

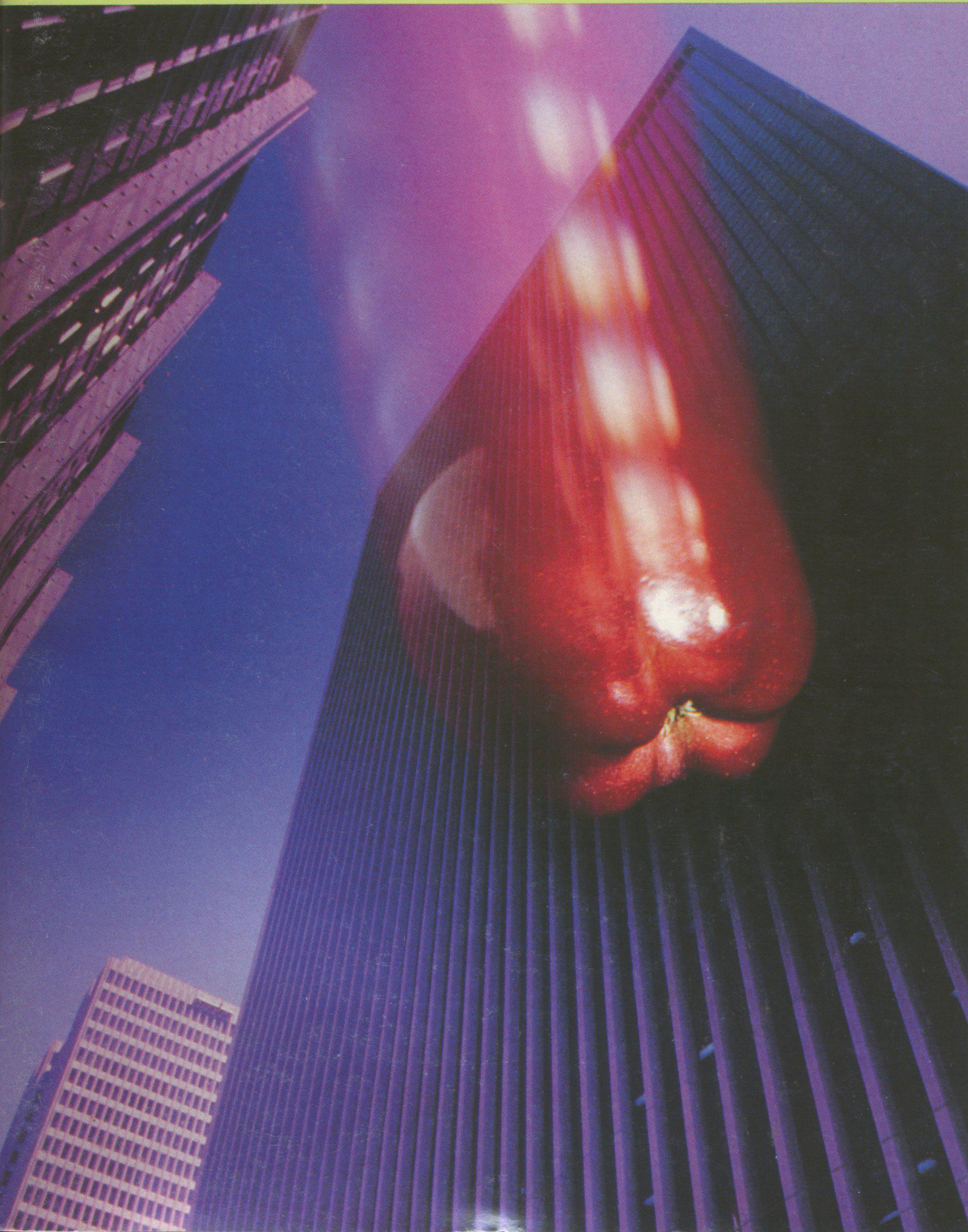
Apple Orchard™

\$3.25 U.S.
OCTOBER 1983

The Premier Magazine for Apple Computer Users

THE SPEEDIEST APPLE

Apple III Drivers
Reading Text Files
WPL



**NEW SOFTCARD IIe
AVAILABLE FOR THE APPLE IIe**



Microsoft SoftCard systems introduce your Apple to thousands of new programs.

More powerful Apples. When you plug a SoftCard™ system into the Apple® II, II Plus, or IIe, you're adding the ability to run thousands of CP/M®-80-based programs. Word processing, data-base management, analysis and forecasting programs—SoftCard gives your Apple access to thousands of software tools for use in your business or home.

Premium capabilities. Apple II or II Plus owners who want even more can add the Premium System. In addition to CP/M-80, it provides an 80-column upper-and-lower case video card and a 16K RAMCard™. Apple IIe owners can have CP/M-80, 80-column video, and 64K of RAM on a single card—the Premium SoftCard IIe.

Two computers in one. Any of these systems turn your Apple into two computers. One that runs Apple software and another that runs CP/M-80. Which means you'll double the utility of your computer.

Complete solutions. The SoftCard, Premium System, and Premium SoftCard IIe have everything you need. Easy-to-install circuit boards. The CP/M-80 operating system. Microsoft® BASIC. And the utilities you need to manage CP/M-80 files.

Why Microsoft? Microsoft was the first microcomputer

software manufacturer. The very first. Today, Microsoft software is running on several million computers, worldwide.

With a reputation for dependability and consistent product enhancement.

Ask your dealer. Ask about the superior application programs the SoftCard and Premium System make available to your Apple—high quality programs for almost every area of home, business and professional use. Then ask for a demonstration of the complete SoftCard, Premium System, or SoftCard IIe. And introduce your system to some of those thousands of new programs.

BETTER TOOLS FOR MICROCOMPUTERS

MICROSOFT™

MICROSOFT CORPORATION
10700 NORTHUP WAY
BELLEVUE, WASHINGTON 98004

Microsoft and the Microsoft logo are registered trademarks, and SoftCard and RAMCard are trademarks of Microsoft Corporation.

A Look at the Future . . .

Apple's Independent Software

Developer's Conference

by Don Norris

Twice a year, Apple Computer puts on a very extensive and thorough series of workshops and seminars for independent software developers. The quality of the information presented by Apple at these four-day workshops makes it very obvious that Apple is committed to supporting and working with the independent software developer. The goal is a broader range of better products for all of us.

The most recent of these conferences took place between August 8 and August 11. System Failure \$06 is in the Apple /// a stack overflow, which is exactly how I felt after attending the sessions. How do you handle a 2 to 3 inch high (paper) stack of information from a four day conference; a stack which could have been even higher if I had picked up *everything* available and been able to attend every workshop.

John Sculley, Apple's new Chief Executive Officer, sounded the general theme. His comments in general: Apple's market positioning is to be the best, not the biggest. Apple would be marketing on a performance basis and not on price. The latter strategy has been pursued by a couple of other computer companies with some disastrous financial results.

Technical

The conference began Monday afternoon with informal sessions on various topics. Of particular interest to me was the session on integrated software, which has become one of the microcomputer industry's buzz words in recent months. (Various manufacturers are claiming that their approximation of Lisa's integrated approach is "imminent". Sure it is.)

This brought up the question of just what is meant by integrated software. Is it having several functions built into one piece of software, such as a spread sheet, database, and text editing all in one package or the ability to move data from within one program to a different program? The developers present indicated that there was no uniform standard in the industry for data transfer. (*Nor for much of anything else.—PCW*)

The only thing close to a standard means of file transfer and format is an ASCII text file. Everyone generally agreed that while it was not perfect it represented a starting point.

New tools are being developed to make software authors' jobs easier. An example is a new language called Classcal, which has evolved as a result of the Lisa technology. It is a marriage of Simula and/or Smalltalk (plus some other language) and Pascal, and is designed to support object-oriented programming.

Improvements have also been made to existing languages, particularly Pascal. Because these languages are loaded into memory rather than being resident in ROM as they are in some machines, they are easier to upgrade and debug. This gives greater latitude to the programmer; you do not have to make a hardware change when a bug has been cured or an improvement (enhancement) is made to the language.

On the question of Rodents on the //e: Evidently rodent (mouse and beyond) technology is being developed for the Apple II-/// product family. This led to an interesting situation later in the morning when several developers started asking

specific questions about the device of Bruce Tognazzini, who was trying valiantly to maintain data security about the new concept. Bruce was unaware that El Presidente had let the mouse out of the bag, so to speak.

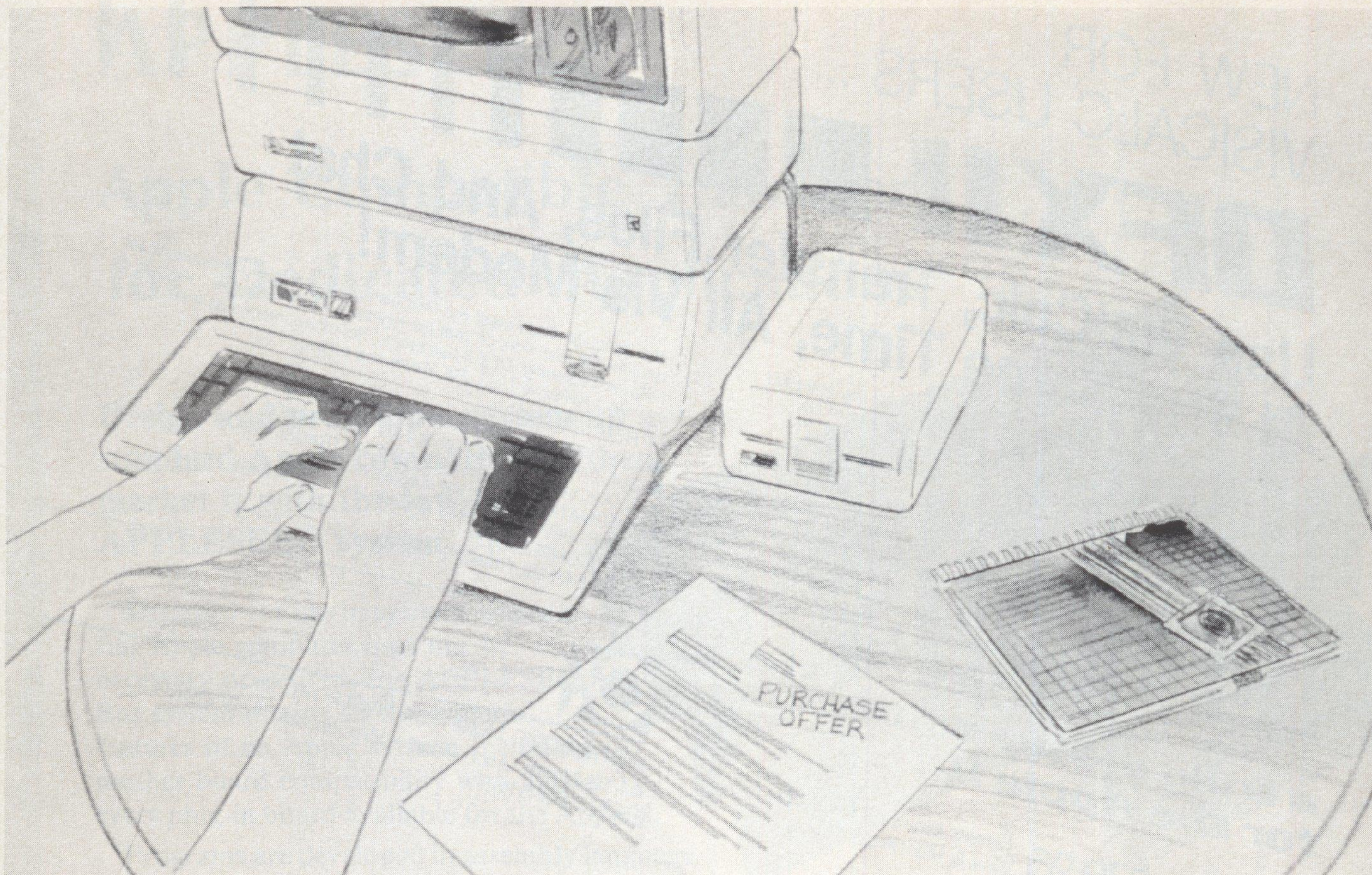
Marketing

The most startling news was in the area of market penetration: only 7% of the potential market for personal computers has been penetrated.

As to whether the Apple //e or /// had been eclipsed by other machines, the answer from John Sculley was that we have not seen the last model of the Apple // or ///. There will be continuing product improvements, not phaseouts, in the product family which includes the // and the ///. You will see more and faster products in this family in the future.

Particular attention was paid to the Apple ///, in light of industry rumors about the machine's future. A special team has been put together to address the marketing needs of the ///. Apple Vice President Ken Zerbe said the /// had been suffering because it was always under the direction of people with other commitments or people yet to be hired (called "TBH" in Apple land), or was assigned to a department which also had several other responsibilities. Consequently, the /// got lost in the shuffle. Many Apple /// owners had expressed the feeling that they were orphans, and now we know why. We *were* orphans. That should change.

Future product enhancements will greatly enhance the portability of data and software within the //e and /// product family.



Demonstrations

Wednesday afternoon, various software and hardware vendors and developers demonstrated new products they were going to unveil or already had on the market. Most of these can't be revealed here, but of note to Apple /// owners were high density 5-1/4 inch floppy disk drives from Rana with 652K Bytes and Micro-Sci with 572K Bytes. Both of these drives can be daisy chained onto the ///.

Tim Gill of Quark, Inc. demonstrated Version 2.0 of Catalyst. This new version is much easier to install and to use. It features an inverse cursor over a menu item. The user can then scroll to the program he wants to use, and merely press «RETURN» to run the desired program. Previously to use Catalyst you had to add every driver you might need for any program onto the Catalyst Boot Diskette SOS.Driver file. This meant that when using VisiCalc, the RS-232 driver and any graphics drivers would have to be on the boot diskette and loaded into memory even though they would not be needed by the program. Another addition to the Catalyst Menu is the ability to exit to Lobotomy (Apple II Emulation)

Mode and turn your 256 K Apple /// into 48K Apple II without rebooting.

One question asked of a software developer was, "Why aren't you developing software for the 'other horse' (presumably Baby Blue) as well?". The reply was that Blue provided no developer support, saying only, "Here is the gray box, you write your software for it."

Another developer criticized Apple's low disk storage capacity, saying that he was working with IBM because, "I will not put on the market software for an Apple which requires multiple disks when I can market for the other machines on one diskette, thus making it much easier for the ultimate user."

Regarding the ultimate end user and disk storage, this was stated to be an item of great priority within Apple now since Unifile and Duofile have been laid to rest. They required too much power, and were too expensive to manufacture. All types of mass storage media are being investigated; Apple does not appear to be sleeping.

