THE CP/M USER'S JOURNAL $2.95

Microsystems

Volume 4 / Number 8 August 1983

"While still a child, nor yet a fool to fame, I LISPed in numbers, for the numbers came."
—A. Pope

LISPing in Numbers

In the first of two articles, Bill Wong gives a brief overview of the Lisp language and reviews three implementations of LISP to run under CP/M: muLisp-80, Supersoft's Lisp, and Stiff Upper Lisp. The second article will contain further reviews.

CP/M Enhancements

Ian Ashdown's XERA (Extended Erase) utility reports filenames as it erases them and asks for confirmation on each file if wildcards are used in the command line.

Ralph Janelli describes how to provide a multiuser CP/M system with some degree of security by means of passwords and log-on/log-off procedures.

Michael Sasso shows how to prevent system crashes caused by booting a disk that has no system on it.

Word Processing

Bob Kowitt reviews "Fancy Font," a software package that provides many different fonts for an Epson MX-80 printer and also allows you to create your own fonts.

Ian Darwin's "50-Line Formatter" is 50 lines of C code that provide "bare bones" formatting of text with ragged right margin.

Jim Gilbreath describes patches to early versions of WordMaster to make the control codes conform more closely with those of WordStar, and also provides a mechanism for automatic dating of document creation and updates.

Peter Radatti shows you how to patch the TP/M operating system to run WordStar.

North Star and Other Topics

Tom Wiens describes enhancements to spreadsheet print programs that avoid awkward round-off errors and improve the placement of numbers in columns.

Doug Anderson and John Nash describe how to find the memory location of a variable in North Star Basic when you want to pass the address of data to a machine language subroutine, instead of passing a value to be processed.

Hardware Reviews

Bill Machrone reviews the Cromemco C-10 computer system, Dave Hardy

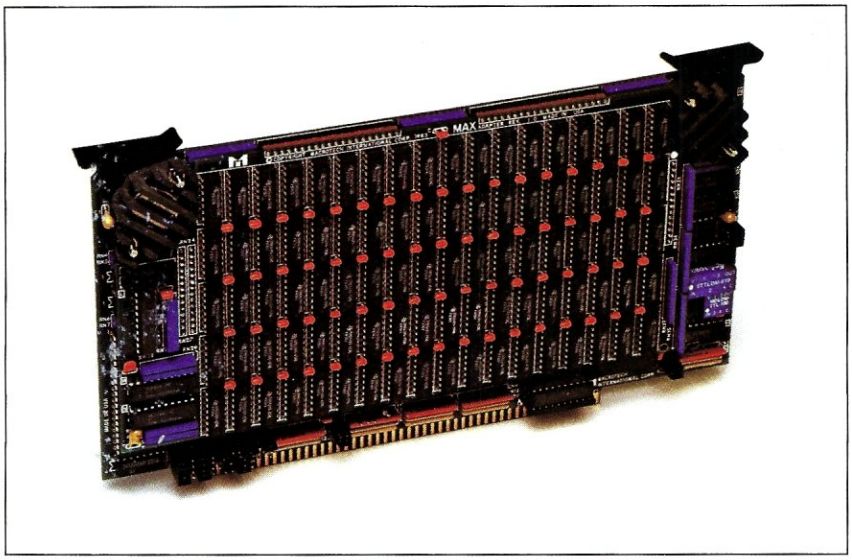
EXTRA

EXTRA

S-100 World News

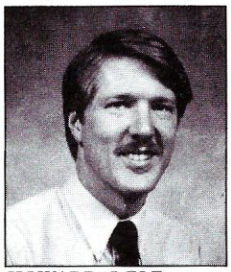
MACROTECH International Corporation 20630 Lassen Street, Chatsworth, California • 213-700-1501

NOW 1 MEGABYTE *MAX* FOR ALPHA MICRO



CHATSWORTH—June 30, 1983—Mike Pelkey, Macrotech International President, announced today that a special version of ***MAX*** is now running in Alpha Micro Systems.

This special version is available only through Soft Machines of Champaign, IL. (217) 351-7199. Howard Ogle of Soft Machines stated, "The new **AM-*MAX*1** runs full speed with all three Alpha S100 machines." Ogle also said, "The **AM-*MAX*1** is not only the most economical memory for Alpha, but the most versatile as well. The system is even faster with Soft Machines' 'GO FAST' disk cache utilities."



HOWARD OGLE

Bob Rubendunst of Soft Machines reports, "Every ***MAX*** is shipped with software that greatly simplifies implementation on bank switched systems. Also included are detailed installation instructions and diagnostic programs."

Dealer inquiries and orders should be directed to Bob at Soft Machines.

VIRTUAL DISK NOW NONVOLATILE

CHATSWORTH—June 30, 1983—Mike Pelkey announced today the release of the latest addition to the Macrotech product family. The **B-Board** is a multifunction system support board, for use with ***MAX*** and **128ST** memories. Used with the **128ST**, this combination creates a complete disk emulation, including nonvolatility. The **B-Board** features include battery backup, power fail monitor, and charging circuitry for on or off board batteries.

The **B-Board** functions also include a time-of-day clock, using a National Semi device for hassle free operation. It also gets early warning at power down, so the time-of-day can't suddenly get creative. An interrupt is available which can be used to turn the system on or off at a preset time.

On board ROM space accepts the users' EPROM based program storage. It can be configured to accept one or a pair of any EPROM type from 2716 to 27256, in 8 or 16 bit format. It supports a wake up jump option with full or shadowed phantom overlay.

The ERROR TRAP feature is designed to support the parity error detection feature of the ***MAX*** series dynamic memories. Any activity on the system's ERROR line causes the trap to record the extended address and data busses and 20 bits of bus status information. Up to 16 events can be trapped; the trap issues an interrupt when it's full.

The **B-Board** is a logical addition to the growing family of Macrotech International's no-compromise S100 boards for no-compromise users.

MAX Split Personality

BURBANK—June 30, 1983—"Many current operating systems permit ***MAX*** to double as both virtual disk and system memory," stated Dan West of Westcom Systems. As an example, an MP/M 2.1* system using ***MAX*-M** could be configured as a 512K system memory and a 512K Vdisk. A typical CP/M 3.0* configuration could be 256K of system memory and up to 768K Vdisk. CP/M 2.2* of course, only permits a 64K system memory, leaving the balance for a virtual disk. With ***MAX***, or the **128ST**, both functions can run simultaneously in a single memory board.

MACROTECH Moves

CHATSWORTH—June 30, 1983—Macrotech has moved to larger facilities located at 20630 Lassen St., Chatsworth, CA 91311. The new phone number is (213) 700-1501. "Due to a healthier marketplace and a

Virtual Disk for CP/M 86*

Dan West, Westcom Systems

BURBANK—June 30, 1983—Most of the CP/M 86* application programs available today fail to take advantage of the possible one megabyte address space. Virtual Disk for

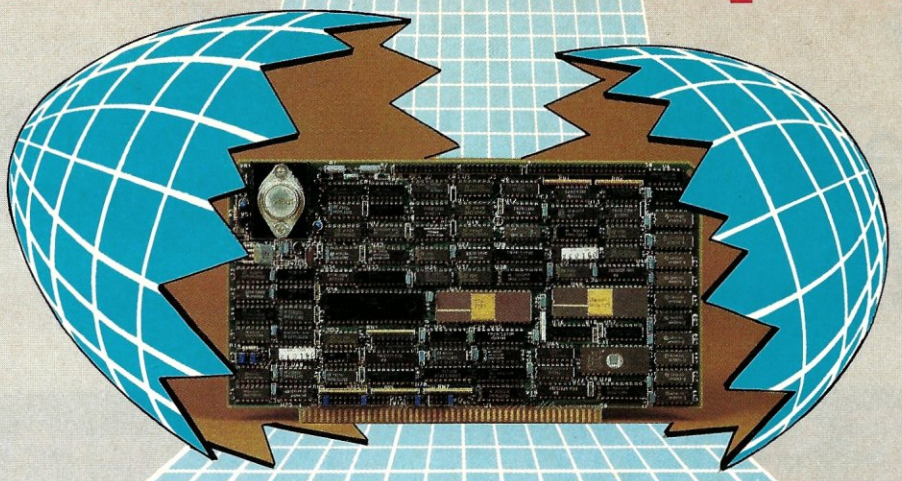
PRICE INDEX

	SIZE	P/N	PRICE
Static Memory	128K	128-ST	$1232
Dynamic Memory	256K	MAX-256	$1108
24-bit	384K	MAX-384	1292
Addressing	512K	MAX-512	1647
	640K	MAX-640	1737
	768K	MAX-768	1815
	896K	MAX-896	1899
	1M	MAX-M	1983

With 16-bit M³ Addressing option, add $91

Software (provided on 8" disk)

Incubation Complete



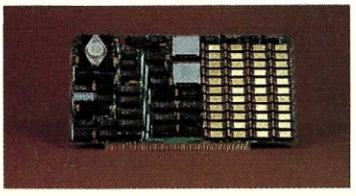
A Third Generation is Born

SBC 300

(Pictured above)

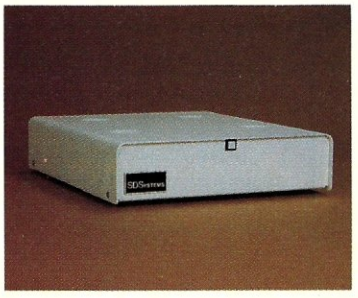
A Z80 based microcomputer board with memory and I/O functions

- ***Fully complies with IEEE 696 Standard***
- 4/6/8 MHz Z80 A/B/H*
- Supports CP/M® Plus
- Operates as bus master/slave for multi-user, multi-processor architecture.
- 64K on board memory, dual ported, parity checked
- 2 serial ports, 1 full SASI port
- All I/O drivers on board
- Memory management
- Full 24 bit address capability
- 3-16 bit CTC's



ExpandoRAM IV—Random access memory board utilizing 64K or 256K NMOS RAM chips

- ***Fully complies with IEEE 696 Standard***
- 256K capacity with 64K chips
- 1024K capacity with 256K chips
- Error checking and correction (2 bit detection, 1 bit correction)
- On board refresh
- Supports both 8 and 16 bit data transfers
- 24 bit addressing



SD300—A new series of compact yet expandable S-100 microcomputers.

- Compact size approximately 4" × 14" × 17"
- 6 Slot motherboard
- Rugged metal enclosure
- Supports up to 5 users

OEM Version: Designed for ease of integration and maximum flexibility

- Z80 CPU
- 256K RAM
- Versafloppy II with free CP/M Plus™

Discless Version: An ideal high performance system for disk intensive applications. Eliminates disk wait states for spread sheets, spelling checkers, and network operation. Utilizes SDSystems RAMDisc and ROMDisc modules.

VFW-3: A single board controller for floppy and Winchester disk drives:

• ***Fully complies with IEEE 696 Standard*** • Free copy of CP/M Plus™ included • Up to 4 floppies and three Winchester drives may be controlled by VFW-3 • Data transfers to and from board under DMA or programmed I/O control • Supports 24 bit address space.

CP/M Plus™: high performance single user operating system.

• CP/M® 2.2. compatible—no modification! • When used with SDSystems 256K memory board speeds are up to 7 times faster than CP/M® 2.2. • High performance file system • MP/M® II file password protection • Time and date stamps on files • Support for 1 to 16 banks of RAM • Support for 1 to 16 drives of up to 512 MB each • Easy to use system utilities with HELP facility • Powerful batch facility • Sophisticated programmer utilities.

RAMDisc 256: A solid state disk emulator that greatly increases system performance by eliminating disk waits in disk intensive applications. Excellent for spreadsheets, spelling checkers and software development.

• 256K capacity • 1 mb total bus capacity • CP/M® 2.2, PLUS™ compatible • I/O port addresses user selectable • Storage locations addressed by on board 20 bit counter • On board refresh.

ROMDisc 128: An EPROM board that replaces a floppy disk drive for the purposes of booting CP/M® and loading application programs.

• Provides non volatile, permanent storage of programs and data • Utilizes 2732 or 2764 EPROMS, (16 max) • 128K capacity per board • 512K system capacity • Use with SDSystems RAMDisc to configure a stand alone or network discless system • CP/M Plus™ available in eproms • Serial port provided.

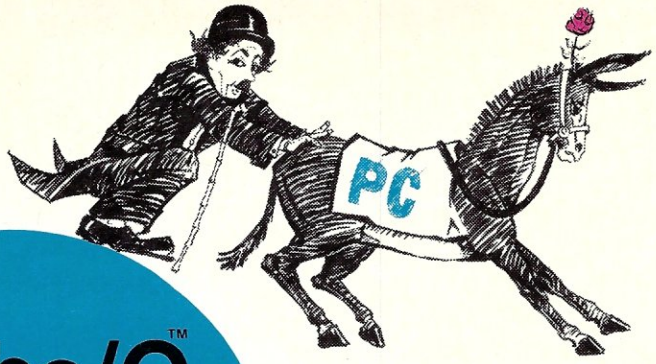
SDSYSTEMS

A Syntech Company

10111 Miller Road • Dallas, Texas • (214) 340-0303 • TLX-682 9016

CIRCLE 3 ON READER SERVICE CARD

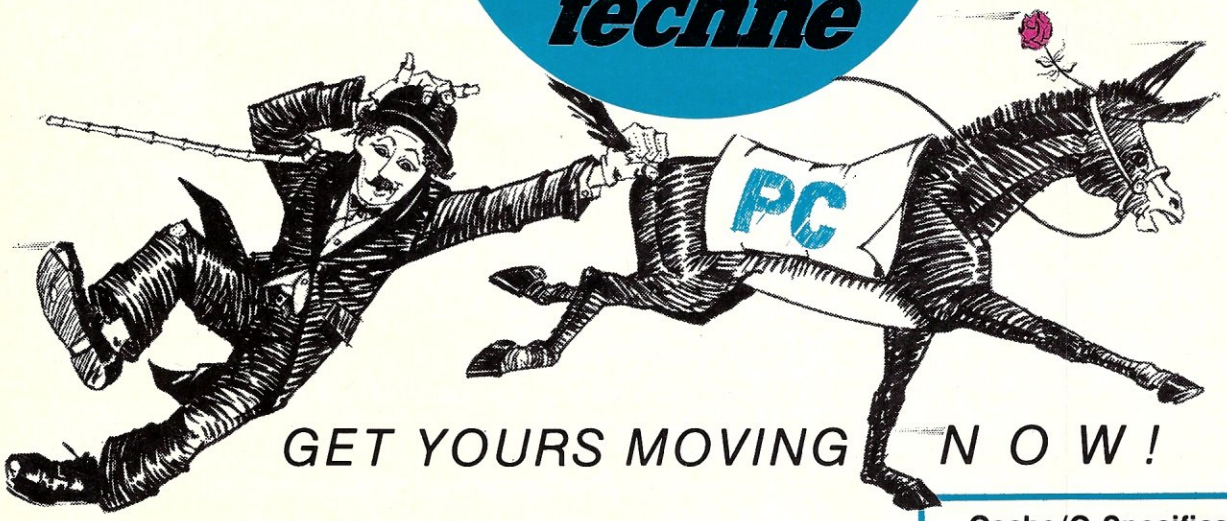
get it in gear with ....



Cache/Q™

THE Software Accelerator

by **techne'**



GET YOURS MOVING NOW!

Why wait longer, to buy it ... to try it?

- speeds up your programs, saves time and money
- faster dBase II, Wordstar and many more
- makes your computer investment really pay
- gets your computer in gear ... gets you in gear

Cache/Q features automatic memory management for your IBM PC providing:

Disk-buffering that can speed up your application programs up to four or more times the speed your are now getting.

Background print-buffering that will allow you unrestricted use of your computer with concurrent operation of up to five attached printers, each with an independent data stream.

Automatic write-thru to your disks protecting your working data from power failure.

Capability for specifying selective buffering of data according to disk-drive, file, even extension.

