have ordered the card with more memory or say have installed his own RAM chips.

Look on the lower right hand area of the rear of your card to see what RAM was shipped with the card. You should see a a "1" (shipped with 256K) or a "2″ (shipped with 512K) stamped in this area.

If the card, as purchased, is not configured as indicated by the code. additional RAM has been added or replaced on the card.

It is normal for dealers to install their own RAM in order to limit the different RAM sizes they stock. Dealer installed RAM does not void the card’s warranty. We cannot, however, guarantee non—Checkmate Technology RAM chips which are added by you or a dealer.

Checkmate Technology guarantees for five years only those RAM chips that are installed and tested at the factory or RAM chips mold in kits by Checkmate Technology, Inc., kits available at your local dealer. Refer to Chapter 9, which explains the terms of Checkmate Technology’s five—year warranty on the card and its components.

User Installed Ram

To install your own RAM, you will need to purchase one or more sets of eight 256K or 64K dynamic RAM chips. RAM on MultiRam C most always be installed in sets of eight in order to generate 256K or 64K of memory.

Chips installed in a bank of sockets must always be all the same type —— 256K or 64K. Although mixing chips will not harm the RAM or the card, you will loan the use of 192K of a bank of 256K chips if one 64K chip is accidentally installed into the hank. Similarly, only 64K of a 256K chip installed into a bank of 64K chips is usable.

Ask for a speed rating of 150 nanoseconds on the dynamic RAM you purchase, although 200 nanosecond 256K or 64K DRAM chips will work in the computer. You should try to purchase a set of eight RAMS that are the same speed and made by the same manufacturer to minimize potential compatibility problems.

If you are unfamiliar with purchasing RAM, ask your dealer for Checkmate Technology’s RAM Kit. The RAM in each (it is guaranteed to work with MultiRam C when correctly installed, and are warranted for five years by Checkmate Technology. If you return a card to Checkmate Technology with defective RAM you inserted into the card (other than from a Checkmate Technology RAM kit), you will be charged a service fee plus RAM replacement cost to repair the card. Refer to Chapter 10 for more information.

Installation Procedure

To install a set of RAM chips lay MultiRam C down on a table with the socket side up. The eight sockets of Bank A will be at the top left end (see Figure 2.1); there should already be a set of 256K RAM installed in that bank. Install the next set of RAM into Bank B on the bottom left of the card.

Most RAM chips have a notch or dot at one end to designate pin one. The chip's notch should match the notch on the chip drawing under the socket. The notch should face up toward the top of the card on both Bank A and Bank B.

> WARNING! IF A RAM CHIP IS INSTALLED BACKWARDS, IT MAY BE DESTROYED WHEN POWER IS TURNED ON. THE MULTIRAM C CARD WILL NOT BE DAMAGED.

Carefully line up both rows of pins with the holes in a socket and press the chip into the socket. seating it firmly. Check that none of the pins are bent out or bent under the chip. Repeat the process with each chip until the bank is filled.

When you have finished installing any additional memory, you are ready to install the MultiRam C Card with the power TURNED OFF in your computer.

If you have any doubts about removing or installing the chips, have a service—person do it for you.

RAM Removal

If you incorrectly insert RAM into the card and need to remove it or if you are removing RAM from another device to insert into MultiRam, you should use an “IC—puller” (an inexpensive tool designed for removing integrated circuits. and widely available at electronics supply stores). Lacking an IC—puller, an ordinary flatware spoon can be inserted between the chip and the socket and the bowl portion can be used to rock each end of the chip lose. If a spoon will not fit, you say insert a flat blade between the RAM and the socket and pry up the RAM chip. Be very careful if you use a knife or other sharp flat blade.

Carefully remove each chip from its socket by gently prying the ends of the chip upward, a little at one end and then the other, being careful not to damage or bend the pins. Be sure you are prying only the chip and not the entire socket; very little pressure is necessary.

Dealer Installed RAM

You may have your dealer install RAM at the time you purchase your card or at a later date. Be certain that the installed RAM passes the test program supplied with the MultiRam C Card (described in Chapter 3).

Refer to Chapter 10 if your card containing dealer installed RAM is in need of service.

Factory Installed RAM

You may return your card to Checkmate Technology to have additional RAM installed. Ill factory RAM is thoroughly tested on the card at the factory and is warranted for five years. Refer to Chapter 10 for core information.

INSTALLING THE MULTIRAM C CARD

The following step—by—step instructions will allow you to easily install and test the MultiRam C Card in your Apple //c.

You should carefully read over all of the instructions before proceeding with installation. Although no technical skills are required to install the card, you must open the case of your Apple to do the installation. Checkmate Technology can not assume responsibility for damages to the Nc case or //c circuitry that are caused by improper installation.

If your //c is under warranty, opening the //c case will void the Apple warranty. If your Apple is under its original 90 day warranty or If you have purchased an extended warranty from Apple Computer, Inc., you should have an authorized Apple dealer install your MultiRam C Card. Most Apple dealers will continue to honor your Apple's warranty period if they perform the installation. Check with your favorite Apple dealer for more information.

If you have any doubts regarding the installation, please have a qualified technician install the card.

Materials Required

To install the MultiRam C Card into your Apple //c, you should gather together the following materials before starting:

Medium size Phillips screwdriver
Small flat blade screwdriver or kitchen butter knife
Small metal kitchen spoon
Soft cloth or mat
Blank disk
Disk copy program
MultiRam disk packed with the card

Preparation

Before opening the case of your Apple to install the card, you need to prepare a work area, a backup Utility disk and the Apple itself.

The area where you will install your Apple should be large enough to lie the Apple down flat and there should also be room for your monitor. A power outlet for both the external Apple power supply and the monitor should be available for later testing purposes.

Clear a space on the work area large enough to place both the Apple and your monitor. Spread a large, soft cloth over the area. You will disassemble the Apple on this cloth.

Make a backup copy of the reverse side of the MultiRam disk, the Utility side of the disk containing a RAM test program. Use COPYA or the ProDOS Filer contained on the Utility side of the MultiRam disk to make the copy. Refer to your Apple manual for information on the copy process if you are unfamiliar with disk copying.

When you have finished making the backup copy, turn off your Apple and remove all connectors from the back panel of the *//c* where it is now sitting. This means you should remove the external power cable and any mouse, joystick, modem, video cable. RF modulator, external disk drive, or printer if these devices are connected.

Insert the backup copy of the MultiRam disk into the //c's internal disk drive. Gently place the //c upside down (the keyboard facing down) on top of the soft cloth at your work area.

Disassembly

Follow steps 1 to 9 to prepare the //c for insertion of the MultiRam C Card.

Before beginning disassembly of the //c case, touch a metal object to discharge any static electricity from your body. This will prevent potential damage to the RAM on the MultiRam C Card.

I. Remove the six screws identified in Figure 2.2 with a small/medium size Phillips screwdriver. These are the screws at the outside edges of the case. Do not remove the four inside screws; these hold the disk drive in place.

   The two screws at the rear of the case have a larger thread size than the four screws at the front of the case. Separate the two larger back screws from the others so that you will not try to insert them into the wrong holes when reassembling the case.

   The two screws at the aides of the case may not come fully out of their holes for removal by hand. If so, turn the //c right side up (keyboard facing up) and gently shake roe *//c* until these screws fall to the cloth below the //c.

   Safely store the screws aside.

2. Place the *//c* on the cloth, keyboard side up. with the back of the *//c* facing you.
3. Remove the back panel. This is the flat plastic panel which identifies the ports at the rear of the computer. The panel must be removed before the top of the //c can come off.

   The back panel is held in place by nine plastic tabs that fit into notches in the //c case. To remove the panel, you must remove seven of the tabs from their notches. The location of the seven tabs are shown in Figure 2.3. If during removal some tabs break off, do nut worry. Only a few of the seven tabs are necessary to firmly secure the back panel to the //c’s case. If you do not apply unnecessary force, so tabs should break.

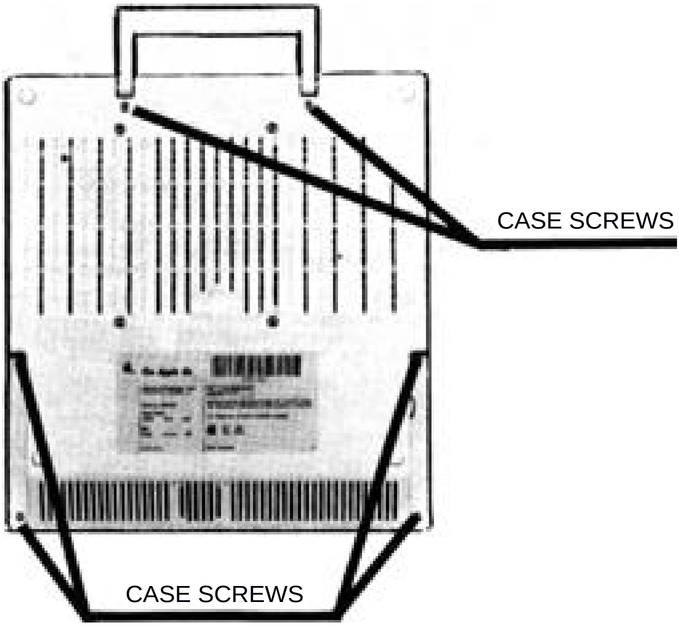


FIGURE 2.2

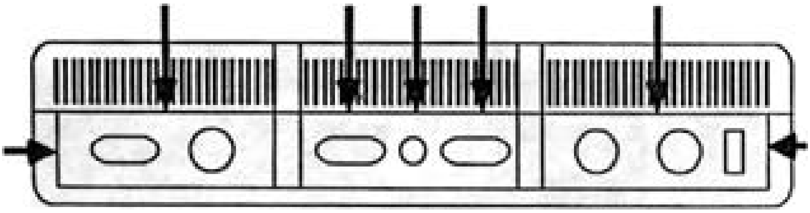


FIGURE 2.3

Insert the tip of a small screwdriver (or thin flat blade like a butter knife) between the outside case and the back panel at point A identified in Figure 2.3. Pry out the back panel until the tab pops free of the notch in the Hr case. Continue the same procedure working up and across until the tab on the side and the one tab on the top right side are free.

You may be able to pull the right side of the panel away from the case at this point as Figure 2.4 shows. If you can, you say be able to free the middle tabs by pulling up on the top of the case with one hand while prying the panel Out with the other hand.

If the top of the //c does our yield, you may need to continuing prying the back panel away from the //c 4th the screwdriver at the remaining tab points. Working from left to right, continue prying until all the top tabs are free.

With the top tabs pulled out of the //c case, pull the back panel away from the case as Figure 2.5 shows. The remaining tab at the left aide of the panel should pop loose. Set the back panel aside.

4. Remove the top cover of the Nc as Figure 2.6 shows. Grasp the top cover at both rear corners and push upward. Push upward until the case has moved up a few inches and then pull back and up on the cover. When you pull the case back, the top cover should be freed from one plastic hook in the middle of the front of the case.

   Put the top cover safely aside.

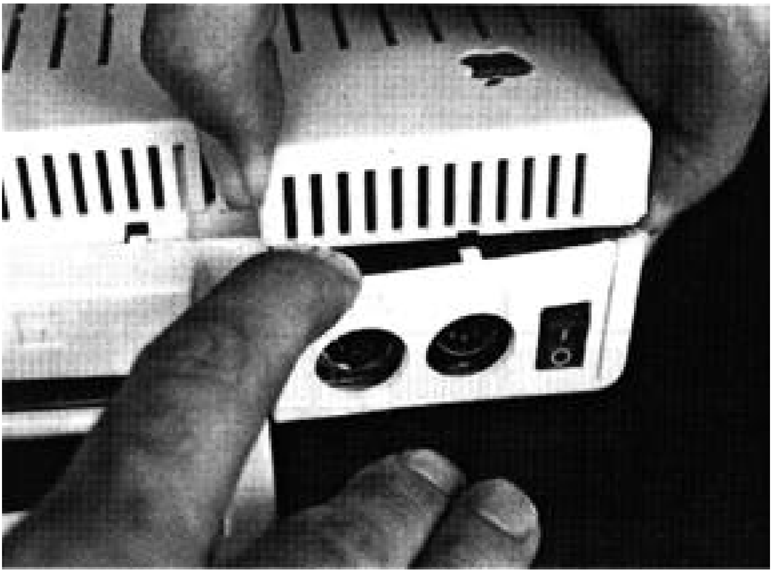


FIGURE 2.4

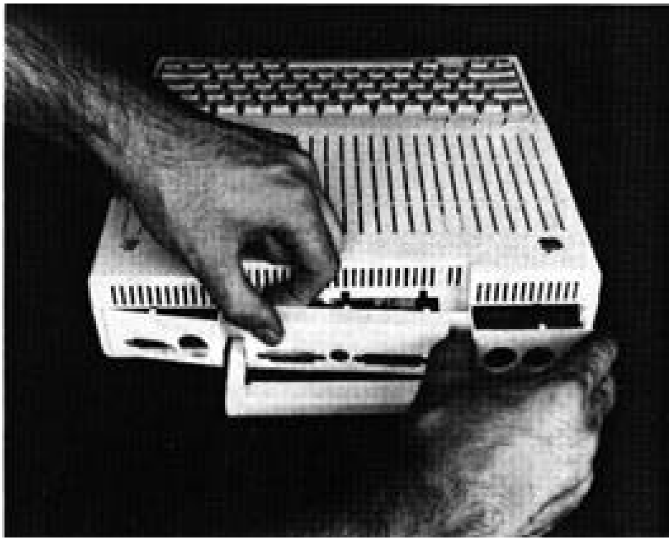


FIGURE 2.6

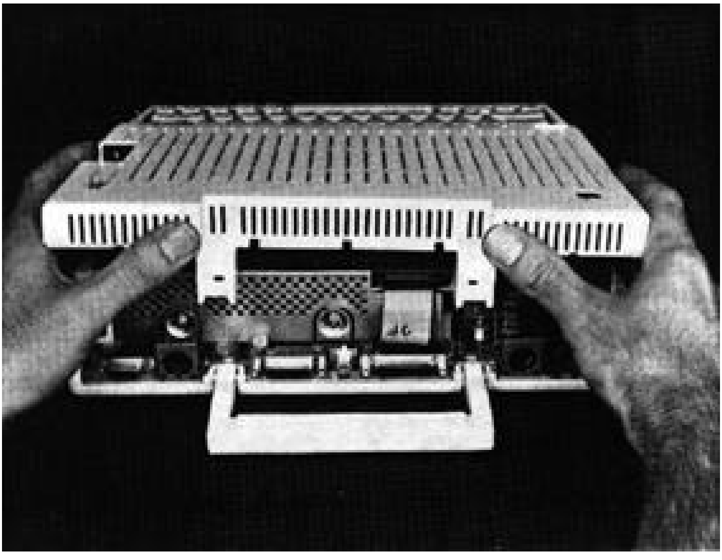


FIGURE 2.5

5. Turn the //c around so that the keyboard end now faces you. The //c should now resemble Figure 2.7.

6. Lift up the front end of the keyboard and pull the keyboard slightly towards you. This will remove the keyboard from the slots in the disk drive case (identified in Figure 2.7) that help to hold the keyboard in place.

   When the keyboard is free of the slots, turn the keyboard upside down and lay it over the disk drive as shown in Figure 2.8. Push the keyboard as far back as the ribbon cable that connects the keyboard to the motherboard allows. This should give you enough clearance to easily install the MultiRam C Card.

7. Remove the keyboard brace. The brace (identified in Figure 2.8) is used to support the keyboard. Because MultiRam C contains its own support posts, this brace will not be needed and should be removed from the computer.

   Two white retaining pins attach the brace to the keyboard, one at each end of the keyboard. To remove each of the two pins, pull upward on the top of the pin (as Figure 2.9 shows) while simultaneously pushing the bottom of the pin away from you with a small screwdriver. The pin should pull free of the back of the keyboard.

   Store the brace sway; you will not need to use the brace as long as the MultiRam C Card is in the *//c*.