Lisa

Thursday was Lisa day. Lisa is the first in a family of personal computers to be

released by Apple based on the 68000 CPU. Developers are being encouraged to write software for Lisa and to take advantage of its powerful integrated software and ease of use for the end user. In this way the developer would have a head start on working with the new machines as they are introduced.

A prime example of the kind of software that has yet to be developed for Lisa is a spelling checker. Most of the material that was handed out at the conference was prepared using Lisa. It contained several spelling errors. It was one more reminder that the imperfect human remains in control, but the tools continue to improve.

An overriding theme of the conference was for developers to "leverage" their efforts with Apple by taking advantage of the research and development Apple has done regarding improved user interface and enhancements to programming languages. Ease of use is an important part of Apple's marketing thrust and was stressed repeatedly during several of the workshops at the conference. The Apple folks stressed their commitment to independent developers as being necessary to Apple's success.



Only one copy system lets you backup 64k programs in 20 seconds flat...



...the Snapshot Copykit™ for Apple IIe and II+

While other manufacturers have been making wild promises, we've been quietly developing the superior copy system you've been waiting for: the **SNAPSHOT COPYKIT**.

THE FLEXIBLE CARD – The **SNAPSHOT COPYKIT** is a board for all systems. It sits in any slot on the Apple IIe, II+ and II as well as software-compatible machines like the Basis 108 and the Franklin Ace. So, you can upgrade your system without paying for another card.

Although **NO LANGUAGE CARD/16K CARD IS REQUIRED**, the **SNAPSHOT COPYKIT** can be used with any memory card and all peripheral cards.

The **SNAPSHOT COPYKIT** is the only copy card that stays up-to-date. Because all its software is in RAM, rather than inflexible ROM, the new enhancements we continually develop are made available to you at nominal cost — immediately.

THE USER-FRIENDLY CARD – We refined backup techniques for two years to make the **SNAPSHOT COPYKIT** just right for you, whether you're new to the Apple or a seasoned pro.

Other copy cards force you to use complex, unreliable procedures and repeated booting of 'utility disks' to copy programs larger than 48K. Outdated 'nibble copier' programs involve frustrating trial-and-error parameter changes. Only the **SNAPSHOT COPYKIT** backs up all 64K memory-resident programs (and most larger programs too) with one simple menu and one press of the trigger.

THE POWERFUL CARD – The **SNAPSHOT COPYKIT** gives you back the power over your computer that copy-protected software took away. You can interrupt a running program, copy it, list it (even if it's 'unlistable'), disassemble it, step and trace it, modify it and resume running it. The program is not disturbed

in any way and never knows it was interrupted. You can view its graphics, print them (if you have a graphics card) and much more. You can even temporarily suspend one program while you use another.

All backups are automatically BRUNnable files that you can move to hard disk. Our state-of-the-art compression facility lets you put a number of backups on one floppy.

With powerful features like these, it's no wonder that **BYTE** magazine called **SNAPSHOT** "The Ultimate Unlock System".

The **SNAPSHOT COPYKIT** has many more features for programmers, gamers and others. Write for our info packet or leave your address and questions with our 24-hour answering service at (413) 584-7600. We regret we cannot return all calls. All **SNAPSHOT** owners receive our 'Technical Hot-Line' number for instant technical guidance.

PRICE: \$139.95

(Snapshot Two is still available for the Apple II+ and Franklin Ace at \$119.95. 16K Card required; please specify brand when ordering).

TO ORDER: Mail check payable to Dark Star Systems. Add \$3 for shipping in the U.S. (\$7.50 to Canada, \$15 elsewhere). Mass. residents add 5% tax. COD costs \$5 extra (not available outside U.S.).

darkStar
SYSTEMS

P.O. BOX 140, Amherst, MA 01004

Apple is a reg. trademark of Apple Computer.

The Speediest Apple

Review by Neil D. Lipson, P.E.

This article will be the first part of a series on how to speed up your Apple II Plus or Franklin Ace 1000. I did not try these techniques on the //e, but most of them should work as well. The following items will be covered:

1. Ramex 128K card by Omega Microware
2. Microseeds Fastload
3. Diversi-Dos
4. Synetix SSD Disk Emulator
5. David-Dos
6. Saturn Accelerator II
7. Routine Machine by Southwestern Data Systems
8. Arithmetic Processor by California Computer Systems

All of these products have a speed-improving impact on your Apple; one or more of the processes will execute more rapidly than on a standard machine. What I wondered was, what would happen if more than one of these devices were used at the same time? Let me tell you that the results are spectacular! First, let's look at some general results; then we'll go into many techniques and describe in detail exactly what I did, and how you can accomplish the same results.

High Speed DOS

In my previous article, *Speeding Up DOS* (Apple Orchard, November-December, 1982), we covered some methods for fast disk access. Here's how to go one step further. I used the Saturn Accelerator II with the normal preboot supplied with the program. I then inserted the Synetix SSD disk and ran the disk emulation program. Some Hi-Res pictures were then loaded into the emulator. Once this was done, I used either Diversi-Dos or Fastload from Microseeds in the patch mode to "patch" the DOS to make it faster. For the most part, DOS then ran instantly. I could load Hi-Res pictures that normally took 10 seconds per picture in about 1/10 of a second, or 10 frames a second. This is twice as fast as without the Accelerator II.

The same thing, using the Ramex 128 card instead of the Synetix SSD, gives about the same results, with the proper patch to the operating system. The Ramex 128 can, of course, also be used for high speed VisiCalc as well as generally extending memory. I used a custom patch, but understand that many of the fast DOS people are developing their own patches. In addition, the Saturn software worked well on the Ramex card. The card is quite versatile.

High Speed Applesoft and 6502 Operation

I then tried the following combination: the Accelerator II, the CCS Arithmetic Processor and some programming tricks. Applesoft ran up to 250 times faster. The Accelerator II made everything in the 6502 mode run faster, including games. Did you ever try to play Choplifter at a three-times-faster speed?

One of the slowest things to do on a computer is sorting. Even with the high speed sort programs, it still takes a while. Well, those days are over. I tried the Routine Machine sorting routines with the above hardware. The sorting routines are only a small part of Routine Machine, which we'll discuss in more detail later.

With the Routine Machine, and the Accelerator, sorts that normally took 5 minutes with my old Applesoft routines were done in about 6 seconds. Routine Machine has small machine language programs that append onto the main Applesoft program, and you use them accordingly. The package is amazing, and this is just one of many of their packages which we'll cover in future issues.

One aspect of Applesoft and DOS that drives everyone crazy is slow loading and saving of text files. There are numerous ways to speed them up. If you have a random access file, use Diversi-DOS. In quoting from their manual, Diversi-DOS contains an optional faster way to specify the record and byte parameters when reading or writing a random access text file from Applesoft BASIC. CALL 1005 sets these parameters to the values in the Applesoft variables name "RE" and "BY" respectively. The following example will illustrate how to use this feature:

Standard DOS way:

```
10 PRINT CHR$(4);"OPEN RANFILE, L500"  
20 BY = 0  
30 FOR RE = 0 TO 100  
40 PRINT CHR$(4);"WRITE RANFILE,R";RE;" ,B";BY  
50 PRINT "TEST DATA"  
60 NEXT  
70 PRINT CHR$(4);"CLOSE RANFILE"
```

Diversi-DOS way:

```
10 PRINT CHR$(4);"OPEN RANFILE, L500"
```

```

20 BY = 0
30 PRINT CHR$(4);"WRITE RANFILE"
40 FOR RE = 0 TO 100
50 CALL 1005
60 PRINT "TEST DATA"
70 NEXT
80 PRINT CHR$(4);"CLOSE RANFILE"

```

The CALL 1005 is over 60 msec faster than the standard DOS PRINT command, which amounts to about a 2 to 4 times speed decrease.

The counterpart to Diversi-DOS is David-DOS. David-DOS, though, speeds up sequential text file loading instead of random access text files. Therefore, you choose the DOS that fits your needs. To accomplish this, use the TLOAD command. TLOAD speed loads a random or sequential text file starting at \$1000 or any address you choose with the "A" parameter. If you want less than the whole file, use the "L" parameter, otherwise, TLOAD always loads the whole track & sector list allotted to the file. This is necessary because DOS never records the last byte of a text file. We can't look for the first "0" as the end mark because we also load random files and they don't use an "0". Therefore, TLOAD without "L" loads the entire last sector even if the file ends in the middle of the sector. You can, if desired, make TLOADS to exact length with "L". Understand? Well, if you don't, examples are given in the documentation with the program.

Now let's go through each item and see what it does, and consider the advantages and disadvantages.

1. The RAMEX 128 card by Omega Microware

This card gives tremendous flexibility to do disk emulation, expand RAM and even act as a large print buffer using Diversi-DOS. It also has a high speed VisiCalc load (20 seconds), which is a pleasure, instead of having to wait up to 10 minutes. It's hardware compatible with the Saturn card, so the Saturn software will work properly with it. The only incompatibility is with the Rana controller card and Super Expander Plus. This appears to be a problem with the Rana controller, which will hopefully be corrected in the future. In addition, you must understand that the card only holds 128K, which is not quite a full disk. This does not include DOS, but the real problem is if the DOS is deleted and the space is used for something else, it just won't fit. This is a minor problem, however.

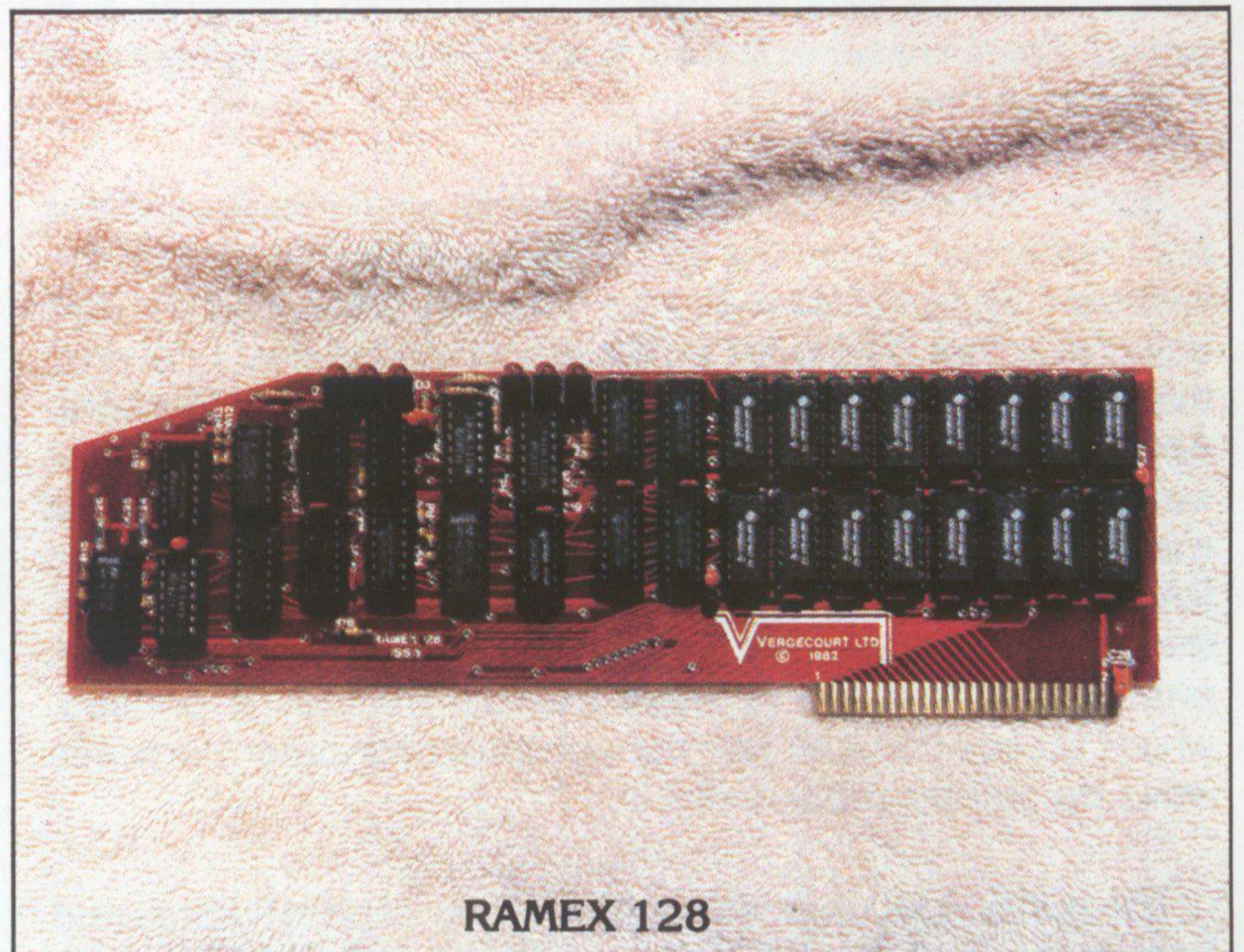
2. Microseed Fastload

This is a good general program that allows fast loading of binary and Applesoft programs. It also has a patch program to "patch" DOS when using it with the Synetix Disk emulator. It is compatible with almost every piece of software and is considered excellent for general use. It will not, however, speed up text files.