VERSATILE • FLEXIBLE • EFFICIENT • EFFECTIVE

Cache/Q Specifications

OPERATING SYSTEMS:
PC-DOS
(Also Available for CP/M 2.2)

BUFFERING TECHNIQUE:
Least Recently Used (LRU) algorithm controls disk-buffering; automatic disk/print buffering dynamically adjusted for highest efficiency.

BUFFERING CAPABILITY:
Utilizing up to 590K of memory. Cache/Q will operate with up to 16 attached mass storage devices, with sector sizes ranging from 128 bytes to 16K bytes and can operate up to 5 attached printers.

INSTALLATION:
No modification of existing programs, operating procedures, or hardware required. A simple interactive configuration program needs no special expertise to operate.

To order CALL NOW: (800) 835-9595. In Calif. (800) 835-9594——Dealer Inquiries Welcome

IN CANADA CONTACT: CYBERTECH TRADING CORP.
Unit 9, 11720 Voyageur Way • Richmond, B.C. V6X 3EI
Telephone (604) 278-4848 • Toll Free 1(800) 663-9767

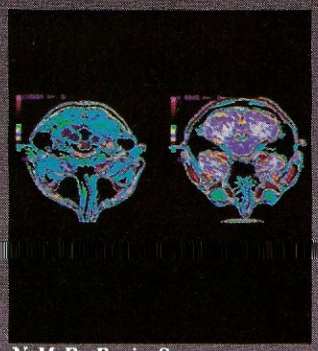
Techne' Software Corp. • 3685 Mt. Diablo Blvd., Suite 210 • Lafayette, CA 94549

techne'

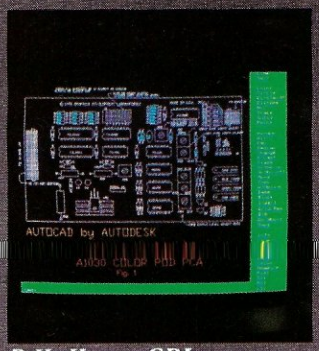
CIRCLE 159 ON READER SERVICE CARD

GDL's A-1000*™

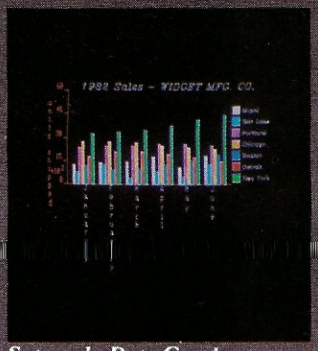
Why Pay More For Less?



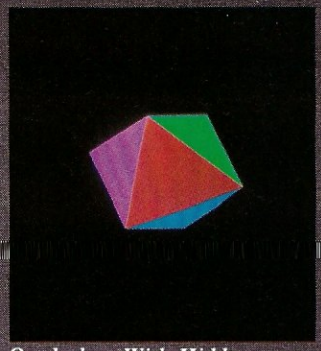
N.M.R. Brain Scan
J. Libove, Dual Systems



R.H. Hymes, GDL

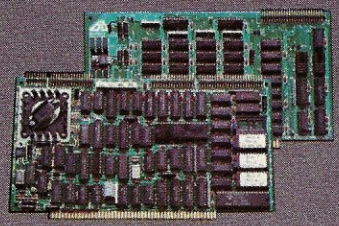


Supercalc Data Graph
Pacific Basin Graphics

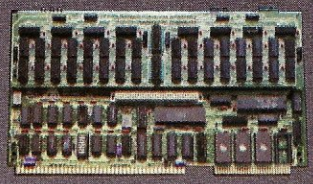


Octehedron With Hidden Surfaces Removed
Dr. C.L. Morgan, Calif. S.U. Hayward

- On-board 16 Bit Processor
- High-level Commands
- S100/696 or Multibus
- High Resolution
- Multiple Display Formats
- Simple Interface
- Extensive Software Support
- Download Capability
- Up to 16 out of 4096 Colors
- On-Board Self-Test



S100/696 A1000



Multibus A1000

Graphics Development Laboratories

has finally made high performance color graphics affordable. These S100/696 and Multibus compatible boards are currently at work in such diverse areas as Medical, CAD, Education, Scientific, and Stock Market Analysis. And it's easy to see why, with their on-board 16-bit 8088 processor and extensive firmware, they act as intelligent graphics sub-systems, relieving the host of time intensive graphics processing, thus maximizing system throughput. Display memory is completely isolated from the host's bus and all communications occur through I/O ports. This simple interface and the high level commands allow for quick intergration into any S100 or Multibus system.

Software Support

The A-1000 command set not only includes pixel and vector draws but also **Polygon Area Fills, 2D rotation, scaling, clipping, dither fills, terminal emulate mode, stroke and raster character sets, circles, windowing and viewporting**. A Microsoft compatible subroutine library and C driver are included with every A-1000, at no extra cost. A **PLOT 10** driver and **GIOS** driver for **GSX** are available.

The A-1000 is supported by extensive third party software including:

AUTOCAD by AUTODESK	A 2D drafting package for use by Architects, Engineers or anyone requiring high quality diagrams.
PBG100	Business software that graphs data from **Supercalc** and **DBASE II** files and keyboard input. Also **Hershey** character sets are available.
PBG100 Library	A **Calcomp** compatible subroutine library.
The Analyst	Stock market analysis.
Graftalk	Business graphics.
Ugraf	Business graphics.

OEM's and Systems Integrators

GDL offers generous discounts to OEM's starting in small quantities. So whether you are a small systems integrator or large OEM, you will find our pricing attractive.

Dealers

Attractive Dealer discounts are available on software/hardware or hardware only packages.

Reliability

All A-1000's must past rigorous functional burn-in process. The self-test command allows the user to verify system integrity and isolate most problems.

To find out more about the A-1000, call or write. **Dealer inquiries welcome.**

A-1000 is the trademark of graphics Development Laboratories, Analyst is the trademark of Kates Komputers, PBG100 is the trademark of Pacific Basin Graphics, Graftalk is the trademark of The Redding Group, UGRAF is the trademark of Transparent Data Systems. Supercalc is the trademark of Sorcim Inc. DBASE II is the trademark of Ashton Tate Inc. GSX is a trademark of Digital Research Inc. Autocad is a trademark of Autodesk Inc.

* Formerly called the **Aurora 1000, AURORA** is the trademark of Aurora Systems.

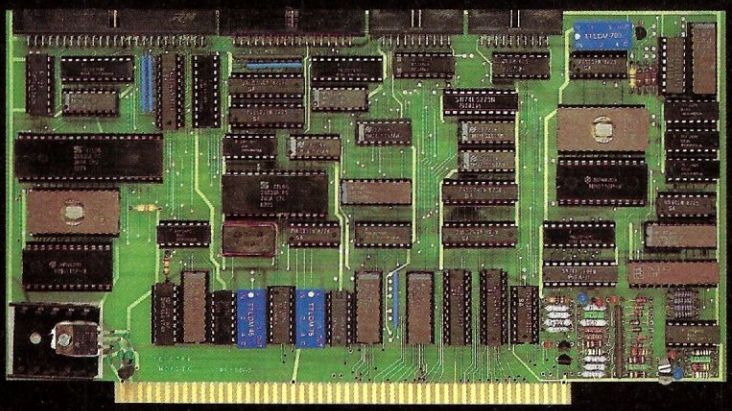
GRAPHICS DEVELOPMENT LABORATORIES

333 Cobalt Way Suite 106
Sunnyvale, Ca. 94086
(408) 980-7135

in San Diego Area contact:
Corky Deeds
Martian Technologies
8348 Center Dr. Suite F
La Mesa, Ca. 92041
(619) 464-2924

CIRCLE 24 ON READER SERVICE CARD

Bored Waiting? Here's The Board You've Been Waiting For.



A hard disk and cartridge tape controller together on one board? Magic? Not really. It's Teletek's HD/CTC. The hard disk and cartridge tape drive controller provide the support necessary to interface both rigid-disk drives and a cartridge tape deck to the S-100 bus.

- A Z-80A CPU (optionally Z-80B) providing intelligent control of the rigid-disk and cartridge tape drives.
- Support of 5¼" rigid-disk drives with transfer rates of 5 megabits per second. Minor changes of the on-board components allow the support of other drive types/sizes and transfer rates up to 15 megabits per second. (Interface to disk drive is defined by software/firmware on-board.)
- Controller communications with the host processor via 2K FIFO at any speed desirable (limited only by RAM access time) for a data block transfer. Thus the controller does not constrain the host processor in any manner.
- Two 28-pin sockets allowing the use of up to 16K bytes of on-board EPROM and up to 8K bytes of on-board RAM.
- Individual software reset capability.
- Conforms to the proposed IEEE-696 S-100 standard.
- Controller can accommodate two rigid-disk drives and one cartridge tape drive. Expansion is made possible with an external card.

Teletek's HD/CTC Offers A Hard Disk Controller, Plus Cartridge Tape Controller, All On One Board.

TELETEK

9767F Business Park Drive Sacramento, CA 95827 (916) 361-1777 Telex #4991834 Answer back-Teletek

CIRCLE 220 ON READER SERVICE CARD

Contents

Microsystems

Volume 4/Number 8
August 1983

DEPARTMENTS

Microsystems

STAFF

Sol Libes	editor
Chris Terry	technical editor
Ian Darwin/Dave Fiedler/Dave Hardy/Bill Machrone/ Ernest E. Mau/Bruce Ratoff/Anthony Skjellum	contributing editors
Andrew Bender/David Gewirtz/Fred Gohlke/ Steve Leibson/Don Libes/Randy Reitz	assisting editors
Ann Ovodow	editorial coordinator
Tom Leander	editorial assistant
Mariano Nicieza	art editor
Jim Beloff	advertising manager

ADVERTISING SALES OFFICES

New England, Midatlantic
Jim Beloff, *Microsystems*
Ziff-Davis Publishing Company
One Park Avenue
New York, NY 10016
(212) 725-3452

Advertising Coordinator
Rosemarie Caruso, *Microsystems*
Ziff-Davis Publishing Company
One Park Avenue
New York, NY 10016
(212) 725-5386

Southeast
Mark Browning, *Browning Publications*
P.O. Box 81306
Atlanta, GA 30366
(404) 455-3430

Midwest
Jeff Edman, *The Pattis Group*
4761 W. Touhy Avenue
Lincolnwood, IL 60646
(312) 679-1100

Southern California, Southwest
Barbara Farkas,
Ziff-Davis Publishing
3460 Wilshire Blvd.
Los Angeles, CA 90010
(213) 387-2100

Northern California, Northwest
Jeff Cohen,
Ziff-Davis Publishing
3030 Bridgeway
Sausalito, CA 94965
(415) 331-7133

Canada
Frank Lederer, *The Pattis Group*
1623 Yonge St.
Toronto, Ontario M4T241
(416) 482-6288

Direct Retail Sales
Lynn Kujawa, *Ziff-Davis Publishing*
One Park Ave.
New York, NY 10016
(212) 725-7679

CONSUMER COMPUTER & ELECTRONICS DIVISION

President	Larry Sporn
Vice President Marketing	J. Scott Briggs
Vice President Circulation	Carole Mandel
Vice President General Manager	Eileen G. Markowitz
Creative Director	Peter J. Blank

ZIFF-DAVIS PUBLISHING

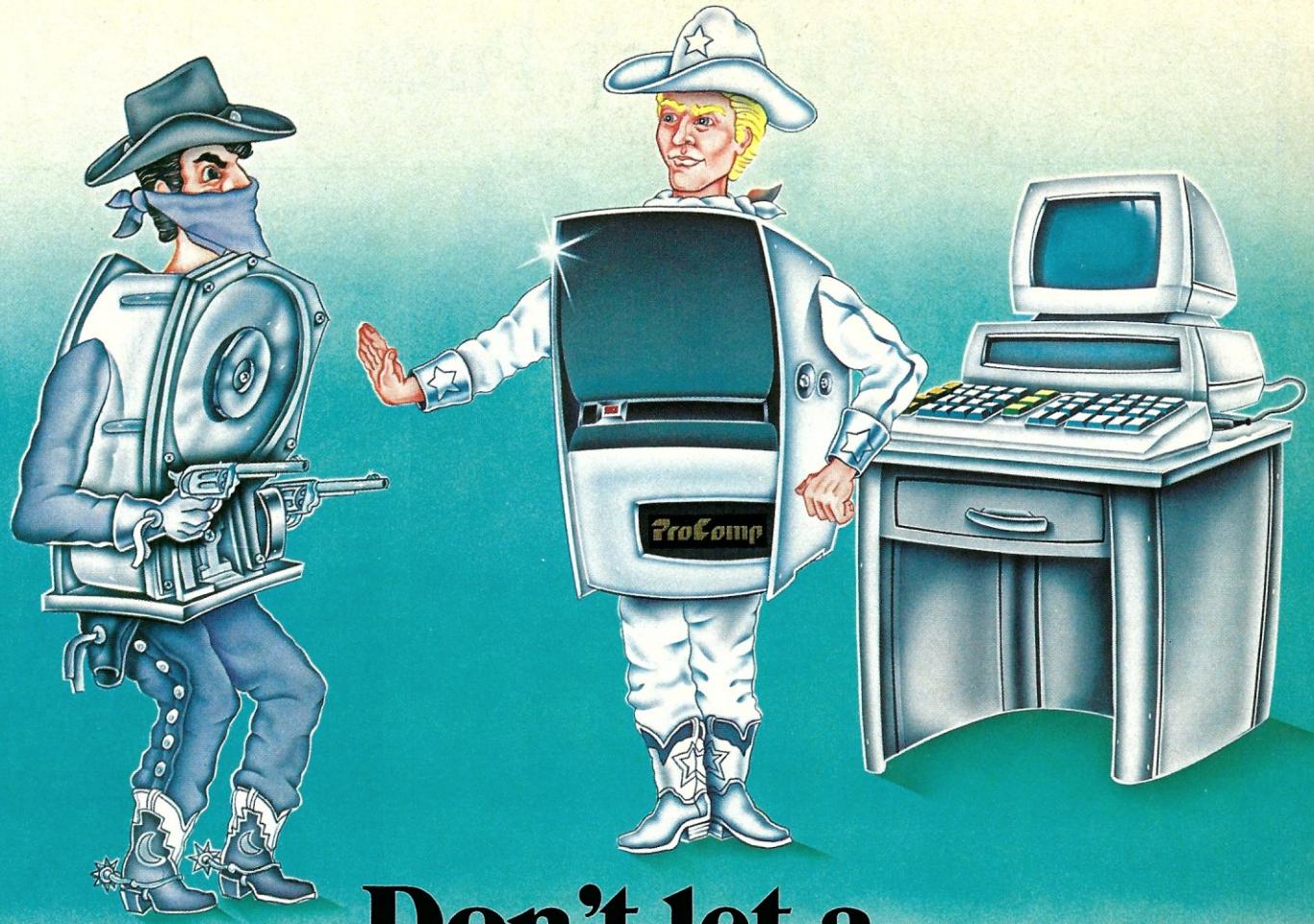
Richard P. Friese	**President**
Albert S. Traina	**President, Consumer Magazine Division**
Furman Hebb	**Executive Vice President**
Paul H. Chook	**Executive Vice President, Marketing and Circulation**
Phillip T. Heffernan	**Senior Vice President**
Sidney Holtz	**Senior Vice President**
Edward D. Muhlfeld	**Senior Vice President**
Philip Sine	**Senior Vice President**
Baird Davis	**Vice President**
George Morrissey	**Vice President**
Selwyn Taubman	**Treasurer**
Bertram A. Abrams	**Secretary**

MICROSYSTEMS (ISSN #0199-7955) is published monthly by Ahl Computing, Inc., a subsidiary of Ziff-Davis Publishing Company, One Park Avenue, New York, N.Y., 10016. David Ahl, President; Elizabeth B. Staples, Vice-President; Selwyn Taubman, Treasurer; Bertram A. Abrams, Secretary.