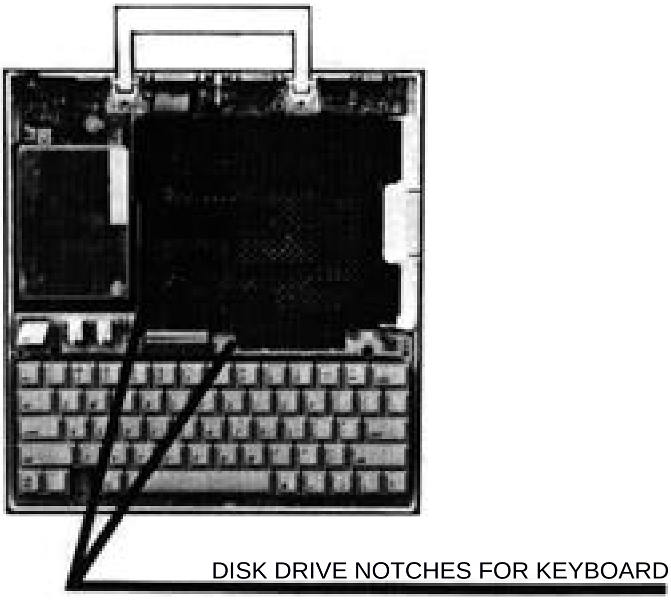


FIGURE 2.7

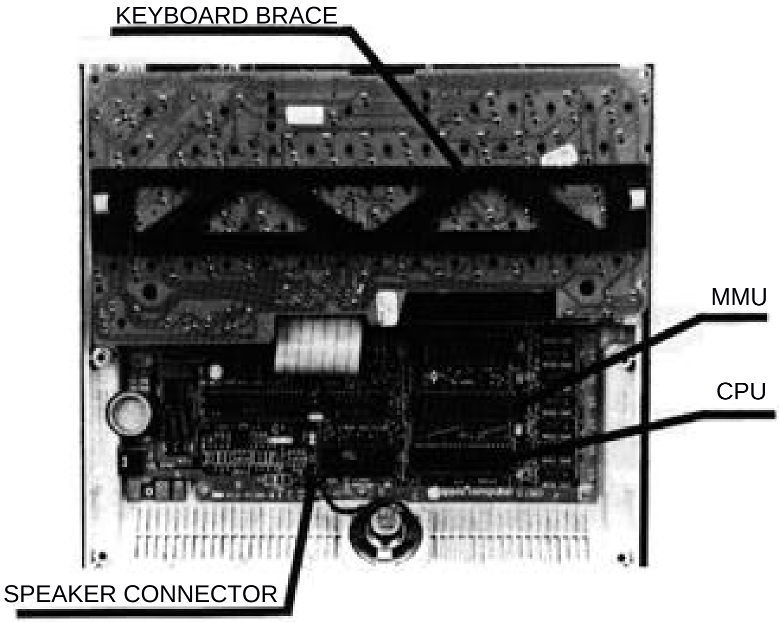
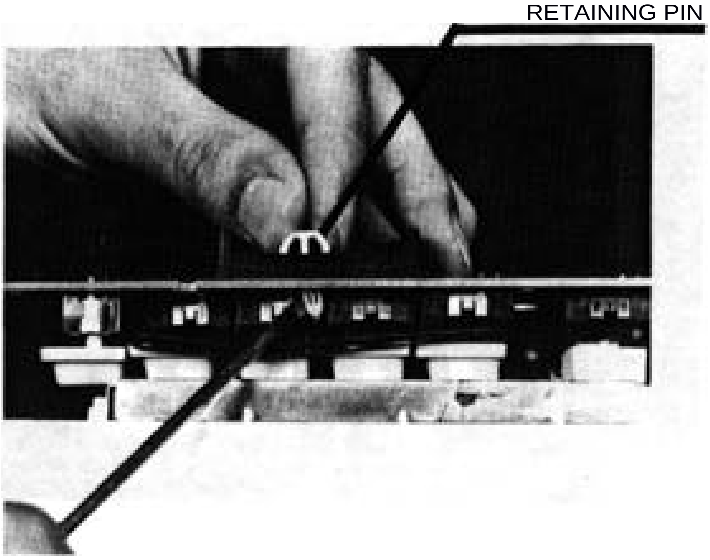


FIGURE 2.8



8. Pull off the speaker ConnecTOR from the speaker pins underneath the connector. The speaker connector is identified in Figure 2.8.

9. Remove the CPU and MMU from the motherboard. These components are identified in Figure 2.8.

   Using a small metal spoon, as Figure 2.10 shows, insert the bowl and of the spoon between the CPU and MMU and their respective sockets. Rock the spoon back and forth to loosen the component from the socket at one end. Repeat this on the other end until the components are loose enough to remove by hand.

   Alternately, you may use a knife or small screwdriver to pry up the chips if a spoon with a small enough lip is not available.

   Be careful when removing the CPU and MMU that you do not accidently hand their pins. If you do accidently bend a pin, carefully use a small needle nose plier to straighten out the pin.

   Plate the removed CPU and MMU on the soft cloth.

   Disassembly is now complete.

Preparing The Card

1. Insert the removed CPU and MMU into the MultiRam C Card. As Figure 2.8 shows, the CPU should be inserted into the lower socket and the MMU the upper socket, just as they are located on the motherboard.

   To insert the components, line up the pins of the CPU and MMU with their respective sockets holes on the card. The notch on both components should face to the left towards the RAN on the card. Insert one aide of the components pins slightly into the socket. Then, using an inward pushing/rocking motion, seat the pins on the other aide of the component.

   If the pins do not easily seat, you may need to very slightly bend all pins on one side of the component inward. Place the component on the table on its side. Lift the component slightly up so that pressure can be uniformily applied to all pins along one side. Apply firm, even, gradually increasing pressure until some small inward movement in the pins is detected. Do not apply too much pressure or the pins will bend too far. Repeat the procedure on the opposite side. Retry inserting the component. Repeat this procedure again if the pins are still too far outside the socket to he moved into the socket holes.

2. Insert the six keyboard support posts into the six holes on top of the card. Insert the two longest posts into the holes at the top left corner and center of the card as Figures 2.11 and 2.12 show. Insert the four short posts into the four remaining holes on the card. Do not change the position of the four short pins which are on the bottom aide of the board.

   Insert the end of each pin with the longer collar into the holes. You can insert the end with the shorter collar into the holes but the pin will be harder to remove later it the pin must be removed for any reason.

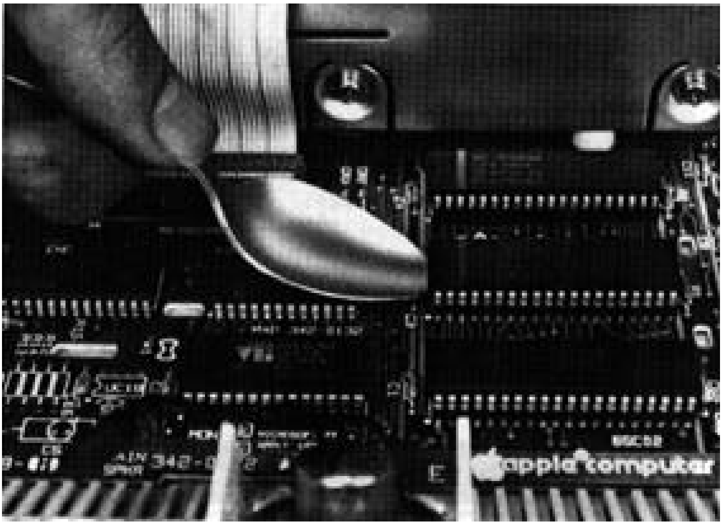


FIGURE 2.10

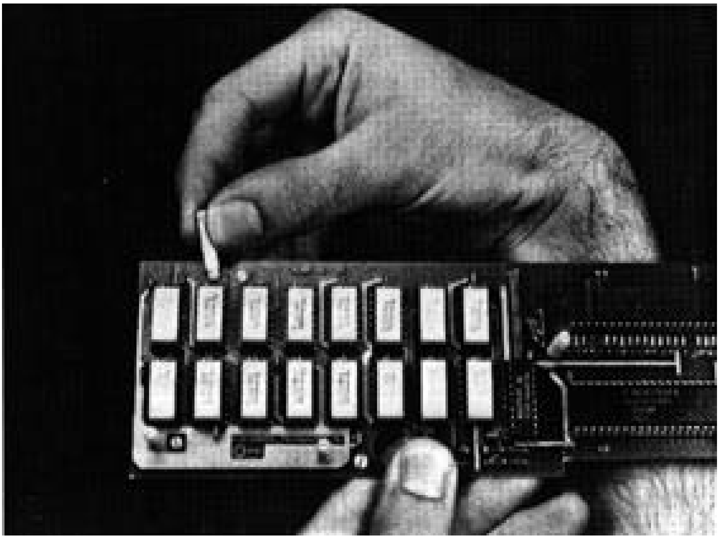


FIGURE 2.11

3. Remove the protective foam from the connecting pins on the bottom of the card.

   The card should now look like Figure 2.12 and is reedy for insertion into the //c motherboard.

Inserting The Card

1. Pick up the card with both hands and insert the card into the CPU and MMU motherboard sockets.

   As Figure 2.13 shows, first position the card directly over the sockets so that you can see the pins that wilt enter the CPU socket. Carefully align the pins with the CPU socket’s holes. Push the card into the sockets using a downward, gentle rocking motion. First push down one side, then the other. With each rock the card should seat deeper into the sockets. When you sense no further downward motion when you rock the card, the card has been successfully seated.

   If the card seems to be higher on the right side than the left side, do not worry; this is normal. If you see a support post that does not touch the bottom of the Apple, this too is normal; the post will touch when the keyboard is put back and the case resealed.

   You should have no problem inserting the card. However, if you feel any resistance or possible bending of the pins, stop immediately. Remove the card and examine the pins and sockets for possible damage.

2. Reinsert the speaker jack. Place it through the round bole in the MultiRam C Card and push it over the two exposed pins. The //c should now look like Figure 2.14.

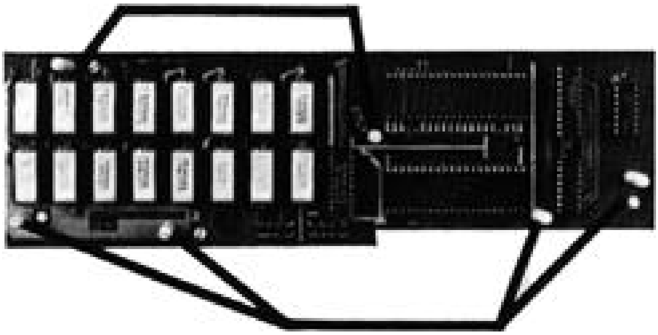


FIGURE 2.12

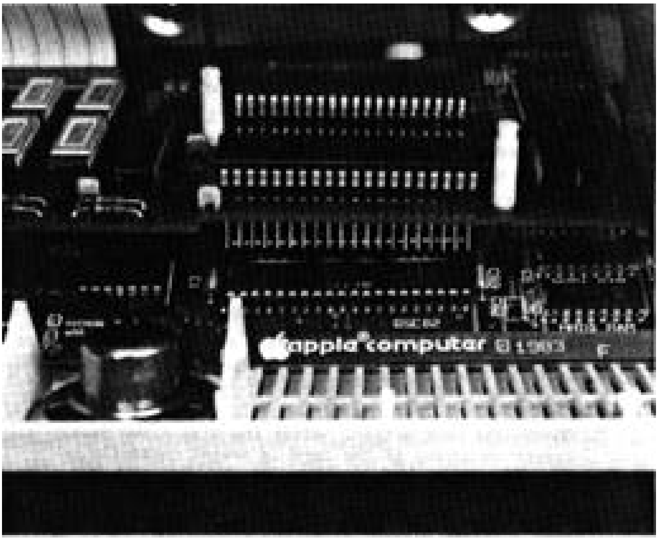


FIGURE 2.13

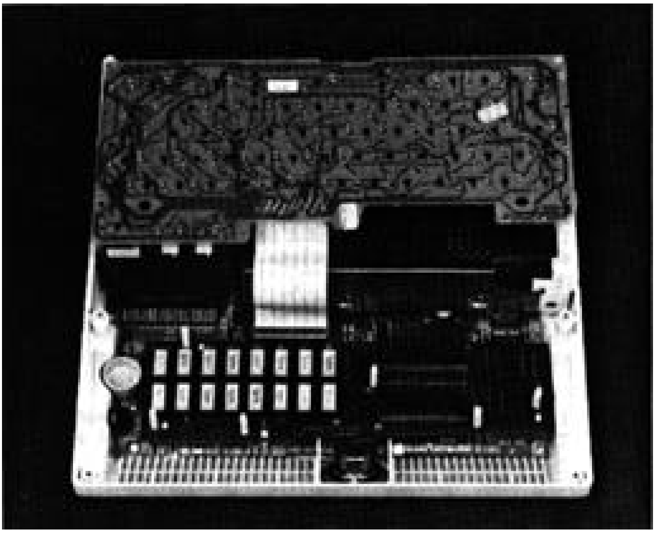


FIGURE 2.14

3. Return the keyboard to its normal position on top of the keyboard support posts.

   The card is now ready to be tested.

Testing The Card

1. Attach the video cable from your monitor to the //c. Attach the external power supply cable to the //c. Turn on the monitor. If you did not insert the backup copy of the Utility side of the MultiRam disk into the internal disk drive, do so now.

2. Turn on the //c. BE CAREFUl! DO NOT TOUCH ANYTHING INTERNALLY IN THE //C OR YOU MAY RECEIVE AN ELECTRICAL SHOCK.

3. Check the boot action. After the disk drive stops, you should see a menu appear on screen if the card is seated correctly.

   If instead you see a blank screen or lines or blocks of inverse characters on screen, then you have probably not correctly inserted the card. If you see a blank screen, first check your monitor to be certain it is on and then check to see that the video cable is properly connected.

   If the monitor is correctly set—up and the screen remains blank or if lines or blocks are on screen, turn off the computer and disconnect the power cable. Place the keyboard back on top of the disk drive and check the card.

   Try rocking the card further into the motherboard sockets. Also check the CPU and MMU components and the RAM chips. If the pins of these components sees to be inserted correctly no pins out of the socket or bent under the socket — press down on them to be certain good contact is being made. Check to be certain that the notches of the RAM chips face the top of the card and the notches of the CPU and MMU face to the left toward the RAM chips.

   Test the card again. Try the preceeding steps again if you do not succeed. If you cannot get the card to work after repeated attempts, contact your dealer or Checkmate Technology for assistance.

4. Assuming a normal boot did occur, select the RAM teat diagnostic from the Utility menu. Refer to Chapter 3 for detailed information on how to use the RAM test and what to do if RAM does not test good.

   If RAM on the card tests good, you are now ready to reassemble the //c. Turn off the computer and remove the external power supply cable and the video cable from the back of the //c.

Reassembly

1. With the front of the //c facing you, insert the back of the keyboard into the restraining slits on the disk drive case. The right end of the back of the keyboard fits into the slit on the disk drive near the power supply and the protrusion to the right of the keyboard connector cable fits into the slit in the middle of the disk drive.

2. Replace the top cover. Insert the hook at the front end of the top cover into the cutout for it on the keyboard. Do so by tilting the front end of the top cover downward. Next, snap the case down at the rear corners. You should hear a snap as the tabs on the cover lock.

3. Turn the rear of the //c toward you. Replace the //c handle if you removed it from the beck of the case. Reinsert the back cover. Snap the tabs of the beck cover into place on the //c case.

4. Reinsert the six screws making sure the two wider thread screws are returned to the two holes at the back of the case. Tighten the screws securely. Don’t overtighten the screws or you may strip the hole's internal threads.

You have successfully finished installing your MultiRam C Card.

You will not have to remove the //c’s case again unless you wish to add more memory to the card. Simply install more memory on the card while it is inserted into the motherboard; there is no need to remove the card from the motherboard to insert an additional bank of RAM.

Chapter 3

TESTING & TROUBLESHOOTING

This chapter covers testing procedures you may use to check the RAM on your MultiRam C Card.

This chapter should be read by everyone who has just bought and installed the MultiRam, C Card. You should always refer back to this chapter when you add or replace RAM on the card or if a problem develops while using the card.

RAM TEST PROCEDURE

On the back side of the disk supplied with your card, the side marked “Utilities”, is a test program that will identify good RAM, bad or missing RAM, and will alert you if a 64K chip has been placed into a bank of 256K RAM or vice–Versa.

Before continuing, please make a backup copy of the Utilities side of the disk if you have not already done so. Use The Filer on the Utility side of the MultiRam disk or COPYA from a DOS 3.3 System Master disk to copy Utilities to a blank disk. See your //c manual for detailed disk copying instructions.

Insert a backup copy of the Utilities disk into disk drive 1 sod boot the disk. You will see a menu. Choose the MultiRam C test from the menu.

If the backup Utility disk does not boot correctly, does not boot at all, no video is seen, or if the //c does not seem to be working correctly with any software you try to boot, turn to the next section “NON–RAM PROBLEMS.” Continue the RAM test if you see the Utility menu.

Test Display

A 40–column screen display known as the “status page” should appear.

You should see eight 64K auxiliary bank indicators on the left side of the screen numbered from 1 to 8. Since the Apple’s 65C02 processor cannot address more than 64K of memory at a time, the Apple swaps memory banks into and out of its addressing range in order to use MultiRam C memory. Therefore, the auxiliary bank indicators show MultiRam C auxiliary memory as the Apple sees it --one bank at a time, not 128K or 512K or any amount of memory larger than 64K.

On the right aide of the screen the usable memory in RAM Banks A and B is identified and the total usable memory available on the card is shown. Upon boot–up, question marks appear instead of numbers on both aides of the screen as the card has not yet been tested (see Figure 3.1).

The bottom two lines of the screen show the keys that say be used to run testing

- Pressing the Arrow keys position a flashing. solid cursor in a hot to the left of the sit applications shown.
- Pressing the Return key tons the application marked by the flashing cursor.
- Pressing the Esc key shorts “full” or “continuous” tests.
- Pressing the Space Bar once pauses “full” or “continuous” tests; pressing it again resumes the tests.