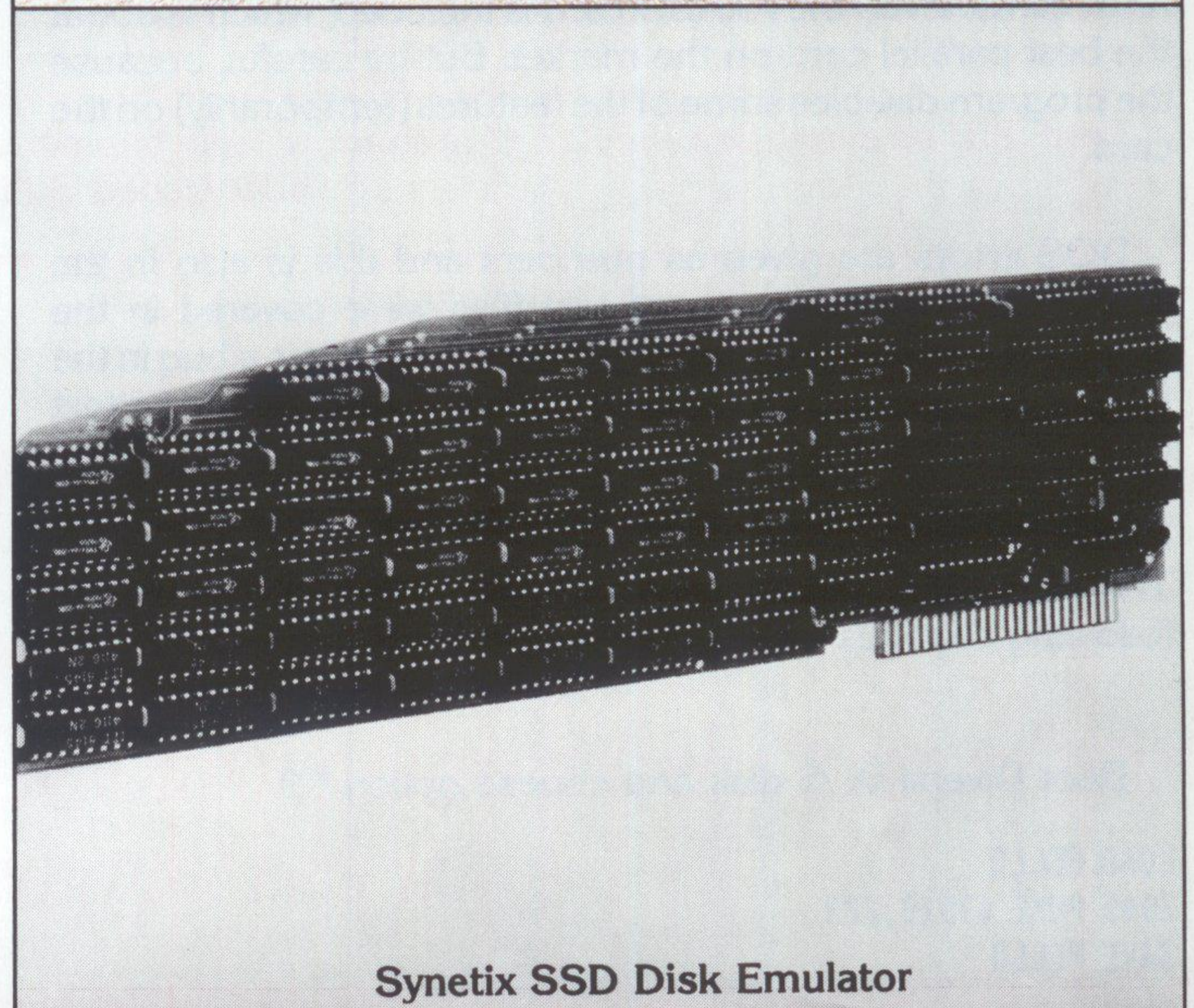
3. Diversi-DOS

Diversi-DOS has one of the most unusual marketing approaches known to man. *They tell you to give their program to everyone*, and if the recipient likes it, then he sends them about \$25. They will send you a sticker to put on your drive. Because the program is so reasonably priced and packed full of features, it is selling well. Bill Basham, head of DSR, Inc. is a programming genius in every sense of the word.

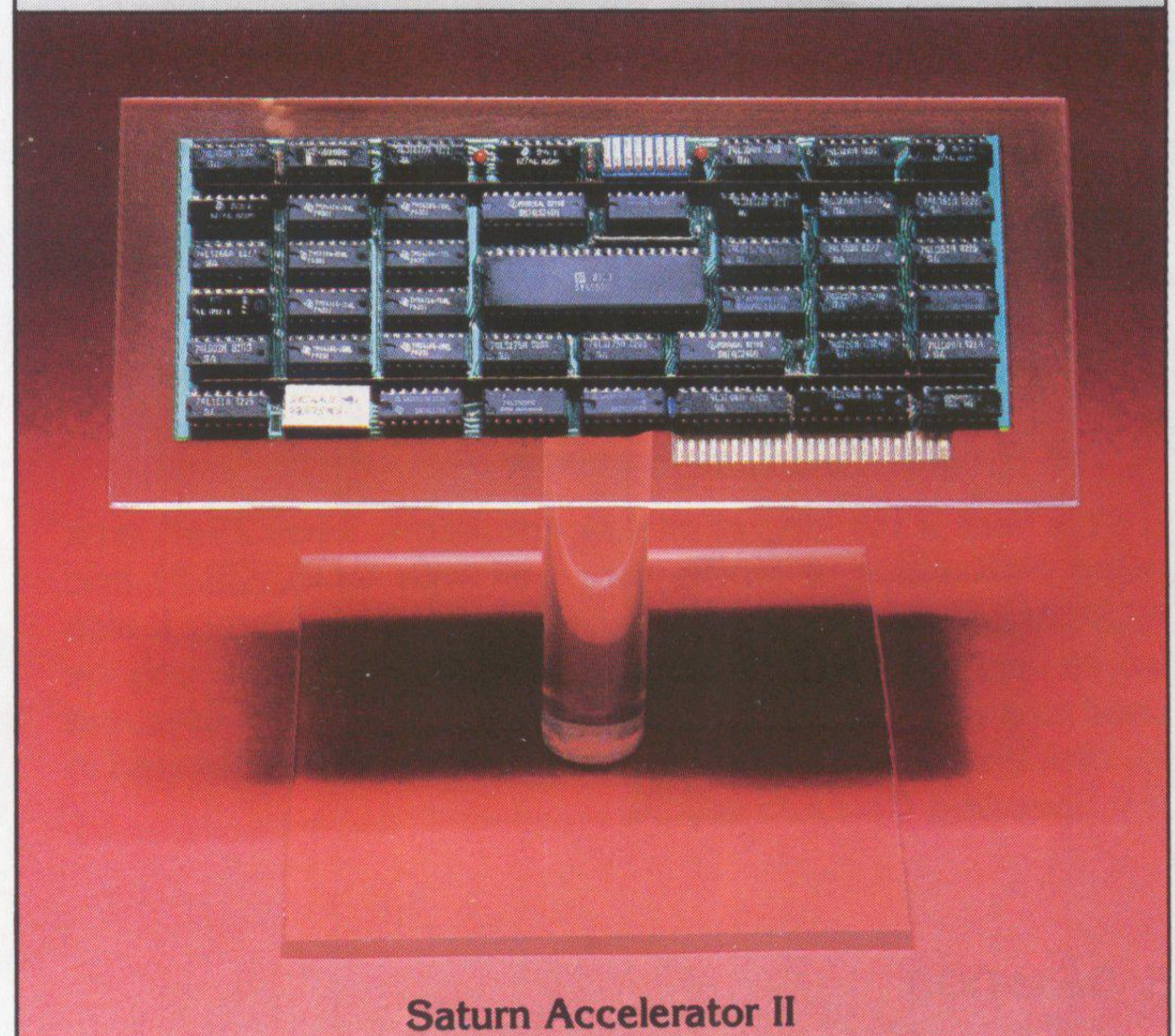
The instructions are on the disk, and will print out automatically if you want. Because Diversi-DOS modifies DOS, any program that has a "trick" DOS may not work properly with the



RAMEX 128



Synetix SSD Disk Emulator



Saturn Accelerator II

program, so be careful. This also applies to protected programs. There is a super fast back-up program on the disk that is a breeze to use.

Some of the more interesting features are the keyboard and print buffer utilities on the disk. This allows you to type on the keyboard while, for example, the drive is running, and not lose any characters. The print buffer is something else. I covered this point in great detail in my buffer article (**Apple Orchard**, April 1983), but for a short explanation, the computer must wait until the printer is finished printing before it sends more material. With the buffer option, the program is stored in the RAM card in Slot 0. If you have a Franklin, it comes with 16K there. If you replace it with the Ramex 128, you have a 128K print buffer! That can be more than 15 minutes of printing, which means you can use the computer for other purposes while the printer prints away.

The buffer is set up to work with a variety of printer cards and RAM cards. Even the PKASO card is included, which is about the best parallel card on the market. But be careful, because the program disables some of the features (temporarily) on the card.

DOS errors are given as numbers and this is also in the instructions. Random access text files were covered in the above, so they will not be repeated here. There was a bug in the first release of Version 2-C: when you read to the end of a text file with an ONERR in effect, you may read some extra garbage data before the ONERR is executed. To see if you have the bug, boot a disk with Diversi-DOS and type **PRINT PEEK (43519)**. The correct value is 223. If you don't see this value, then the following changes will fix the bug:

Boot Diversi-DOS disk and choose option #9

```
LOAD HELLO
2045 POKE 43519,223
SAVE HELLO
```

Use option 2 to add Diversi-DOS to all your disks again.

Interrupts are also discussed, but we'll not cover them here as very few of you will use them. Licensing information is also included along with many other commands. This program is top shelf in every respect.

4. The Synetix SSD Disk Emulator

This card is the king of all disk emulators. It has 294K on it using 64K chips. It works very well with DOS, CP/M and PASCAL, along with many of the fast DOS programs. It also works very well with the Saturn Accelerator II. It can only be used as a disk emulator, and not for buffers or other uses, but for disk emulation, it is untouchable. This card was covered in my Speeding Up DOS article.

5. David-DOS

David-DOS is another fast DOS program with some interesting features. Listed below are these added commands:

/	CATALOG
TLOAD	SPEED LOAD ALL TEXT FILES
TLIST	LIST ALL TEXT FILES
DUMP	BINARY/ASCII
DISA	DISASSEMBLE BINARY
AL	ADDRESS/LENGTH OF LAST LOAD
HIDOS	MOVE DOS IN MEMORY TO CARD #0
FIND	FIND HEX CHARACTER SEQUENCES
DATE	PRINT MOUNTAIN HARDWARE DATE

The documentation is on the disk and can be printed out to the printer. It is quite comprehensive, and the program is a marvel of human engineering. David Weston, the author, is of the same genius caliber as Bill Basham of Diversi-DOS and Dave Winzler of Microseeds. It is a pleasure to have these three experts in the Apple field of programming. I also admire that all three have chosen not to protect their disk. This does wonders if you want to use it with other drives, disk emulators and the like.

The program is easy to use, and the FIND command is interesting. FIND searches for any hex sequence up to 31 bytes long. Hit return before entering the bytes. Put a space between each byte. Example:

Keyboard Entry

Screen

```
FIND
FIND
HIT RETURN
FIND HEX:
11 3C 45
FIND HEX:11 3C 45
HIT RETURN
(FOUND ADDRESSES)
```

The program is packed full of uses and I strongly recommend it.

6. The Saturn Accelerator II

Since I purchased my Apple in 1977, I have been waiting for a card like the Saturn Accelerator II. The card has a fast (3.6 Mhz) 6502 and 64K of memory. It does all of the calculations while the Apple or Franklin does the video display. You now have both running at once, and the speed is incredible.

There are switches on the card which allow you to slow down certain peripheral cards, such as the drives. The card has a built-in language card, and the program is put into the memory of the card. A pre-boot is provided with the card with four options on it:

- Phantom 0 - You would use this for programs written in Integer or Pascal and for programs which require a language card, such as the arithmetic processor.
- Fast Applesoft - This option is specifically set up for speeding up Applesoft programs.
- Disable - This option is for all programs which run under CP/M when using the Microsoft Z80 Softcard. It is not needed if

Hitch your Apple to a Star.

OFFER
EXTENDED
THROUGH OCTOBER 31ST

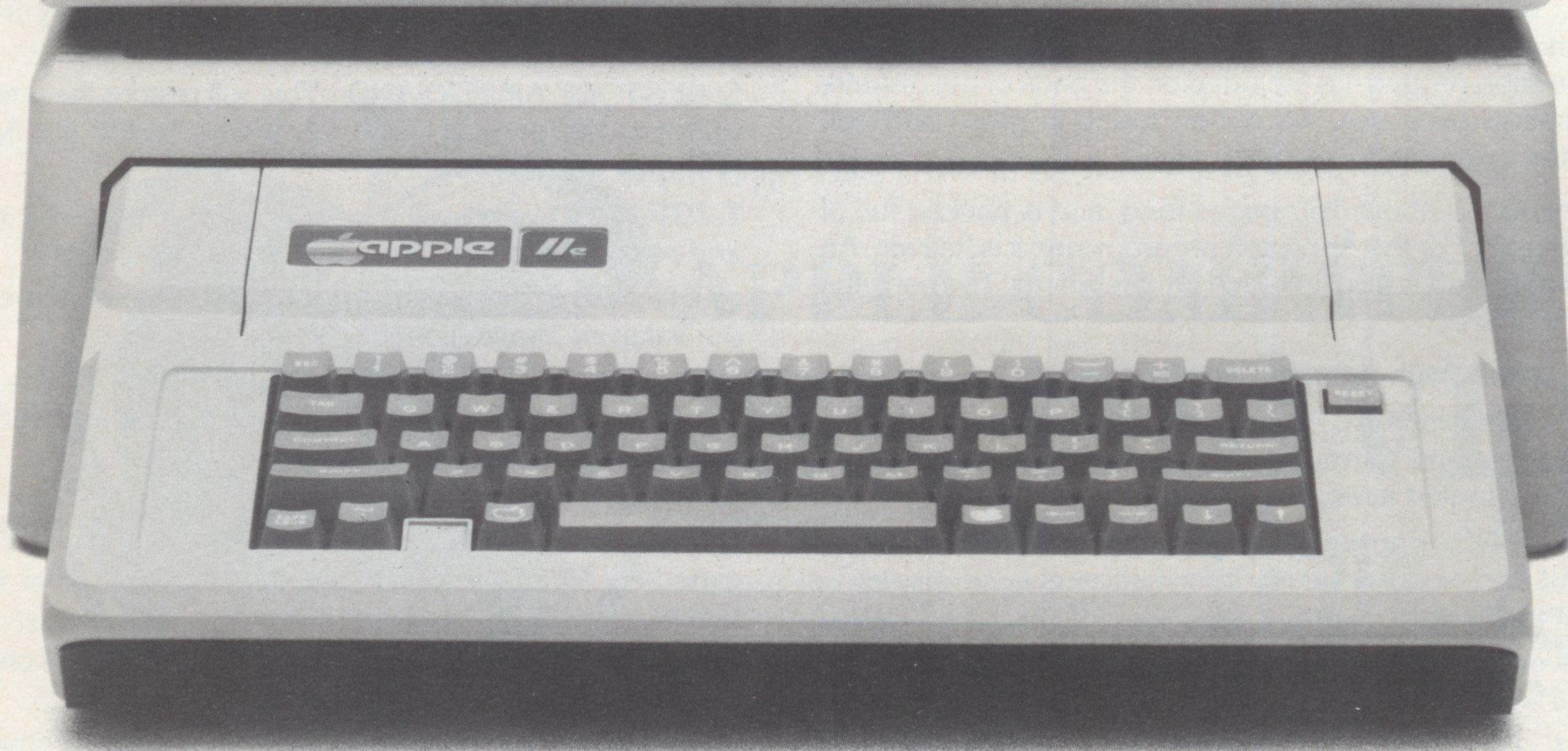
FREE CP/M[®] BOARD WORTH \$375

With MicroPro[®] WordStar[®] or InfoStar[™]

MicroPro now helps you expand the capabilities of your Apple computer with a free CP/M board. If you already have CP/M, we'll give you \$100 off

WordStar or InfoStar to install on your Apple computer.