Second Class postage paid at New York, NY 10016 and at additional mailing offices. POSTMASTER: Send address changes to MICROSYSTEMS, PO Box 1987, Morristown, NJ 07960. Subscriptions are $26.97 for 12 issues. Canadian prices are $5.00 per year additional; other foreign $8.00 per year additional (U.S. currency only). For information or questions about subscriptions phone: (800) 631-8112.

Editorial correspondence is welcomed and should be sent to: MICROSYSTEMS, One Park Avenue, New York, NY 10016. Phone: (212) 725-6856.

For information on commercial advertising, write to: MICROSYSTEMS Advertising Dep't., One Park Avenue, New York, NY 10016, or call Jim Beloff at (212) 725-3452.



Don't let a hard disk hold up your computer!

A hard disk can quite literally hold up your computer system. Lower cost hard disk subsystems do not have the capacity or performance of modern software and operation systems especially on multiuser systems where the disk subsystem is the most heavily used part of your computer.

Others will try to tell you that all hard disk drives are the same. Don't believe it. Average access times vary by over 800% between ProComp disk drives and the slowest drives and by 200% between ours and our competition who claims to make fast subsystems.

Don't be held up by an unreliable or fragile disk.

For reliability ProComp's high speed subsystems use the latest technology thin film plated media which eliminates the dangers of a fatal head crash through power failure or a dead drive through shipping damage or even someone dropping the computer. Our drives use direct drive motors and thermal compensation so that they give you the same data on hot days that they give you on cold ones. We take the same care with our cabinets; oversize power supplies and fans, and sturdy steel construction create the most reliable hard disk you can buy.

Don't be held up by a disk that is too small.

ProComp subsystems support up to 31MB (unformatted) per drive. These drives are available NOW, not next month or six months from now, with bigger and faster drives to follow.

And most importantly, don't be held up by a disk that is difficult to use.

ProComp has more experience writing CP/M disk software than anyone. Our software can be installed by even the most novice user in less than 10 minutes and provides all the features that you could ever want.

So don't get held up by a hard disk, call us — FAST.

ProComp systems inc.

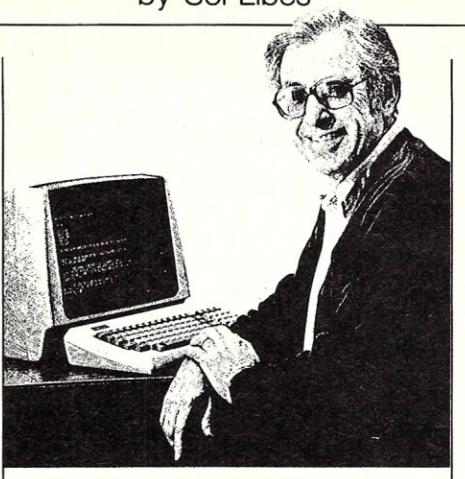
333 West 52nd Street, New York, NY 10019 (212) 246-0074

Dealer Inquiries Invited

CIRCLE 38 ON READER SERVICE CARD

Editor's Page

by Sol Libes



Microcomputer software has already gone through many generations, whether we realize it or not. In the early days of microcomputers (the early '70s), microcomputer software was very primitive because micros were almost exclusively thought of as process controllers, not as general-purpose computers. Thus, software was developed on either a mini or large computer, using a high-level development language (e.g., Intel's PL/M) and a simulator for debugging. Then the program was compiled and put on paper tape or into a PROM or ePROM. If paper tape was used, the micro contained a simple paper tape interface and a "load and go" loader program either on tape or in ROM.

With the advent of the 8008 microprocessor, Intel introduced a program called a "monitor" or "executive," which could be loaded into memory via tape or ROM and could be used to write and debug programs on the microcomputer using hexadecimal notation. Often a hexadecimal keypad was the only means for the user to talk to the computer. Many micro system designers actually wrote and debugged their programs using this technique. Some small high-level interpretive languages were actually developed this way to run on the 8008.

The introduction of the new 8080 chip into the Altair 8800, in early 1975, started people thinking that micros could be used for applications other than process control. Thus, in late 1975 MITS (the producer of the Altair) introduced a Basic interpreter written by Bill Gates and Paul Allen, who later decided to market the interpreter on their own and formed Microsoft. Also in 1975, Processor Technology Co., a manufacturer of S-100 compatible boards, put into the public domain an editor/assembler/debugger program that took up 4K of RAM (or ROM) and now allowed users to develop programs on their micros.

However, it was in mid-1976, when IMSAI (a MITS competitor) introduced floppy disk hardware along with a disk operating system called CP/M, that we entered the next generation of micro software. In as little as 16K of RAM a user had a very powerful software development system that enabled him to develop both application and further system software right on the micro itself.

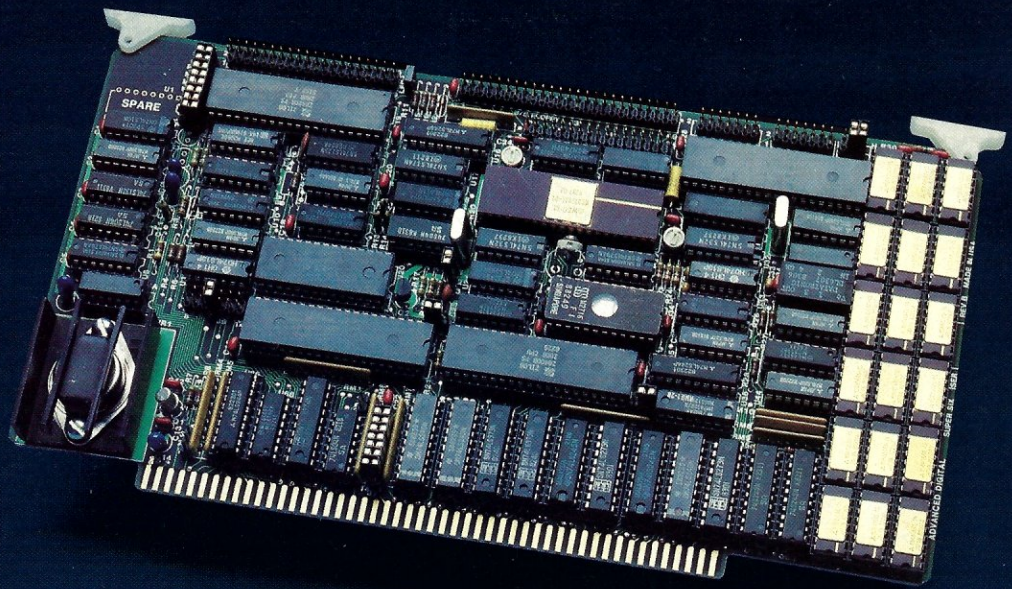
Memory sizes increased dramatically to accommodate larger and more powerful programs. Assemblers and debuggers got more powerful and convenient to use. More languages were written, so that by 1978 we had a large number of utilities and software development tools which made possible the introduction of powerful applications software. 1978 saw the introduction of the first word processor, called "Electric Pencil," and the following year brought us "WordStar," the all-time word processing best seller. This was the third generation of micro software.

This generation was also marked by many applications packages in the area of database management, spreadsheets, graphics, and communications. All these were made possible by the availability of larger memory size and the introduction of even more powerful software development tools such as relocatable assemblers, symbolic debuggers and high-level development languages such as Pascal, C, and Forth, and their associated libraries of subroutines.

Last year marked the beginning of what I consider to be the fourth generation of microcomputer software—the generation that will finally move the computer into the office and home on a grand scale. This generation is distinguished by the introduction of integrated software packages that put many applications immediately at the beck and call of the user and allow data and text to be easily moved back and forth between applications. The best example at present is the new Tandy 100 portable computer, which has an integrated software package in ROM that can do word processing, manage a small data base, do spreadsheet analysis, communicate with other computer systems, and allow the user to write and execute Basic programs. And data and files created can be easily moved between applications. There is no doubt that this type of software is the wave of the future for most computers.

Further, the increase in power of microcomputers (more memory and 16-bit processing) is leading to the use of micros in areas that have traditionally been those of the mainframes. The latest such area to be transferred to micros is that of Artifical Intelligence (AI). The application of AI techniques will lead to the creation of "expert systems" in which we have a knowledge base which programs can use to solve problems for the user rather than require that the user define how to solve the problems. The primary language of the AI community is LISP, which is now available on both 8-bit and 16-bit micros. Therefore, in this issue we begin a two-part series on LISP and the LISP packages that run under CP/M.

SUPER SIX™

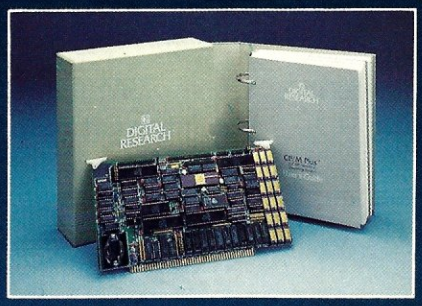


SUPER SIX, THE FIRST 6MHz S-100 SINGLE BOARD COMPUTER TO SUPPORT BANKED CP/M™ 3.0

SUPER SIX FEATURES:

- 128 KB of Bank — selectable RAM
- 6 MHz, Z-80B CPU
- DMA Controller
- 6 MHz, Z-80B DART (2 Serial RS-232 Ports)
- 6 MHz, Z-80B PIO (2 Parallel Ports)
- 6 MHz, Z-80B CTC (Clock Timer)
- Double/Single Density Floppy Disk Controller — Supports 8″ and 5-1/4″ Drives Simultaneously
- 2/4 KB of Monitor EPROM
- S-100, IEEE 696 Compatible

*CP/M is a trademark of Digital Research Corp. TurboDOS is a trademark of Software 2000 Inc. SUPER SIX and SUPER SLAVE are trademarks of Advanced Digital Corp.

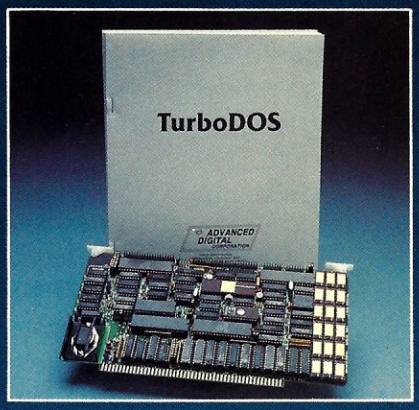


SUPER SIX & CP/M™ 3.0 A PERFECT MATCH

Advanced Digital has found the perfect match to its powerful, high-speed SUPER SIX single board computer. It's Digital Research's new CP/M 3.0. Because of SUPER SIX's 128 KB of RAM, it is the only S-100 board to support CP/M 3.0 in the banked mode; or run CP/M 2.2 with 64 KB of extra buffer.

SUPER SIX & TurboDOS™ ANOTHER PERFECT MATCH

When you combine the TurboDOS multi-user operating system with the 6 MHz SUPER SIX, you'll find your system running 1-1/2 times faster than



before. Add the 4 MHz or 6 MHz SUPER SLAVE™ processor board(s) and you will have the fastest multi-user, multi-processor system available today.

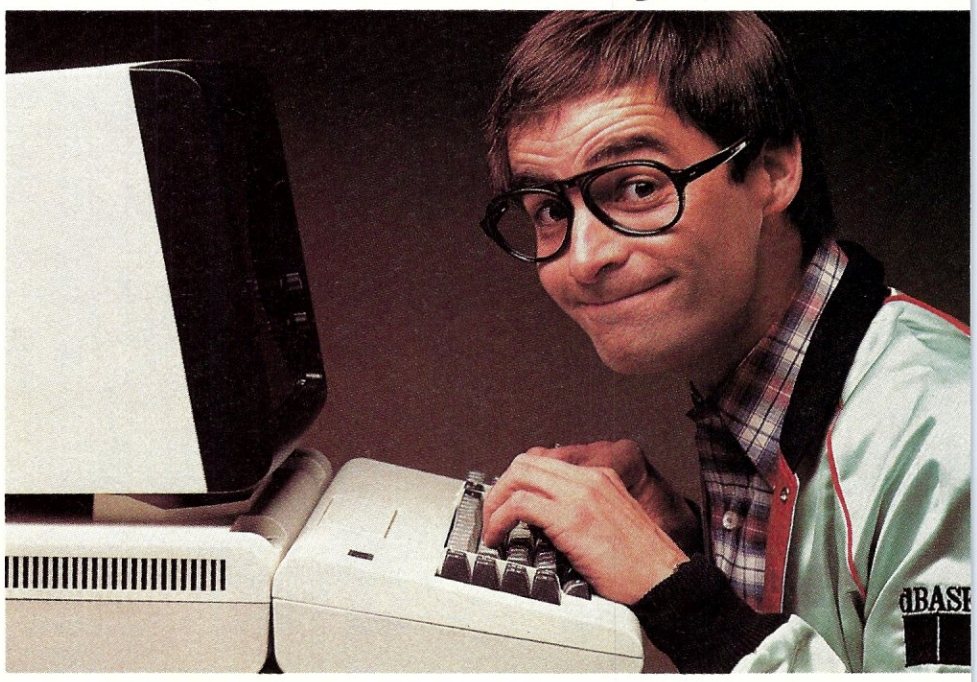
See the SUPER SIX at your quality computer dealer or contact:

ADVANCED DIGITAL CORPORATION

5432 Production Dr.,
Huntington Beach, CA 92649
Phone: (714) 891-4004
Telex: 678401 TAB IRIN

CIRCLE 148 ON READER SERVICE CARD

All you dBASE II™ hotshots are about to get what you deserve.



You've written all those slick dBASE II programs.

Business and personal programs. Scientific and educational applications. Packages for just about every conceivable information handling need.

And everybody who sees them loves them because they're so powerful, friendly and easy to use.

But that's just not good enough.

Uh-uh.

Because now you can get the gold and the glory that you really deserve.

Here's how.

We've just released our dBASE II RunTime™ application development module.

And it can turn you into an instant software publisher.

The RunTime module condenses and encodes your source files, protecting your special insights and techniques, so you can sell your code without giving the show away.

RunTime also protects your margins and improves your price position in the marketplace. If your client has dBASE II, all he needs is your encoded application. If not, all you need to install your application is the much less expensive RunTime module.

We'll tell the world.

With your license for the dBASE II RunTime module, we provide labels that identify your program as a dBASE II application, and you get the benefit of all the dBASE II marketing efforts.

We'll also provide additional "how to" information to get you off and running as a software publisher sooner.

And we'll make your products part of our Marketing Referral Service. Besides putting you on our referral hotline, we'll publish your program descriptions and contact information in *dBASE II Applied*, a directory now in computer stores world-wide.

Go for it.

But we can't do any of this until we hear from you.

For details, write RunTime Applications Development, Ashton-Tate, 10150 West Jefferson Boulevard, Culver City, CA 90230.

Or better yet, just call (213) 204-5570. And get what you deserve today.

dBASE™ II

ASHTON·TATE

CIRCLE 13 ON READER SERVICE CARD

Editor's Page
continued . . .

COMING IN FUTURE ISSUES OF MICROSYSTEMS

SEPTEMBER - UNIX ON MICROCOMPUTERS
A second look at Unix-like hardware and software for micros

OCTOBER - LOCAL AREA NETWORKING
The in's and out's of LANs, how they are implemented, and a review of CP/NET.

NOVEMBER - 16-BIT MICROCOMPUTERS
Microsystems looks at the new super 16-bit processor chips and some of the computers that use them.

DECEMBER - CP/M SOFTWARE DIRECTORY
The most complete directory of CP/M-based software ever published in a magazine.