Three tests are shown on the menu: a "quick” test, a “full” test and a “continuous” test.

Test Selection

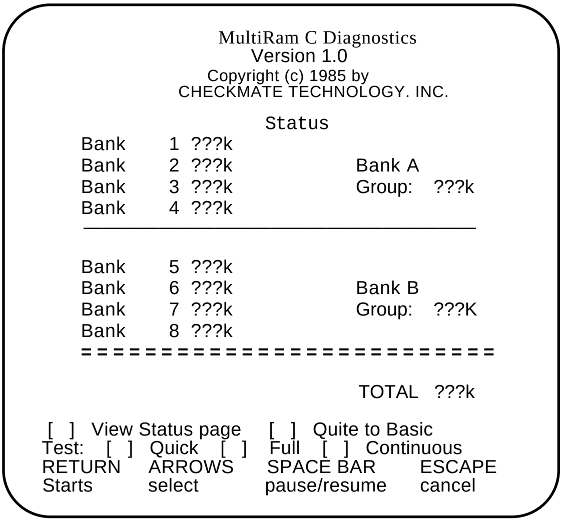
The “quick” teat checks a very small percent of the memory in each RAM chip ina short time. The test will run in three seconds. Use this test to quickly find major RAM failures. If RAM in bad In an area other than the few places the test checks, a bad chip may be undetected by this test.

The “full” test exhaustively tests all 64K of each auxiliary bank available on the card. The test runs approximately three seconds per 64k. Yogi should always run the full test after adding or changing RAM on the card or when first installing the card. The “full” test runs the “quick” test and then exhaustively checks all banks found good by the “quick” test. Bad or incomplete banks are not tested as they could not be used by any program.

The “continuous test runs the full test over and over until you press the Escape key. Use the continuous test if a problem develops with the card after it has been in use for awhile as many problems with RAM show up as intermittent problems. Ogre run of the “full” test might not disclose such a problem whereas its continued running would.

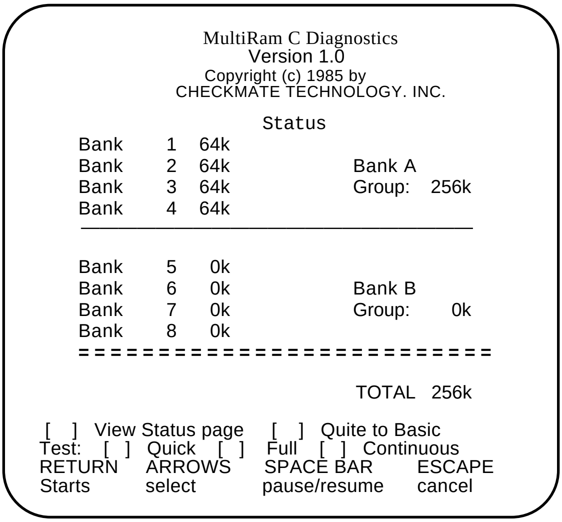
Running A Test

Running the “quick”, “full”, or “continuous” tests shows a high resolution picture of the MultiRam C Card. Each RAM socket is shown exactly as mid oat on the card.



```
              MultiRam C Diagnostics
                   Version 1.0
               Copyright (c) 1985 by
          CHECKMATE TECHNOLOGY. INC.

                      Status
    Bank     1  ???k
    Bank     2  ???k              Bank A
    Bank     3  ???k              Group:   ???k
    Bank     4  ???k
    ____________________________________________

    Bank     5  ???k
    Bank     6  ???k              Bank B
    Bank     7  ???k              Group:   ???K
    Bank     8  ???k
    ============================

                                  TOTAL   ???k

 [  ]  View Status page    [  ]  Quite to Basic
Test:   [  ]  Quick   [  ]   Full   [  ]  Continuous
RETURN     ARROWS      SPACE BAR      ESCAPE
Starts     select      pause/resume   cancel
```

FIGURE 3.1



```
              MultiRam C Diagnostics
                   Version 1.0
               Copyright (c) 1985 by
          CHECKMATE TECHNOLOGY. INC.

                      Status
    Bank     1   64k
    Bank     2   64k              Bank A
    Bank     3   64k              Group:   256k
    Bank     4   64k
    ____________________________________________

    Bank     5    0k
    Bank     6    0k              Bank B
    Bank     7    0k              Group:     0k
    Bank     8    0k
    ============================

                                  TOTAL   256k

 [  ]  View Status page    [  ]  Quite to Basic
Test:   [  ]  Quick   [  ]   Full   [  ]  Continuous
RETURN     ARROWS      SPACE BAR      ESCAPE
Starts     select      pause/resume   cancel
```

FIGURE 3.2

Each socket is divided into four squares. Each socket is shown as four squares because MultiRam C can contain eight 64K auxillary banks in its two RAM banks (Figure 3.3). The Apple. through MultiRam C circuitry, addresses these eight 64K banks as follows:

BANK A RAM	BANK B RAM
1	5
2	6
3	7
4	8

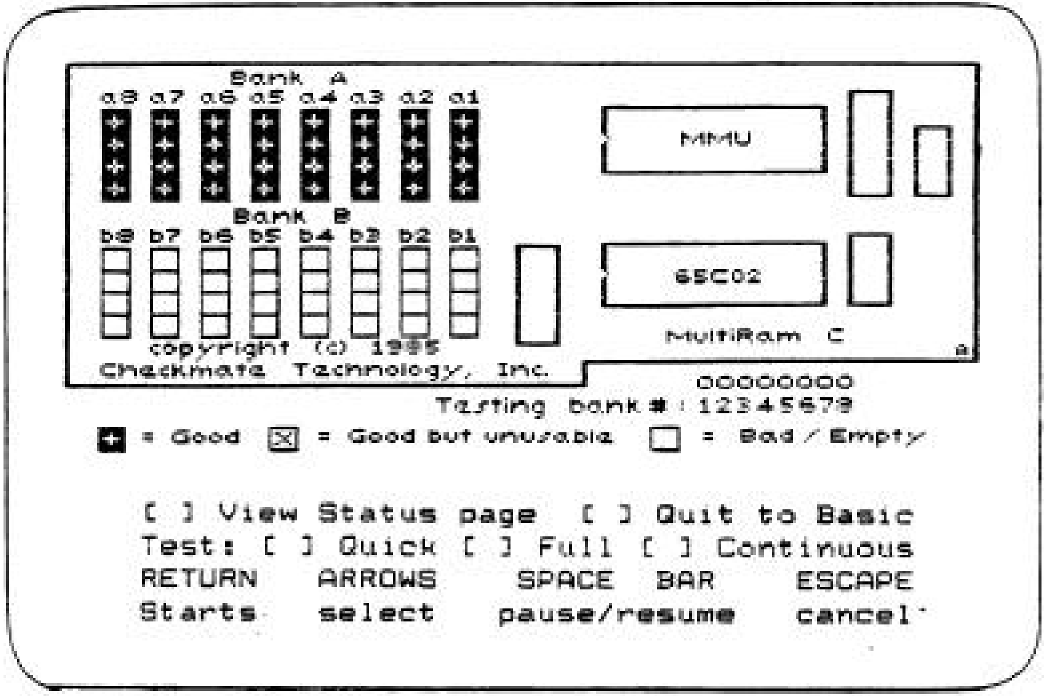
Running any of these tests, you will see a spinning cursor appear in place of the flashing cursor in the bottom four lines of the screen. The cursor will spin until the test Is completed.

foe will also see eight tights blink off ned on on the bottom right side of the picture. These lights show which of the eight banks of auxiliary 64K Is being currently tested. auxiliary banks are tested from 8 to 1 in descending order.

At the end of a test. you will see one of three markings appear in each quarter of the sockets

- A “+“ on a black background indicates the RAM in the 64K auxiliary memory segment, whether found on a 256K RAM chip or a 64K RAM chip, tested good.
- An “X” on a white background indicates the RAM in the 64K auxiliary memory segment Is probably good but can’t be completely tested as another chip in the same group of sockets tested bad, was missing, or was not the sane size as the other RAM in the group.
- An empty square indicates bad RAM or missing RAM.

After a test is run, you say select the option “SHOW STATUS’ and see a summary of usable memory (Figure 3.2). The eight banks show 64K for good RAM, 0k (zero) if RAM is missing or bad is the bank, bank having 256k chips. where one or more 256K chips are “partially” defective, can show 64K available is that bank.



```
                    00000000
Testing bank # : 12345678
■ = Good   ☒ = Good but unusable   □ = Bad / Empty

[ ] View Status page   [ ] Quit to Basic
Test: [ ] Quick [ ] Full [ ] Continuous
RETURN    ARROWS    SPACE  BAR    ESCAPE
Starts    select    pause/resume   cancel
```

FIGURE 3.3

Test Results

Several reasons may lie behind an appearance of a blank square or a “X” in a square. If these results are shown as the result of your testing, write down the socket number of those chips that test “bad.” Turn off the Apple //c, open its case following the instructions in Chapter 2 remove its cover, lift back the keyboard and visually inspect the MultiRam C Card.

If an entire bank shows “bad”, there is probably no RAM in that bank. This is easily verified. if one chip in a bank shows bad, it is also easy to see if that chip is missing.

If there is a chip in the socket marked “bad”, first inspect the chip. Make certain it matches the memory size of the other chips in its bank, If it does, neat check to see that all of its pins are firmly inserted Into the socket. A chip say read “had” if one of its legs is bent under the chip or if a pin is sticking outside the socket.

If none of the preceding are true, then the PAM identified as “bad” is probably defective. You may try swapping the chip with other RAM shown as good and rerun the test to confirm the first test results.

If the defective RAM was originally shipped with the card (see Chapter 2 for the shipping code on the card) or is RAM from a Checkmate Technology RAM iii, then turn to Chapter 10 for warranty service information.

If the RAM is not from Checkmate Technology, check with the store that sold you the RAM regarding replacement. As discussed in Chapter 10, Checkmate Technology will replace other company’s bad RAM for a service tee.

If your card does not work properly although the RAM test indicates RAM on the card is good, please call the Customer Service Department as outlined in Chapter 10 for help.

NON—RAM PROBLEMS

Common sources of problems other than a RAM malfunction exist. If your card or your computer is not working correctly with the MultiRam C Card installed, open the //c case following the instructions in Chapter 2 and check for the following conditions.

Visually inspect both the 65C02 CPU and MMU you removed from the //c motherboard and placed onto the MultiRam C Card. Each pin of these two chips must be firmly socketed in order for the //c to work properly. If only one pin of either of these two components is outside the socket or bent under the socket, the entire Apple may not work. If any pins are out of their sockets or best under the sockets, remove the chip, straighten the pins not seated and reinstall the chip.

Check to be certain the notch of both the CPU and MMU face left toward the RAM on the card. If not, remove the component and reinsert it with the notch punting to the left.

If all components are correctly seated, next check to be certain the card is firmly inserted into the CPU and MMU sockets. If the card is not making full contact in these sockets, the card will not work. Apply gentle rocking downward pressure on the card near the area of the CPU and MMU on the card. You should feel further downward movement of the card if it was not fully seated.

If the proceeding steps do not solve the problem, remove the card from the //c. Inspect the pins that fit into the CPU and MMU sockets. Check to see if the pins have bent during the insertion process. If any of these pins are not seated into the motherboard the card will not work. Straighten the bent pins and reinsert the board.

If none of these troubleshooting steps works, please call the Customer Service Department for further help. Refer to Chapter 10 for more information.

Part 2

USING THE MULTIRAM C *CARD*

Using MultiRam C shows you how to use MultiRam C for massive Desktop space with AppleWorks, how to use the card as a RAM fast disk drive, and describes hardware and software that you can use to enhance the usefulness of MultiRam C.

Chapter 4

APPLEWORKS & MULTIRAM

The MultiRam C Card releases the power locked up within AppleWorks!

AppleWorks has become the best selling program for the Apple //c and //e. AppleWorks real potential, however, has not been realized as AppleWorks usefulness is constrained by the small 55K Desktop apace an ordinary 128K Apple //c provides the program. With MultiRam C enhancements made to AppleWorks, AppleWorks can now live up to its real potential.

Running AppleWorks with the MultiRam C Card you may:

* Expand Desktop space up to 413K!
* Handle Individual files more than 7 times the previous desktop size!
* Enter up to 5,100 records into a data basal
* Save large files over multiple disks automatically !
* Speed execution of AppleWorks many times over!
* Minimize the need for an external standard floppy drive!

This chapter will help you get the most from your new AppleWorks and MultiRam combination with:

1. Detailed instructions for modifying AppleWorks to make use of all the memory on your MultiRam C Card.
2. Explanation of basic restrictions within AppleWorks and MultiRam modifications to remove sot restrictions.
3. Suggestions for effectively working with large files.
4. Instructions for automatically saving large files.

APPLEWORKS CONFIGURATION

You will need to configure backup copies of the AppleWorks Startup disk and Program disk to allow AppleWorks to use all of MultiRam C’s memory.

The MultiRam C configuration is a one time operation; once it is done, the configured AppleWorks disks are ready to use. You will not have to reconfigure the disks even if you later add more memory to the card. Simply boot AppleWorks and use all the new added MultiRam features.

You will need three (preferably five) blank disks, your AppleWorks Startup and Program disks, and the MultiRam disk to prepare your modified AppleWorks disks. Once you have gathered these disks together, follow these steps.

1. Make backup copies of the Startup disk, Program disk, and AppleWorks memory expansion disk (the front of the MultiRam disk). For your convenience, the Filer, Apple's ProDOS disk copy program, is included on the Utility side of the MultiRam disk (beck side) to make your backup copies. You may use COPYA on a DOS 3.3 System. Master disk or any other disk copy program for this purpose, however.

   To save rebooting a copy program again in order to see the affect of different MultiRam options explained later, we recommend you sake two copies of both the Startup disk and Program disk now if you have two additional blank disks available.

   If you use a custom printer setup, have made a printer patch to your copy of AppleWorks, or have modified the AppleWorks Startup disk in any way, make your Startup backup copies from your already modified Startup disk, not the original.

2. Boot the backup copy of the AppleWorks memory expansion disk. Press the Return key to begin the program and again to advance to the configuration option.

3. Press "Y" or "X" when you are asked "Do you wish to have all the program load during the hoot?"

   If you press "N", AppleWorks will load and operate normally — only key parts of the program are loaded into memory when you boot the disk, not the whole program.

   When using AppleWorks, you may have noticed that the disk drive reading the Program disk turns on frequently when moving between a file and the main menu, sending material to the printer, etc. This occurs because only part of the AppleWorks program is in memory at one time. AppleWorks uses an overlay manager that loads program segments from disk into memory to do specific tasks only when those tasks are first called. Because program routines are swapped into and out of memory as they are needed. Desktop memory is conserved for data files. The first option maintains this convention.

   If you press "Y", the entire AppleWorks program loads into memory during the bootup process.

   With all the program in memory, each time you manipulate a file, AppleWorks acts immediately, and you will find yourself using the Escape key. Control I) key, etc., to literally skip through files with quick, single keystrokes. Achieving this immediate response makes AppleWorks one of the fastest integrated packages available.

   When using the AppleWorks Startup disk configured with option two, an extra one to two minutes is required during the boot to load the entire program into memory depending upon the AppleWorks version used. After inserting the Program disk, you will be prompted for the date as usual. The program is then loaded entirely into memory.

   A minimum of 256K of RAM, preferably more. should be installed on your MultiRam C to best use option two. Larger memory area are best used with this option as approximately 130K of the Desktop space is filled with the AppleWorks program. The remaining apace indicator sees in the lower right corner does not show this loss because the overlay manager does not reserve apace for all AppleWorks program segments. AppleWorks will allow data files to overwrite all but a few key program segments option two loads in during the boot. Program segments can be overwritten because AppleWorks will search for the Program disk and load the required program segments if it does not find what it needs in memory.

You may want to run the AppleWorks memory expansion program twice and configure one set of disks with the “Y” option and one set with the “N” option. You will then be able to see how each version operates and decide which better fits your needs.

4. Insert your backup Startup disk into disk drive 1 for modification when prompted to do so by the instructions on screen. If AppleWorks is on hard disk, enter the pathname leading to the program as described on screen.

   The floppy drive will stop in approximately twenty seconds. When it stops, your backup Startup disk will be reedy for use with MultiRam C. Be sure you label this disk appropriately to avoid later confusion.

   If AppleWorks is on hard disk, both Startup and Program files will be modified and you will see a message at the end informing you that AppleWorks has been successfully modified. Enter BASIC and type the pathname that leads to your modified AppleWorks program “APLWORKS.SYSTEM.” You may want to descriptively rename the “/APPLEWORKS” program name to avoid any future confusion.

5. Insert your backup Program disk into disk drive 1 for modification when prompted to do so. The disk will atop in approximately one minute. You will see a screen informing you that AppleWorks has been successfully modified. Label your modified Program disk to avoid any future confusion.

After you have modified your disks, the only way you can change the configuration is to start at the beginning with a new backup copy of your AppleWorks disks and then re—configure the disks with the desired Option.

Insert the new Startup disk into disk drive 1 and press Control+[Open~Apple]+Reset to boot the new, more powerful AppleWorks MultiRam provides you!