OFFER GOOD JUNE 1 THROUGH OCTOBER 31, 1983



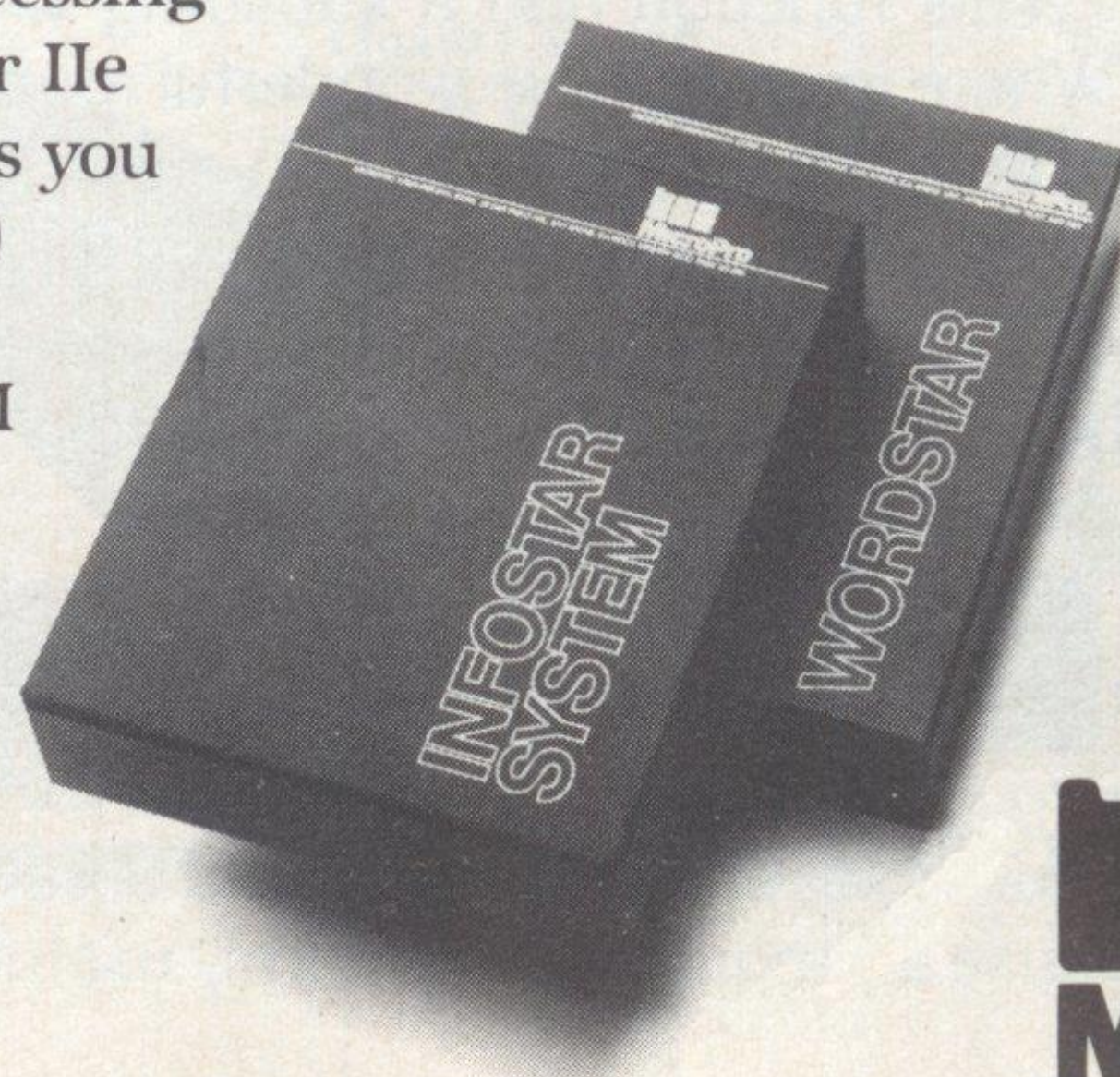
Apple, CP/M & IBM PC are registered trademarks of Apple Computer Inc., Digital Research Inc. & IBM Corp. respectively.

Now when you buy our best selling WordStar word processing software or InfoStar data base manager for your Apple II, II+ or IIe computer, we'll throw in a free CP/M board. (The one that gives you full implementation of WordStar in one board instead of three.)

With it, you'll be able to take full advantage of the advanced features that already made WordStar No. 1 on the IBM PC. And you can use InfoStar data base manager to enter, sort, and report information without programming in code.

To take us up on our offer, hurry down to your local computer store between June 1 and October 31 and pick up WordStar or InfoStar.

Either one will make your Apple shine.



Technical Data: Board Includes 6 MHz Z-80B Processor, 64K RAM, CP/M 2.2, 70 Column High Resolution Capability, Shift Key Modification Device, And Expansion Interface Connector.

MicroPro[®]

you use the Appli-Card from Personal Computer Products.

D. Standard - This uses no preboot disk for binary programs which do not require a Language card.

The manual is written in two parts. The first part is for a quick introduction and quick set up. It allows you to use the board instantly for those of us that can't wait. The second part goes into all the detail that you need once you get over the initial excitement of trying the device. I happen to like the arrangement. Also given in the appendix is a complete list of programs and the recommended options to use. Many programs, such as VisiCalc, do not need the preboot.

At this writing, this card is the only one of its type, so there is no competition. If you want to speed up the computer, this is a sure-fire way to do it.

7. The Routine Machine

Southwestern Data Systems has always had a reputation of having some of the finest programs available for the Apple. This goes way back when they came out with Apple Doc, and they haven't stopped giving us excellent software since then. Their packages seem to get better with each addition. One of their latest selections is The Routine Machine.

The Routine Machine is a series of programming utilities which can be used with Applesoft programs. You can have the speed and versatility of machine language in Applesoft without knowing machine language, and you can pass variables back and forth from the utility to the program with no problems whatsoever. You decide where in memory to locate the routines, and more than one routine can be used. No debugging is necessary, and use of these routines is simple. A demonstration program is included on the back of the diskette for illustration purposes. An option menu is given and you choose the appropriate selection.

The manual is about 160 pages long, and is packed full of ideas and uses. For the Applesoft programmer, it is heaven. An excellent appendix is also included for references. The package is a piece of art.

Some of the main points of the package:

A. SWAP - This swaps two Applesoft variables without requiring a third variable and without generating garbage which would have to be handled later.

B. PRINT USING - Formats numbers for screen, printer or disk file output. It makes monetary information easy to display.

C. TEXT OUTPUT - This will print text to screen or printer without word breaks at the right margin. It is similar to word-wrap in a word processor.

D. STRING INPUT.RM - This wonderful routine will allow you to enter in commas and colons in input strings. (Try doing this without this routine!)

E. STRING SEARCH - This function find a substring within another larger string. (Where were all these wonderful routines when I began programming the Apple in 1977?)

F. ARRAY SEARCH - This is used in searching a one-dimensional string array for the occurrence of a specified search string, or for the occurrence of a string standing in a certain relation to the search string.

G. BUBBLE SORT - Here we sort the elements of a one dimensional string array, placing all the empty strings at the end.

H. BEEP - Generates a pure tone of a given pitch and duration. It can also be used to pause a program. Extremely useful for

generation of music.

I. SOUND EFFECTS - Simply lets you use sound effects in your program.

J. FIX LINK FIELDS - This recalculates and establishes the link fields in an Applesoft program.

K. ERR - This function fixes the stack pointer in preparation for continuing the operation of a running Applesoft program when RESUME will not be used. It will also optionally return the error code and line number of the error.

L. ERR MSSG - This will print the usual Applesoft or DOS error message from a running program without halting program execution. (Can you believe all this for \$64.96?)

M. GOTO - This allows the equivalent of Applesoft's GOTO statement with the line number specified being given by a variable.

N. GOSUB - This allows the equivalent of Applesoft's GOSUB statement with the line number specified being given by a variable.

O. LINE DATA RESTORE - This performs a similar function to Applesoft's RESTORE command, with the exception that the line number to which the DATA pointer is restored can be specified.

P. DATA ELEMENT SELECT - This advances the DATA pointer a given number of positions relative to its current position. This gives a random access-like aspect to DATA statements.

Q. XNUM - This will convert numbers between decimal and hexadecimal notations.

R. MEMORY MOVE - Moves a block of memory from one location to another.

S. RESTORE AMPERSAND - This will restore the ampersand vector to its original value, as it was before the Applesoft program was run.

T. PTR READ - This will read any two byte pointer in memory and return the decimal value in a numeric variable.

U. PTR WRITE - This sets any two byte pointer in memory to the value or address specified.

V. SHAPE GOBBLER / SHAPE VIEWER - This is a utility provided to convert existing Applesoft shape tables into usable Routine Machine Modules.

W. SHAPE TABLE VIEWER - This utility is provided to allow you to examine raw shape tables. this can be useful, especially in cases where you might not even be sure if a binary file is in fact a shape table.

X. HIRES ASCII - This is a shape table of 95 ASCII characters used for printing test on the HIRES screen.

Y. BLOAD - This BLOADs binary files about 4 times faster than normal. It will not, however, speed up fast DOS diskettes that I have seen.

Z. BINADR - Returns the length and load address of any binary file on a disk, without loading the file.

AA. RESET NORM - This sets the RESET vector back to its normal condition, so that pressing RESET will put the use in direct command mode with an Applesoft prompt.

BB. RESET ONERR - This sets the RESET vector so as to generate an Applesoft error code when RESET is pressed. If an ONERR GOTO statement is in effect, control will then pass to the error-handling routine as in the case of a normal error occurrence.

CC. RESET RUN - This sets the RESET vector so as to re-run the current Applesoft program when RESET is pressed.

DD.RESET BOOT - This will boot the computer when RESET is pressed.

EE. FREE SECTOR COUNT - This allows determination during program execution of the number of free sectors on a disk in a selected drive.

It should be mentioned that not all of these programs speed up the software, but because they are written in machine language, they certainly do run faster than a normal Applesoft program. The other benefits are additional.

8. The CCS Model 7811 Arithmetic Processor

The 7811 Arithmetic Processor Unit from California Computer Systems is a powerful floating-point hardware unit which, under the control of the CCS Applesoft on the supplied diskette, significantly improves the execution speed of mathematical functions. Program execution speeds can be increased by as much as 70 to 1, depending on what function you are working with.

The only drawback is that you have 6 significant digits instead of the usual 8. This, in my opinion, is a small sacrifice. If you must have 9 digits, turn the card off and operate Applesoft normally.

I have not tried this card out on the Apple //e, but California Computer Systems recently informed me that the Arithmetic Processor has been tested on the //e and is compatible. The card plugs into any slot, and you simply boot the enclosed diskette.

The following benchmarks were done to illustrate the speed difference of this card:

Function	Speed Ratio
MUL	1:1.76
FDIV	1:2.56
EXP	1:10.07
SQR	1:71.24
RND	1:3.50
LOG	1:7.98
COS	1:9.48
PWR	1:4.80
TAN	1:15.50
ATN	1:13.55
INT	1:1.18
ACOS	1:32.20
ASIN	1:27.62
LOG10	1:23.98
PI	1:12.44
INV	1:3.50
SINH	1:10.57
COSH	1:10.66
TANH	1:8.08

As you can see, some of the improvements are substantial. The card works well with the Saturn Accelerator in the limited testing that I did.



5 1/4" Floppy Disk Drive in Apple Colored Case with Cable

APPLE COMPATIBLE DISK DRIVES

Compatible with Apple Language card,
Z80 Software, Fortran, Pascal, & CP/M.

From
\$225⁹⁵

Serial Board & Phone Modem
in Assembled & Kit form
Call for Prices.

Quality Tested
90 Day Warranty
VISA & Mastercard Accepted

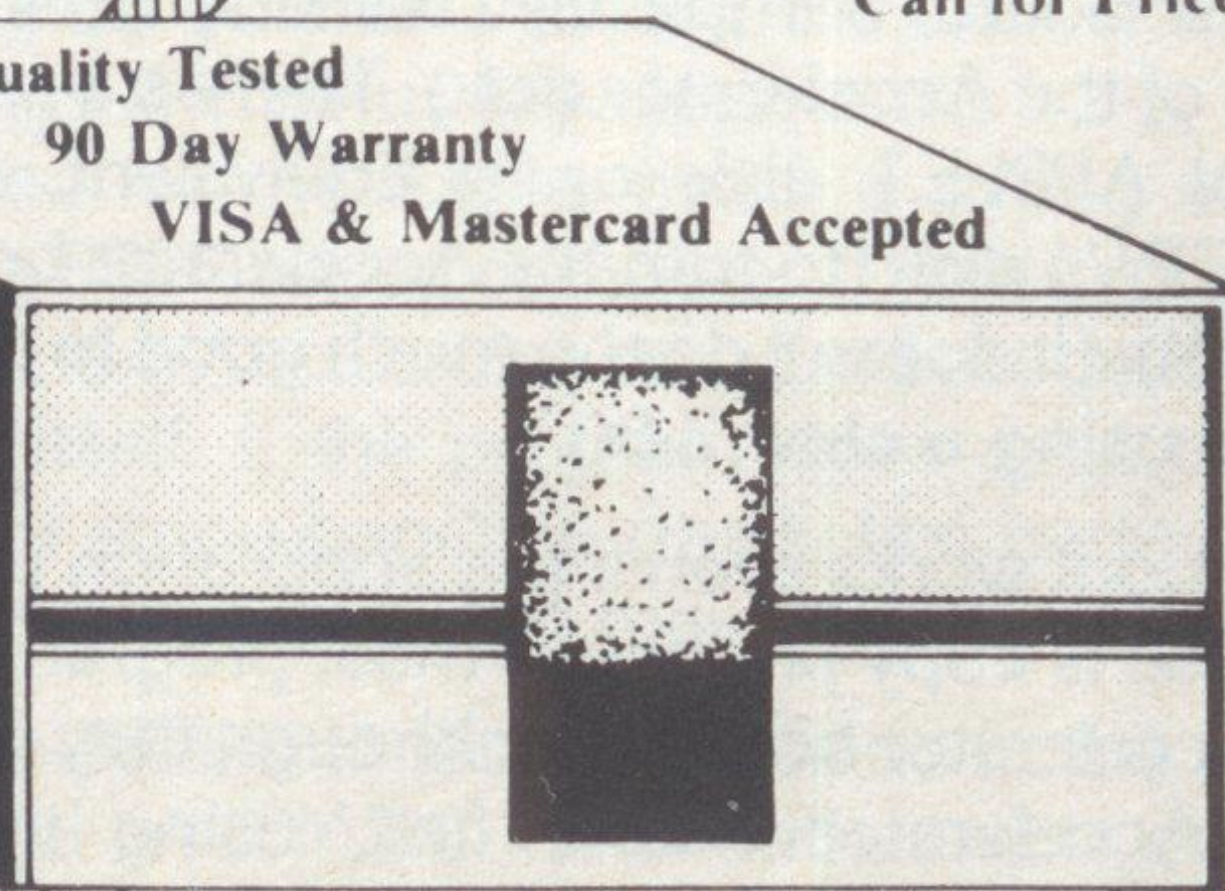
FOR FURTHER INFORMATION CALL (TOLL FREE) OR WRITE FOR FREE LITERATURE

MICRO-DESIGN

6301 MANCHACA ROAD, SUITE B. AUSTIN TEXAS. 78745
TEXAS RES. CALL 512/441-7890



1-800-531-5002



The Accelerator II

Review by Barry D. Bayer

Saturn Systems, Inc., of Ann Arbor, Michigan has moved from high density memory expansion boards and VisiCalc pre-boots to a \$600 board with a "fast" 6502C and 64K of "fast" RAM (Random Access Memory), that will make your Apple II really move.