News & Views

by Sol Libes

Random rumors

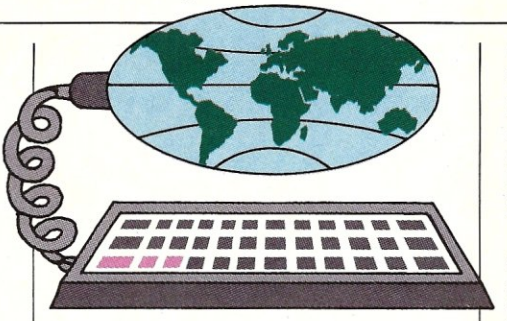
It is rumored that Microsoft began beta testing version 3 of MS-DOS in June, with release expected in September or October. It is expected to fix some of the bugs (features?) in version 2 and implement a multiprogramming (a la Concurrent CP/M) multiuser system (a sort of stripped-down Xenix). . . . Digital Research, ever ready to do battle, is expected to shortly announce an upgraded version of CP/M-86 that includes many of the features recently incorporated in CP/M Plus. . . . Digital Research is also rumored to be doing the operating system for the under $3,000 Lisa-like machine in development at Commodore Business Machines. The machine is expected to be offically announced at Comdex this fall.

Public domain software news

The SIG/M subgroup of the Amateur Computer Group of New Jersey has released six new volumes of public domain software. This brings their total up to 118 volumes. SIG/M has also released a new edition of their printed catalog, which lists all their software up through volume 118. The cost is $2 domestic ($2.50 foreign). Write to SIG/M, Box 97, Iselin NJ 08830.

The contents of the new volumes are the following:

Volume	Contents
113	Kermit—Communications package
114	Pistol—Language in the footsteps of Forth and Stoic
115	Miscellaneous utility programs
116	Forth, ZCPR2 upgrades, and miscellaneous utilities
117	Macro library, catalog program update, and Modem-7 overlays
118	XLisp—A complete Lisp with message-passing features



The CPMUG group did not release any new software. However, they have raised the price for their software. Charges are now $13 per 8″ disk volume domestic ($18 foreign), and $17 per 5.25″ North Star and Apple disk volume domestic ($21 foreign). CPMUG, 1651 Third Avenue, NY NY 10028; (212) 860-0300.

In order to speed up the availability of public domain software and reduce the cost of CPMUG volumes, a commercial service is offering SIG/M and CPMUG software. Catalogs of SIG/M and CPMUG public domain software ($4) and IBM-PC public domain software ($1) are available. Write: Turner Systems, 362 Lincoln Dr., Kenilworth NJ 07033; (201) 276-1638.

UNIX news

Western Electric has announced an agreement with Intel, Motorola, National Semiconductor, and Zilog, whereby the latter four companies will offer UNIX Version V on their systems and Western Electric will provide support to the companies.

Forty-five new UNIX systems were shown at the National Computer Conference held in May. Of these 45, 80% were based on the Motorola 68000 microprocessor.

The number of UNIX systems in operation may double in one fell swoop with Tandy's decision to supply Microsoft's XENIX to all past, present, and future purchasers of the Tandy 16 Z80/68000 dual-processor system. It is estimated that Tandy has already sold 15,000 Model 16 systems. The question now is, what percentage of the systems are multiuser systems that will run XENIX? Until now Model 16 owners could only run TRS-DOS on the Z80 and had nothing to run on the 68000.

Also, Microsoft, unable to convince Apple to sell XENIX to purchasers of the 68000-based Lisa, has decided to sell it direct through dealers. Regretfully, Microsoft will not be selling a generic version for those who wish to their own implementation. Now, if some enterprising hacker could come up with an implementation package.

Radio Shack news

Tandy Corp. has introduced the new TRS-80 Model 4 with 80 x 24 display; CP/M Plus is available as an option. A two-disk version with 64K of RAM (expandable to 128K) is $2000. This includes RS-232 and parallel printer interfaces, plus a new TRS-DOS (Version 6) operating system. The new DOS includes features such as: memorydisk, JCL, type-ahead, I/O redirection, print spooler, modem support, and forms tailoring.

Aton International Inc., 260 Brooklyn Ave., San Jose CA 95128 is offering a software product to allow users of the Radio Shack TRS-80 Model II 12- and 15-multiuser systems in multiuser configurations to run both CP/M and XENIX on the same hard disk.

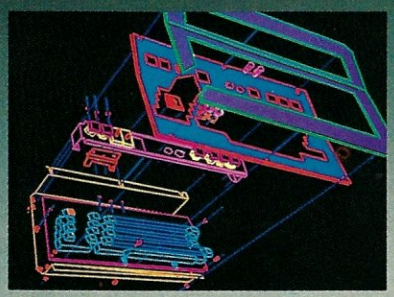
DRI news

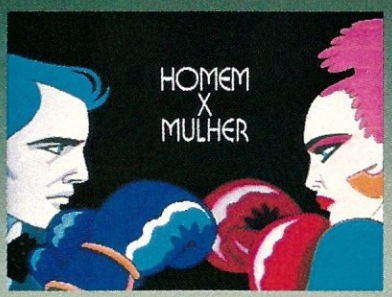
Digital Research Inc. has announced an agreement with VisiCorp to adapt Visi^{ON} software for CP/M systems. Visi^{ON} is an interactive, graphics-oriented software environment that provides an Apple Lisa-like environment (e.g., windows and mouse).

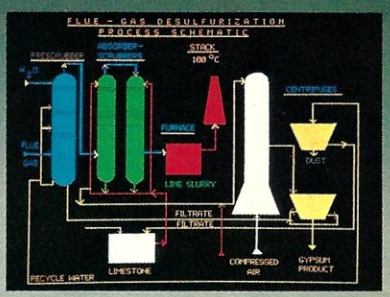
As yet Visi^{ON} has not been delivered to customers and is not expected to become available until October. It has been demonstrated at shows such as Comdex and CP/M '83, run-

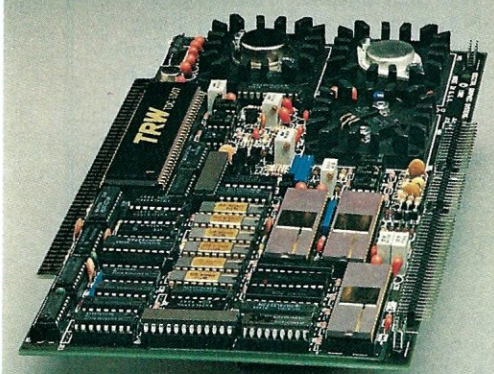
The Image Solution

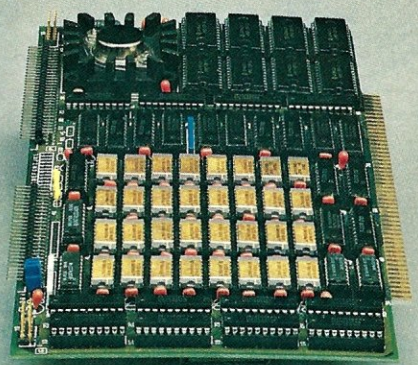
The CAT 1600 Series lets you take full advantage of your color graphics potential.

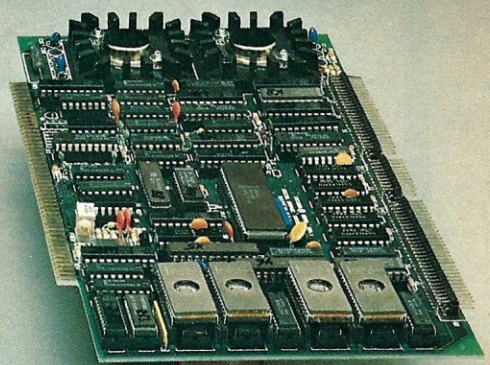












Plug this powerful color video graphic system into your IEEE-696 bus and watch your computer open its eyes. Exercise your creativity developing new ways to study your world and discovering the flexibility of video imaging. Our real time frame grabber gives you instant availability of the image to be processed. The CAT 1600 is the creative link between machine, man and the world.

Resolution is the name of the game, and we've got it. Physically you're looking at 512 × 512 pixels up to 24 bits deep. And that's real color. Now, center on a pixel, any pixel, and roam the screen through an image space as large as 1K × 2K. Zoom in and explore a close-up of 32:1, not in the usual quantum leaps of integer zooms, but in smooth logarithmic steps of 1.1%. A smooth zoom... that's human engineering.

At the heart of the matter is a dedicated 8086 image processor. It blazes a 16 bit wide path through the various memories, lookup tables and image parameters as it executes high level commands from your host processor. Up to 48K of static RAM makes the image processor useful for downloading custom programs from the host.

When it came to adequate memory, we didn't forget. 768KB of dynamic memory gives you plenty of image. Our PROMs have a library of 64K organized into over 130 sophisticated graphics commands such as continuous live digitization, character and shape generation, global image manipulation and animation effects, to relieve the host computer from low level primitives.

At your disposal is a palette of 16.7 million colors and 256 shades of gray. Quantized lines or free-hand sketching completes the picture. Use a variety of pen widths, brush strokes, even airbrush! Now imagine what you can do with a superb quality image captured in real time from a color video camera. Contact us for an eye opening demonstration: 935 Industrial Avenue, Palo Alto, CA 94303. 415/856-2500.

DIGITAL GRAPHIC SYSTEMS, INC.

What You See Is What You Get.

CIRCLE 249 ON READER SERVICE CARD

ning on the IBM-PC.

DRI is expending much effort on making application programs portable between machines. this involves unifying the various runtime packages for each high-level language.

DRI has also announced a new Basic interpreter for 16-bit systems, called "Personal Basic", which is compatible with Microsoft MBasic. The language offers enhancements, such as syntax checking during statement entry, and debugging aids, such as variable tracing, breakpoints, and single-step operation.

DRI has also disclosed that it will soon put an on-line data base on the Source (McLean VA) timesharing system. This database will include technical support and marketing information.

DRI now also has a coordinator of CP/M User Groups. He is Gary Gysin, Mail Station 8, DRI, Box 579, Pacific Grove CA 93950.

Ada Standard adopted

ANSI and the Department of Defense have both approved the standard for the Ada programming language. Also, the DOD has announced that it has released the Ada validation test procedures. DOD trademarked the name Ada, and it will not permit any language vendor to use the name until the DOD has tested the product and certified that it has passed their tests for compliance with the standard. Both Western Digital and Rolm Corp. have announced that they are ready to submit their compilers for testing. The test results should become known next year.

Hardware disk interface standard

Twenty-six companies, including Control Data Corp., Memorex, NEC, IMI, Micropolis, Western Digital, AMD, and National Semiconductor have endorsed a 5¼" Winchester interface standard. Noticeable by their absence are Seagate Technology, Tandon and IBM. . . . TI, National Semiconductor, General Instrument, and Motorola are sampling single-chip modem ICs for both 300- and 1,200-baud use. Expected the next generation of personal computers will have modem circuitry and software as built-in functions.

Software checkout service

Tymshare has disclosed that it plans to offer a service whereby software packages can be checked out via their timesharing network, and then users can down-load a package if they decide to purchase it.

New book publishers

Microsoft and VisiCorp, two of the leading micro software houses, have decided to go into the book publishing business. Why not—virtually all of the book publishers have gone into the software business! MJ

VEDIT

CUSTOMIZABLE
FULL SCREEN EDITOR

The Professional's Editor for Program Development Word Processing Source Code Translations

Widely acclaimed as an editor, VEDIT has evolved to be much more. Only VEDIT offers the combination of a versatile full screen editor integrated with a powerful command language. For the first time you'll be able to perform complex, yet useful, text manipulations that are virtually impossible with other editors or word processors. Plus, its customizability and hardware support ensure that VEDIT will be perfectly matched to your individual needs and to any microcomputer you are ever likely to own.

With two modes of operation, VEDIT never compromises its speed or ease of use for its power and sophistication. As one reviewer (Bradford Thompson, BYTE) wrote: 'If this review gives you an appetite for simplicity while editing, then VEDIT is well worth considering.' Its command language, based on TECO, is virtually a text oriented programming language, allowing command macros to be created, loaded and saved on disk. Yet its simplicity allows even a novice to perform tasks beyond the capabilities of any word processor.

VEDIT cuts programming time in half - with multiple file handling, macro capability and special features for Pascal, PL/1, 'C', Cobol, Assembler and other languages. And it can help with source code translations (example ZILOG to/from INTEL translator macros are included). A complete line of translators will be available by the year's end.

Word processing is a snap with word wrap, paragraph and print functions. Command macros free you from tedious search/replace operations. Hundreds of search/replace on dozens of files can be performed by VEDIT without waiting or intervening.

VEDIT easily configures to your favorite keyboard layout. Use any function or cursor keys you wish. It optimally supports nearly every 8080, Z80 and 8086 computer.

Go ahead and expect a lot from VEDIT. Its performance and our support will satisfy your most exacting needs.

To order, please specify your 8080, Z80 or 8086 microcomputer, operating system and disk format.

COMPARE VEDIT'S FEATURES

- True Full Screen Editing
- Horizontal scrolling
- Edit files one disk in length
- Automatic Disk Buffering
- Compact (only 16K) and Fast
- Display of line and column #
- Set/Goto text markers
- 'Undo' key to restore line
- Automatic Indent/Undent
- Adjustable Tab positions
- Repeat function key
- Text Move and Copy
- 10 Scratchpad Buffers
- Load/Save buffers on disk
- Powerful command macros
- Directory display
- Edit additional (small) files simultaneously
- Insert another disk file
- Unlimited file handling
- Recovery from 'Full Disk'
- Change disks while editing
- Word wrap, format paragraph
- Simple Printing
- 150 page indexed manual
- Startup command file
- Menu driven installation
- Program CRT function keys
- Support newest CRT terminals
- Support smart CRT functions
- Flexible Memory Mapped support
- Customizable keyboard layout

IBM PC, Displaywriter ° Zenith Z100 and Z89 ° NEC APC ° DEC Rainbow and VT180 ° Televideo 802
TRS-80 I, II and 16 ° Xerox 820 ° Apple II Softcard ° SuperBrain ° NorthStar
MP/M ° CP/M-86 ° MP/M-86 ° Concurrent CP/M-86 ° Cromix ° Turbo DOS ° MSDOS ° PCDOS

VEDIT - Disk and Manual
8080, Z80 or IBM PC $150
CP/M-86 or MSDOS $195
Manual only $18

CompuView
PRODUCTS, INC.

1955 Pauline Blvd., Suite 200 ° Ann Arbor, Michigan 48103 ° (313) 996-1299

CIRCLE 35 ON READER SERVICE CARD

The CP/M Bus

by Anthony Skjellum

The first topic for this installment of CP/M Bus is a discussion of user area specifications and the different ways in which existing programs allow them to be specified as part of filenames. Two technical points about user areas are also mentioned.

In the remainder of this column we discuss a public domain utility system: LU. The LU system is a general-purpose librarian utility. Possible applications for library files are also mentioned.

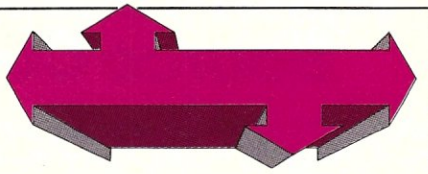
User area specifications

Starting with CP/M 2.0, Digital Research began supporting user areas as part of the CP/M file structure. The user area part of the filename was designed for compatibility with MP/M, in which each user has his own file area. With the advent of hard disks, the availability of user areas has become important to many CP/M users because CP/M provides no subdirectory structure within its filesystem. However, Digital Research's support of this feature for CP/M has been very limited. PIP and STAT reference user areas in limited cases, but CP/M utilities in general do not recognize a filename suffix or prefix to override the current user area.

Two points about user areas

Two points about user areas are in order. The first is that there are 32 available areas, and not 16. Only 16 are accessible through the console command processor USER command, but the full 32 may be accessed from transient programs. These additional upper areas are supported by utility software such as "The Unica," marketed by Knowlogy.

The second point about user areas concerns closing files. It is not possible to close a file if it is not in the current (default) user area. Be aware of this fact and program around it.