APPLEWORKS RESTRICTIONS

AppleWorks has a number of internal restrictions with regard to file sizes, memory capacity, etc. This section discusses theme restrictions and changes the MultiRam C modification makes regarding the limitations. where applicable. Modifications to overcome restrictions not changed by the current AppleWorks memory expansion disk are planned. Software updates will be available to registered MultiRam C owners. See Chapter 10 for details.

Desktop Space and File Limits

AppleWorks supports a total of 12 files and 55K of memory for the Desktop. The MultiRam C modification permits the Desktop to expand to 413K. The number of files allowed remains at 12.

Clipboard Line/Row Limit

The Clipboard Is a temporary storage area in AppleWorks that allows the transfer of data between files. Currently, the Clipboard is limited to accepting a maximum of 295 lines or rows at one time.

Disk Capacity

The standard 5 1/4 inch Apple floppy disk is hailed to storing 136K of AppleWorks files, less than the size a file may grow to with MultiRam C memory on line.

MultiRam modifications allow AppleWorks to automatically break a large file into segments that can be sawed onto a series of floppy disks and allow AppleWorks to automatically load and restore the segmented file back into memory when needed. See ‘Working With Large Files In this chapter for further details.

If you will extensively create and use files in excess of 135K, consider purchasing a hard disk drive that supports the ProDOS operating system. Any file size that can be created with MultiRam can he saved to hard disk.

Program Segment Overlays

AppleWorks is a complex, large program. Only the most frequently used portions of AppleWorks are loaded into memory from the program dIsk when required due to the normal memory constraints of a 128K Apple //c. The result Is frequent accessing of the Program disk and program delays.

The MultiRam modification to AppleWorks eliminates those annoying delays by loading the full AppleWorks program into memory when it Is first booted. Complete Instructions and requirements for loading the program are provided earlier In this chapter.

Word Processor Documents

AppleWorks word processor documents are limited to a maximum of 2,250 tines, or approximately 45 pages. Please refer to “Working With Large Files” in this chapter for sore information.

Database Files

The standard AppleWorks database is limited to a maximum of 1,350 records. MultiRam modifications allow 5,100 records, almost a 300% increase in capacity

While this increase makes AppleWorks much more useful for small to medium size records, files with large record sizes (AppleWorks allows records to be as large as 1K) can exceed even MultiRam's memory before reaching the 5,100 record limit. Please read the section "Working With Large Files" in this chapter for more information.

Spreadsheet Worksheets

Worksheets are limited to 127,873 cells in the AppleWorks spreadsheet. Maximum columns are 127, maximum rows are 999

The AppleWorks Reference Manual States that an Apple //c with 128K RAM can hold approximately 6,000 filled cells. Although there is a considerable difference in the memory requirements to store labels and values versus formulas, exceeding the cell limits in AppleWorks, even with 413K of Desktop space, is unlikely. Refer to "Working With Large Files" in this chapter for more information.

WORKING WITH LARGE FILES

MultiRam C makes very large AppleWorks files possible. You will notice some differences in AppleWorks when working with large files. This section explains some general considerations you should keep in mind when working with large files as well as specific comments about the word processor, spreadsheet, and database segments of AppleWorks.

General Considerations

Three areas most affected when working with larger files are the time AppleWorks will require to perform various actions, the space remaining for the program if you chose to load all of AppleWorks into memory when configuring your disks, and the file size shown by AppleWorks.

Time

The larger a file becomes in memory, the longer many AppleWorks functions will take to produce their expected results.

Searches, sorts, calculation times, etc. all will take more time as a file grows in size. Most notably, a large spreadsheet file can take a very long time to recalculate a large number of complex cells. The speed of the Apple //c 6502 processor is the key factor in most calculation times.

AppleWorks will show you an approximate time for performing many operations when you first select the operation. For example, a message will appear telling you how long it will take to sort records in the database when you request a sort. These approximations were designed for a 128K AppleWorks; they are not accurate for larger files.

The actual time an action on a large file takes say be as little as half the time AppleWorks shows or as much as twice the stated time. The differences in actual versus approximate stated times will vary depending on the type of file in use and the operation called for.

For example, a sort on a large database file could be double the time Indicated if the sort is being performed on several fields with each record having information in the fields selected. A sort on the same database with only a few records having information in the fields being sorted could take half the time AppleWorks approximated. Moat of the time, however, the time stated by AppleWorks will closely approximate the actual time required to perform the selected operation on the large file.

Some activities may take what seems to he an extraordinarily long time to finish. If. this is the came, before restarting AppleWorks, give the action at least five minutes to finish. This should be more than enough time for any simple action on a large file to have been completed. More complex actions. like a recalculation on a complex, large spreadsheet may take somewhat longer to finish. In general, the larger the file and the more complex the task, the longer the time required. Look at the benchmark samples later ii, this section for some sample times to perform simple tasks.

MultiRam C **Size**

If you configure the AppleWorks Startup disk so that the entire program resides in memory **at** once (thereby speeding access to overlay portions of AppleWorks), approximately 130k will be used for the program. With a 256k MultiRam C, that would leave approximately 99k for large files: with 512k. approximately 283k would he left for files.

If the cumulative size of files on the Desktop grows larger than the available Desktop memory (total memory less 130k), then floppy users should have the Program disk in a disk drive as parts of the program will be Overwritten by data and AppleWorks will have to go to the Program disk to return the overwritten program segments to memory when they are needed.

Do not load AppleWorks into memory at bootup if you intend to consistently fill all of your desktop space with files. If data files fill most of the desktop, most of the program segments that took two minutes to load during bootup will be overwritten. You will have lost the advantage that you sought.

As suggested in the configuration section. you may want to configure two copies of AppleWorks. Use the copy that loads the program into memory if you will he working for a long session and do not anticipate filling the Desktop. The two minute wait during bootup will be offset by time savings achieved from the program being in memory. Boot the copy that doesn’t load AppleWorks into memory if you will be using the Apple a short while and/or will be working with a full Desktop.

File Size Differences

AppleWorks shows you the size of tiles on the Desktop and on a disk drive. The file sizes shown are an approximation of the actual sizes.

The sizes AppleWorks will report for files greater then 255K will be incorrect.

Figure 4.1 shows a catalog of a ProDOS hard disk subdirectory with one AppleWorks file “SAMPLE” in the directory. ProDOS reports the size of a file in blocks. A block is 512 bytes or 1/2 of 1K. Therefore the size of the file is 621 blocks divided by 2 or 310K.

The sase file when cataloged by AppleWorks shows 53K on disk end 67K when loaded onto the Desktop as Figures 4.2 and 4.3 show.

The reason for the discrepancy is that AppleWorks was not designed for files larger than 255K. the upper limit a single byte can indicate. A future modification to AppleWorks may be possible to correct this problem.

To tell how big a file is when saved on floppy disk is easy; an Apple 5 1/4 inch floppy can hold only 136K and a large file must be split over several disks. Count the number of disks the file spans and you can easily approximate its true size.

To tell how big a file is on hard disk is more difficult. When you save a file to a hard disk, you may wish to add the real size to the file name to avoid later confusion . From ProDOS, after exiting AppleWorks, catalog the AppleWorks files subdirectory, get the file’s block count, divide by two, and rename the file with the results. At minimum, you should at least add some indicator to the file’s name to inform you that the file is larger than 255K.

If you know a file is larger than 255K on disk, add 256 to the file size shown, and the result should approximately equal the right size. In our example. 53K plus 256K equals 309K, approximately the 310K determined earlier.

When a large file is loaded to the desktop, as Figure 4.3 shows, AppleWorks will continue to show an incorrect size. The size will always be larger then the size shown on disk as AppleWorks uses a compaction scheme when it saves files to disk. If the amount of space remaining on the desktop is small and the file on the desktop also shows a small size, you will need to add 256K to the file size to get a correct size.

The available space shown remaining on the Desktop is always accurate. The size of the large file shown on the Desktop, corrected by 256K, and the amount shown remaining on the Desktop may not equal the original empty Desktop, however. Every record/row of a file has some unusable bytes associated with it. The larger the number of records or rows in a file, therefore, the larger the discrepancy. Most certainly, you will not be able to load a file that is larger than the amount shown remaining on the desktop.

Benchmarks

Following are sample values describing some of the time requirements or capacities for certain operations in AppleWorks. Because of the many variables involved, some values will very significantly in different test situations.

General

Desktop space with different MultiRam C configurations:

MultiRam size	Total Apple	Desktop size
256K	384K	229K
320K	448K	275K
512K	640K	413K

Time to load/save a full disk (135K) of information: 1:15 minutes

Word Processor

number of lines in document	500	1000	2250	2250
platen width (PW), inches \ a	8.5	8.5	8.5	13.0
left margin (LM), inches > typical	1.0	1.0	1.0	0.0
right margin (RM), inches / page	1.0	1.0	1.0	0.0
memory size	27K	55K	119K	130K
disk size	25K	53K	111K	122K
load/save time, minutes	:17	:31	1:03	1:08
time to insert 20 lines at beginning	–	–	:08	:12
time to insert 20 lines at end	–	–.	:22	:26

Spreadsheet

contents of cell	label/ value	label/ value	simple formula	simple formula
number of filled cells	5.000	10,000	5,000	10,000
memory size	58K	116K	126K	252K
disk size	57K	113K	125K	250K

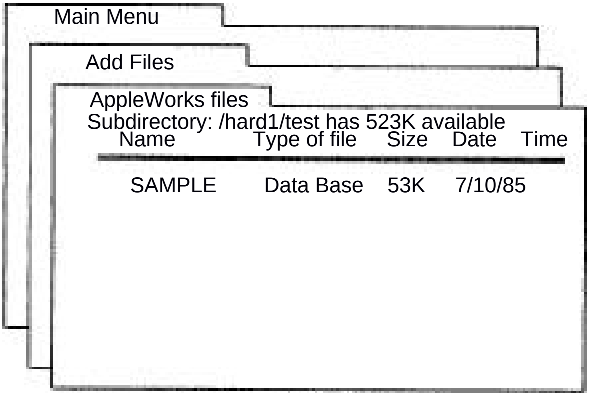
Time to recalculate 10,000 formulas of the type “n=n+1”	: 2:10 sin.
Time to change cell size ([Open–Apple]+V) for 20,000 cells	: neglible
Time to insert 250 rows at. top of large (250K) file	: 0:08 sin.
Time to insert 250 rows at bottom of large file	: 0:48 mm.

Database

number of records	100	1000	3300
total size of file	6K	57K	310k
individual record size (no. of characters)	100	100	100
time for alpha sort, minutes	0:02	0:14	:38
time for numeric sort, minutes	0:02	0:30	1:30

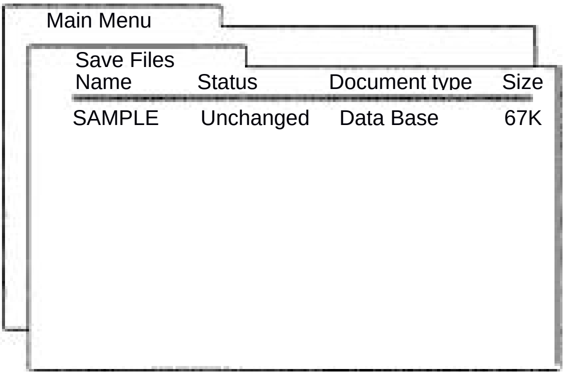
```
cat/hard1/test

TEST

NAME          TYPE     BLOCKS     MODIFIED

SAMPLE        ADB         621     10—JUL—85

BLOCKS FREE:  1047     BLOCKS USED:  2089
```

FIGURE 4.1



```
Subdirectory: /hard1/test has 523K available
   Name          Type of file    Size   Date    Time
   ------------------------------------------------
    SAMPLE         Data Base     53K    7/10/85
```

FIGURE 4.2



```
Name          Status          Document type     Size
----------------------------------------------------
SAMPLE        Unchanged       Data Base          67K
```

FIGURE 4.3

Word Processor Flies

AppleWorks limits word processing files to 2,250 lines. This is the equivalent to 45 single—spaced, typewritten pages with standard margins. While AppleWorks supports print lines up to about 140 characters, it only displays about 74 characters on a screen line. Given these restrictions, you can create a word processor file that uses 130K of memory. 122K on disk, and takes 1:08 minutes to load or save from floppy disk.

If you wanted to create a larger file, you could save the current file on disk when you reached its limit and then create a new file from scratch by choosing selection I, "Add a file to the Desktop' from the main menu. Continue entering text into the new file from where you left off with the first file. Be sure to use the same file specifications (platen width, top margin, left margin. etc.) as on the first part of the file. If you leave both files in memory, you say jump back and forth between files using the [Open Apple]+Q command and you can transfer text between them using the clipboard and either [Open—Apple]+C (Copy) or [Open-Apple]+M (Move).

Several peculiarities occur in the AppleWorks word processor when the file size approaches 2,250 lines. When the cursor is on line 2,250, the "Delete" key usually will not work. Also, with 2,250 lines in memory, you will be unable to use the [Open—Apple]+O (Print options) or [Open—Apple]+P (Print copy) commands. To be safe, you should end your word processor files a few lines short of 2.250 lines.

There is no discernible slowing of most commands when working on a large word processor file. The [Open—Apple]+1 and [Open—Apple]+9 cursor commands bring you to the top or bottom of the file in a flash. Speed of text entry is also uneffected. However, copying and moving text around, along with re-formating text (ie., changing platen width from 8.0 inches to 8.5 inches) can take a long time. Inserting text from the clipboard can take from 8 to 26 seconds, depending on how much or where the text is placed. Certain commands such as [Open—Apple]+D (Delete) and (Open—AppleA]+S (Save) may also take a bit longer to function.

Keep these potential delays in mind, and don't panic if the keyboard refuses to respond. Wait a minute or so. and allow the program to finish its task.

Spreadsheet Files

In the hands of an experienced user, a spreadsheet can grow to enormous proportions. The AppleWorks spreadsheet is limited to 126,873 cells, which is less of a limitation than available memory.

Before creating large spreadsheet files, please keep in mind that speed and efficiency suffer with larger spreadsheet files. For example. a spreadsheet file consisting of 12,000 celia (primarily formulas of the form "n=n + 1") took up 309K of memory and required over 2:40 minutes to recalculate. Spreadsheet files should be kept as small and compact as possible for best results.

When working with a very large worksheet in memory, it is a good idea to leave an empty border of 2 or 3 rows or columns at the top, sides and bottom. As with the word processor, the date moving commands (Hove and Copy) won’t always operate as expected at the edges of the worksheet.

For example, if you fill every cell of a worksheet that stretches from Al (upper left) to A999 (lower left) to K1 Copper right) to K999 (lower right), and then move the bottom 250 lines to the clipboard using [Open—Apple]+M (Move), an error message “Not enough space to move aM Ia existing data” would result when you try to move the same 250 lines back. Delete one row (A999), however, and you can move the 250 row back back and forth to the clipboard with no problem. Moral: Stay away from the edges.

Database Files

MultiRam modifications to the AppleWorks database allow 5,100 record database files to be created and saved to disk,

The actual number of records that can be entered into a file will depend upon the size of the records. The larger the record, the fewer records can be put into a file. AppleWorks maximum record size is 1K which would permit approximately 400 records to be saved. Such large records are not typical, however. A typical address file consisting of of a persons name, address, telephone number and a few fields of other information could easily hold 5,100 records.

You say have a number of data base files that you split into small segments to fit the old 55K Desktop space. These split files can now be merged into one large file if they contain exactly the same categories in the same format, i,e,, the names of the categories (fields) must be in the same alignment, like templates:

Example:

FILE 1 — Name	Address	City	State	
FILE # 2 — Name	Address	City	State	
FILE # 3 — Company name	Name	Address	City	State

In the above example, files can be moved between file ONE and file TWO interchangeably, but merging file THREE with file ONE or file TWO would seem to create difficulties, In fact, it is possible to combine these, but some changes must be made.

You may merge the similar files, ONE and TWO, and then fit file THREE into the combined file (ONE and TWO).

Instructions for merging files ONE, TWO and THREE follow. These instructions can be used to serge any database files with AppleWorks. When working with your own files, don’t be afraid of making a mistake when combining files. No damage or change can possibly be done to your files on disk while you are working with them in memory. If you are wary of a problem, sake a backup copy of your data disk before you experiment.

To merge files

1. Load all files (in this example, files ONE, TWO, and THREE) into memory. Press [Open–Apple]+Q (Desktop index). A window will appear in the center of the screen showing the names of all the files in memory.

2. Decide which of the files you went to merge all of the records into. We will use ONE am the main file in this example, moving first TWO, then THREE into it. Move the cursor to file TWO and press RETURN to select it.

3. Make sure the display is in the “MULTIPLE RECORD LAYOUT” format on the screen ([Open–Apple]+Z toggles this display). Place the cursor on the first record in file TWO and press (Open–Apple]+M (Move). Use the arrow keys to choose “To the Clipboard (cut)”. The entire top line will turn inverse. Follow the prompts, using the down arrow to move the cursor to cover the records you want to move.