The Accelerator board installs with very little problem by dropping into Apple Slot 0, 1, 2, 3 or 4. But which Slot? Slot 0 is not recommended for various reasons, Slot 1 contains my printer interface, 2 my modem, 3 my 80 Column Card, and 4 my 128K card. My modem-terminal program (Hello Central!) is very easily configurable, so I banished the modem to 5, liberating Slot 2 for the Accelerator board.

Because the board does not speed up speed dependent tasks such as modem communications and disk I/O, I was also required to set DIP switches and to notify the Accelerator board which slots should operate at standard speed. And that's all the installation there was. No screwdriver, no IC puller, no soldering iron.

Many programs (there's a list in the manual) will boot and run without further ado. For others, it is necessary to pre-boot a disk which will allow you to make the Accelerator board a phantom Slot 0, to add machine code which speeds up the language in ROM on your motherboard, or to disable the Accelerator board, completely (but only until powerdown or rebooting of the Accelerator disk). There's a routine to modify the Pascal APPLE1 disk, or its equivalent, and another to modify a CP/M disk (to turn the Accelerator board off if you're using CP/M). (It doesn't do too much good to add a fast 6502 if you're still using a slow Z-80.)

The preboot works even if the application program you are trying to use is copy protected. Most programs seem to work with the Accelerator board on, although they crash unless you turn the Accelerator board, first, using the preboot. (For example, the Videx 80 Column card preboot for Applewriter II doesn't work properly unless you turn the Accelerator card off. Why would anyone need an Accelerator board for a word

processor? Well, it scrolls faster, searches faster, and completely eliminates any typeahead. Its not crucial, by any means, but it is handy.

The first thing I did after installing the Accelerator board was to boot VisiCalc, and try it out with a large template which used to take over a minute to recalculate. With the new installation it took 22 seconds. I then ran some tests with time consuming formulae, and things did speed up considerably. (One change that is immediately noticeable is that the VisiCalc "beep", which is undoubtedly generated by a machine language subroutine which "toggles" the speaker for a given length of time at a given frequency) is much shorter, and octaves higher. (It turns out that all such sounds including "music" and "buzzes" end up much higher and shorter, leading to some amusing and unanticipated effects.) VisiCalc loading speeds up a little; as previously stated, disk access speed is not altered, but accelerated VisiCalc processes information coming off the disk so quickly that the drive never stops. But VisiCalc is so quick, in most cases, that it's really difficult to say that it "feels" different with the Accelerator.

But try a pre-boot such as Omega Microware's Super Expander Plus, with a Videx 80 Column card, and a 128K card, and you get a program which "feels" slow. Clearing memory, using the /C command takes about 10 seconds without the Accelerator, and 3 and a half seconds with the Accelerator. Recalculating a 136K file takes a long time without the board, but a third or less as long with the board. Scrolling the 80 column card can be jerky and slow, but not with the Accelerator board, which speeds things up so much it is almost like 40 column video. The expanded program is so different that the speedup is immediately noticeable.

So much for impressions. From time to time certain "benchmark" programs are published to indicate how fast a particular computer is. One set was published in the January, 1982 issue of *BYTE Magazine*, in Gregg Williams' article on his first glimpse of the IBM Personal Computer. The five programs constituting the Benchmark are shown in Figure 1.

	Byte Tests		Bayer Tests			
	IBM PC	Apple II	Apple II w/o Acc.	Apple II with Acc.	% of Time of:	
					Apple Time	IBM Time
Test 1	6.43	6.66	6.2	2.1	.34	.33
Test 2	23.8	29.0	26.3	8.7	.33	.37
Test 3	12.4	13.9	12.3	4.3	.35	.35
Test 4	23.0	32.3	27.7	9.3	.35	.42
Test 5	190.0	241.0	194.8	66.4	.34	.35

Figure 2

```

1 PRINT CHR$ (7)
60 A = 2.71828
80 B = 3.14159
100 FOR I = 1 TO 5000
320 NEXT I
1000 PRINT CHR$ (7)
      Test 1

1 PRINT CHR$ (7)
60 A = 2.71828
80 B = 3.14159
100 FOR I = 1 TO 5000
120 C = A / B
320 NEXT I
1000 PRINT CHR$ (7)
      Test 2

1 PRINT CHR$ (7)
2 SIZE = 7000
3 DIM FLAGS (7001)
4 PRINT "ONLY 1 ITERATION"
5 COUNT = 0
6 FOR I = 1 TO SIZE
7 FLAGS(I) = 1
8 NEXT I
9 FOR I = 0 TO SIZE
10 IF FLAGS(I) = 0 THEN 18
11 PRIME = I + I + 3
12 K = I + PRIME
13 IF K > SIZE THEN 17
14 FLAGS(K) = 0
15 K = K + PRIME
16 GOTO 13
17 COUNT = COUNT + 1
18 NEXT I
19 PRINT COUNT, " PRIMES"
1000 PRINT CHR$ (7)
      Test 5

```

Figure 1

(For an explanation of what these programs do and why they were chosen, see the Williams article on Page 54 of January 1982 *BYTE*. The programs here are the same, with the addition of a line to ring the Apple's "bell" at the beginning and end of each program, to help with starting and stopping the timings.) In general, the *BYTE* article showed the Apple to be between 4 and 27 per cent slower than the IBM PC

Figure 2 shows the raw results of my tests. For reference I have included the *BYTE* Apple and IBM results. For some reason, my figures for the Apple without the Accelerator board are less than *BYTE*'s, but the real contrast comes with the Accelerator board. In all cases, the accelerated Apple took about one third the time of the regular Apple, which means that the Accelerator board makes the Apple a *lot* faster than the PC.

But the Accelerator board is not an unmitigated pleasure. Saturn Systems has set the Accelerator board to automatically slow down for disk access and modem operation, but games are not automatically slowed. (Pinball or PacMan are simply impossible going three times faster.) If you move around a lot between one program and another, you're going to be turning off your computer and doing a cold boot, more often than without the board. But you can live with that. What you might not be able to live with is a 500 milliamp draw. (That's 25 per cent of the maximum current rating of Apple's power supply.) Now my Apple is still working with the lineup shown in Figure 3, but I really wonder how close I'm getting to my power supply's limit.

- Slot 0 - 16K Memory Expansion
- Slot 1 - Mountain Hardware CPS Multifunction Card
- Slot 2 - Saturn Accelerator card
- Slot 3 - Videx Videoterm
- Slot 4 - Hayes Micromodem II
- Slot 5 - 128K Memory Expansion
- Slot 6 - Disk Controller
- Slot 7 - blank (*Barry, how could you?—PCW*)

Figure 3

Last, but certainly not least, the Accelerator board has a manufacturer's suggested retail price of \$600. That's a good percentage of the cost of an Apple system. Is the board worth the money, or should you use it to "upgrade" to a new, modern computer, or even half a 5 magabyte disk. That's a very personal decision and, I suppose it depends on how you use your Apple. If it's strictly a game machine, the answer is no. (Let me take back a little bit of that. If the game the computer is playing is chess, the answer might well be yes.) If you use a lot of large memory 80 column VisiCalc, the answer might be yes, although things like Lotus Development's 1-2-3, which is not being provided for the Apple, make a strong argument for purchasing an 8088 based computer. If your programs tend to be disk bound, the Accelerator board will probably not speed it up much by itself. If the program does a lot of calculating and/or a lot of string handling (really fast garbage collection with the Accelerator board) it may help.

Finally, if you'd like to look your friend in the eye, (the one who purchased an IBM PC last month, and is chiding you to junk your "toy" and get a "real business computer"), and explain that your Apple is two and a half times faster than his pokey PC, then the Accelerator board is a must.



WPL - The Sleeping Giant

Part One - Introduction

by Richard Loggins

Hidden deep inside your AppleWriter is an enormously powerful force just waiting to be unleashed. Once aroused from its dormant state, its power can be harnessed to perform feats of magic. It's called Word Processing Language (WPL for short), and is probably the most overlooked feature of AppleWriter.

Once under control, this power can be used to print any number of form letters, specialized reports, quotations, and more. It can personalize letters, insert financial information, and even replace other software you would normally purchase. WPL won't replace VisiCalc or Multiplan, but it can provide an acceptable alternative to mailing lists and small databases.

This series of articles will remove the mystical shroud surrounding WPL. It will help you to take advantage of what you already have - hopefully without frustrations.

We'll start with simple WPL commands and build on them a little at the time. WPL programs will be developed that will allow some pretty neat and useful applications. You'll be surprised to discover that you've already used many of the WPL commands that we'll discuss.

As an added benefit, the WPL programs should work for AppleWriter /// and AppleWriter //e. Just pay attention to the file name notations and you'll do fine.

Command Origins

WPL commands can originate from (1) the keyboard, from (2) a text file, or from (3) *within* a text file as it is being printed.

When the command originates from the keyboard, it is known as originating from the *immediate* mode. Commands that come from a text file, or a text file as it is printed, are known as originating from the *deferred* mode.

If you have used AppleWriter, at one time or another you have used the **CTRL-P** command to view or change the current printing parameters. By changing any of the printing parameters, you have already been using WPL, although you didn't know it!

In fact, you use WPL each time you load a file, save a file, change character sets, etc. All of the AppleWriter immediate mode commands are actually WPL commands. That ends the tutorial on WPL immediate mode. Easy, wasn't it?

Now that you have mastered the immediate mode (and even if you haven't), it's time to conquer a little of the deferred mode via printing a text file.

If you have printed letters or other files using Apple Writer, you have probably embedded printing format commands within the text. The most common command used is the ".ff" (or ".FF" if you prefer) to issue a form feed. This command causes the printer to advance the paper to the top of the next page. Surprise: this is also a WPL command!

The rest of this article will be devoted to the WPL commands that originate from a text file on disk, how they are used, and WPL programs.

Text Commands

Commands are one or more of a specified set of instructions that tell your computer what action to take. Commands are nothing more than a predetermined sequence of characters, and are easier to learn than a foreign language or shorthand.

WPL commands are composed of one to three letters. Each command will cause a specific action to take place. The shortest command is simply the letter "P", which is used to insert a comment into the program.

Comments are brief statements that explain portions of the program. They are not required, and are totally ignored during program execution.

Many of the commands used in a WPL program are identical to the commands used during normal operation of AppleWriter. The command to load a file **CTRL-L** is the same in normal operation and in a WPL program. Both must be followed by a legal file name, and both will load a file.

The primary difference between normal operation and execution of a WPL program lies in the understanding of what

you can't see. Under normal operation, you can see the CONTROL key being pressed; under WPL you can't.

This brings forth the first rule:

Rule 1: The first character in a line following a blank character is treated as a control character.

This means if the first letter on a new line is a space, and the space is followed by the letter "L", the letter "L" is treated as CTRL-L.

So a command such as

```
L .D1/MYFILE
```

with a space preceding the "L", would be interpreted as meaning "LOAD from Drive 1 the file named MYFILE", and would act accordingly. This allows use of all the CONTROL commands normally used in AppleWriter from the immediate mode.

Which brings up the question of why it was mentioned in the first place. Under WPL program control, AppleWriter uses two different displays, both of which can be viewed on your monitor.

One display is the normal display you see whenever you use the text editor portion of AppleWriter. The other screen display we will call the WPL display. It is used to inform you what is taking place, to request information from you, etc., while a WPL program is being executed.

To clear the text editor display, you enter CTRL-N followed by Y, and press «RETURN». This will erase the text editor display, along with any file you currently have in memory.

The WPL display is another matter. It is erased by using a CTRL-«Backslash». In a WPL program, this command would appear as:

```
PPR«Backslash»
```

with a space preceding the first P. More on this later.

Simply having all of the CONTROL commands normally used by AppleWriter isn't enough to accomplish very much. There are two more essential items we must include to make effective use of WPL.

The two items we must add are labels and variables. Fortunately, both of these items exist, leaving only the problem of understanding them.

We can now say that a WPL program consists of only three items: labels, commands, and variables.

WPL Labels

Labels are to WPL programs what road maps are to travelers, or addresses are to mail carriers. Labels provide WPL with a method to identify a unique location. And like addresses, no two should be the same.

Labels are not required in a WPL program, but they are used to control the sequence of program execution. Labels are the creation of the person writing the program, and generally are descriptive of the function performed by a part of the program.

For instance, suppose you have a label named "LOOP". If the program is directed to go to "LOOP", it will do so and carry out the instructions following the label. This brings up the rule for labels:

Rule 2 If the first item in each line is not a blank character (a space), all of the following characters up to the first blank character are treated as a label.

For our program listings, we'll use a format that should make it easy to identify labels from commands. Each program will have the heading shown below, followed by one blank line. Labels will start to the left of the space following "WPL", and commands will be indented to start at the right of the space.

```
WPL Program: [Program Name]
```

```
LABELS will start here.
```

```
COMMANDS will start here.
```

WPL programs are created by using the text editor portion of Apple Writer. When you enter these programs, be sure to start your program with the first label, and *not* our program indicator (the part that says "WPL Program").