Specifying user areas

Many software packages now support the specification of user areas. Unfortunately, there are two standards: a prefix notation and a suffix notation. The latter notation is used by programs such as "The Unica," while the BDS C Compiler's runtime library (and other software) supports the prefix form. Here is an example of each:

```
ls C:*.*;5        --  Unica style
pgm 5/C:*.*       --  BDS C style
```

The prefix form seems to derive from the Unix notation for subdirectories.

Since the use of the BDS C compiler is more widespread than "The Unica," the prefix format may eventually win out as the standard notation. However, we can expect to use both formats in the foreseeable future. We hope that no one will invent any additional notations for user area specifications.

The existence of two standards is inconvenient, but inevitable considering that Digital Research defined no standard. The fact that not all software permits any form of user area specifications is also unfortunate.

Library files

In a CP/M disk directory, each filename listed in the directory corresponds to one file. However, it is sometimes convenient to lump files together as a collective file. This is most often done for archival storage, or when sending many files via a modem program that will not transmit multiple files on a single execution. The individual file identities are maintained in the collective file, so that member files can be extracted later. Use of libraries can also save disk space (and directory entries.) Disk space is saved, since CP/M allocates sectors to files in 1K increments.

Virtually none of the programs currently available understand the LU library format, so files will have to be extracted from the library before they can be used with most programs. This limits the usefulness of libraries as pseudo-subdirectories.

Library files can actually be counterproductive if used incorrectly. For example, continual deletion/updating of library member files creates unused space in the library, which naturally consumes disk storage. To remove such wasted space, the library must be periodically repacked. This can be a lengthy process for large libraries. Because of these drawbacks, I recommend that libraries be used for archival storage of files that are updated only occasionally. LU's most important use is in creating convenient file bundles for modem transmission.

The LU system

The LU system consists of two programs: LU, the librarian, and LRUN, a program to execute .COM files from a library. We'll discuss each of the programs in turn.

The library utility (LU)

The LU program operates either interactively or on command line arguments supplied at execution. Commands are entered as dash options followed by filename parameters.

The most convenient way to use LU is interactively. In order to run the program, the following command is entered at the console command processor level (CCP):

```
A> LU <CR>
```

where <CR> indicates, as always, either Carriage Return or Enter. LU will sign-on:

```
Library Utility x.yz
```

where x.yz is the version num-

HOW TO CONTROL YOUR CP/M™ MICRO–AT ONLY $3. PER COMMAND.

A Remarkable Program For CP/M Users.

Of course, CP/M is a wonderful operating system. That's why so much serious business software has been created for it.

BUT, CP/M is not easy to work with. That's why you need to take the **POWER!** trip.

POWER! is a super-power-packed, user-friendly program that lets you take immediate and complete control of CP/M. And at a cost of only $3. per command, it's the software buy of the year.

Over 55 Housekeeping Utilities.

POWER! is over 55 prompted, user-friendly CP/M utility programs all rolled into one 15k package. It takes care of all of these frustrations and more:

—BDOS errors? POWER! ends BDOS errors and gives you a way out.

—Accidentally erased a file? If you accidentally erase a program or disk file, **POWER!** restores the erased files.

—Can't remember file names? POWER! assigns a number to each file on your disk. So, to copy files from disk to disk, you don't have to fiddle with PIP anymore. You just pick the file from a numbered menu and **POWER!** copies it for you. No more typing errors! **POWER!** also marks original files and their copies for you; and you can compare files to find identical copies regardless of name.

—Lose data on a glitched disk? If a glitched disk makes it impossible to call up a long word processing text, **POWER!** can fix the glitch. This means you may have to retype only a couple of sentences instead of losing 20 pages of text.

—Trouble with "bargain" disks? POWER!'s disk testing function gathers any bad sectors of the disk into a special file so that CP/M thinks those parts of the disk are already used and never attempts to write to them. The rest of the disk is then safe to use.

—CP/M scrolls too fast through text files? POWER! spools through files page by page, file by file, or line by line with instant halt by touching the space bar.

—Need to reorganize files? POWER! sorts and formats the directory in 4 different ways. And you can easily copy or move files from user area to user area. **POWER!** creates 32 user areas instead of CP/M's 16.

—Need to change memory? POWER! searches, displays and lets you change memory wherever you want. You can even automatically run software anywhere in memory. And you can inter-mix your search with as many wild card jokers as you need to find, for instance, all occurrences of "Sam Jones" and "Sid James" just by typing "S??J??". And **POWER!** also lets you read or write to any sector or track very simply.

—Changing disks? You can forget the ubiquitous Control C to change disks. **POWER!** can do it for you automatically. And **POWER!** doesn't require a system disk in any drive, so Drive A is open for use, when **POWER!** is in control of CP/M.

—Afraid of HEX numbers? POWER! automatically converts Hex to Decimal, Binary or ASCII.

Special Password Protection, Too.

POWER! now includes a special program that lets you lock sensitive files, so that only you can access them. Without the secret PASSWORD which you can create and change at will, no prying eyes will ever know your secret file even exists. A great way to protect financial or scientific data from unauthorized eyes. Just this single program alone would be worth the price of **POWER!,** but there are over 55 more just as valuable programs in this power-packed-package.

At $169., It's A Bargain.

Space doesn't permit describing all the wonderful ways **POWER!** can put you in complete control of your CP/M micro. But see for yourself. There's a Money Back Guarantee. At the low price of $169., each powerful command costs you less than $3. A true bargain!

InfoWorld
Software Report Card

Power Version 2.55

	Poor	Fair	Good	Excellent
Performance	☐	☐	☐	☑
Documentation	☐	☐	☑	☐
Ease of Use	☐	☐	☐	☑
Error Handling	☐	☐	☐	☑

POWER! Is Better Than Ever!

Eventhough "InforWorld", "Microsystems" and "Interface Age" call **POWER!** great, we have improved **POWER!**—including a completely rewritten 120-page easy-to-read documentation. (Previous purchasers of **POWER!** may exchange their original disk for an updated version with the new commands and a brand new manual—for only $35.)

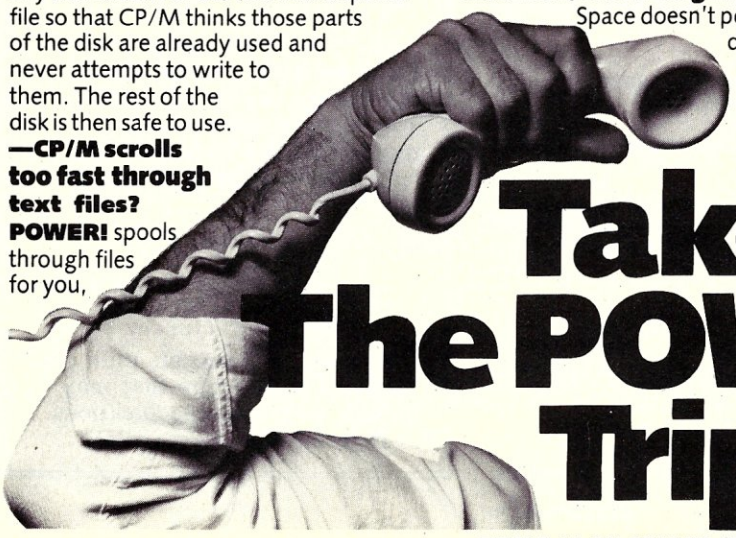
Take The POWER! Trip Today!

POWER! will operate in any standard CP/M or MP/M system, including CP/M-86, IBM PC, Apple (Z80 card), Osborne, Kaypro, HP, TeleVideo, TRS-80 conversions, S100's including NorthStar, Vector, Morrow, CompuPro, etc. Up and running immediately, no configuration necessary—for hard disks and floppies.

At only $3. per command, you can afford to **Take the POWER! Trip.** Call or send in your order today.

NOW AVAILABLE FOR MS-DOS, TOO.

ONLY $169. Money Back Guarantee.
Charge & COD Orders Welcome.
TOLL FREE (800) 428-7825 Ext. 96AG
IN CA: (800) 428-7824 Ext. 96AG
DEALERS AND OEM'S (415) 567-1634 Ext. 96AG



Take The POWER! Trip.™

COMPUTING!
2519AG Greenwich San Francisco, CA 94123
TOLL FREE (800) 428-7825 Ext. 96AG
IN CA: (800) 428-7824 Ext. 96AG
DEALERS AND OEM'S (415) 567-1634 Ext. 96AG

ONLY $169. Calif. add 6½% sales tax
☐ **CP/M $169.** ☐ **CP/M-86 $169.** ☐ **MP/M $249.**

Card No. _____
Exp. Date _____
Computer _____
Your Name _____
Company Name _____
Address _____
City/State/Zip _____

CIRCLE 14 ON READER SERVICE CARD

ber. The version that I currently use is 2.12, so this overview reflects that version. LU subsequently displays the command-line prompt:

```
-?   Ø/A:>
```

The question mark indicates that there is no default command at present. The notation 0/A: indicates that the default user area and disk are zero and A:, respectively (these will be used when LU reads or writes files.) The '>' is the command prompt. In response to this prompt, we can enter a command line or a carriage return by itself. The latter causes the program to terminate.

Command lines consist of dash options (minus sign followed by a single letter) and filenames. For example, to open a new or current library file, the following command is given:

```
-O FILENAME
```

LU will open the file FILENAME.LBR. If the file is new, the user will be prompted for the number of directory entries for the library file. This determines the maximum number of files that may be stored in the library. This number is fixed and cannot be increased unless the file is reorganized (i.e., repacked).

Several operations can be performed on an open library. The first we will consider is file addition. To add files, we enter the following in response to the '>' prompt.

```
>-A <file 1.ext>,[<file2.ext>],...,[<fileN.ext>]
```

where file1.ext through fileN.ext are ambiguous or unambiguous filenames separated by spaces. The BDS C user area prefix may be prepended to any of the file specifications. After entry of the −A command, LU will inform the user of its progress in appending the files to the library. When appending is complete, LU displays its prompt once again. Now −A is the default command and further files could be specified for appending without typing −A again. Here is a sample fragment of a session:

```
Library Utility 2.12

-?   Ø/A:>-O LIB

     -- open library LIB.LBR

-O   Ø/A:>-A CPMBUS1.TXT

     -- add file CPMBUS1.TXT

-A   Ø/A:>CPMBUS2.TXT  CPMBUS3.TXT

     -- add  two more files

-A   Ø/A:>...

     -- session continues
```

Once we have a set of files in our library, we could conceivably wish to delete one or more. This is effected via the −D option. Here is an example:

```
-?  Ø/A:>-D *.FIL

    -- remove all *.FIL from

       current library
```

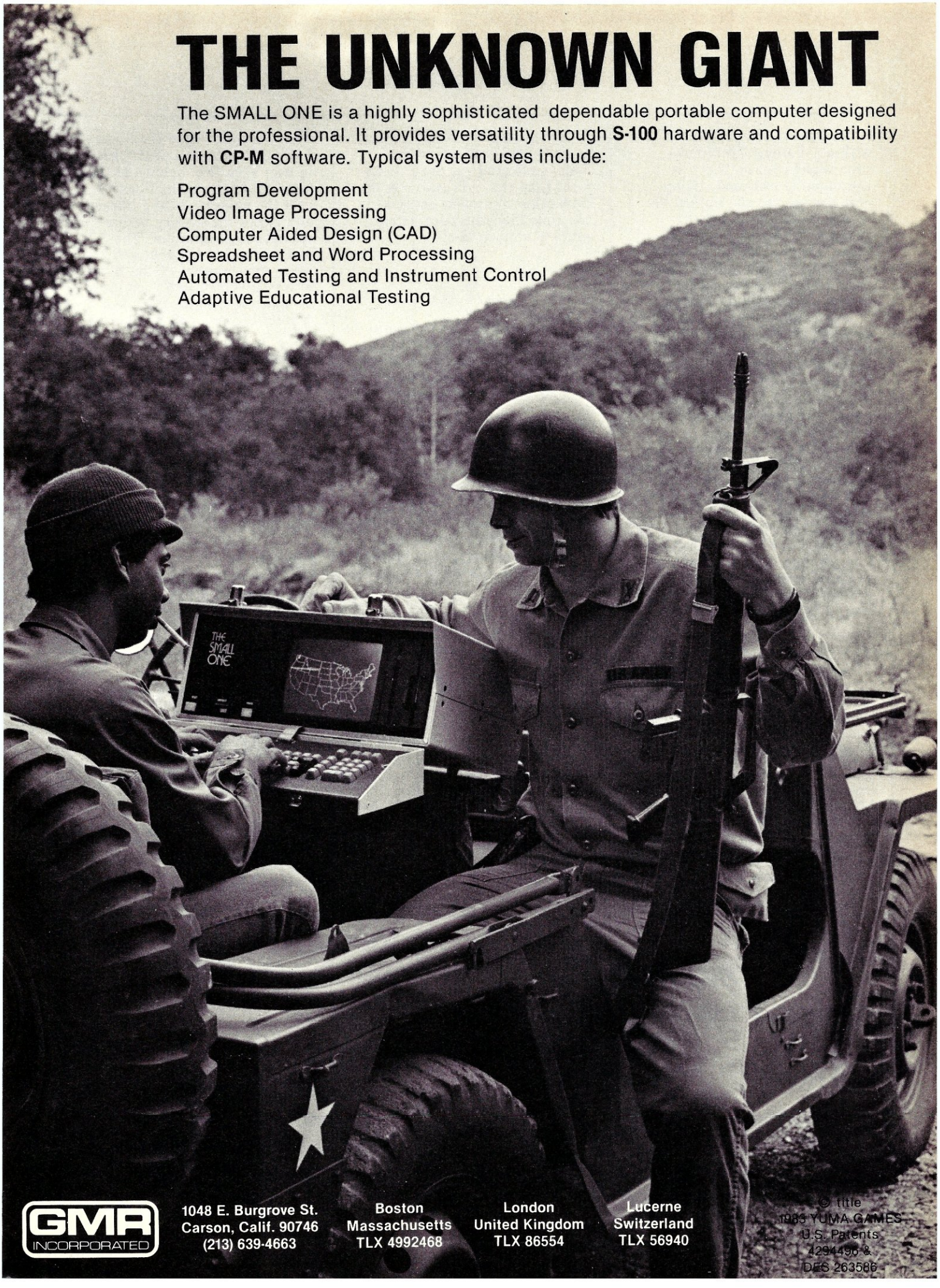
Like −A, −D accepts multiple wildcard parameters that may include user area specifications.

Deleting files leaves allo-

THE UNKNOWN GIANT

The SMALL ONE is a highly sophisticated dependable portable computer designed for the professional. It provides versatility through **S-100** hardware and compatibility with **CP-M** software. Typical system uses include:

Program Development
Video Image Processing
Computer Aided Design (CAD)
Spreadsheet and Word Processing
Automated Testing and Instrument Control
Adaptive Educational Testing



GMR INCORPORATED

1048 E. Burgrove St.
Carson, Calif. 90746
(213) 639-4663

Boston
Massachusetts
TLX 4992468

London
United Kingdom
TLX 86554

Lucerne
Switzerland
TLX 56940

cated but unused disk space within a library. To remove such wasted space, the −R command is provided. Using −R also allows the library directory to be increased. Repacking the library requires about as much free disk space as the size of the library itself because the files are copied during the repacking process.

It is also possible to list the contents of the current library via the −L command. This also displays pertinent information about space allocation, library size, file sizes, wasted space, and the number of free library directory slots.

When work on a given library is complete, closure of the library is accomplished via the −C command. This does not cause termination of the program, so that new libraries may be opened via the −O dash-option. Simply typing <CR> in response to the command prompt closes the current library and terminates execution.

Eventually, we will want to extract files from a library. This is effected via the −E option. Like the −D and −A options, this command accepts multiple filename parameters. Here is an example:

```
-?   Ø/A:>-E *.*

     -- extract the whole
```

One final note. The default 0/A: may be changed with the −U command, which takes the new default user area and/or disk name as its argument.