   Pressing [Open–Apple] key in conjunction with the arrow keys will allow you to move the cursor one whole screen (15 lines) at once: holding this key combination down will repeat, screen after screen, until all files are covered with this Inverse cursor. Alternately, you may press the [Open–Apple! end a number key and you can select up to 250 records at a time.

   From one to 250 records may be moved at one time. If you attempt to move more than 250 records at once, no records will be copied into the buffer and you are returned to your starting point.

4. When all of the records that you selected are covered with the inverse cursor, press return and those records will disappear from the file. The selected records are now on the Clipboard, a temporary storage area that cannot be seen.

5. Press [Open–Apple]+Q (Desktop index), choose file ONE sad press return. File ONE will replace file TWO on the screen. Press [Open–Apple]+M (Move), choose “From the Clipboard (paste)” end press return. All the files from the clipboard will appear on the screen.

6. Repeat this process until all of the records are moved from file TWO to file ONE. Although only three files are used in this example, twelve files can be on the DESKTOP at one time and can each be worked with individually.

When merging dissimilar database files, the files suet he altered so that the ease number of categories is reflected in both. This means that either one or sore categories must be added to one file, or one or sore categories deleted from the other.

AppleWorks is quite forgiving. The type of information in each category is relatively unimportant; as long as the number of categories match in both files. no information will be lost. It is critically important to arrange the categories in the same order in each file (i.e.. name, address, city, and state in our example). If this is ignored, you will have a laborious task editing each odd record individually.

In our example, the destination file (ONE) is modified by the addition of another category. so that none of the data in the merged file (THREE) is lost. Further, the categories in ONE will be rearranged so that it is in the same order as THREE. Note that they don't have to be the same names nor match upper or lower case, but they must be in the sequence. The category names in the destination file (ONE) will prevail and the other category names will be lost.

7. With file number ONE on the screen, press [Open–Apple]+N (Category Name change), use the arrow keys to move the cursor to the first Category position (NAME in our example). Press [Open–Apple]+I (Insert) and make a new Category above NAME. Type in "COMPANY NAME" and press "Return", then "Escape". You have now matched the format of file THREE, i.e., COMPANY, NAME, ADDRESS, CITY, and STATE.

8. Press [Open–Apple]+Q (Desktop index), choose file THREE and press "Return". Repeat the process discussed in steps 3 through 6, using [Open–Apple]+M (Move) to transfer the records from file THREE to file ONE, until all records are transferred.

Now that all three files are now merged into one, you can manipulate that file as you normally would any AppleWorks database file. Change the name of the final file using [Open–Apple]+N, then save it under the new name.

This concludes our discussion of techniques used in manipulating very large Appleworks files. We at Checkmate Technology welcome any tips or suggestions that you have in managing large files. We will pass along your tips as they accumulate to registered owners through our newsletter (see Chapter 10).

SAVING LARGE FILES

MultiRam C makes it possible to create very large AppleWorks files. These files may be saved to floppy disks or a hard disk drive.

Segmenting Floppy Files

AppleWorks data disks prepared on the typical 5 1/4" Apple Disk II floppy drive hold a maximum of 136K of AppleWorks data. Therefore, an AppleWorks file larger than 136K in memory must be segmented or divided into parts and saved over a series of disks. MultiRam modifications allow AppleWorks to do this automatically for you.

When you save a file that is larger than the apace remaining on a data disk, the file will automatically be split when the disk is full. On-screen prompts will allow you to save the file over a number of floppy disks. When you are ready to reload the file, AppleWorks will prompt you to load each segment of the file that has been split over several disks.

Preparing for Segmentation

Saving very large files does involve some planning and time when you are using standard Apple 5 1/4 inch disk drives.

The first requirement for using large files and floppy disk drives is disks and lots of them. Don’t consider starting a session with one or more large files without having at least three to five blank or recycled disks per large file you will be working with. Any file over 260K will require three blank disks to save the file.

Although you can save a modified large file back to the same disks you originally loaded it from, we do not suggest you do this.

If you overwrite the original file and something happens before you finish saving the file — the power goes out, the disk drive develops a problem, you accidentally open the disk drive door, etc. — you will have lost some, most, or all of the large file you worked long and hard on. Forever. For this reason. please always save your newly modified large files to blank or recycled disks. This will protect the time investment you made in creating your file.

You should consider overwriting an existing set of disks only if you have made one or more backup copies of the set of disks involved.

The second requirement is time sod peace of mind. It will take over a minute to save each full 5 1/4 inch floppy and an equal amount of time to load the segmented file back into memory. A 300K file can take from 3 to 5 minutes to save and load. Don’t be in a hurry when saving large files as you can make mistakes that say later cost you more time. Forgetting to label each disk to show which segment is on what disk, starting to save s file without having bought enough disks, and forgetting to format disks as required causing you to start all over again are but a few mistake you can make when you are impatient.

Saving Files

To save a large file, whether for the first time or after the file has been loaded from disk, modified and is to he rewritten, follow these steps

1. Determine the file size and number of disks needed to save the file.

   Press the Escape key to return to the main menu from the file you are working on. Press “3” to save Desktop files. The save menu shows the size of the file to be saved. Make the size adjustment described under “File Size Differences” in the previous section. Divide the corrected size of the file by 1301 to determine the number of disks required to save the file. Always round up to the next highest number. For example, to save our 3101 film SAMPLE discussed earlier, three disks would be needed.

   Press Escape to return to the main menu. D0 NOT save at this time.

2. Format the required number of data disks with the same volume name.

   From the main menu choose “5” to go to other activities and then select “5” to format a blank disk. Enter a disk name no more than 15 characters long.

   After the first disk is formatted, press the space bar to return to the disk formatter. Remove the first disk and place the next blank disk in the drive. Label the first disk with a “1” to indicate that you will use it to save the first file segment.

   Continue formatting data disks with the same name until you have formatted the required number of disks. Also continue labeling the disks in sequence. Press the Escape key to return to the main menu when you finish.

3. Save the file. Choose “3” from the main menu to save Desktop files to disk. Use the arrow keys to select the correct file.

   The name you save the file under should not be sore than 13 characters long: ProDOS allows only 15 characters in a file name and the segmentation routine will add to the end of your file name a period and a number from 2 to 9 indicating the segment sequence number. If your file name is 15 characters long, to satisfy ProDOS’s 15 character limit, the last two characters vi 1 be deleted and replaced with a period followed by the segment number. The first segment will not have a “.1” appended but segments after the first disk will start with a *“.2”* suffix.

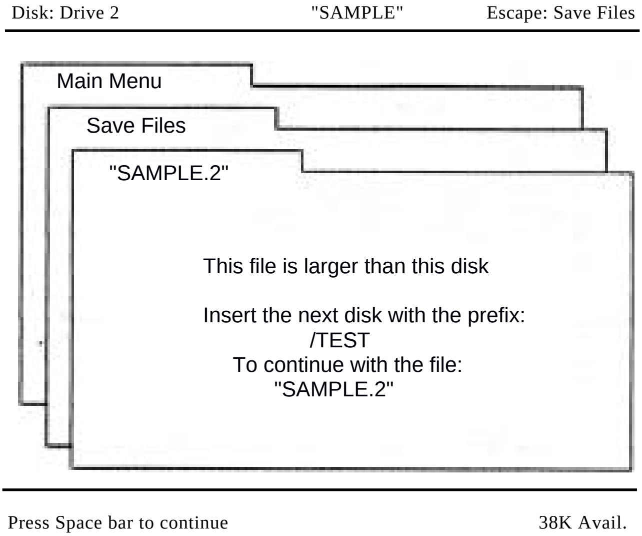
   Insert the blank, formatted disk labeled “1” (and also labeled with the disk name and any other descriptive material) into the data disk drive and press the Return key. Assuming the file is larger than one disk, like our example “SAMPLES”, it will take just over one minute to save the first segment of the file.

   When the first disk is full, you will be prompted to insert the second disk as Figure 4.4 shows. When that disk is full, you will be instructed to insert the next disk as Figure 4.5 shows. This will continue until the file is finally saved.

   The disks in our example named “TEST” will have the file “SAMPLE’ saved to them. Disk I will have saved to it the first segment “SAMPLE”. disk 2 the segment “SAMPLE.2”, and disk 3 the last and smallest segment “SAMPLE.3”.

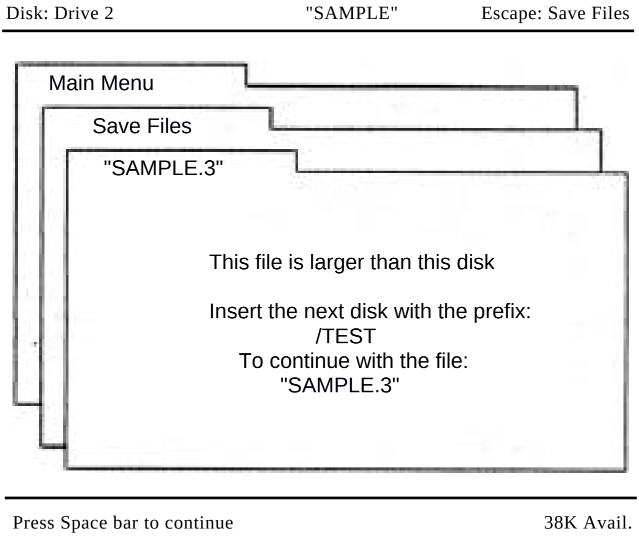
AppleWorks normally first saves the desired file in memory to a temporary file on disk. Assuming you decided to save the file under its existing name on disk and the temporary file is successfully saved, AppleWorks then renames the temporary file to the original file name and deletes the old file. This way, an updated file is created only after it has been safely saved. If any problem occurs while trying to write the disk, the old file still remains on disk as a backup. Part A of Figure 4.6 illustrates this action.

Segmentation is activated only when AppleWorks detects no more room on a disk but still has part of a file left to save. The segmentation routine will cause a file to be saved over two disks if it can not fit on the space left on one disk.



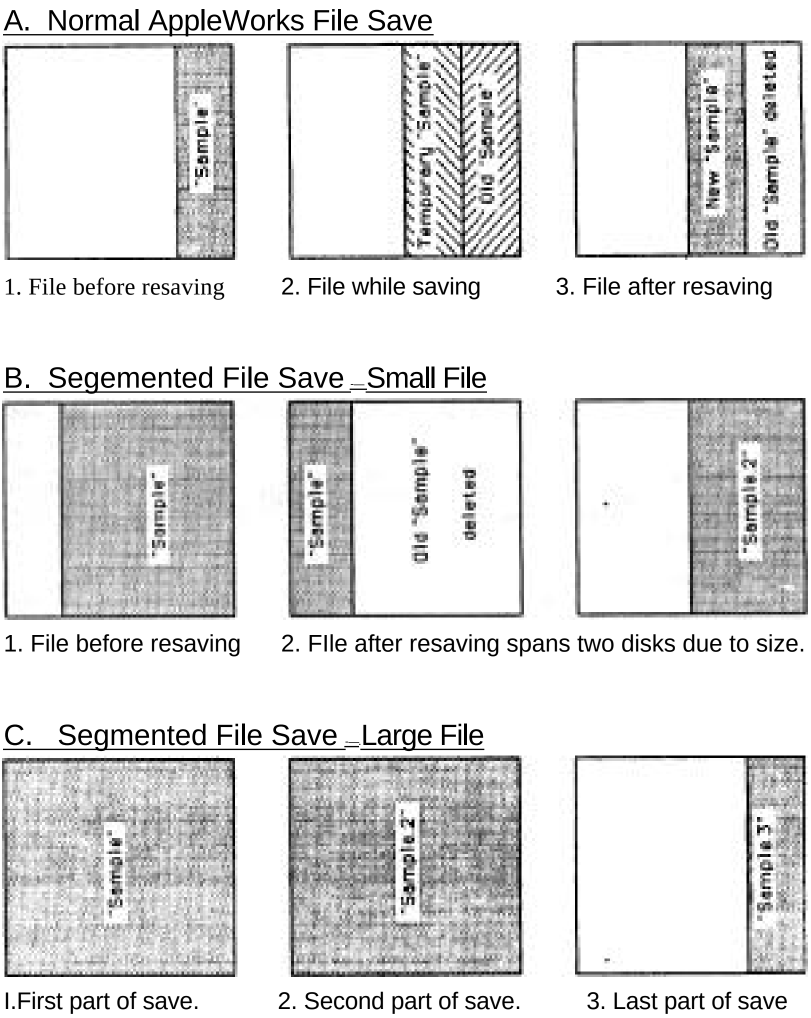
```
Disk: Drive 2              "SAMPLE"          Escape: Save Files

Main Menu
  Save Files
    "SAMPLE.2"

              This file is larger than this disk

              Insert the next disk with the prefix:
                          /TEST
                 To continue with the file:
                        "SAMPLE.2"

Press Space bar to continue                          38K Avail.
```

FIGURE 4.4



```
Disk: Drive 2              "SAMPLE"          Escape: Save Files

Main Menu
  Save Files
    "SAMPLE.3"

              This file is larger than this disk

              Insert the next disk with the prefix:
                          /TEST
                 To continue with the file:
                        "SAMPLE.3"

Press Space bar to continue                          38K Avail.
```

FIGURE 4.5



A. Normal AppleWorks File Save

1. File before resaving 2. File while saving 3. File after resaving

B. Segemented File Save – Small File

1. File before resaving 2. FIle after resaving spans two disks due to size.

C. Segmented File Save – Large File

I.First part of save. 2. Second part of save. 3. Last part of save

FIGURE 4.6

□ Free disk space □ Disk space used by file

The MultiRam file segmenting procedure does not affect AppleWorks internal backup safeguard as long as there is room on the disk for the temporary file being saved and the old, existing file. But if there is no inns on the disk to save the temporary file, even though the file could easily be saved to a blank disk, the segmenting routine will save the first part of the file to the disk, delete the old file, and then request a new disk to continue the file. Part B of Figure 4.6 shows this result.

If you do not want this to occur, you should save the file to a new disk with room for it or you may save the file under another name and let the file span more then one disk.

If you find yourself with no disks left, you might consider deleting the old file first using the “Other Activities” menu and then save the file all on the existing disk. You run the risk of losing the file this way, however. as was described earlier.

Part C of Figure 4.6 shows that there can be useable space left on a disk in the last part of a segmented save. To eliminate any possible problems in the future with the file “SAMPLE”, you should not try to use this remaining apace for saving other files.

Loading Files

Loading segmented files back into memory is easy.

Insert the first segmented disk into the data drive. Select “Add Files” and choose the first segmented file. You will see a message just like that in Figures 4.2 and 4.3 to let you know which file segment to load next. The file will be ready for use when the last segment loads into memory.

Error Messages

Two types of error messages are likely to occur if you are not correctly saving or loading segmented files.

The message Can’t write on disk at Drive 2” can occur if you insert a disk during a segmented save whose name does not match that of the first disk.

To resolve this problem, find the data disk with the proper name and insert it to continue the save. If you have not prepared enough data disks to handle the entire file, you will have to press Escape to abandon the present file save. Reread the section pertaining to disk preparation and format enough disks for the file. You will not lose any of your data but you will have to Start saving from the beginning again.

The message “Getting errors trying to read <file segment name> on Drive 2” can occur if you try loading a disk out of sequence. Simply remove the incorrect disk and insert the disk requested by the prompt.

If the first disk you load is not the first disk in the sequence, unpredictable results can happen as only the first file contains critical information relating to the segmentation. At best, error messages that make no sense will appear.
At worst, AppleWorks will hang or crash.

Be very careful to always correctly and boldly label the first disk of a segmented file to avoid potentially disastrous results.

Hard Disks

Using a hard disk drive with MultiRam C and AppleWorks can greatly ease your file management efforts. All hard disk drives can easily store a large number of files without any special preparation by you.

An additional advantage of a hard disk is that it takes much less time to store and retrieve AppleWorks files then does a floppy drive. The same 300K file that would take 3 to 4 minutes to load from standard 5 1/4 inch floppy would take from 30 to 60 seconds to load from a hard drive.

If you do use a hard disk, one caution: do not allow the area you have partitioned for your AppleWorks files to become too full. If you run out of room on the hard disk while saving a file, MultiRam modifications that allow large files to span multiple floppy disks will be activated and will instruct you to insert a new “floppy” disk for the next file segment. You cannot do this on a hard drive. Always check the remaining space you have available on your hard drive before saving a file to avoid this potential problem. Refer to the preceding section “File Size Differences” regarding inaccuracies in AppleWorks measurement of hard disk file sizes greater than 255K.

You should seriously consider purchasing a hard disk drive if you intend to regularly use large AppleWorks files. See Chapter 6 for more information.

SMALLER FILES

Undoubtedly, creating truly large word processor, database or spreadsheet files is one of the biggest appeals of installing the MultiRam C. However, even if you never need to make a file larger than 55K, you will realize many benefits from the MultiRam C and AppleWorks combination.

Speed of Execution

Loading the entire AppleWorks program into memory significantly speeds the operation of AppleWorks by eliminating the need for reading in segments that normally stay on the Program disk until needed. Remember that the more memory installed on MultiRam, the better you will be able to use this feature. A 256K MultiRam C will provide sufficient memory for AppleWorks storage, besides nearly doubling the standard 55K Desktop space.