Variables

WPL supports two different types of variables. The first type is called *numeric* variables, and the second type is called *string* variables.

Numeric variables are used for representing numbers. There are only three numeric variables in WPL, represented by the letters X, Y, and Z.

String variables are used to represent a series of characters. This can be letters, numbers, or a combination of both. WPL provides four string variables: \$A, \$B, \$C, and \$D.

If you have experience programming in BASIC, don't be alarmed by the position of the "\$" in the string variable name. In WPL, the "\$" always comes before the variable.

WPL Program Construction

Now that the components of a WPL program have been introduced, we'll cover a few basic rules about WPL programs, and put this information to use. As the man said, a picture is worth a thousand words.

After each program listing, a line by line discussion will examine each command and what it does.

Our first WPL program will be simple. It will load a file from the disk in drive one and print the file. For Apple /// owners, drive one is the internal drive. For Apple II owners, drive one will be Slot six, Drive one.

Conventional Apple /// pathnames will be used (".D1/") to refer to the drive. If you have an Apple II, make sure you change this reference to agree with DOS and Apple Writer //e.

Before a WPL program can be written to load and print our test file, we must first have a file to work with. The letter shown in Figure 1 will be used as our test file, and will be modified occasionally to suit our needs.

Get your Apple Writer up and running; type the letter, and save it with the name of "TEST.LETTER" on the disk in drive one.

When you have the letter saved on your disk, erase memory and enter the following WPL program.

```
WPL Program: PRINT.LETTER
```

```
INITIALIZE
```

```
    PND
```

```
    NY
```

```
LOAD
```

```
    L .D1/TEST.LETTER
```

```
PRINT
```

John Johnson
123 Oak Street
Denver, CO 88888

Dear John:

We are having a party to celebrate Independence day, and would like you and your wife to attend.

We'll supply all of the food and drinks, so all you have to bring is yourself and your wife.

The party will start at 4 p.m. and last until everyone is tired.

Sincerely,

Sam Smith

Figure 1

```
PNP
QUIT
PQT
```

Check each label to make sure it starts at the very first position in each line, and that the commands are indented. Remember, WPL won't recognize the labels if there is a blank character (space) in front of them.

When you are sure the program is correct, name it "PRINT.LETTER", and save it under that name on your disk.

Let's analyze the program and see what it will do. The first line of the program is a label. This label describes this part of the program as "Initialize", and is where our initial procedures are located.

The second line contains the command **PND**, which will turn off the text editor display (N for NO and D for Display). This only turns off the text editor, and not the entire video display.

By turning off the text editor display, our WPL program will have control of what is displayed on the monitor. With the text editor turned off, the WPL program will also execute about five times faster.

Line three, **NY**, is a command to clear memory. Since WPL treats the first character (N) following a blank character as a control character, this command is equivalent to entering CTRL-N followed by a Y from the keyboard in the immediate mode. The result is to erase memory and clear the text editor. That's why you saved both the letter and the WPL program to disk first.

Line four is another label. It describes the portion of our WPL program that will load the file.

Line five is a command. Again, the "L" is treated as a control character, making the command CTRL-L. This is followed by the location (.D1), a delimiter (the slash), and the name of the file to load. That's Apple ///, remember: on the Apple II//e, it would be L TEST.LETTER,D1.

Next is another label. This label describes the routine that will print the letter.

In line seven we use the command that will cause the letter to be printed. This is equivalent to entering **CTRL-P**, followed by **NP** from the immediate mode, and will result in printing the letter.

The last two lines of our program provide a formal ending. The label **QUIT** is followed by the command **PQT**, the command that will terminate a WPL program.

Before we use the WPL program, check your printing parameters and your printer. Get back into the text editor portion of Apple Writer, and we'll use our program (WPL programs *cannot* be executed from the printing parameters portion of Apple Writer).

Now, enter CTRL-P from the keyboard. The lower portion of your monitor should display:

[P]rint/Program:

Enter **DO .D1/PRINT.LETTER**, followed by the «RETURN» key. This will cause our WPL program to be executed. If you have entered everything correctly, the WPL program will load the letter, print it, and return control back to the text editor.

Since we ended the WPL program without clearing the text editor memory, the letter that was loaded by the WPL program will still be in memory, and should be displayed on your monitor.

If the WPL program does not execute as it should, press the «ESCAPE» key. The «ESCAPE» key will abort any WPL program, and return control to AppleWriter.

Play it Again

Printing a single copy by using a WPL program is more difficult than just printing a letter directly from AppleWriter. But our simple WPL program has provided the foundation for more complicated applications that will overcome this limitation.

By changing this WPL program, we can print up to 65,535 copies of any file, all automatically. This is accomplished by using a variable and a loop.

Clear memory, and enter the following WPL program. This

program will print five copies of our invitation, and demonstrate how variables and loops can be used.

WPL Program: PRINT.5

```
INITIALIZE
  PND
  NY
  PSX 5
LOAD
  L .D1/TEST.LETTER
PRINT
  PNP
  PSX - 1
LOOP
  PCP
  PSX - 1
  PGO LOOP
QUIT
  PQT
```

After entering the program, give it the name of "PRINT.5", and save it on your disk.

The first three lines of this WPL program are the same as our first WPL program. The text editor display is turned off, and memory is cleared.

In line four, we encounter a new instruction, **PSX 5**. This command ("PS") can only be used with numeric variables. Here, it sets the variable X to equal five. This is the number of copies we want our WPL program to print.

Line six loads our invitation, and line eight, **PNP**, prints the first copy.

In line nine, the **PSX** command is encountered again, but is used differently. Instead of setting the variable X to a predetermined value, the number contained in X is decremented by one, leaving the number four in X (five minus one is four).

The **PSX** command can be used to set the variable X to a specified value, subtract a specified quantity from the present value of X, or add a specified quantity to the present value of X.

Line ten introduces a new label, **LOOP**, that will be used to control the printing of the remaining four copies of our invitation.

Line eleven is a new command, **PCP**, which causes the next invitation to be printed. This command is the same as the "Continue Printing" command used in the immediate mode. **PCP** is used instead of the **PNP** command. This takes into account the top margin and top line options of the printing parameters in the event they are used.

Line twelve decrements the variable X again, leaving the number three in X.

The instruction in line thirteen, **PGO LOOP**, is a branch command. When this command is encountered, WPL will search the program for the label "LOOP", and transfer control of the program to the instructions following the label. In effect, this command says "GO to the label named LOOP."

This will cause the command **PCP** to be executed again, followed by the **PSX** command. Thus, we have set up a loop that will print several copies of our invitation.

To terminate the loop, we rely on the variable X. When the variable X is decremented to zero, the **PGO LOOP** instruction will be ignored, and control of the program will proceed to **QUIT**.

This action is known as a "controlled error" situation. Whenever a variable in WPL is decremented to zero, an error is generated. Whenever an error is generated, the next instruction is ignored.

In this program, the error will be generated after five copies of our invitation have been printed, breaking the loop and ending the program.

To use this WPL program, again enter CTRL-P followed by **DO .D1/PRINT.5** from the text editor.

Pick a Number

As the last WPL program ("PRINT.5") is written, it will print five copies of the invitation. All well and good if all we will need is five copies. However, there may be times when it is desirable to print more or less than five copies.

By introducing three new commands, the WPL program can be modified to print as many copies as needed.

Clear memory, load the WPL program named "PRINT.5", and make the following changes. Save this program with the name of "PRINT.NUMBER" on your disk.

WPL Program: PRINT.NUMBER

```
INITIALIZE
  PND
  NY
  PPR<Backslash>
GETNUMBER
  PIN How many copies? =$A
  PCS/$A/0/
  PGO QUIT
  PSX $A
LOAD
  L .D1/TEST.LETTER
PRINT
  PNP
  PSX - 1
  PGO LOOP
  PGO QUIT
LOOP
  PCP
  PSX - 1
  PGO LOOP
QUIT
  PQT
```

The first three lines of the modified WPL program are the same. Line four of the program introduces a new command **PPR«Backslash»** that will clear the video display.

This command is composed of two items. The command **PPR** tells AppleWriter to print on the WPL video display the information that follows. In this case, it is printing a **CONTROL-«Backslash»**.

To enter this command, you must press **CTRL-V**, followed by a **CTRL-«Backslash»**, followed by another **CTRL-V**.

The result you should see on your monitor is the backslash displayed in inverse (dark letter in white block.) The CONTROL-V allows control characters to be entered into your text that will be passed to a device without affecting AppleWriter. Be careful entering this command.

Line five, **GETNUMBER**, is a new label that begins the routine to determine how many copies are to be printed.

Line six contains a new command, **PIN** that will wait for an entry. When this command is executed, the message "How many copies?" will be displayed on the video monitor.

When a number is entered and «RETURN» is pressed, the number will be placed in the "\$A" variable. Notice there is no space between the equal sign and the \$A variable.

Line seven introduces another new instruction, the **PCS** command. Notice there are no spaces in this line (other than the spaces preceding the command). This instruction is used to compare the information contained between the first and second slash marks, with the information contained between the second and third slash marks.

In this case, it is comparing the number contained in \$A with the number zero. If the two items being compared are identical, the result is said to be valid, and no error is generated. Therefore, if a zero was entered for the number of copies to be printed, the result of the comparison would be valid, and no error would be produced.

The result of this comparison will determine what action the program will take. If the two items being compared are *not* identical, the result is said to be invalid, and an error is generated. So if a number was entered that was not zero, the result will be invalid, and an error will be generated. As with numeric variables, when an error is produced, the next instruction, to quit, is ignored.

If the user entered a zero, no error will be generated, and the **PGO QUIT** command will be executed. This will cause the WPL program to end, as there are no copies to be printed.

Let's try that again. If a number greater than zero was entered, the comparison result will be invalid, an error will be generated, and the **PGO QUIT** instruction will be ignored. Sounds like the long way around, but it works.

This action will allow the **PSX** command to be executed, assigning the number in \$A to the variable X.

Lines ten through thirteen of the program operate as before. The invitation is loaded, and the first copy is printed.

In line fourteen, X is again decremented to account for the first copy being printed.

The next two lines (fifteen and sixteen) are used to determine if more than one copy was desired. If a number greater than one was entered, no error will be generated when X is decremented in line fourteen. This will cause the program to branch (**PGO LOOP**) to the label **LOOP**.

If the number "1" was entered, an error will be generated when X is decremented in line fourteen. This will cause the **PGO LOOP** instruction to be ignored, and the **PGO QUIT** command will be executed.

The rest of the program will execute as did our previous WPL program, **PRINT.5**.

Choose Your File

So far, so good. The last example will allow you to specify how many copies of the invitation you wish to print.

If we make just one more change to the program, we will be able to load any file we wish, and print as many copies as desired.

Once again, clear memory and load the last WPL program. Make the following modifications, name it "YOUR.CHOICE", and save on disk. This will be our last accomplishment for this article.

WPL Program: YOUR.CHOICE

INITIALIZE

PND

NY

PPR<Backslash>

GETNAME

PPR

PIN Name and Location of file: =\$B

PCS/\$B//

PGO QUIT

PPR

PPR

GETNUMBER

PIN How many copies? =\$A

PCS/\$A/0/

PGO QUIT

PSX \$A

LOAD

L \$B

PRINT

PNP

PSX - 1

PGO LOOP

PGO QUIT

LOOP

PCP

PSX - 1

PGO LOOP

QUIT

PQT

We have inserted a new routine ("GETNAME") into the WPL program. This routine will print a blank line (PPR). It will display the message "Name and Location of file: ", and wait for a file name to be entered. The name that is entered will be assigned to another variable, \$B.

Again, we will use the **PCS** command. This time, we will compare the information in \$B with "nothing", to see if the «RETURN» key was pressed without a name being entered. To compare \$B with "nothing" (or a null string), make sure there is no space between the second and third slashes.

If the «RETURN» key was pressed with no name being entered, the result of the comparison will be valid, and the **PGO QUIT** command will be executed.

If a name was entered, the result of the comparison will be invalid, an error will be generated, and the **PGO QUIT** command will be ignored.

This does not mean the program will operate correctly if you enter an illegal file name or location. If you do this, when the

WPL program attempts to load the file, a system error will be produced, and the WPL program will terminate.

After the completing of the comparison test, control will pass to the following command (PPR), which will print a blank line. The next command (another PPR) will be executed, and the program will resume with GETNUMBER.

The number of copies to be printed is determined in the routine named GETNUMBER. This is the same routine used in our last program.

The routine named **LOAD** will load the file name you assigned to \$B into memory.

The rest of the program will print the desired number of copies, and terminate the program.

Now you have a small, general purpose WPL program that will allow you to load and print as many copies of a file as you need.

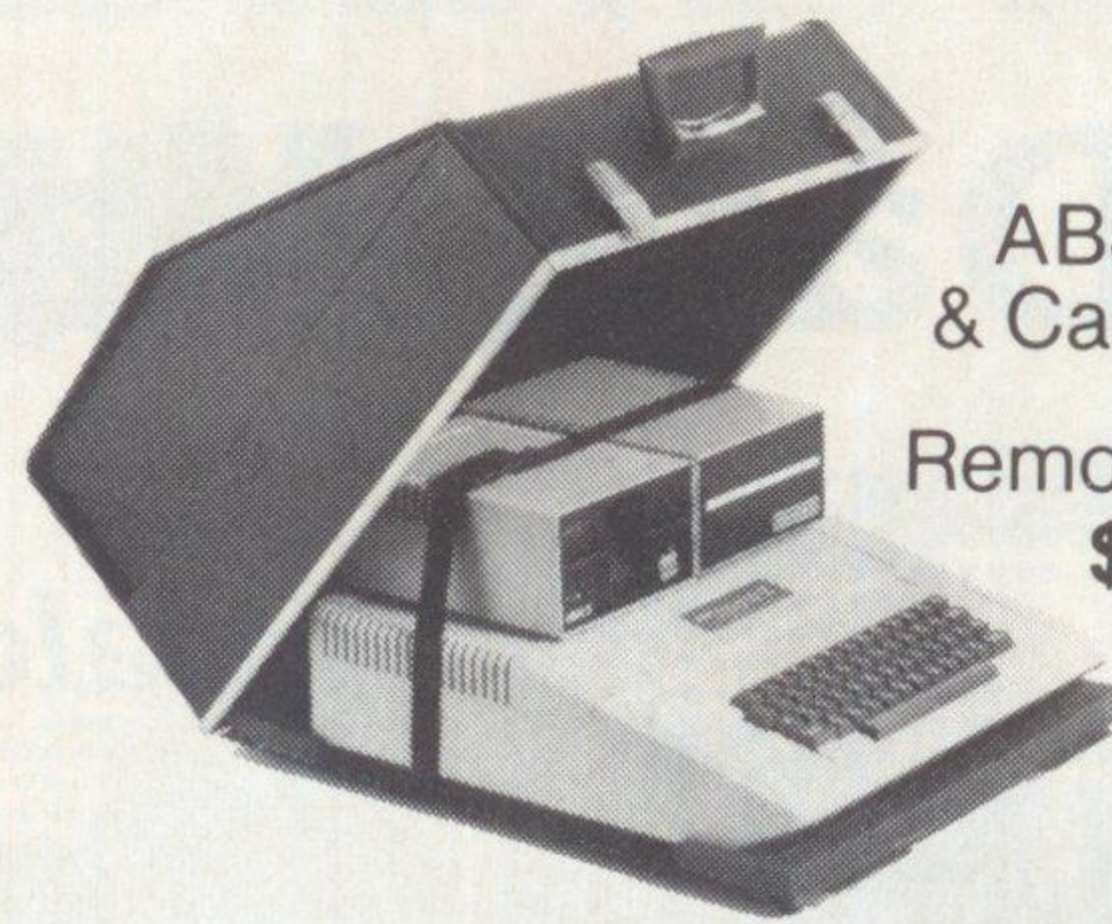
In this first article, the structure of a WPL program, the basic commands, and the flow of control have been examined. With some thought, a little work, and a dash of creativity thrown in, you should be able to modify these samples to perform some more useful tasks.