Multiple commands

When running the program, multiple commands may be entered on the same command line. I don't do this, since I prefer to see the action of each command in turn. In the noninteractive mode, multiple commands are, of necessity, entered at execution. Here is an example:

```
A>LU -O LIB -A 5/B:*.FIL -D *.TXT -L -C
```

This example opens the library LIB.LBR, adds all of B:*.FIL in user area five, deletes all the files *.TXT from the library, lists the contents of the library, and closes the library before terminating execution. The trailing −C is not required here, since closure of the library on return to CP/M is automatic; it is shown just as a sample of what can be done.

Other features

The LU program is written in BDS C and uses the directed input/output package (DIO). Therefore, both input and output may be redirected in a UNIX-like fashion. This permits interactive commands to come from a file, and output from the program to be logged in a file.

LRUN

The LRUN program allows execution of .COM files from a single library. When many

CP/M Bus continued . . .

small utilities are combined into a library, disk space savings can be substantial. Using LRUN is straightforward. Here is an example (the default library COMMAND.LBR is used when none is specified):

```
A>LRUN ERAQ B:

 -- run ERAQ with parameter B:
```

Conclusions about LU

The LU utility provides a useful library filing system for CP/M. If used properly, it can be very convenient. LU is a copyrighted program released for noncommercial use by author Gary P. Novosielski. It is available on many RCP/M systems nationwide.

Conclusion

This installment of CP/M Bus has included a discussion of user areas under CP/M 2.x, as well as an overview of the LU librarian system.

Suture columns will include material on CP/M-86.

In the Public Domain

by Chris Terry

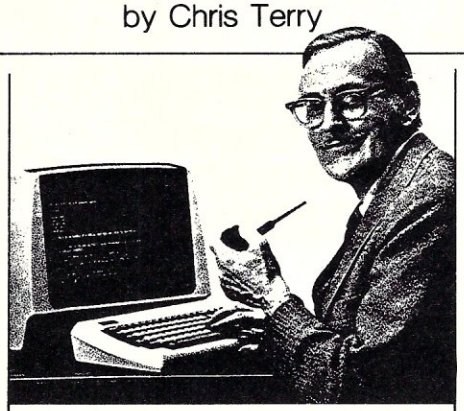
This month I shall be discussing catalogs, assemblers, disassemblers, cross-assemblers, and cross-reference utilities that are available in the CPMUG and SIG/M libraries.

Catalogs

A catalog of the CPMUG volumes up through 91 (last issued up to June 1, 1983) is available from the CP/M Users Group, 1651 Third Avenue, New York, NY 10028. The most recent catalog of SIG/M volumes, which covered up to Vol. 107, rapidly went out of print; a new catalog is in preparation and will include the latest volumes up to 114. In addition, it will include a list of distribution points for SIG/M (both clubs and individuals). You can order it from SIG/M Software, Box 97, Iselin, NJ 08830 at $2 (mailed, USA) or $2.50 (mailed, foreign). Both of these are brief catalogs, containing the program names for each disk and a half-line indication of their purpose. Complete documentation for the programs is included on the disks, but can also be found in the seven catalogs of CPMUG and SIG/M volumes issued by the New York Amateur Computer Club, P.O. Box 106, Church St. Station, New York, NY 10008. The price is $10 (postpaid) for each book ($15 for overseas airmail).

Catalog of Public Domain Software for CP/M

Book 1: CPMUG 1 49, SIG/M 1-18 ..... 204 pp
Book 2: CPMUG 50-54, 78-79; SIG/M 19-42.. 214 pp
Book 3: SIG/M 43-60 .. 217 pp
Book 4: CPMUG 80-84, SIG/M 61-76 ..... 210 pp
Book 5: SIG/M 77-104
Book 6: ZCPR2 User & Reference Manuals (SIG/M 105)
Book 7: S106, 107 ZCPR2 & SYSLIB
IBM-PC Book 1: 1-26



Assemblers

One of the most useful assemblers in the public domain is **LINKASM.COM,** contained in CPMUG Vol. 36. This is an enhancement by Ward Christensen of the standard CP/M ASM.COM, which I no longer use. LINKASM runs slightly faster than ASM, allows any number of source code files to be assembled into a single .HEX file, and can produce a symbol table suitable for use with the SID debugger. Each source code file to be linked ends with the statement LINK <filename>, where LINK is in the opcode field and <filename> is the name of the source file to be linked (.ASM is assumed) in the operand field. The last file in the series must be terminated with the usual END statement—no other file in the series may contain the END statement. The invocation of LINKASM is similar to that of ASM, i.e., LINKASM file.shp t:, where **file** is the source filename, **shp** specify the drives on which to find the source, put the hex file, and put the listing, and **t:** specifies the drive on which to put the symbol table. The symbol table is only partially sorted on the first letter of each symbol. An external sort program must be used if the symbol table must be in exact alphabetic order. This assembler was originally concocted for the specific purpose of assembling the modules of the MODEM program which, with comments, amounted to over 100K of source code.

Another assembler for 8080 code is **ML-80** in CPMUG Vol. 4. This contains (according to the documentation) high-level control structures, a recursive macro-processor, a relocating compiler, and a linking loader. It is supplied with Submit files to run source code through the various modules. The wide-directory programs SDIR and XDIR were written by Bruce Ratoff with this assembler, perhaps also using the patched versions of the M81 and L82 modules on CPMUG Vol. 36. Documentation is somewhat sparse, but when Vol. 4 was released a language manual was reputedly available from Digital Research for the cost of copying; I do not know if it is still available.

Several Z80 assemblers are contained in CPMUG Vol. 16. The **ASMX.COM** assembler recognizes Zilog mnemonics, and is reputed to work well if you restrict source, .HEX, and .PRN files to the A drive. Specifying a different drive, or faulty syntax in the command line have been known to destroy the disk directory.

MACASM is a macro assembler similar to one issued by Intel, and recognizes Zilog mnemonics, while running on an 8080. The nesting of macros, however, sometime clobbers locations 5, 6, and 7, which of course is catastrophic. The fix is not known.

Z80ASM is another assembler that recognizes Zilog mnemonics and can be run on an 8080. This one is reputed to work well, but may be hardware-sensitive. I tried it several times, but each time it went out to lunch leaving no messages. Admittedly, my system had a few quirks in it, but I had no trouble running other programs in the same volume.

Cross-Assemblers

One of my former colleagues, while vainly struggling to produce runnable code for a Quotron 800 minicomputer from an assembler written in IBM Fortran, defined cross-assemblers

as "Assemblers that make the user cross!" One hopes that this definition does not apply to the two cross-assemblers I found in the public domain catalog. One of these is **XLATE** (SIG/M Vol. 91), which translates 8080 code to Z80 code using Zilog mnemonics. The other is **A68K** (SIG/M Vol. 92) which, as one might expect, runs on an 8080/Z80 machine but assembles source code for the Motorola 68000 16-bit CPU.

Disassemblers

The prime 8080 disassembler is **RESOURCE** (CPMUG Vol. 42), by Ward Christensen. This was based on a stand-alone disassembler published in DDJ and supplemented by features to meet Ward's later needs. RESOURCE is so powerful that there were nervous outcries from some software vendors when it was first put in the public domain. The documentation (as with all of Ward's programs) is excellent. With care and experience it is possible to disassemble even a complex program with RESOURCE, reassemble the result, and find that the new object code is identical to the original.

REZ (SIG/M Vol. 10 & CPMUG Vol. 64) is a version of RESOURCE modified to disassemble Z80 code to assembly language using either TDL or Zilog mnemonics.

DASM (SIG/M Vol. 91) is another Z80 disassembler based on RESOURCE, and likewise handles both TDL and Zilog mnemonics.

DISAMB (CPMUG Vol. 75) is a disassembler written in MBasic, to handle 8080, 8085, and Z80 code (TDL or Zilog mnemonics). Anyone wishing to use this program should be sure to read DISAMB.DOC first. This file contains operating procedures and an explanation of the numerous small files associated with DISAMB.

Auxiliaries

COMBINE (CPMUG Vol. 36) concatenates assembly language source modules into a single large file, removing all comments to save space.

NOTATE (CPMUG Vol. 78) fetches an assembly language source file and allows the insertion of comments. It scans each line in the file and, if no comment is present, places the cursor at the start of the comment field ready for insertion from the keyboard.

XREFASM (CPMUG Vol. 6) prints an assembly language source file with line numbers and provides a cross-reference index (by line number) of all symbols.

XREFPRN (CPMUG Vol. 78) scans a .PRN assembly listing and outputs a cross-reference index (by address) to the List device.

Next month I shall be exploring some of the many CP/M enhancements that are to be found in the public domain libraries. These will include not only major systems such as ZCPR and SUPERSUB, but also smaller items such as the null file.

Letters to the Editor

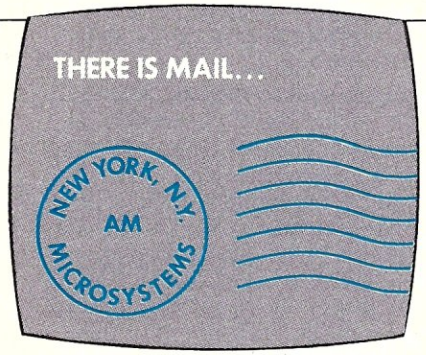
Dear Sir:

The public domain program CATALOG, by Ward Christensen, is one of the most useful available to the serious hobbyist and/or professional. Anyone who needs to keep more than half a dozen discs on file will find this program a must. There are various catalog programs for sale in the commercial software market, but none that I have tried can compete with CAT.

For those unfamiliar with CAT, the package consists of a group of related programs: CAT.COM, the main retrieval program, FMAP.COM, needed to sort and catalog files on the selected disc, and UCAT.COM to update the main catalog file (while is labeled MAST.CAT). All of these .COM files are the end product of years of loving extensions and modifications, a process that continues today. Like all good things, it keeps getting better and better. It is of course free, being in the public domain, but even nicer is that there is a CATALOG.DOC, which tells one all that one would ever wish to know about CAT but were too lazy to ask. Check with your local user's group.

The main catalog file, MAST.CAT, has the very nice capability of being told which programs to process and which to ignore. Thus common files such as PIP, STAT, SUBMIT, etc. will not clutter up every file, and such unneeded (or totally undesirable) programs as *.PRN, *.BAK, *.$$$, and so forth will not appear. MAST CAT is very easy to set up initially, but be sure to leave enough DUMMY.ARG statements at the end to allow later changes. Otherwise it is very difficult to modify!

The use of CAT is simplicity itself. Any disc, once added to MAST.CAT, can be updated at any time simply by running the CAT commands needed for processing. Any new files added since the prior update will be added to MAST.CAT, any deleted from the disc will be automatically deleted from MAST.CAT.



To make the update process totally painless, or as painless as possible at any rate, I have written several short and simple SUBMIT routines to do the drudgery. One of these, called CAT.SUB, will update the disc in the B: drive with no further attention. Another, called CATME.SUB, catalogs the disc in the A: drive—the master disc itself with the CAT programs on it. A third, called CAT1.SUB, allows naming and initial cataloging of a new disc. The programs are listed on separate printouts accompanying this letter.

There is a still later version, which I have just received, that has automatic labeling facilities: i.e., it will keep track of the discs already named and will supply the next number in sequence for the new disc. I have not had a chance to check this out yet, and in any event I am dubious of its value. The ability to supply my own numbers and names, regardless of sequence, is better, I should think.

Incidentally, I note from the Letters section that a rather large percentage of micro users like to rename standard programs: substituting S for SUBMIT, for instance. I used to do this myself, until I found how much confusion it causes among other users. They always want to know "What's this program here—," which is bad enough, but worse when I've forgotten what the shortened or renamed label means myself! Much better to leave the standard and accepted names as they are.

CAT.SUB

```
:Catalog disk b:
era *.bak
era b:*.bak
fmap b: f
ucat
```

CATME.SUB

```
ERA *.BAK
FMAP A: F
ERA NAMES.BAK
UCAT
```

CAT1.SUB

```
SAVE 0 B:-$1
ERA MAST.BAK
ERA B:*.BAK
FMAP B: F
UCAT
ERA NAMES.BAK
```

```
Note: Calling sequence is

A> SUBMIT CAT: filename:num

where filename.num is the
name AND number of the disk
to be cataloged.  Do NOT
include the hyphen in the
filename.
```

Gerald L. Hewett
Star Route, Box 108
Inyokern, CA 933527

Dear Sir:

I was very excited by Mr. Jon Lindsay's article entitled "Restoring Unsaved MBasic Programs" (May 1983, page 94).

Immediately after reading the article, I rushed to my computer, typed in the BASIC program (Listing 1.), and, having, lost my share of BASIC programs already, I saved the program.

I then typed SYSTEM to re-

FORTH for Z-80®, 8086, 68000, and IBM® PC

FORTH Application Development Systems include interpreter/compiler with virtual memory management and multi-tasking, assembler, full screen editor, decompiler, utilities, and 130+ page manual. Standard random access files used for screen storage, extensions provided for access to all operating system functions.

Z-80 FORTH for CP/M® 2.2 or MP/M II ..... $ 50.00
8080 FORTH for CP/M 2.2 or MP/M II ..... $ 50.00
8086 FORTH for CP/M-86 or MS-DOS $100.00
PC/FORTH™ for PC-DOS, CP/M-86, or CCPM $100.00
68000 FORTH for CP/M-68K $250.00

FORTH + Systems are 32 bit implementations that allow creation of programs as large as 1 megabyte. This is the only language that supports the entire memory space of the 8086/88 directly for programs and data!

PC/FORTH + for PC-DOS or CP/M-86 ..... $250.00
8086 FORTH + for CP/M-86 ..... $250.00

Extension Packages for FORTH systems

Software floating point (Z-80, 8086, PC only) $100.00
Intel 8087 support (8086, PC only) $100.00
AMD 9511 support (8086, Z-80 only) $100.00
Color graphics (PC only) $100.00
Symbolic interactive debugger (PC only) $100.00
Cross reference utility ..... $ 25.00
PC/GEN™ (custom character sets, PC only) ..... $ 50.00
Hierarchical file manager ..... $ 50.00
B-tree index manager $125.00
B-tree index and file manager $200.00

QTF + Screen editor and text formatter by Leo Brodie, for IBM PC with IBM or Epson printer ..... $ 50.00

Nautilus Cross Compiler allows you to expand or modify the FORTH nucleus, recompile on a host computer for a different target computer, generate headerless code, and generate ROMable code with initialized variables. Supports forward referencing to any word or label. Produces load map, list of unresolved symbols, and executable image in RAM or disk file. No license fee for applications created with the Cross-Compiler. Prerequisite: one of the application development systems above for your host computer.

Hosts: Z-80 (CP/M 2.2 or MP/M II), 8086/88 (CP/M-86 or MS-DOS), IBM PC (PC-DOS or CP/M-86), 68000 (CP/M-68K)
Targets: 8080, Z-80, 8086/88, 6502, LSI-11, 68000, 1802, Z-8

Cross-Compiler for one host and one target $300.00
Each additional target $100.00

AUGUSTA™, ADA subset compiler from Computer Linguistics, for Z-80 computers under CP/M 2.2 ..... $ 90.00
LEARNING FORTH computer-assisted tutorial by Laxen and Harris for CP/M, includes Brodie's "Starting FORTH" ..... $ 95.00
Z-80 Machine Tests Memory, disk, printer, and console tests with all source code in standard Zilog mnemonics $50.00
DATA ACE, fully relational data base system from CSD, for the IBM Personal Computer. Faster and more powerful than dBASE II $595.00

FORTH application development systems require 48 kbytes RAM and 1 disk drive, Cross-Compilers require 64 kbytes. All software distributed on eight inch, single density, soft sectored diskettes except PC/FORTH on 5¼ inch single sided double density diskettes. Prices include shipping by UPS or first class mail within USA and Canada. California residents add appropriate sales tax. Purchase orders accepted at our discretion.

Laboratory Microsystems, Inc.