Switching Between Files On The Desktop

The multi--file Desktop is one of the beat features of AppleWorks.

With only 55K of standard Desktop apace, however, it can be a nuisance to have to remove a 45K database tile from the Desktop in order to sake room for a 12k word processor document or spreadsheet. And with 55K. transferring data between applications can became a laborious chore that spoils the “integrated” aspect of AppleWorks.

With MultiRam C’s larger Desktop sizes, however, these old problem disappear. A Desktop of sufficient size will allow you to stop work on one sizeable file, switch to work on another good size file, and then return to the first file without going to the bother of saving your data.

With MultiRam C, the AppleWorks Desktop works for you much mote like a real office desktop, s big desk that allows you to spread out your files and work more efficiently between them.

By increasing the effectiveness of the Desktop, MultiRam C truely allows AppleWorks to fulfill its promise as one of the best integrated programs.

Chapter 5

RAM DISK EMULATION

The MultiRam C Card can be made to look like a high capacity disk drive to DOS 3.3, ProDOS, and Apple Pascal operating systems.

After MultiRam C is setup as a RAM disk, it can be used like any disk drive except it will work many times faster since it is RAM based and has no mechanical parts. Depending upon the type of operating system and commands used, disk access times can be forty times faster than normal.

To transfer a program to a RAM disk, the program must be available on disk as an unprotected file, a file that can be cataloged using a normal, unmodified operating system. Most copy protected programs cannot be transferred to a MultiRam C RAM disk although some may be able to use the RAM disk as a fast data disk.

DOS 3.3 RAM DISK

The MTULTIDRIVE.DOS program converts your MultiRam C Card’s memory and the 64K auxiliary memory of the //c into fast electronic disk drives for the DOS 3.3 operating system. The DOS RAil disk drive is used as up to two slot 3 disk drives.

Specifications

MultiRam C memory and the 64K auxiliary bank of the //c is allocated by the RAM disk program to provide up to two large capacity disk drives with MultiRam C’s full 512K of memory. With a 256K standard RAM configuration, the RAM disk program creates one 320K RAM disk (256K MultiRam memory + 64K //c auxiliary memory = 320K). With 256K in Bank A, 64K in Bank B, and 64K auxiliary memory on the //c, one 384K RAM disk is created. With a 512K MultiRam C, one full 384K RAM disk and one partial 192K RAM disk are crested.

Maximum RAM disk capacity	:	384K
RAM disk sizes possible	:	320K, 384K, 192K + 384K
Maximum number of drives	:	2
Sectors per track	:	32
Tracks per 64K	:	8
Maximum tracks per drive	:	48
Slot assignment	:	3

Apple 5 1/4 inch floppy drives have a total usable memory of 128K (140K total less four tracks reserved by DOS) so that MultiRam C’s full 384K disk drive is the equivalent of three standard Apple floppy disks,

Preparing The Program

The RAM disk program is on the Utilities side of your MultiRam disk. You will need to save this program to another disk as it cannot be used directly from the Utilities disk. The Utilities disk uses the ProDOS operating system. Programs stored on a ProDOS disk cannot be directly transferred to a DOS disk. You will need to use the Convert program, also found on the Utilities disk, to transfer the RAM disk program for you.

Before you start, please initialize a blank disk with your DOS 3.3 System Master disk. You will transfer the RAM drive program to this disk. You may want to use this disk for an auto—boot disk as discussed later in this chapter. A DOS formatted disk with files already on it say also be used if at least 10 sectors remain on the disk. Place the DOS formatted disk in drive 2.

Boot the Utilities disk (the back of the MultiRam disk). Choose the MULTIDRIVE.DOS from the menu. Follow the on—screen instructions and press the appropriate key to run the Convert program which will convert the ProDOS based binary file into a DOS based file.

The Convert screen should show “DIRECTION: PRODOS —----> DOS 3.3 S6,D2” at the top of the screen. Press “R” to reverse direction if it does not. If you are not using an external drive, press “C" to change to drive 1.

Press “T” to transfer the file. Use the right arrow key to copy over the prefix than enter “MULTIDRIVE.DOS” and press the Return key. After the disk drives stop, your DOS RAIl disk will be ready for use.

You may place your DOS disk with the “MULTIDRIVE.DOS” program on it in drive 1 and press Control+[Open—Apple]+Reset to boot the disk.

To rerun the Utilities menu. leave the Utilities disk in drive 1. Press “Escape” to return to the Convert main menu. Press “Q” to quit the Convert program. Press the “Return” key at the “LOAD WHAT SYSTEM PROGRAM?” prompt.

Installation

Install DOS 3.3 by booting your DOS SYSTEM MASTER diskette, or any other DOS 3.3 disk that leaves you with an Applesoft prompt. “I”. on the screen The DOS RAM disk can be installed now in slot 3 from the keyboard by typing “BRUN MULTIDRIVE.DOS.’

To install the RAM disk from within an Applesoft program, enter this line: PRINT CHR$(4)”BRUN MULTIDRIVE.DOS.” The RAM disk program should be on the same disk as the Applesoft program using this line. When the program is run, a message appears telling you the number and size of RAM disk drives initialized.

Transferring Files

Files must be copied to and from the RAM disk using a file copier. FID. found on the DOS 3.3 System Master, is easily used for this purpose. A disk copy program, such as COPYA, will not work. If you use a file copier other than FID, it must be able to copy files to and from disks that have more than 35 tracks.

Always end a session with the RAM disk by copying any files on the RAM disk that you wish to permanently save to a floppy disk. Programs can also be directly saved to the RAM disk from memory. Similarly, programs can be loaded into the Apple’s memory from the RAM disk.

CAUTION: When using a RAM disk, DO NOT create a file greeter in size than the disk you can save it too. A standard 5 1/4 inch Apple disk drive can save 128K of data with DOS on the diskette, or 140K without DOS. A hard disk drive is recommended for large files.

Auto—boot Disks

You may create disks that will automatically install the RAM disk, transfer to the RAM disk those files you would like to have there, and then run a selected program from the RAM disk.

To create an auto–boot disk, you may use the disk onto which you transferred the RAM disk program. That disk already has the RAM disk drive program on it. You will need to copy one other file to the disk, FID. Use FID to make a copy of itself onto the auto–boot disk.

Next, after clearing memory with the “NEW” command, enter this Applesoft line: 10 PRINT CHR$(4)”EXEC AUTO–BOOT”. Save this short program under the name HELLO, the boot file of the disk you initialized earlier. This program will automatically run when you boot the disk.

Last, create an EXEC file to control the auto–boot process. An EXEC file is a text file that can control the Apple just as you would from the keyboard. You may create a text file with your favorite word processor or from an Applesoft program like the following. Consult your DOS 3.3 manual for sore information on EXEC files.

EXEC File Maker **Comments**

```
10  D$= CHR$(4)                  Set DOS command prefix
20  PRINT D$”OPEN AUTO–BOOT”     Create a text file
30  PRINT D$”WRITE AUTO–BOOT”
31:
32: REM EXEC PILE CONTENTS
33:
40  PRINT “BRUN MULTIDRIVE.DOS”  Run the RAM disk program
50  PRINT “BRUN FID”             Run the FID program
60  PRINT “1”                    Select “Copy” from the FID menu
70  PRINT “6”                    Slot 6 for program to copy from
80  PRINT “1”                    Drive 1 for program to copy from
90  PRINT “3”                    Slot 3 for program to copy to
100 PRINT “1”                    Drive 1 for program to copy to
110 PRINT “=”                    Select all programs for copying
120 PRINT “N”                    no prompt, copy all
130 PRINT “B”                    Begin copying (any key would start it)
131:
140 PRINT D$”CLOSE AUTO-BOOT"    Save EXEC file to disk
```

Running the preceding program will create and save a text file to disk that when EXEC’d first runs the RAM disk installation program, runs FID, and then passes values to FID so that all the files on the disk. including P10 and the RAM disk drive program, are copied to the RAM disk. The only thing to add to the disk are the files you would like to see moved onto the RAM disk.

You my customize this EXEC file maker to control FID to do other actions than an automatic transfer of all files on the disk.

DOS Commands

All DOS 3.3 commands will work with the RAM disk except the INIT command. You will see “I/O ERROR’ if you attempt to initialize the RAM disk with this command. The RAM disk will not be affected if you accidentally do try to INIT the RAM disk.

Disconnecting The RAM Disk

The RAM disk can be destroyed by turning off the Apple, by doing a Control+[Open—Apple]+Reset wars boot, or by running a program that uses auxiliary bank zero, the //c’s own auxiliary emory. Running the MULTIDRIVE.DOS program after a RAM disk is installed will reforest the RAM disk eliminating any files that did exist on the RAM disk.

Using 80—column or double hi—rem displays will not affect the disk. These areas of memory are not included in the RAM devoted to the RAM disk.

Pressing the “Reset” key or “Control C” has no effect on what is in memory on the MultiRam C Card.

Non—Standard DOS

Many forms of customized DOS exist. The RAM disk will work with most customized versions that have not made major changes to DOS. For example, ProntoDOS and Diversi-DOS which speed up disk access times work with the RAM disk. Other more radically altered DOS say not work with the RAM disk (for example, David—DOS).

If you have made or intend to make patches to DOS yourself, almost all of the most common modifications will work with the RAM disk. Increasing the number of files per disk, increasing the size of binary files, custom catalogs, custom error messages, and hello files that can be BRUN or EXEC’d are examples of modifications that should work with the program.

Programming Tips

DOS 3.3’s Read/Write/Track/Sector routine, RWTS, can be used to directly read or write to the RAM disk. If you wish to use RWTS with the RAM disk, follow all standard protocols and do not call into RWTS; enter RWTS from the top of the routine. The standard or custom RWTS Input/Output Block (IOB) and its sector buffer must lie in the $200 to $BFFF area of main motherboard memory. Programs written to run in the language card area of memory must have their IOB table and sector buffet in the lower 48K main memory area.

Auxiliary 64K bank 0 Ia always in place. The only time other 64K banks are activated is when RWTS calls them.

DOS Updates

The DOS RAM disk software will be updated in the future adding new features. Some new features planned include a user re–definable slot option and a customization utility to allow you to modify the RAM disk itself.

Registered owners will be notified of updated RAM disk software.

PRODOS RAM DISK

The MULTIDRIVE.PRO program converts your MultiRam Cs memory into a fast electronic disk drive for the ProDOS operating system. The ProDOS RAM disk is recognized by ProDOS as one large ProWS volume.

Specifications

The MULTIDRIVE.PR0 RAM disk program creates one ProWS volume named “/RAM,” located in slot 3, drive 2. The size of the “/RAM” disk depends upon the amount of memory on the MultiRam C Card. RAM disk memory can be as small as 256K or, with a full 512K of memory on the MultiRam C, as large as 512K.

The “/RAM" disk created by the MULTIDRIYE.PRO program will take the place of the 64K “/RAM" drive automatically created by the ProDOS operating system in the 64K auxiliary bank of the //c. You will always have a larger RAM drive using the MultiRam “IRAN” disk than the RAM disk ProDOS automatically creates.

The MultiRam ProWS RAM disk uses only memory on the MultiRam C Card to create a RAM disk so that the RAM disk directory and files stored on the RAM disk cannot be destroyed by booting a program that uses main memory end the standard //c 64K auxiliary memory bank.

The ProWS RAM disk can be used with ProDOS version 1.1.1 or any older version of ProDOS and is compatible with floppy and hard disk drives.

The storage capacities of the ProWS RAM drive are as follows:

MultiRam C Size	RAM Disk Size	Total Blocks	Blocks Free	Blocks Used
256K	256K	512	495	17
3201	320K	640	620	20
512!	512K	1024	1024	29

In comparison, a standard Apple 5 1/4 inch floppy disk formatted by the Filer with the ProWS operating system and the BASIC.SYSTEM file saved to it (which are already in memory whenever the ProDOS RAM disk is used) has 2B0 total blocks available. 221 blocks free and 59 used.

Installation

To install the ProDOS RAM disk, simply boot the Utilities side of the MultiRam disk. Select the ProDOS RAM disk option from the main menu and the RAM disk will be automatically installed. You will then see a message telling you the disk has been activated and you will also see the memory size used by the ProDOS RAM disk.

Using the ProDOS Filer, which is included on the Utility side of the MultiRam disk, you may transfer the MULTIDRIVE.PRO program to any ProDOS formatted disk. To run the program, simply type “MULTIDRIVE.PRO” after the disk the program is on has booted and has displayed the Applesoft “]” prompt. If the program is on a disk other than the one you originally booted, first type the disk s prefix and then the program name. For example, “—/USERS.DISK/MULTIDRIVE.PRO” would install the RAM disk if you had transferred the program to a copy of your ProDOS Users Disk.

You may also have the RAM disk automatically installed upon bootup if you have the MULTIDRIVE.PRO program and the BASIC.SYSTEM and PRODOS system files saved to a ProDOS formatted disk. To automatically create a RAM disk upon bootup, save this one line program under the name "STARTUP”:

```
          10 PRINT CHR$(4)”BRUN MULTIDRIVE.PRO”
```

To install the RAM disk from within any Applesoft program, enter the preceding line. Be certain the program is on the same disk as the Applesoft program using this line; otherwise you must precede the program name with the volume name where it can be found.

Using The RAM Disk

The RAM disk drive can be used like any other disk drive with ProDOS. You may transfer programs into the RAM disk from other disks and from the RAM disk to other disks; you may save programs to the RAM disk from memory; and you can load programs from the RAM disk into memory. Unlike other disk drives, however, you should end each session with the RAM disk by saving any files desired to a standard floppy or hard disk as the RAM disk’s memory is lost when the Apple is turned off.

Because the RAM disk can become quite large, you may wish to use subdirectories with the RAM disk. ProDOS allows only 51 file entries in its volume directory. With subdirectories, you may have 51 files stored per subdirectory.

The ProDOS RAM disk’s volume name is “/RAM” by default. Using the Filer volume command “Rename”, the volume name can he changed to any legitimate ProDOS volume name.

To clear the RAM disk of all files, you could format the disk using the Filer’s “Format a Volume” command. If you have renamed the RAM disk, the RAM disk’s volume name will be changed back to “/RAM” after using the Filer format command.

Transferring Files

Files must be copied to end from the RAM disk using a file copier. The Filer’s “Copy Files” commend is easily used for this purpose. The “Copy a Volume” command, found under the volume commands, will not work. The Filer program is included on the Utilities side of the MultiRam disk.

Always end a session with the RAM disk by copying any files on the RAM disk that have been modified and will be used again to a floppy or hard disk.

CAUTION: When using s RAM disk, DO NOT create a file greeter In size then the disk you can save it to. A standard Apple disk drive can save approximately 256 to 280 blocks to a disk. A hard disk drive can save far more, usually 10 megabytes or more. Use the Filer to determine the amount of memory you have left on any storage device before trying to save a large file stored on the RAM disk to it. Although you can easily create very large files if your card has 256K or more memory, remember that you still must save the files you create to non—volatile storage media. Plan ahead so that you do not find yourself with a file that is larger than any storage device you can successfully use.

ProDOS Commands

All ProDOS and Basic commands will work with the ProDOS RAM disk. If you have added any disk commands to Basic following Apple’s guidelines, these new commands should work with the RAM disk.

Disconnecting The RAM Disk

The ProDOS Ram disk can be disconnected end the contents of the RAM disk can be destroyed only by turning of f the Apple or by running a program that uses more than the first auxiliary 64K bank of the MultiRam C Card (such as AppleWorks modified for MultiRam use).

Using the 8O—column or double hi—resolution features of the //c will not affect the RAM disk. These areas of memory are not included in the RAM devoted to the disk.

Pressing the Reset key will not affect the files stored on the RAM disk. To restore the RAM disk with all files intact if you accidentally press the Control+[Open—Apple]+Reset key combination, simply reboot a ProDOS disk using the same key combination and run the MULTIDRIVE.PRO program again.

Similarly, you can boot a disk using the ProDOS, DOS, or Pascal operating systems after you have used the ProDOS RAM disk, and you can then easily reconnect the ProDOS RAM disk with all its files intact using the following steps.

First, boot the other disk you would like to run using the Control+[Open—Apple]+Reset key combination. DO NOT turn off the Apple and then turn it on again to boot the desired disk aa all files on the RAM disk will be lost. Run the desired programs from the disk you booted.

Second, when you are ready to return to use the ProDOS RAM disk, remove the disk you have been working with from your disk drive. Insert a ProDOS disk, preferably one with the file MULTIDRIVE.RAM on it. Press the Control+[Open—Apple]+Reset key combination to exit the program you have been working with and boot the ProDOS disk. If you were using a ProDOS program, DO NOT use that program's "Quit" routine instead of the Control+[Open~Apple]+Reset key combination; using the ProDOS 'Quit" routine will destroy the RAM disk files.

Third, run the MULTIDRIVE.PRO program. This will restore the ProDOS RAM disk and all files on it will reappear as you last left them.

Programming Tips

The ProDOS RAM disk can be used like a normal disk from within Applesoft programs. All Applesoft disk related commands function correctly and no special steps are required to use "IRAN."