In our next article, we'll move on to more advanced techniques that will allow letters to be personalized, printing of multiple files, and some other interesting features.

Before we're finished, we'll write a WPL program that will create and maintain a mailing list, do automatic insertion, and who knows what.



Travel With Your Apple... Call (800) 847-4176!



ABS Storage
& Carrying Case
with
Removable Cover
\$69.50*

*UPS Shipping Included.

Apple • IBM • Epson

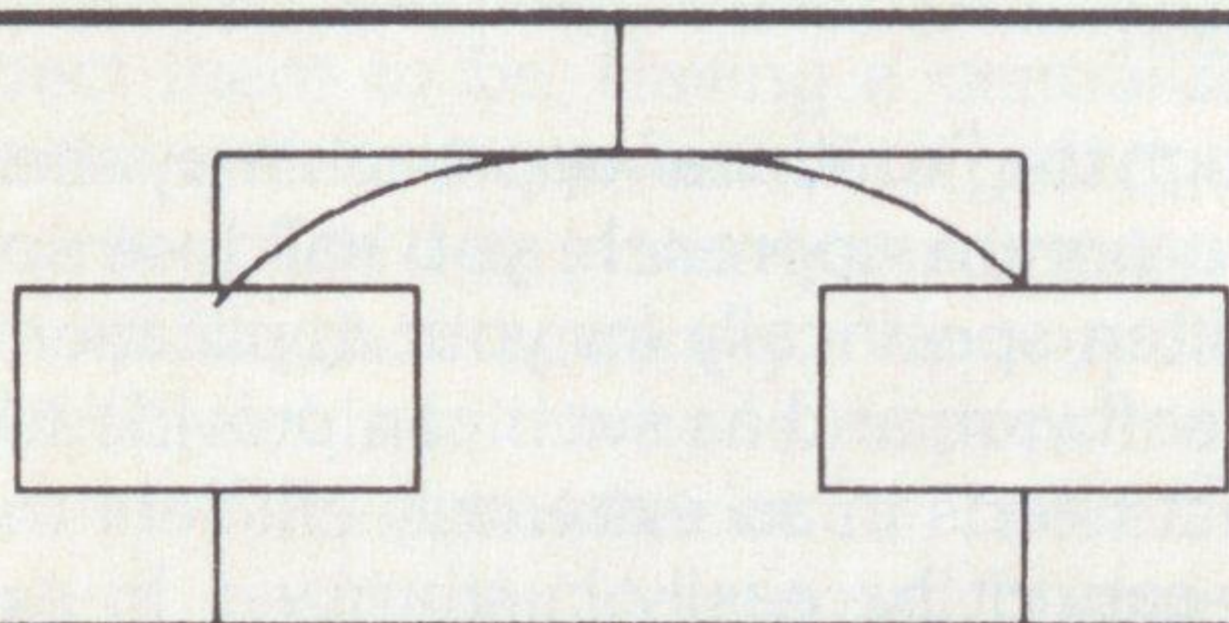
Call for Complete Computer Case Catalog.
Credit Card Customers Call Toll Free

Fiberbilt

Ikelheimer-Ernst, Inc.
601 West 26th Street
New York, New York 10001-1199
(212) 675-5820 (N.Y.C. area)



SOFTDRAW for the Apple III

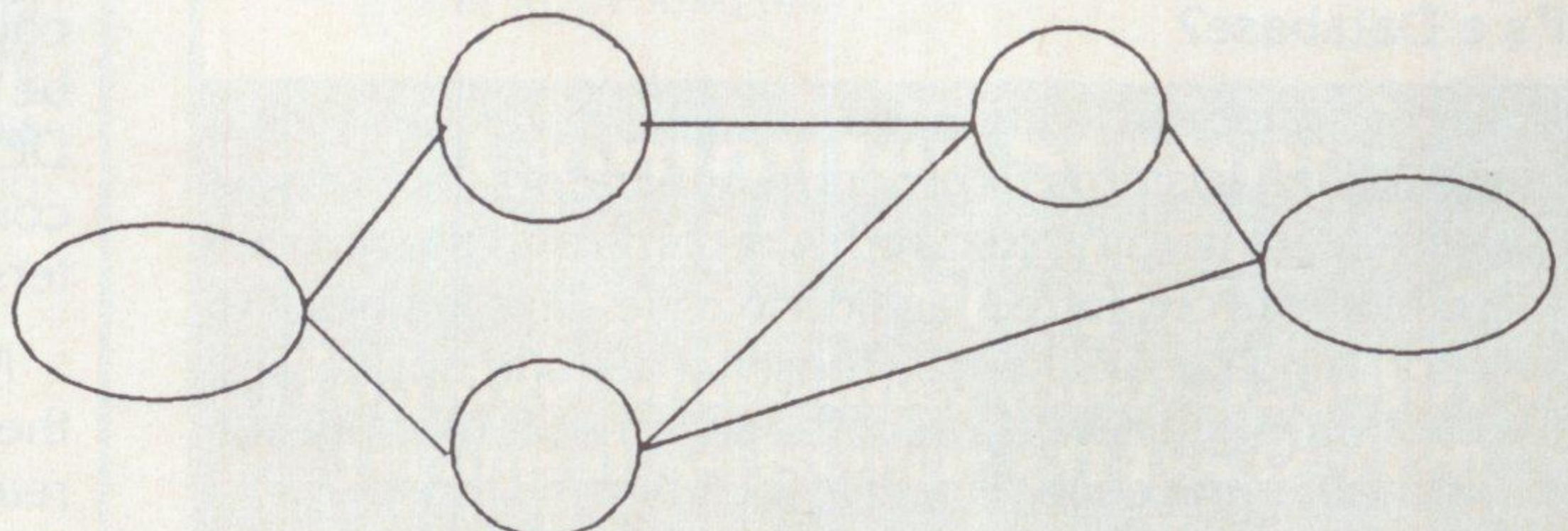


95\$ check or M.O. 15\$ for manual with demo disc that has all features except save, load, and print.

MDS Software Associates
1103 Horseshoe Lane
Blacksburg, Va 24060
(703) 435-4674

Softdraw is a professional illustration development system, ideal for making flow charts, schematics, and storyboard pictures. Draws lines, rectangles, ellipses, curves, and text. Edits by stretch, rotate, and magnify. Works in both color and B&W. Makes full 8 1/2 x 11 in. hi-res illustrations -- the screen is a window into a larger picture. Makes and saves templates of corporate logos and other frequently used symbols.

Prints on Epson, IDS 460, and Prism printers. Plots on the HP 7470A plotter. Requires 128K RAM, 2 disc drives.



Why Your Apple Needs DAVID-DOS™

Your Apple is easier to use and runs much faster with this new, licensed by Apple, DOS 3.3 update.

DAVID-DOS (rhymes with moss), is rated AA, (top of its class), by the highly respected software review magazine PEELINGS II.

Novices and Pros

DAVID-DOS is licensed by programmers for inclusion in the software they sell. 30 out of 60 bought it on the spot when DAVID-DOS was demonstrated at the Original Apple Corps, UCLA. DAVID-DOS receives applause from Apple novices. That's why you need DAVID-DOS.

What's Going On

Every Apple II/IIE boots up with DOS 3.3 software.* DOS 3.3 is right there on the first three tracks of all your program disks. Your Apple is supervised constantly by DOS 3.3.

Easier and 5 Times Faster

Our Installation Program goes in and makes specific permanent improvements to DOS 3.3 on all your disks in a few seconds. (The programs on your disks are not touched). From then on your programs load up to 5 times faster. And you have new features that make your Apple easier to use.

*Exceptions are Pascal & CPM Operating Systems. Apple II and Applesoft are trademarks of Apple Computer inc.

100 Sectors in 7 Seconds

Speed Load Applesoft, Integer & Binary 100 sector programs in 7 seconds. Tload Random and Sequential Text Files at the same speed with a simple command.

Automatic Card Support

Automatically supports an Integer or Applesoft Rom Card in Any slot (without configuring). Single key stroke Catalog and Catalog Abort. Shows Free Sectors Left on each Catalog.

Nine New DOS Commands

1. TLOAD speeds loads all Text Files.
2. TLIST lists all Random/Sequential Text Files.
3. DUMP Binary/Ascii to screen or printer.
4. DISA disassembles Binary to screen/printer.
5. AL prints program Address & Length.
6. / Single keystroke, second Catalog command.
7. HIDOS moves DOS to the Language Card.
8. FIND hex group in 64K memory in 3 seconds.
9. DATE prints out. Use with Mtn. clock card.

Commands 8 & 9 in Hidos only.

10K More Memory

These nine commands operate identical to existing DOS commands. Use A or A\$ for address and L or L\$ for length. Enter new commands on the Keyboard and use them in Basic programs with the familiar D\$. Use HIDOS in Hello program for turnkey startup, adding 10K free memory.

Variable Speed Scrolling

TLIST, DUMP, and DISA Features are:

1. Print with PR#.
2. Key operated variable speed control.
3. Instant pause with Space Bar.
4. Exit anytime with Return Key.

Compatible

All DOS entry addresses have been preserved. DOS is Same length and compatible with most software. David-Dos is copyable and creates fully copyable updated disks. Init areas were used for David-Dos. Works with all Apple IIs including IIE 80 Col. Requires 48K. Complete documentation for screen or printing and many utilities are on the disk.

California residents add 6½% sales tax.

\$39.95

Foreign add \$2.00.

To Order: Send Check To

12021 WILSHIRE BLVD., SUITE 212D
LOS ANGELES, CA 90025 (213) 478-7865

DAVID DATA™

THUNDERCLOCK PLUS™



Supported by
the best names in the business

Many of the best hardware and software products on the market today are compatible with Thunderclock Plus. By design. Because Thunderclock Plus makes them much more versatile.

Just plug Thunderclock Plus into your Apple® II, IIE or III and it can do any number of tasks automatically. In the office, the lab or at home. For instance, with business or communications software, your Apple can access a data base or send electronic mail when the rates are lowest.

Thunderclock Plus can also time and date stamp your disk files to the minute,* time experiments and even water your lawn.** But that's just the start. No matter how you use

your system now, it can work much more efficiently with Thunderclock Plus.

Thunderclock Plus comes with a one-year warranty. On-board batteries keep it running accurately for up to four years without battery replacement.

If you want the convenience and versatility of running your system around the clock, get the clock it's designed to run around. Thunderclock Plus. Supported by the best names in the business.

See your dealer for a demonstration or contact us.



THUNDERWARE, INC.
44 Hermosa Ave., Oakland, CA 94618
(415) 652-1737

®Apple is a registered trademark of Apple Computer, Inc.

*Requires Thunderware's Dos Dater™ Software.
**Sprinkler application requires BSR X-10® Home Control System and our Scheduler software.

INTRODUCING ULTRATERM™

YOU'LL LOVE THE VIEW!

Budget for 1983												Budget for 1984											
JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
2101	268	268	300	300	300	320	320	320	320	320	320	400	400	400	450	450	450	450	400	400	400	400	400
5175	100	100	100	100	100	150	150	150	150	150	150	200	200	200	200	200	200	200	250	250	250	250	250
5210	75	75	75	75	75	85	85	85	85	85	85	100	100	100	100	100	100	100	110	110	110	120	120
5225	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
5230	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
5250	125	125	125	125	125	125	125	125	125	125	125	0	0	0	0	0	0	0	0	0	0	0	0
10450	0	0	0	600	600	600	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11475	20	20	20	20	20	20	20	20	20	20	20	0	0	0	0	0	0	0	0	0	0	0	0
12400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Code listings: 100...199 rental & supplies.												300...399 misc. equip.											
400...499 misc. expenses.												600...699 unused.											
700...799 unused.												900...999 unused.											

Code listings: 100...199 rental & supplies. 200...299 Equipment rentals and purchases. 300...399 misc. equip.
400...499 misc. expenses. 500...599 2nd office supplies. 600...699 unused.
700...799 unused. 800...899 unused. 900...999 unused.

128 column VISICALC for use with your ULTRATERM card!

Coming this spring from VIDEX, software to use VISICALC with the ULTRATERM; now you can display four times as much of your worksheet with upper and lower case characters in your labels. You now can see more of your worksheets than ever before; a full 128 characters across by 32 lines down. Simply boot VISICALC via our ULTRATERM pre-boot disk and you're up and running. The ULTRATERM VISICALC pre-boot works with the Apple II, Apple IIe, Apple IIc and the Franklin.

With UltraTerm, the revolutionary new card from Videx, you'll enjoy sweeping panoramas of spreadsheets that you've never seen before: 128 columns by 32 lines, 132 columns by 24 lines and even 160 columns by 24 lines. You'll revel in the scenics of a whole year of records stretching out across your screen.

You'll also delight in the new horizon of 80 columns by 48 lines—double the lines you normally have. So your word processing will reveal a "depth of character" never possible before!

Another breath-taking view of UltraTerm—it delivers absolutely flicker-free, state of the art display, with 8 x 12 character matrix giving you preposterously clear, readable characters. Not only will you see **more** characters on your screen (a whopping 4096 possible), but they'll also be larger and more readable than the characters you read every day in your news-

paper! And you can differentiate those characters in several modes: normal (white on black), inverse (black on white), bright intensity and dim intensity.