4147 Beethoven Street
Los Angeles, CA 90066
(213) 306-7412

Z-80 is a registered trademark of Zilog, Inc.
CP/M is a registered trademark of Digital Research, Inc.
IBM is a registered trademark of International Business Machines Corp.

Augusta is a trademark of Computer Linguistics
dBASE II is a trademark of Ashton-Tate
PC/FORTH and PC/GEN are trademarks of Laboratory Microsystems Inc.

CIRCLE 26 ON READER SERVICE CARD

Letters continued . . .

turn to the CCP. Then, following his instructions, I typed DDT to enter the debugger. So far, everything was going exactly as stated in the article. Then, again following the instructions, I typed D4990 to display a block of memory at

Table of address equates

MBasic Version 4.51 Location	MBasic Version 5.2 Location	Identity of the data residing at the address listed
4990H	61B0H	Starting address of the block of memory you want DDT to display
4997H	61BBH	00—Start of Basic source
4998H	61BCH	Low byte value pointing to address of the beginning of the next line of source code
4999H	61BDH	High byte value pointing to address of the beginning of the next line of source code
499AH	61BEH	Low byte value of the line number of the first line of the MBasic source
499BH	61BFH	High byte value of the line number of the first line of the MBasic source
499CH	61C0H	Location of token for first line reserved key word

ERG/68000

MINI-SYSTEMS

☐ Full IEEE 696/S100 compatibility

HARDWARE OPTIONS

- ☐ 8MHz, 10MHz or 12MHz 68000 CPU
- ☐ Memory Management
- ☐ Multiple Port Intelligent I/O
- ☐ 64K or 128K STATIC RAM (70 nsec)
- ☐ 256K/512K or 1MB Dynamic RAM, with full parity (150 nsec)
- ☐ 5¼"-8" DID, D/S floppy disk drives
- ☐ 5MB-40MB hard disk drives
- ☐ Full DMA Disk Interface
- ☐ SMD Interface
- ☐ ¼" tape streamer
- ☐ 10 to 20 slot backplane
- ☐ 20 or 30A amp power supply
- ☐ Desk top or Rack mount cabinets

SOFTWARE OPTIONS

- ☐ 68KFORTH[1] systems language with MACRO assembler and META compiler, Multi-user, Multi-Tasking
- ☐ Fast Floating Point package
- ☐ Motorola's MACSBUG
- ☐ Unix[2]Type Operating System with C,PASCAL, FORTRAN 77, 68K-BASIC[1], CIS COBOL[4], RDBMS
- ☐ CP/M-68K[3] O/S with C, Assembler, 68K-BASIC[1], 68KFORTH[1], Z80 EMULATOR[1], FORTRAN 77
- ☐ Unix[2] System III with C, etc.
- ☐ VED68K[1] Screen Editor

Trademark [1]ERG, Inc.
[2]BELL LABS [3]Digital Research
[4]Micro Focus

30 day delivery
with valid Purchase Order
OEM prices available
For CPU, Integrated Card Sets or Systems.

Empirical Research Group, Inc.
P.O. Box 1176
Milton, WA 98354
206-631-4855

ERG SOFTWARE AVAILABLE FOR:

CP/M-68K[3] O/S: 68K FORTH[1], 68K-BASIC[1]
VED68K[1] Screen Editor,
Z80 Emulator[1], FORTRAN 77

Motorola KDM Board: 68KFORTH[1]

Motorola MVME 110: 68KFORTH[1]

NEW PRODUCTS IN DEVELOPMENT

68000/16032 Processors

For The

VME Bus

10MHz ERG/VME 16032
/w 16082 MMU $1995
/w 16081 FPU $2495

12 MHz ERG/VME 68000
/w 68451 MMU $2495
/w 16081 FPU $2695

ERGFORTH[1] (68KFORTH[1]) Kernal and Compiler in resident ROM. Complete ERGFORTH[1] Systems Language available for both processors. Call for additional software availability.

AVAILABLE: 68000—Now
16032—August

Trademark [1]ERG, Inc. [3]Digital Research

Empirical Research Group, Inc.
P.O. Box 1176
Milton, WA 98354
206-631-4855

Letters continued . . .

location 4990H. That was where the problem began. As you can see from Listing 2, there was no correlation at all with my Basic program. Since Mr. Lindsay's instructions were so precise, I knew that something was wrong.

That was when I realized that I was using the new version 5.2 of MBasic. When I looked at the status of MBASIC.COM, I found that instead of 18K, the program had grown to 24K. So, with pencil in hand and some hints from Mr. Lindsay's article, I went hunting for the elusive Basic source program.

After a few minutes, there it was. Instead of lurking in the area of 4990H, it was now residing around 61B0H (Listing 3.) With that bit of information, it was easy to construct the attached table of address equates so that Mr. Lindsay's instructions may be followed with the new version of MBasic.

By substituting the addresses given in the table for the addresses in Mr. Lindsay's article, you should have no trouble restoring your MBasic programs as he explained.

In the program shown in Listing 1, all you have to do is go back into MBasic and issue the commands:

POKE &H61BC, &HD0
POKE &H61BD, &H61

This will reset the pointers. You should then be able to list the program and save it using the ASCII option.

John L. Dilbeck
Microcomputer Systems Specialist
The Lamson Colleges, Inc.
2300 East Broadway
Tempe, AZ 85282

```
10 REM  Test program
20 PRINT "It is easier to recapture a "
30 PRINT "lost program than you think."
40 REM  Test calculation
50 LET A=23
60 LET B=67
70 PRINT A+B, A+B
```

LISTING 1.

```
D4990
4990 6F CD BD 48 D1 CD CE 48 C3 1E 47 CD E3 49 D1 D5 o..H...H..G..I..
49A0 1A 90 C3 6E 49 EB 7E CD C8 49 04 05 CA 4A 14 C5 ...nI.~..I...J..
49B0 CD 05 4B F1 E3 0A 72 49 C5 3D B6 06 30 D0 4F 7E ..K...rI.=..0.O~
49C0 91 BB 47 D8 43 09 CD FC 48 0A EF 1D 5F 23 7E 23 ..G.C...H..._#~#
49D0 66 6F E5 19 46 72 E3 C5 2B CD 32 13 CD CD 2E C1 fo..Fr..+.2.....
49E0 E1 70 C9 EB CD 07 44 2C C1 D1 C5 47 C9 CD 32 13 .p....D,...G..2.
49F0 CD 19 1A CD 7F 1D 3E 01 F5 CA 0F 4A F1 CD 8E 23 ......>....J...#
4A00 B7 CA 4A 14 F5 CD 07 44 2C CD 1D 1A CD 3A 29 CD ..J....D,....:).
4A10 07 44 2C E5 2A 27 0C E3 CD 1D 1A CD 07 44 29 E5 .D,.*'.......D).
4A20 CD C7 48 EB C1 E1 F1 D5 01 5C 27 C5 01 EF 1D C5 ..H.....\'.....
4A30 F5 D5 CD CD 48 D1 F1 47 CD 4F BE 3E 00 D0 1A B7 ....H..G.O.>....
4A40 78 C8 7E 23 46 23 66 68 06 00 09 91 47 C5 D5 E3 x.~#F#fh....G...
-
```

LISTING 2.

```
D61B0
61B0 28 C9 F5 C5 D5 E5 15 00 CD BE 61 00 D0 61 0A 00 (.........a..a..
61C0 8F 20 20 54 65 73 74 20 70 72 6F 67 72 61 6D 00 .  Test program.
61D0 F5 61 14 00 91 20 22 49 74 20 59 73 20 65 61 73 .a... "It is eas
61E0 59 65 72 20 74 6F 20 72 65 63 61 70 74 75 72 65 ier to recapture
61F0 20 61 20 22 00 1A 62 1E 00 91 20 22 EC 6F 73 74  a "..b... "lost
6200 20 70 72 6F 67 72 61 6D 20 74 68 61 6E 20 79 6F  program than yo
6210 75 20 74 68 69 6E 6B 2E 22 00 32 62 28 00 8F 20 u think.".2b(..
6220 20 54 65 73 74 20 63 61 6C 63 75 6C 61 74 69 6F  Test ca'culatio
6230 6E 00 3D 52 32 00 88 20 41 F0 0F 17 00 48 62 3C n.=b2.. A....Hb<
6240 00 88 20 42 F0 0F 43 00 56 62 46 00 91 20 41 F2 .. B..C.VbF.. A.
6250 42 2C 41 F4 42 00 00 00 1A 20 69 73 20 65 61 73 B,A.B.... is eas
6260 69 65 72 20 74 6F 20 72 65 63 61 70 74 75 72 65 ier to recapture
-00
A:MBASIC
```

LISTING 3.

GENESIS
COMPUTER CORPORATION

introduces...

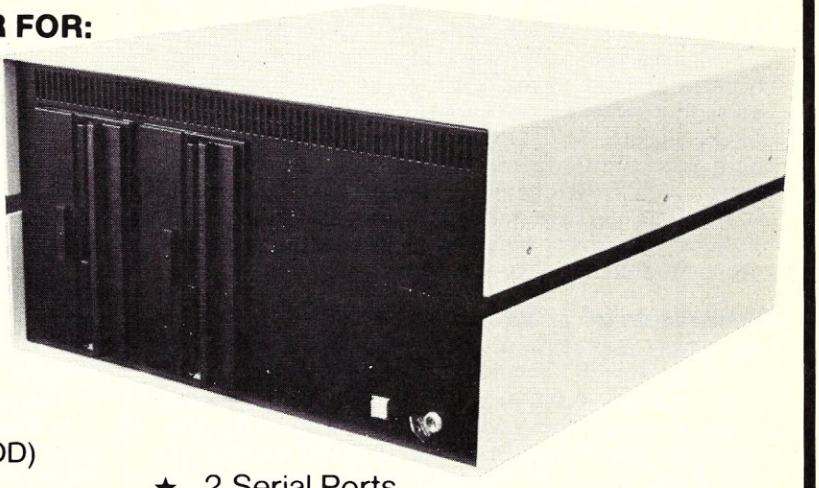
the Price & Performance Leader

the GENESIS 8D S-100 micro

A COMPLETE DUAL 8" DISK DRIVE S-100 MICROCOMPUTER
FOR ONLY... **$2395.00**

THE IDEAL MICROCOMPUTER FOR:

- ★ Business
- ★ Education
- ★ Industry
- ★ Development
- ★ Networking

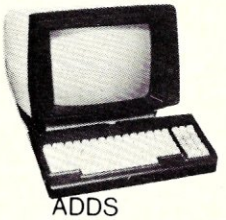


FEATURES:

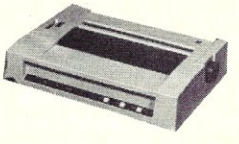
- ★ Z80A CPU 4MHz
- ★ 64K Dynamic RAM
- ★ 2 1.024 MB 8" disk drives (DSDD)
- ★ Floppy Disk Controller
- ★ CPM 2.2© Operating System (MPM© and TURBODOS® Available)
- ★ 10 Slot IEEE 696 Mainframe
- ★ 2 Serial Ports
- ★ 2 Parallel Ports
- ★ DMA
- ★ Keylock ON/OFF Switch

COMPLETE THE SYSTEM WITH OUR INEXPENSIVE TERMINAL AND PRINTER COMBINATION FOR ONLY

$925



ADDS
VIEWPOINT



C. ITOH PROWRITER

GENESIS
COMPUTER CORPORATION

1444 LINDEN ST.
P.O. BOX 1143
BETHLEHEM, PA 18018
(215) 861-0850

SCHOOLS AND BUSINESSES ASK ABOUT OUR LEASE/PURCHASE ARRANGEMENTS

A MICROSYSTEMS TUTORIAL

LISP for CP/M

The favorite language of the Artificial Intelligence community is now available to run under CP/M

by William G. Wong

LISP (LIST Processing) is the major development language for the computer science Artificial Intelligence (AI) community. It was first developed in the early 1960s by John McCarthy at the Massachusetts Institute of Technology (MIT). Lisp has been available on large mainframe computers, but it is now available under CP/M and CP/M-86 from a number of sources. This opens the world of Lisp and AI to everyone who has access to a microcomputer.

This review covers three implementations of Lisp for CP/M. The first is muLisp-80, created by the Soft Warehouse; it is distributed by the Microsoft Corporation and is available from LifeBoat Associates. The second is SuperSoft Lisp from SuperSoft, and the third is Stiff Upper Lisp by Harry Tennant, which is available from Tennant & Tennant Computing (3537 Ridgemoor Drive, Garland, TX 75042). The muLisp package is available for 16-bit machines; however, only the 8-bit CP/M versions were evaluated for this article.

Overview of Lisp

Lisp syntax and semantics are very simple; however, Lisp is not as well known as Fortran or Basic. For this reason, a quick discussion on Lisp is in order, since the examples in this review are given in Lisp. This discussion should not be taken as a full tutorial on Lisp. Reading one of the books in the bibliography should be a prerequisite for purchasing any version of Lisp, since the documentation provided with all of the packages reviewed here will be incomprehensible without one.

Lisp has two basic components: atoms and lists (of course). An atom is a number or name (sometimes called a symbol). More extensive implementations of Lisp have other types of atoms such as arrays and strings. A list is a vector whose elements are either atoms or lists. A list is printed within a set of parentheses, with the elements separated by spaces. The following are examples of three element lists:

```
(1 2 3)
(ELEPHANTS ARE PINK)
(1 (2 3) 4)
(+ (+ A B C) (+ 1 2 3))
(ONE (((TWO))) THREE)
```

Notice that the last three examples have lists nested as elements. Lisp does not place a limit to either the length of a list or the depth of nested lists.

The number of parentheses in a deeply nested list is one reason why Lisp is often called the language of Lists In Stuttered Parentheses. This should not be a problem, however, because implementations of Lisp will typically "pretty print" lists with indentations to make them easier to read. The following examples show a function definition, first with no special format and then in a "pretty printed" version. I will let you decide which of the following is easier to read.

```
(DEF EXAMPLE (X Y) (COND ((> Ø X)
(+ X Y)) ((= Ø X) Y) (T(- X Y))))
```

```
(DEF EXAMPLE (X Y)
     (COND ((> Ø X) (+ X Y))
           ((= Ø X) Y)
           (T (- X Y)) ) )
```

Lists are actually CONStructed from a simpler object called a "CONS" cell or a "dotted pair." "(FIRST . REST)" is a dotted pair where we have the FIRST part of a list separated from the REST of the list by the dot. Thus (1 . (2 . (3 .())))) is the dotted pair representation of (1 2 3), where () is the "empty list." The empty list is also printed as NIL, so the dotted pair representation could also be (1 . (2 . (3 . NIL))).

List syntax is used instead of the dotted pair syntax, since it significantly reduces the number of dots and parenthesis. A list that does not end with the empty list still retains the trailing dot, so (1 . (2 . 3)) is shown as the list (1 2 . 3). Essentially, the ". (" and the corresponding ")" can be removed from the dotted pair syntax to get the list syntax.

Everything is built around the lists and atoms, including expressions, function definitions, and data. An expression is what you find on the right side of a Basic assignment statement. Lisp expressions are sometimes called "symbolic expressions," or simply "s-expressions." Lisp uses a prefix syntax for expression, so the list (+ 1 2) represents the operation 1 + 2. In fact, unlike Basic and Fortran, which have expressions and statements, Lisp has only expressions.

Although the prefix syntax may be foreign to many people, it does have significant advantages. One advantage is that Lisp has no operator precedence to remember, since each expression is a list. The other advantage is that functions and parameters are easily identified, because the first item of a list will be the function and the rest will be the parameters for all lists being evaluated. This is also true for system-defined functions as well as user-defined functions.

The prefix notation is actually not too different from the typical function syntax in languages such as Pascal or Basic. The only difference is that the left parenthesis is moved to the left of the function name.

William G. Wong, 902B Merritt Drive, Somerville, NJ 08876

For example, Pascal and Basic would use "SIN (45)", while Lisp syntax would be "(SIN 45)". Note that parameters and results can be lists as well as atoms, depending upon the function.