From within assembly language programs, you may directly manipulate files on the "IRAN" disk using a ProDOS MLI (Machine Language Interface) callable subroutine in the ProDOS Kernal or by passing commands using the Basic Interpreter's command handler. Avoid using direct calls to the device driver for block access as unpredictable results may sometimes occur.

ProDOS Updates

The ProDOS RAM disk program may be modified in the future. Allowing the ProDOS generated "/RAM" file to co—exist with the MultiRam C's RAM disk is now planned. Registered owners will be notified of updated software.

OTHER RAM DISK SOFTWARE

Apple Pascal RAM disk emulation software will soon be available at a nominal charge. Contact your dealer for more information if you are interested in a Pascal RAM disk emulator.

CHAPTER 6

HARDWARE & SOFTWARE COMPATIBILITY

MuItiRAM C’s operation can be enhanced by using other manufacturer’s hardware and software products along with it. Some of the more useful products are described in this chapter.

HARD WARE

All hardware products tested with the MultiRam C card have been compatible : modems, mouses, printer cards and printers. etc.

The two moat useful hardware additions that could be made to increase MultiRam C’s usefulness are a large capacity disk drive and a non–interruptable power supply.

Disk Drives

MultiRam C’s large on–board memory, especially if expended to a full 512K, is far larger than the standard Apple II 5 1/4 inch floppy disk drive’s unformatted capacity of approximately 144K.

To better take advantage of software that can use MultiRam C’s large memory, you may want to consider purchasing larger capacity disk drives. Using larger capacity drives, you may more easily save and load large data files as well as keep many more programs on one disk.
Large capacity floppy disk drives are currently available for the Apple //c and others will soon be available. Larger floppy disk drive capacities range from 160K to BOOK and come in 5 1/4 inch and 3 1/2 inch sizes.

Herd disk drives have greater storage apace available and faster access times to information stored on them than floppy disk drives. Hard disk drives for the Apple //c range from 10 megabytes to 20 megabytes.

Whether used in a single–user setting or networked for multiple users, a hard disk drive makes a very effective companion to the MultiRam C Card. As more ProDOS software like AppleWorks is made available for the Apple, a hard drive and MultiRam C will make an even stronger combination as ProDOS is capable of creating 16 megabyte files, more than enough to handle large files that can be manipulated with MultiRam C.

Currently, the Quark ~ 10 and QC 20 are the only hard drives that can be directly connected to the Apple //c. No tests have been run with the Quark drives and the MultiRam C Card to date. Please call Customer Service if you have a question regarding compatibility as the Quark drives and the MultiRam C Card will have been tested by the time this manual is published. No compatibility problems are expected.

Before buying a large floppy drive or hard disk drive, check to ace that all the operating systems you would like to use on your Apple (ProDOS, DOS 3.3, and Pascal) work on the drive. Further, if you use copy protected commercial software. be certain that the drive will run the software you now use or check with the software publishers to see if they have a version that will run on the drive in question.

Non — interruptable Power Supplies

In a business environment, a power failure that occurs while you are working with a large data file stored all in memory could be disastrous. At best, data must be re—entered costing lost time and aggravation. At worst, if a power failure occurs while information is being written to disk, a power surge or other disruption accompanying the power failure could cause damage to the file being saved so that the file is unusable.

A non-interruptable power supply offering quick response time to a loss of power as well as having sufficient power to operate the computer long enough to allow information stored in MultiRam C’s memory to be saved to hard or floppy drives would be ideal if you are using MultiRam C in a business setting.

Checkmate Technology plane to offer to MultiRam, C owners a non—interruptable power supply that meets these specifications.

SOFTWARE

Checkmate Technology plane to offer for sale modifications to popular existing software (like the AppleWorks memory expansion program) that can best take advantage of MultiRam C’s large memory.

Revised spreadsheet and data base software packages that will use MultiRam C’s large memory are also expected to be released by third party software publishers later this year.

Some programs currently on the market will use more than 64K of MultiRam C’s memory, though not all of the memory on a card may be recognized. FlashCalc, MagiCalc, and Magic Office are examples of programs which will recognize 64K banks 0, 3 sod 5 of MultiRam C memory.

Unprotected programs that return to a program or data disk frequently can benefit now from MultiRam, C RAM disk emulation software. Most popular, unprotected programs can run out of a RAM drive and use the same RAM drive to store their data, thus greatly speeding up these programs. These programs would boot—up almost instantaneously. When running them, their disk access time would be almost unnoticeable. RAM fast disk speed would also allow you to very quickly and easily switch between programs as needed.

Part 3

PROGRAMMING THE MULTIRAM C CARD

Programming the MultiRam C card explains how to use the //c’s standard auxiliary memory and how to manage multiple auxiliary bank addressing for the MultiRam C Card.

chapter 7

PROGRAMMING AUXILIARY MEMORY

MultiRam C's memory is addressed by The Apple as standard auxiliary memory. Therefore, when any 64K bank of the MultiRam cards memory has been switched into place as active auxiliary memory, standard auxiliary memory addressing techniques can be used to access MultiRam C memory.

This section briefly reviews standard auxiliary memory programing techniques for the Apple //c. For a more detailed description of auxiliary memory programming for the Apple //c, please refer to The Apple //c Reference Manual published by Apple Computer, Inc.

If you are familiar with programing auxiliary memory, you may wish to go directly to Chapter 9. To fully use all of the MultiRam card's memory as auxiliary memory requires switching from one bank of MultiRam 64K auxiliary RAM to another, by means of one soft switch. This is detailed in Chapter 9.

To access auxiliary memory from a program, to transfer program control or move data to auxiliary memory, or to employ double high resolution graphics, requires assembly or machine language programming skills. The following discussion assumes these skills.

GENERAL INFORMATION

The Apple //c uses a 65C02 microprocessor to control its operation. This microprocessor can address only 64K of memory at one time.

The //c contains two banks of memory: 64K of main memory and 64K of auxiliary memory. To use the 64K auxiliary memory, the //c uses a set of soft switches, contained in its Memory Management Unit, to control the address bus and switch between the 64K of main memory and the 64K of auxiliary memory. The soft switches are in turn controlled through software.

The 64K of auxiliary memory may be used for program or data storage, except for 1K which is used to control the 8O—column display. If you are going to use only 4O—column mode, you say use the full 64K.

The video display of the //c is controlled by routines located in specific areas of memory. In 4O—column mode. Text Page 1 and Text Page 2 control text and l0—res graphics display in the main memory (see Table 8.1 for locations). For 80—column mode, Text Page 1 of both main and auxiliary memory control the display.

Double High—Resolution Graphics may be turned on via soft switches. This doubles the horizontal resolution of Graphics Page 1. To turn on the double hi—rem graphics, write any value to soft switch 80COL ($C00D), TEXT Off ($C050), HIRES On ($C057). and MIXED Off ($C053).

BANK SWITCHING

Because the 65C02 can address only 64K of memory at one time, in order to utilize the I/c’s auxiliary memory, the computer must switch between the 64K RAM of the main memory and the 64K of auxiliary memory. This is called “bank switching.

Soft switches are locations which control which of the banks of memory is being addressed at a given time. Each soft switch has two control locations assigned to it as Table 8.1 shows. Write to the location that has the effect you desire with a STA, STX, STY instruction or a POKE command from Applesoft.

The soft switch called RANRD selects either the main or auxiliary memory for READ operations. The soft switch RAMWRT selects either main or auxiliary memory for WRITE operations. Read and Write may be set independent of one another so that s program may read data from a location in one bank and write it to another location in the other bank. To access main memory, both RANRD and RAMWRT must be Off; to access auxiliary memory, both must be On.

ALTZP selects locations in one bank or the other for both Read and Write operations in Page 0, the stack, and the language card.

The 8OSTORE switch controls the effects of some of the other switches, and it is particularly important to set it correctly. 8OSTORE must be OFT if RANRD and RAMWRT are to control the entire memory from $0200 to $8FFF. With 80STORE On. the PAGE1 switch can select either main or auxiliary memory. If HIRES is turned Off. PAGE2 will select Text Pagel from either main or auxiliary memory; if HIRES is On. PAGE2 selects Hi—rem Graphics Page 1 and Text Page 1 from main or auxiliary memory. RAMRD and RAMWRT do not affect the Text Page while 80STORE is On. Table 8.1 shows the relationships of these three switches.

Table 8.2
Functions of soft switches 8OSTORE, PAGE2, and HIRES

80STORE:	OFF	ON
PAGE2 OFF	Use RAMRD,RAMWRT	Access MAIN Memory (RAMRD RAMWRT no effect)
HIRES OFF	(no effect)	Text Page 1
HIRES ON	(no effect)	Hi—res Page 1
PAGE2 ON	Access Hi—Res Psge 2	Access AUX. Memory (RAMRD, RANWRT no effect)
HIRES OFF	(no effect)	Text Page 1
HIRES ON	(no effect)	Hi—res Page I

In Table 8.1, the first address given for each soft switch is the location at which the status of the switch can be read. When this location is read in machine language, a 0 (aero) in the high bit (7) indicates the switch is Off, a 1 shows the switch is On. If the location is read from BASIC, a value 0f 127 or less indicates the switch Is Off, 128 or greater indicates On.

Table 8.1
Soft switches

Switch Name	Location Hex	Location Decimal	Function
RAMRD	$C013	49171	Read RAMRD switch
	$C002	49154	OFF: Read Main 48K ($0200—$BFFF)
	$COO3	49155	ON: Read Aux. 48K
RAMWRT	$C014	49172	Read RAMWRT switch
	$CD04	49156	OFF: Write to Main 48K
	$C005	49157	ON: Write to Aux. 48K
ALTZP	$C016	49174	Read ALTZP switch
	$C008	49160	OFF: Use Main memory, Page 0. stack bank—switched (16K) memory
	$C009	49161	ON: Use Aux. memory. Page 0, stack bank—awitched memory
80STORE	$C018	49176	Read 80STORE switch
	$C000	49152	OFF: Enable RAMRD, RANWRT switches for video pages
	$C001	49153	ON Override RAMRD, RAMWRT switches for video pages (use PAGE2 switch)
80COL	$CD1F	49183	Read 8OCOL switch
	$COOC	49163	OFF: Display 40—colunma
	$COOD	49164	ON : Display 80—columns
TEXT	$CO1A	49178	Read TEXT switch
	$C050	49232	OFF: Display graphics
	$C051	49233	ON: Display text
MIXED	$CO1B	49179	Read MIXED switch
	$C052	49234	OFF: Full graphics
	$C053	49235	ON: Mixed text and graphics
PAGE2	$CO1C	49180	Read PAGE2 switch
	$C054	49236	OFF: 1. Display video page 1 using RAMRD.RAMWRT (80STORE off) 2. Display video page 1 using main memory (8OSTORE on)
	$C055	49237	ON : 1. Display video page 2 using RAMRD, RANWRT (80STORE off) 2. Display video page 1 using AUX. memory (80STORE on)
HIRES	$C01D	49181	Read HIRES switch
	$C056	49238	OFF: Low—res graphics selected
	$C057	49239	ON: Hi—res graphics selected

AUXMOVE AND XFER

No built–in assembly language subroutines may be used to transfer data and program control between main memory and auxiliary memory. These are AUXMOVE (starting location $C311) and XFER ($C314).

AUXMOVE is used to move blocks of data between main and auxiliary memory. Three pairs of bytes are used to pass addresses to AUXMOVE. Al is used for the starting address of the source block (AlL at $3C for low–order byte; A1H at $3D for high byte); A2 for source ending address (A2L at $3E for low byte; A2H at $3F for high byte): and A4 for the destination starting address (A4L at $42 for low byte; A4H at $43 for high byte).

Data addresses must be put into byte pairs in Page 0, and the carry bit set to select a move from Main to Auxiliary (SEC) or Auxiliary to Main memory (CLC). This must be done before calling the AUXMOVE routine. AUXMOVE cannot be used to copy data in the memory space switched by ALTZP (Page 0, stack, bank–switched memory).

When AUXMOVE is called by a JSR, the block of data specified by the A registers is copied in the direction specified by the carry bit. When complete, the x and y registers and accumulator are back as they were when the subroutine was called.

XFER is used to transfer program control between programs in main and auxiliary memory. To use the subroutine XFER to transfer control to and from segments in auxiliary memory, you must first set up three parameters: 1) the address of the routine being transferred to; 2) the direction–Main to Auxiliary memory, or vice versa; and 3) which Page 0 and stack to use.

The low byte of the program starting address is placed at $3ED, the high byte at $3EE (low byte first). The carry bit is set (SEC) to transfer from main to auxiliary, or cleared (CLC) to go from auxiliary to main memory. Set the overflow bit to use Page 0 and stack in auxiliary memory, or clear (CLV) the overflow bit to use Page 0 and stack in main memory.

Be sure to save the current stack pointer in current memory space before using XFER and restore it afterwards.

Control is passed to the XFER routine by a jump instruction rather than a subroutine call. The accumulator and transfer address are saved on the current stack, the soft switches set for the parameters selected, and the jump to the new program is executed.

Chapter 8

AUXILIARY MEMORY MANAGEMENT

The MultiRam C Card can contain up to 512K of memory. The Apple //c’s 65C02 processor can use this memory as eight banks of ordinary 64K auxiliary memory. How to use these different memory banks from within control by machine language routines is the subject of this chapter.

This chapter should be read by programmers who intend to write assembly language programs using the MultiRam C Card’s memory or assembly language subroutines that can be called by higher level languages.

SELECTING AUXILIARY MEMORY BANKS

Because the Apple’s 65C02 processor is an 8 bit processor, it can directly address only 64K of memory. The Apple addresses the 64K memory in a standard auxiliary bank by “hank switching” using software controlled switches. These soft switches cause different memory areas to be switched into the address range the Apple can read and write to and are discussed in Chapter 7.

Each of MultiRam C’s eight possible 641 memory banks can be moved into the address area reserved for normal auxiliary memory by writing the number of the desired 64K auxiliary bank —— $1 to $8 —— into a soft switch located at $C07X (X is any hexadecimal number $0 to $F). The Apple then can use this memory bank as it normally would use auxiliary memory. See “Softawitch Selection” in the following section for more information on use of the soft switch.

MultiRam C’s eight auxiliary banks are associated with fixed RAM Bank locations on the card. As Table 9.1 shows, Banks A and B have four 64K auxiliary memory banks associated with each of them. Four auxiliary banks can be associated with each RAM Bank as a set of 256K RAM chips can be placed into a RAM Bank and a set of 256K RAM chips is the equivalent of four 64K banks.

Table 9.1 Auxiliary Memory Bank Location (Hex notation)

MultiRam C RAM Bank	MultiRam C 64K Aux. Bank Number
A	1, 2. 3. 4
B	5, 6, 7, 8

A $0 written to $C07X selects the //c’s own auxiliary 64K memory. Values $9 to sET written to $C07X also select auxiliary hank $0.

Under program control, the last 64K bank selected will remain enabled as auxiliary memory until another bank is selected.

PROGRAMMING GUIDELINES

The following programming guides will will help you in writing your own software to effectively use the card’s memory. These guides are not comprehensive but should cover the most common questions you may have regarding card programming.

Softeswitch Selection

As previously explained, writing a value associated with a MultiRam C 64K auxiliary memory bank to any of the $C070 to $CO7F memory locations causes that 64K memory bank to be selected as the auxiliary bank. As Figure 8.1 shows, writing a SO to $C07X selects 64K bank 0, the //c’s own 64K auxiliary memory bank. Figure 8.2 shows bank 4 selected when a $4 is written to $CO7X.

Writing anything to the $C07X locations also causes the Apple a game paddle timers to be activated. If you are writing a program using a joystick, game paddle or other device attached to the game controller port, you should read the $CO7X address with a LDA, LDX, LDY instruction if you only intend to trigger the timer for the joystick. If you write to the $C07X register with a STA, STX, or STY instruction, you may switch to an undesired auxiliary bank. Also, remember that the paddle timer must time out before you can use the $C07x address again. Allow 3 to 4 milliseconds to elapse before you use $C07X again.

You should not write to $C070, $C077, $C07E, and $C07F as theme switches perform video related operations on the //c. Writing to any other location such as SCUll, $C073, or $0078 should cause no problems.

As previously mentioned, a $0 written to $C07X selects the //c’s own auxiliary 64K memory. Values $9 to $FF written to $C07X also select auxiliary bank $0 (not true with MultiRam //e and MultiRam RGB memory cards discussed later).

Under program control, the last 64K bank selected will remain enabled as auxiliary memory until another bank is selected.