UltraTerm features a built-in soft video switch and has complete firmware support for BASIC, Pascal and CP/M®. Use it with the Apple® II, Apple IIe, Apple III and Franklin.

UltraTerm. Come on over and enjoy the view.

Suggested retail price: \$379



897 N.W. Grant Avenue, Corvallis, OR 97330
503-758-0521

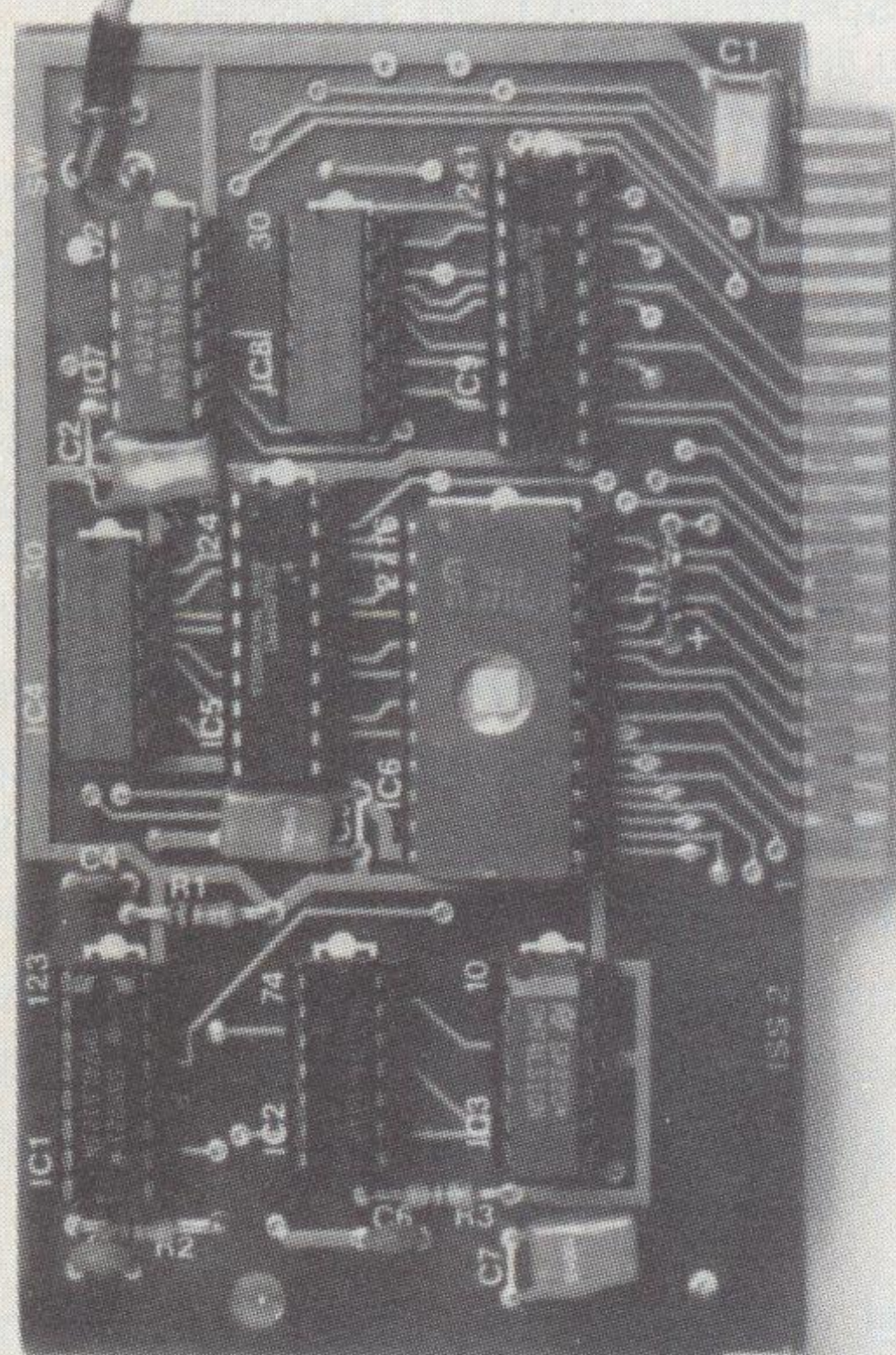
NEW FOR APPLE II AND IIe

WILDCARD™

MAKES BACK-UP COPIES OF PROTECTED SOFTWARE QUICKLY, EASILY, WITH JUST A PUSH OF A BUTTON.

New software locking schemes have rendered even the latest generation of copy programs virtually unusable. Locksmith™, Nibbles Away™ and other "Nibble copiers" require complicated parameter settings, much patience and great effort to use. More often than not, the results are disappointing. WILDCARD is different. Rather than copying disks track by track, WILDCARD ignores the disk and any copy protection encrypted on it. Instead, WILDCARD takes a snapshot of memory in your Apple® II.

Now you can make back-up copies of protected software with the push of a button.



FEATURES

- Hardware copying device... push button operation.
- Copies 48K memory resident software, most 64K software.
- No programming experience or parameters necessary.
- Backs up DOS 3.2 and DOS 3.3 disks.
- Creates DOS 3.3 unprotected and autobooting disks.
- WILDCARD lives in any slot. Undetectable by software.
- Produces autobooting disk in 2 minutes.
- Copies are DOS 3.3 compatible.
- Copies become accessible for alterations.
- Simple, easy-to-use software included.

WILDCARD Utility Disk 1 also included, featuring:

- Automatic program compression and BRUN file maker.
- Multiple programs can be placed on the same disk.
- Recreates basic files to load and save.
- Files can be placed on a hard disk...and more.

Software is not copy protected. System requirements: Apple II Plus with 64K and DOS 3.3 or Apple IIe. Franklin Ace also supported.

*Wildcard does not operate with CP/M® or other microprocessor based software.

WILDCARD **\$139.95**

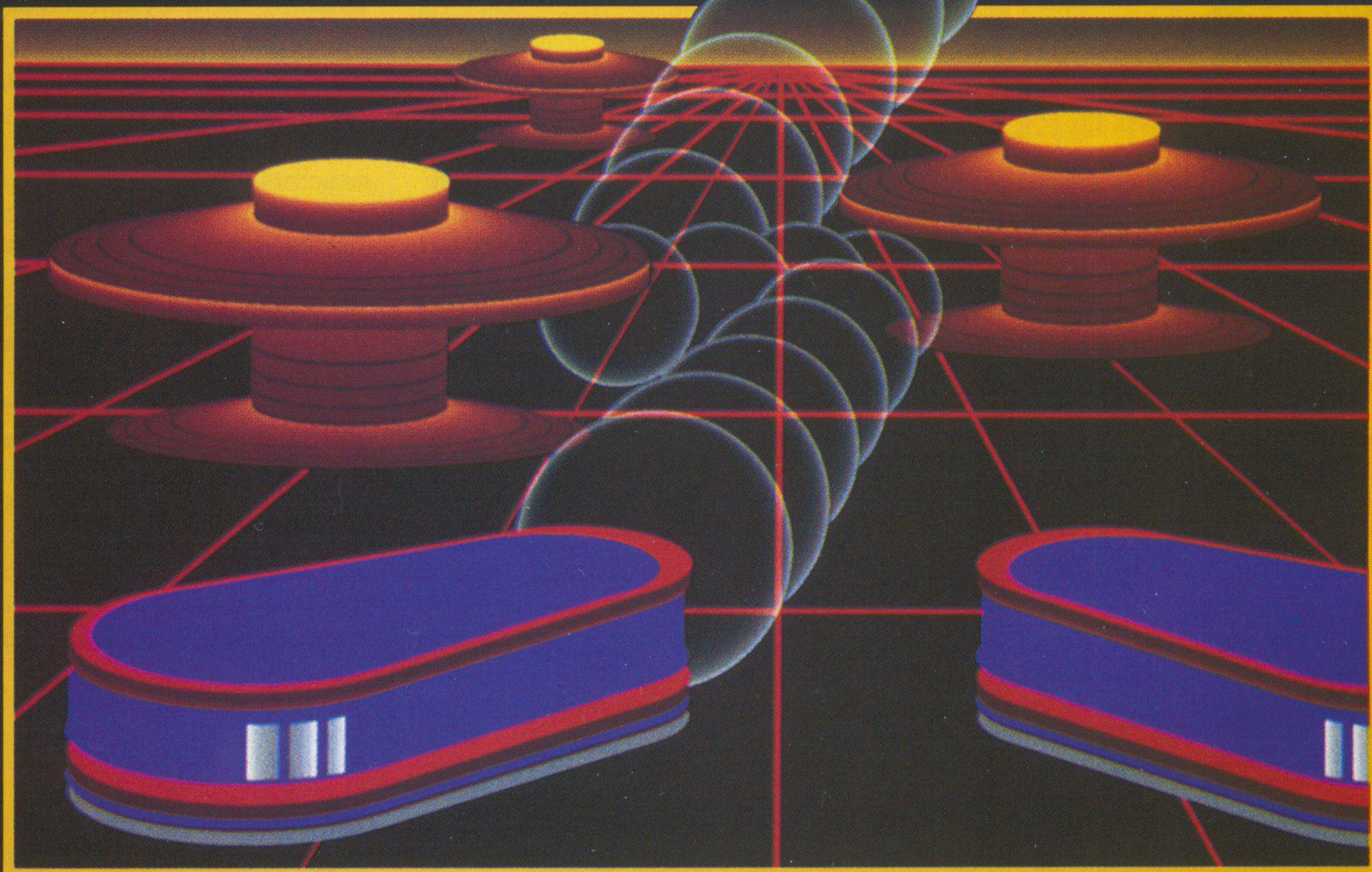
Order direct from East Side Software Co., 344 E. 63 St., Suite 14-A, New York City 10021, 212-355-2860. Please include \$3.00 for shipping and handling. Orders outside continental U.S. please add \$10.00 for shipping and handling. Mail and phone orders may be charged to MasterCard and VISA. N.Y. State residents add sales tax. Dealer inquiries welcome.

IMPORTANT NOTICE: The WILDCARD is offered for the purpose of enabling you to make archival copies only. Under the Copyright Law you, as the owner of a copy of a computer program, are entitled to make a new copy for archival purposes only and the WILDCARD will enable you to do so. The WILDCARD is offered for no other purpose and you are not permitted to utilize it for any other use, other than that specified.

Apple and the Apple logo are registered trademarks of Apple Computer, Inc.—CP/M—trademark of Digital Research, Inc. Locksmith—trademark of Omega Microwave, Inc. Nibbles Away—trademark of Computer: applications.

For Apple II,
Apple II+,
and Apple IIe

TAKE A BREAK!



WITH NIGHT MISSION **PINBALL**

You deserve the best. You've earned it. Now reward yourself with a session of **Night Mission PINBALL**, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put **Night Mission PINBALL** in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with **Night Mission PINBALL** from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.



See your dealer . . .

or write or call for more information. For direct orders please add \$1.50 for shipping and specify UPS or first class mail delivery. Illinois residents add 5% sales tax. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800/637-4983

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

Your Apple's telephone. Hayes



"Thanks for the prompt reply. Sure was a lot faster than waiting for the mail!"

"Gary: The pedigrees for next week's auction are as follows..."

"Attn. Prod., Sales, Purch.: Recommend 50% blue, 30% red screen for closest match!"

A complete plug-in communications system for Apple® computers. From Hayes, the established telecomputing leader: the simple but sophisticated Micromodem II™ plug-in board modem and its companion software, Smartcom I™. Everything you need to expand the world of your Apple II, IIe, II Plus and Apple III. In one, convenient communications package.

With Micromodem IIe and Smartcom I, you can access data bases, bulletin boards, and the varied resources of information services. Plan your travel itinerary via computer, including flight numbers, hotel and rental car reservations. Retrieve and analyze daily stock and options prices. Work at home and send reports to your office. You can even do your gift shopping by computer!

Micromodem IIe. Think of it as your Apple's telephone. It allows your computer to communicate with any Bell-103 type modem over ordinary telephone lines, at 110 or 300 bits per second. Micromodem IIe installs easily in an expansion slot, and requires no outside power source. It connects directly to either a single or multiline modular phone jack, to perform both Touch-Tone® and pulse dialing.

Micromodem IIe dials, answers and disconnects calls automatically. And, unlike some modems, it operates in full or half duplex, for compatibility with most time-sharing systems.

A built-in speaker lets you monitor your



calls when dialing. That way, you'll know if a line is busy. With Smartcom I, Micromodem IIe automatically redials your last number.

Discover how Micromodem IIe can help maximize the capabilities of your Apple. While Smartcom I software will minimize your efforts.

Smartcom I companion software. For effortless communications. Whether you're a newcomer to personal computing or a seasoned professional, you'll appreciate the ease and speed with which you can perform any communications function. Thanks to Smartcom I!

Let Smartcom I guide you through a few easy-to-answer questions to tailor the program to your particular needs. Then you're ready to go!

Make a selection from the Smartcom I menu to manage your communications, files or printer. Program prompts guide you along the way. And menu selections let

you easily make a call, end a call, or answer a call. When you're on the receiving end, your Micromodem IIe answers automatically, even if you're not there!

Convenient! And so is the Smartcom I memory for phone numbers. Smartcom I stores three of your most frequently called telephone numbers and one prefix. Plus, it also remembers the last number dialed.

Smartcom I also provides a directory of the files stored on your disk. And lets you create, list, name, send, receive, print or erase files right from its menu.

Smartcom I is as versatile as you need it to be. It accepts DOS 3.3, Pascal, CP/M™ 3.0 or CP/M Plus™ operating systems. And accommodates up to six disk drives and several printer interface cards.

Like all our products, Smartcom I and Micromodem IIe are backed by excellent documentation and full support. Including a two-year limited warranty on Micromodem IIe and a 90-day warranty on Smartcom I!

See your dealer today. Then plug into the exciting world of telecomputing.

Hayes Microcomputer Products, Inc.,
5923 Peachtree Industrial Blvd., Norcross,
Georgia 30092. 404/449-8791.



FCC approved in U.S.A.
©1983 Hayes Microcomputer Products, Inc.
Micromodem IIe and Smartcom I are trademarks of Hayes Microcomputer Products, Inc. Apple Computer is a registered trademark of Apple Computer, Inc. Touch-Tone is a registered trademark of American Telephone and Telegraph. CP/M is a trademark of Digital Research, Inc. CP/M Plus is a trademark of Advanced Logic Systems.