Figure 8.1

MultiRam C Auxiliary Memory bank Selection

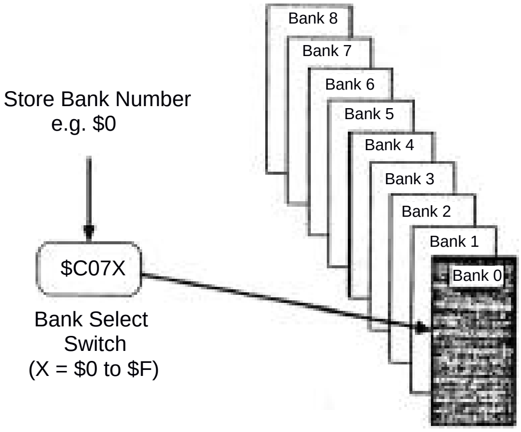
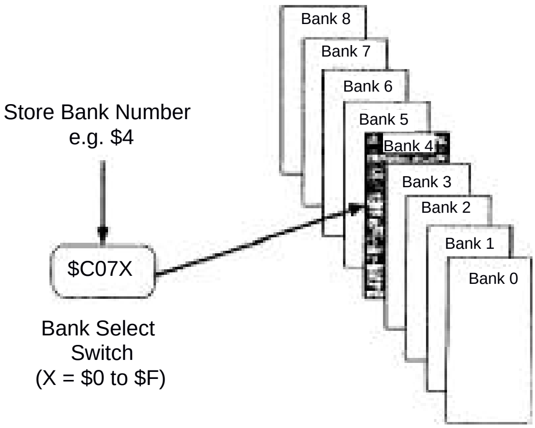


Figure 82

MultiRam C Auxiliary Memory Bank Selection



Initializing The Card

When the Apple is first booted from a cold start, auxiliary bank 0 (zero) is selected. This is the //c’s own built-in auxiliary 64K memory.

A program, however, should not assume bank 0 (zero) will he immediately available to it. A program may have been run previously that left another bank selected. Make it a practice to select bank 0 (zero) when first addressing the card and when exiting your program. This will ensure error free operation for your program and programs run after it.

Identifying MultiRam C Memory

After selecting auxiliary bank 0 (zero), your program should next determine how such memory is available on the card for use by your program.

Memory banks can be identified by writing a particular bit pattern into a specific location or locations in a memory bank. Next read the memory location that the pattern was written into and compare what is found there against the original pattern. A 64K auxiliary memory bank exists if there is a match.

Many different techniques can be used to perform the write/read/Compare operation. A preferred method is to use a descending loop that selects the highest memory bank available on the card, writes the expected auxiliary bank number and a checksum value into two bytes of that bank, decrements the bank counter and then repeats the process ending with bank 0 (zero). This would be followed by a descending loop that reads the bytes from the two bytes of each bank and compares them against the expected value.

A descending identification loop is preferable as it consistently identifies useable 64K banks from the highest possible bank number to the lowest.

Using a descending loop starting with $18 rather than $8 would ensure the software developed for MultiRam C would also work with MultiRam //e and MultiRam ROB memory cards. These cards can be combined in the //e to give op to 1.5 megabytes of memory. The ending 64K bank of their 1.5 megabyte combination is defined as siB. In the I/c. selecting non—existent banks from $1B to $9 and writing the bank number associated with that bank into memory causes no problems; bank 0 is selected for all values above $8 and the last value written to bank $0 in a descending loop ending with $0 would be $0.

An example of a routine to identify useable memory is as follows:

```
        LDX BANK#          Bank number to start. $1B for //e compatibility
MARKALL  STX BNKSELCT       Select bank with the $C07X bank switch
        STX ALTZP          Write to $C009 for aux. zero page
        STX TEST1          Write the bank number to first location
                             in zero page of selected bank
        TXA                Prepare for EOR
        E0R #FF            Reverse the bank’s bits setting
        STA TEST2          Write the checksum byte to second location
                           in zero page of selected bank
        DEX                Decrease the bank counter
        BPL MARKALL        Loop through to bank zero

        LDX BANK1          Bank number to start, same as MARKALL start
READALL  STX BNXSELCT       Select bank with the $C07X bank switch
        CPX TEST1          Compare bank number against memory
        BNE MAPFAIL        Memory map marked showing bank n with no memory
        TXA                Prepare for EOR
        EOR #FF            Prepare the checksum for the double check
        CMP TEST2          Compare the EOR’d byte for the double check
        BNE MAPFAIL        Memory map marked showing bank n with no memory
        BEQ MAPPASS        Memory bank exists, mark bank n into memory
READNEXT                   MAPFAIL, MAPPASS return here
        DEX                Decrease the bank count
        BPL READALL        Loop through to bank 0
        STX MAINZP         Select main memory zero page
        RTS
```

Memory Bank Mapping

You should keep a record 0f available memory for your program’s use after you have identified the memory on the card. Either a byte map or a bit map could be used for the memory map. You may wish to construct a map that not only shows which 64K banks are useable, but like DOS and ProDOS sector bit maps, also keeps track of the availability of smaller increments of memory. The auxiliary bank memory map could then become part of a larger memory manager.

You will also need to use a byte or bytes to track the memory bank in current use. The $CO7X strobe cannot be read to see which bank was last called.

Memory Transfers

Data can be transferred from main memory to auxiliary memory and back using standard //c auxiliary memory transfer commands. The RAMRD and RANWRT soft switches and the AUXMOVE routine described in Chapter 7 all function normally. The only difference is the additional need to determine which of the MultiRam C’s 64K banks should be used for the transfer.

Data can also be transfered from one MultiRam C 64K auxiliary bank to another 64K auxiliary bank. This could be done by selecting the source bank, copying the source bank’s data into main memory using AUXMOVE or your own memory move routine, selecting the 64K destination bank, and then writing the data to the destination bank from main memory.

Another method is to transfer data directly from one bank to another, one byte at a time, through the accumulator. The memory move routine doing the transfer must always be located in the part of memory not being transfered. The safest method of doing this type of transfer is to locate the move routine on the motherboard main or bankswitched memory areas.

80—Column Addressing

The 80—column display is derived from the //c’s 64K auxiliary bank 0 only. When another bank is active, the 80—column display will still come from bank 0; the RAM in bank 0 is automatically refreshed during the //c’s video refresh cycle. Therefore, you do not have to keep a mirror image of the bank 0 text page area in other banks in order to maintain the 80—column display when you switch banks.

To read and write to the 80-column screen, auxiliary bank 0 (zero) should be selected. If you write to the text video page area with some other bank selected while an 80-column display is active, information displayed will become garbled. You may only safely write to the text page area of an auxiliary bank ——other than bank 0 — if an 80—column display is not active.

To avoid potential problems, you may want to exclude the 80—column text page area ($400—$7FF) from use on all banks.

Similar warnings apply to double hi—res displays.

Reset Key Trapping

Pressing the Reset key will not automatically switch in auxiliary bank 0 (zero). If you would like to direct your program to a specific auxiliary bank location when the Reset key is pressed, you must change the reset vectors $3F2 and $3F3 to point to a routine that will write the desired bank number into $CO7X.

Interrupt Handling

Interrupts are handled in the Apple //c by an interrupt vector located at $FFFE. The $FFFE location is normally RUM area.

If an auxiliary bank’s bank—switched memory is active when the interrupt occurs, that bank most have a vector at the $FFFE location in its memory pointing to an interrupt handler routine on motherboard main memory. The interrupt handler routine should be able to return to the bank that was active when the interrupt occurred if a “bank active” temporary byte is being maintained by your program in maim motherboard memory ($200—$BFFF) or a defined location in another area.

An alternative method to handling interrupts is to disable interrupts with a SEI instruction whenever auxiliary bank–switched memory is in use. When switching out of auxiliary bank–switched memory, interrupt requests could then be enabled with a CLI command.

Part 4

MULTIRAM C SERVICE GUIDE

The MultiRam Service Guide describes the MultiRam C warranty and explains Checkmate Technology’s service policies.

Chapter 9

THE WARRANTY

The MultiRam C Card, like all of Checkmate Technology's peripheral cards, carries a 5 YEAR LIMITED WARRANTY.

CHECKMATE TECHNOLOGY, INC. warrants products against defects in material and workmanship for a period of 5 years (90 days for software) after purchase. During the warranty period, CHECKMATE TECHNOLOGY, INC. will at its option, repair or replace at no charge, or refund the purchase price to the purchaser, products that prove to be defective provided that the CHECKMATE TECHNOLOGY product is returned, shipping prepaid, to CHECKMATE TECHNOLOGY. INC. For products returned by other than U.P.S., the sender assumes greater risk of loss and delays.

Your sales receipt is your warranty validation. The receipt must be provided when requesting warranty work to be performed unless a valid product registration card is on file at the company headquarters. This warranty does not apply if, in the sole opinion of CHECKMATE TECHNOLOGY, INC., the product has been damaged due to abuse, misuse, misapplication, accident or as a result of service or modification by other than an authorized CHECKMATE TECHNOLOGY, INC. service center.

THIS WARRANTY IS EXPRESSLY IN LIEU OF ANY AND ALL WARRANTIES EXPRESSED OR IMPLIED. CHECKMATE TECHNOLOGY DOES N(Yt WARRANT THAT GOODS ARE MERCHANTABLE OR FIT FOR ANY PARTICULAR PURPOSE. CHECKMATE TECHNOLOGY SHALL NOT HE HELD RESPONSIBLE FOR ANY CONSEQUENTIAL DAMAGES OR LOSSES ARISING FROM THE USE OF THIS PRODUCT.

This warranty applies to CHECKMATE TECHNOLOGY products purchased in the United States. The warranty may vary for products purchased outside the continental United States. Contact CHECKMATE TECHNOLOGY, INC. for warranty service information.

SECOND HAND OWNERSHIP

Purchasers of used Checkmate Technology products should register the transfer with Checkmate Technology. The cost of registration is $10.00 and is a wise investment: any remaining unexpired warranty period will be transferred into your name and you will receive newsletters providing product information. including news of product updates.

To transfer the unexpired warranty period, the original owner must have filled out and returned the Registration/Warranty card to Checkmate Technology at the time of purchase.

If the original owner failed to register the card, for a one—time fee of $30.00 you say transfer the unexpired warranty period (based on either the original purchase receipt showing date of purchase or by date of manufacture indicated by serial number on the card). Checkmate Technology reserves the right to decline any warranty transfer.

In order to transfer registration you must provide Checkmate Technology with:

1. Product name
2. Serial number on the card
3. Name of previouS owner (sellers name)
4. Your name, address, and telephone number
5. Date of sale
6. Your check for $10 ($30 if the card was not registered)

EXTENDING THE WARRANTY

If you would like to extend the warranty on your newly purchased or used card, Checkmate Technology offers an EXTENDED WARRANTY PROGRAM.

New owners can take advantage of an additional five year warranty by sending a check for $25.00 together with the Product Registration/Warranty Card. This offer is limited to new owners and must be sent to the company within thirty days of purchase.

Second hand owners can extend the unexpired warranty to a maximum of five years. The cost of extending the warranty is $15.00 per year in addition to the transfer registration fee. For example, if there is only one year left on the unexpired warranty, a check for $70.00 would be sent: $60.00 for 4 years plus a $10 registration fee ($30 if the original owner did not register the card). Checkmate Technology reserves the right to limit the duration it accepts for extended warranties.

chapter 10

SERVICE POLICIES

This chapter explains Checkmate Technology’s repair and product update policies and other pertinent customer information.

HOW TO GET HELP

If you are unable to get your MultiRam C Card to work properly after following the procedures in Chapter 3, please contact the Customer Service Department. We will try to solve your problem as quickly as possible.

Telephone Help

Before sending in your MultiRam C Card, you may want to call a Customer Service representative. Some problems may be solved without returning your card.

The telephone number and hours for technical assistance are:

Customer Service: (602) 966—5802
Hours: Mon. to Fri., 9am to 5pm (Mountain Standard Time)

If you need to cell us, please hove the following information available to save time:

1. Approximate date you mailed in your registration card.

2. Serial number of the Apple //c used (found on the bottom of the //c case).

3. Peripherals in use when the problem occurred.

4. Nature of the problem with the card.

If your card needs to be returned to the factory for repair, you will be issued a Return Merchandise Authorization number during your call. Write this number on the outside of the package that you will use to ship your card.

Telephone Policy

We want to be as helpful as possible with your questions and problems. We understand that the problem you are having is important, and we will provide technical assistance to try to answer it. We ask that you use our Customer Service telephone number for assistance. Support vs not available through the toll—free order desk telephone number; collect calls are not accepted by Customer Service.

HOW TO RETURN A CARD

If it is necessary to return your MultiRam C Card, follow these steps:

I. Please wrap your card in anti–static material. If none is available, wrap the card in aluminum foil. Returning a card without wrapping it in protective anti–static material or aluminum roil say cause further damage to the card and will VOID YOUR WARRANTY. Insert the pins on the bottom of the card into the anti–static foam shipped with the card to protect the pins from damage.

2. Include a note with your name, address, home telephone number, a daytime telephone number, plus:

   A. Detailed description of the problem.

   B. Serial number of the Apple /Ic used (found on the bottom of the case).

   C. Peripherals in use when the problem occurred.

3. If you are a registered owner, meaning that you completed and returned your warranty registration card to Checkmate Technology, you are not required to send in proof of purchase. If you have not registered your MultiRam C Card, include proof of purchase.

   NOTE: ANY CARD RETURNED WITHOUT PROOF OF PURCHASE, OR WHICH IS NOT REGISTERED, WILL BE TREATED AS A PRODUCT OUT OF WARRANTY AND WILL BE CHARGED FOR NON–WARRANTY REPAIR AT RATES THEN IN EFFECT.

4. Pack the wrapped card, note, and proof of purchase (if needed) in a sturdy box and cushion the card with non–static packing material much as crumpled newspapers.

5. Ship the card, postage pre–paid. to:

   Checkmate Technology, Inc.
   511 South Rockford Drive
   Tampa, Az. 85281–3021

   Attn: Repair Dept.

Checkmate Technology, Inc. assumes no responsibility for cards either damaged or lost in shipping to Checkmate Technology for service. Please insure your valuable card. The coat of shipping insurance is low compared to replacing a lost card. We suggest U.P.S. as they insure the first $100.00 at no charge and require the addressee to sign for each package.

RETURNING YOUR REPAIRED CARD

After repair or replacement, your MultiRam C Card will be shipped to you in the U.S.A. by U.P.S., regular delivery, postage paid. Returns outside the U.S.A. may require a handling and shipping charge. Contact us regarding repairs outside the U.S.A.

RAM INSTALLATION & REPLACEMENT POLICY

You may return your card to Checkmate Technology if you would prefer to have the factory install added RAM on your card. Installed RAM will be teated at the factory on your card before return to you.

The cost of factory installed RAM is $65.00 for a set of eight 64K DRAM and $120.00 for a set of eight 256K DRAM. Checkmate Technology will return your card postage prepaid. These prices are subject to change. Contact the Customer Service Department for current pricing.

Cbeckmate Technology warranties only that RAM on the MultiRam C Card that was installed at the factory or RAM from a Checkmate Technology RAM Kit that was installed later. Such RAM is replaced at no charge if a defect in the RAM occurs during the 5 year warranty period.

Cards returned to the factory with defective RAM that was not supplied by Checkmate Technology will be charged a $25.00 service fee in addition to an $8.00 charge for replacement of each defective 64K RAM chip and a $15.00 charge for replacement of each defective 256K RAM chip. The repaired board is returned to you C.0.D. you pay repair charges upon receipt of your board. Return shipping charges are included in the service fee. Face are subject to change. Contact the Customer Service Department fur current pricing.

REPLACING DAMAGED DISKETTES

All Checkmate Technology disks are warrantied for 90 days and will be replaced free of charge if a defect is discovered within the warranty period.

If you damage or wear—out a CHECKMATE TECHNOLOGY, INC. program disk after the 90 day warranty period and are a REGISTERED OWNER, we will replace your old program diskette for $15.00, return postage prepaid. Non—registered software owners will be required to pay $25.00. The additional ten dollars covers the costs of setting up a new owner registration.

Return the damaged or worn—out program diskette to the Customer Service Department, along with a note detailing the problem, and a replacement disk will be sent to you.

SOFTWARE UPDATES

Checkmate Technology may from time to time upgrade the standard features of the MultiRam disk and other related MultiRam software. Registered owners will be notified of updates through our newsletter or special mailings.

Send $5.00 to cover disk, shipping and handling costs, and an updated disk will be sent to you. You will not need to return the older software version.

NEWSLETTERS

Checkmate Technology will publish informative newsletters for registered owners of its products. "Check—It—Out” Newsletters will contain Information on updates to products, information on modifications to software, general information on product usage, and news of product developments at Checkmate Technology, Inc.

If you would like to share any discoveries, programming tips, comments, or suggestions with other owners, please write to us and we will include them in the newsletter.

INDEPENDENT SOFTWARE DEVELOPERS

We actively solicit your help in making as many programs MultiRam C compatible as possible. Owners submitting the most useful program modifications will receive, at Checkmate Technology’s discretion, products or cash rewards for their efforts.

In the newsletter we will also list the names of users of the MultiRam C Card that have joined our “staff” by submitting original software that we can offer for sale. Anyone writing an original software package featuring the MultiRam C Card is invited to call to arrange a product review. Please do not send your original software without first contacting us. Cell the Customer Service Department for further details.

SUGGESTIONS

If you have suggestions, comments, or questions regarding Checkmate’s products please contact us. Address correspondence to:

CHECKMATE TECHNOLOGY, INC.
509 South Rockford Drive
Tempe, Arianna 85281—3021

Attn: Andrew P. Niemic President

Our goal is a satisfied customer!