Lisp's use of lists and atoms for data as well as for expressions and functions is very important, because it allows functions to be treated as data and data to be treated as functions. Languages like Basic and Pascal do not have this type of commonality between programs and data. Thus, a Lisp function can generate new functions that can be used immediately. In addition, a function definition can, if necessary, be imbedded in a data structure.

A function definition list is called a "lambda" expression. Lambda is a mathematical term used to designate a method of substitution. Assume the definition of "EXAMPLE" is the sample lambda expression:

```
(LAMBDA (X Y) (LIST 1 Y 2 X 3))
```

Evaluating (Example 10 20) would cause X and Y to be bound to 10 and 20 respectively. Lisp then evaluates the body of the lambda expression, which is the last item in the list. "LIST" is obviously the function in this case, and it is a Lisp function that evaluates its arguments and puts them in a list. In this case the list would be (1 20 2 10 3) because the variables X and Y are replaced by the values bound to them. So the substitution process is simply a matter of matching the parameters to the parameter list and evaluating the body.

Lisp functions can be called recursively because the parameter-binding process is always done with a new and unique set of variables, just as it is in more common languages like Pascal. The difference between Lisp and Pascal, with respect to function calls, is in the area of variable-name "scope." Pascal variable names have a "static scope," while Lisp names have a "dynamic scope." References to statically scoped variables are known at compile time, but dynamically scoped variable references can only be known when a program is actually running. In fact, starting from a different initial state can change the variable referenced in a dynamically scope environment such as Lisp. This difference in variable-name "scope" must be taken into account whenever you use Lisp.

Luckily there are some things that you do not have to worry about when using Lisp. In particular, you do not have to worry about "garbage." This is not the garbage referred to by the phrase "Garbage in—Garbage out." In Lisp, garbage consists of lists that cannot be accessed. To understand how lists can become inaccessible, a few Lisp functions need to be examined.

These functions are CONS, CAR, and CDR (pronounced "cudder"). These are the most basic functions in Lisp and will be found on all implementations. The CONS function takes two arguments and returns a "CONS" cell whose first element is the first argument and whose second argument is the rest of the cell. Therefore, (CONS 1 2) would return a list of (1 . 2). CONS can be used to build larger lists, as shown in the following examples:

```
Expression:     (CONS 1 (CONS 2 3))
Result:         (1 . (2 . 3))
List Syntax:    (1 2 . 3)

Expression:     (CONS 1 (CONS 2 (CONS 3 NIL)))
Result:         (1 . (2 . (3 . NIL)))
List Syntax:    (1 2 3)

Expression:     (CONS (CONS 1 (CONS 2 NIL)) (CONS 3 NIL))
Result:         ((1 . ( 2 . NIL )) . (3 . NIL))
List Syntax:    ((1 2) . (3))
```

Obviously the LIST function mentioned before uses the CONS function to build a list; however, CONS builds a list that does *not* make a list into garbage. This leads to the use of the CAR and CDR functions, which are the complement of CONS.

The CAR and CDR functions are used to access the first and remaining portions of a "CONS" cell respectively. The CAR and CDR functions are sometimes called "selector" functions because they "select" a part of a list. The following examples show how the CAR and CDR functions can be used to select portions of a list.

Expression	Result
A	(1 (A B C) (2 3 . X) 4)
(CAR A)	1
(CDR A)	((A B C) (2 3 . X) 4)
(CAR (CDR A))	(A B C)
(CDR (CDR A))	((2 3 . X) 4)
(CAR (CDR (CDR A)))	2
(CDR (CDR (CAR (CDR (CDR A)))))	X
(CONS (CAR A) (CAR (CDR A)))	(1 A B C)

One interesting about CAR and CDR is that what they return is not a copy of a list but a pointer to the element. This is different from the string selector functions in Basic like MID$ or LEFT$, which return a copy of a string. The operation of CAR and CDR allow Lisp to generate lists that share elements with other lists. The last result (1 A B C) shares the list (A B C) with the variable A.

But how does the use of CAR or CDR make garbage? Well, actually, this is not always the case. Garbage is created only when the list passed to one of these functions is not referenced elsewhere. Assume that the variable A in the previous example is a local variable in a function and that result of (CDR A) was returned by this function. The "CONS" cell referenced by A becomes garbage because A is not needed anymore. However, element 1 may not be garbage because there may be another reference to it in another list.

This leads to a simple definition of garbage in a Lisp system. "Garbage" is any "CONS" cell or atom to which there is no reference that is not itself garbage. This of course means that there is some root which is not garbage. In Lisp, the root is the list of objects that you want to keep. This list is called the OBLIST (i.e., object list).

Now I mentioned that in Lisp you do not have to

Lisp's use of lists and atoms for data as well as for expressions and functions allows functions to be treated as data, and data to be treated as functions. Languages such as Basic and Pascal do not have this type of commonality between programs and data.

worry about garbage. What happens to it, then? Well, Lisp has a "garbage collector," of course. The garbage collector will go through the Lisp universe and find all the garbage. It gives all the garbage "CONS" cells back to the Lisp system so they can be used in new lists generated by CONS. In fact, Lisp will normally call the garbage collector whenever it runs out of space.

In Pascal, there are pointers that could be used to make lists as in Lisp. However, you must keep track of all the lists and the garbage. This process is simple if each element has only one reference to it, as is the case in a basic tree structure. There are, however, many useful structures that are not trees; in these, an element may have more than one reference to it. An element is garbage only when *all* references to it are removed.

Pascal requires *you* to build your own garbage collector if you use lists. Building such a program is not too bad if there is only one reference to each element. But much of the power of Lisp lies in the fact that lists can be manipulated and referenced in arbitrary ways. In Pascal, keeping track of what is garbage would be very difficult; in addition, Pascal's strict type checking would require a different garbage collector for every data structure used.

In Lisp, there is just one garbage collector that is available to all Lisp programs, and that operates effectively on all data types. Nevertheless, garbage collection does not come free in Lisp any more than in Pascal. It takes time to do the operation; the amount of time depends upon the number of lists and the amount of garbage, but can become quite large. The Lisp garbage collector is automatically started when you run out of space, which greatly simplifies the use of lists. This is important because lists are Lisp's bread and butter.

In fact, BREAD and BUTTER can be atoms and, in Lisp, atoms can have "properties" that are not found in procedure-oriented languages such as Basic or Pascal. Lisp keeps the properties of an atom on a "property list"—that is, an ordinary list that can be accessed using special functions that retrieve, add, or delete properties from the list.

Allowing atoms to have characteristics that are accessible through the property list is one way in which Lisp allows us to model the real world. For example, BUTTER could have the properties COLOR, SALTED, and WHIPPED, the current values being YELLOW, T (yes), NIL (no). T and NIL are the Lisp boolean values for true and false. Of course BUTTER can have other properties and need not have the ones listed here.

Although this "short" overview is getting a bit long, there are two other functions, found in all implementations of Lisp, that need to be mentioned. The first is the QUOTE function. QUOTE takes one argument and returns this argument unchanged, without evaluation. For example, (QUOTE (+ 1 2)) returns the list (+ 1 2), *not* the atom 3. QUOTE can sometimes be shortened to an apostrophe so that '(+ 1 2) is the same as (QUOTE (+ 1 2)).

The second function is the COND function, which is equivalent to a Pascal "IF" statement, but more powerful. It is sometimes called a "McCarthy conditional" after John McCarthy. The COND function takes one or more lists as arguments, and evaluates them in a special way. It evaluates the first element of the list, which is the "test." If the result is NIL, then the next argument is tested in the same fashion. Otherwise, the remaining portion of the list is evaluated, the result of this evaluation being the result returned by the COND function. COND returns NIL if all tests return NIL. An example would be:

```
(COND ((> A Ø) 1)
      ((= A Ø) Ø)
      ((< A Ø) -1) )
```

This is really the absolute value function definition, where A would be the parameter. The result of the first test would be T (true) if A is a positive number, in which case the result would be the number one. The results of the other tests should be obvious. In addition, a non-numeric result would return NIL if the comparison functions returned NIL for non-numeric arguments. The last comment is implementation dependent.

COND provides a compact syntax, since it eliminates the extraneous IFs and ELSEs found in Pascal. COND also replaces the GOTO branching statements typically found in Basic. The power of COND is enhanced by fact that the conditional part of a COND function can be any Lisp expression, as in the following example, which returns the absolute value of X. Negative numbers are those which are not positive, and the absolute value of a negative number is found by negating it.

```
(COND ((< Ø X) X)
      (T (- X)))
```

You should now be able to take a look at a set of Lisp functions and be able to get a good idea of what is going on. Of course, having a description of the built-in functions would help immensely. This information can typically be found in the books on Lisp and in the user documentation supplied with the various Lisp packages.

With the basics of Lisp well in hand, it is now time to take a look at some of the implementations of Lisp under CP/M. The best place to start is with the documentation supplied with each package.

Documentation

Unfortunately, documentation is the one area where very few software packages shine. Someday, perhaps, software developers will get it into their heads that the documentation is as important as the software itself. The manuals supplied with these three Lisp

A single garbage collector is available to all programs. It starts automatically when you run out of space, which greatly simplifies the use of lists.

Move up to Microsystems

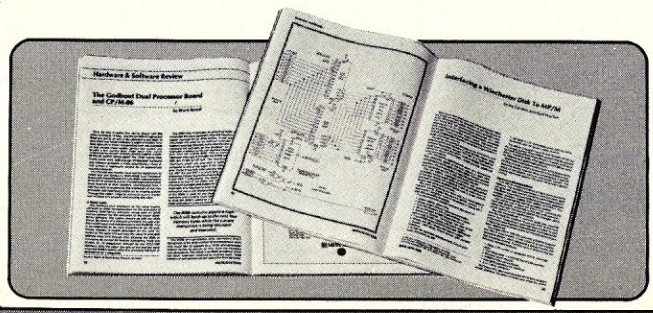
THE JOURNAL FOR SOPHISTICATED MICROCOMPUTER USERS.

Microsystems is not, in any sense, for beginners. Unlike "personal computer" magazines, *Microsystems* is written *exclusively* for expert users and designers in the software and support hardware field.

If you need a reliable, high-level, up-to-date journal devoted to CP/M,® MP/M,® MS-DOS,™ UNIX,® XENIX,™ OASIS and other operating systems, then *Microsystems* is for you. We feature tutorials on the modification and special use of systems, offer short-cut routines, show you debugging procedures, techniques for interfacing, and much more. Our one purpose is to keep the advanced microcomputer professional at the leading edge of DOS development and use.

In a few recent issues of Microsystems you would have seen:

- Reviews of four C compilers: BDS C, Small C, tiny c TWO™ and Whitesmiths C.
- A programming style comparison: Digital Research PL/I-80™ vs. Microsoft BASIC.
- Twenty-seven 16-Bit DOS compared.
- Virtual segment procedures under UCSD Pascal.
- An in-depth look at MP/M-80 II.™
- A simple 6-byte hexadecimal ASCII conversion routine.
- Using CP/M's undocumented "Autoload" feature.
- An explanation of double density disk controllers.



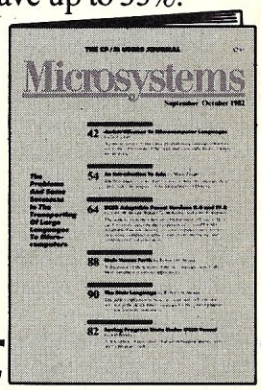
- A comparison of five popular S-100 disk controller cards.
- OS-1—a UNIX-like DOS with a CP/M adapter.

Every issue of *Microsystems* keeps you at the forefront of the industry in customizing software and adapting hardware to meet new requirements and standards. You'll also find industry news, book and new product reviews, plus our regular *Software Directory* listing newly available programs.

To get the high-level systems information you want, join the club of serious microcomputer professionals who subscribe to *Microsystems.* Use the coupon below to order, and save up to 33%!

PROFESSIONALS PLEASE NOTE:

Your subscription to Microsystems may be tax deductible. Check with your accountant.



Microsystems CN 1987
Morristown, New Jersey 07960

YES, enter my subscription to *Microsystems* for:

☐ 12 issues at $19.97—I save 20%.

☐ 24 issues at $36.97—I save 26%.

☐ 36 issues at $49.97—I save 33%.

Savings based on full 12-issue subscription price of $24.97.

Mr.
Ms. _____ 4S23
(please print full name)

Company _____

Address _____

City _____

State _____ Zip _____

CHECK ONE: ☐ Payment enclosed. ☐ Bill me later.

Offer valid in U.S. and possessions only. Please allow 30 to 60 days for delivery of first issue. Satisfaction guaranteed or a prompt refund for unmailed issues.

NEW SUBSCRIBERS ONLY

packages are very short; they also assume that you know quite a bit about Lisp. None of the manuals can be classed as more than barely adequate. Given this major criticism, the internal quality of the documentation is still the bottom line, and it differs significantly among the three packages.

muLisp-80 documentation is better than most in terms of organization. It includes a good index and bibliography, which no documentation should be without. In fact, it contains separate indices for Lisp functions, variables, and concepts, along with a description of what is on the distribution disk. It also tells the user (near the *beginning* of the manual) how to back up the original disk and how to enter and exit muLisp-80.

Unfortunately, muLisp-80 uses the Backus-Naur Form (BNF) syntax to describe Lisp, instead of using Lisp itself. This is extremely confusing and makes the manual difficult to use, since BNF is different from the Lisp syntax. The descriptions are also devoid of useful examples that would help first-time Lisp users. Calling documentation a "user's manual" is no excuse for omitting examples or for using unhelpful variable names like X, Y, and Z. Because of these faults, the description of the internal data structures and machine language interface is confusing to the point of being almost useless. Luckily, interfacing muLisp-80 to the machine should not be required.

muLisp-80 documentation also mentions a support program called CONTINUE. This "program" must be used with *extreme care*! It is actually a kludge that is required because muLisp-80 can exit to CP/M very easily, especially when you really do not want it to! Even worse, you could lose a lot of work in the process. CONTINUE.COM is actually a zero-length program file that can be run immediately upon exit from muLisp-80. The result is that no program is actually loaded; hence the previously loaded program, hopefully muLisp-80, will start to run again. In fact, muLisp-80 configures itself so as not to reinitialize itself; therefore, if you re-enter muLisp-80 via CONTINUE.COM, your Lisp program and its data will be preserved.

This would all be fine and dandy if the procedure were explained well, but it is not. CONTINUE was actually put in so that you could use built-in CP/M commands like DIR and ERA, which are operations not supported by muLisp-80. The only alternative to using CONTINUE.COM would be to exit the current Lisp session, use the commands, and then reload muLisp-80 along with your functions. There is a catch, however. CONTINUE works properly only for the built-in CP/M commands. Using CONTINUE after running a .COM utility (PIP, for example) will restart that utility, *not* muLisp-80—thus you will lose your Lisp program and all its data.

The **SuperSoft Lisp** manual is one of the worst I have ever seen, and the nondisclosure agreement is almost frightening. This agreement indicates that none of the underlying concepts, documentation, user interface, or code fragments can be disclosed, so I am not sure if I can even review this package without violating the agreement. This alone would probably keep me from using it for anything useful.

The rest of the documentation is as bad as the nondisclosure agreement. It contains no index or bibliography. It does not tell you how to back up the distribution diskette (or even why you should do so), how to start the program, or how to exit. Even a directory of the contents of the distribution disk is absent.

The document requires a *very* good background in Lisp just to understand it. Examples are almost nonexistent, and those which are present have no decent explanation. The text is disorganized, and even the text alignment is terrible. Margins seem to have gone the way of the dodo bird. The definitions of the Lisp functions are not "pretty printed" but mashed into an entire line, making them unreadable. The sections on internal structure and machine language interface suffer not only from these problems but also from a lack of clarity.

The **Stiff Upper Lisp** manual is one of the better ones. The implementation was based upon a commonly used dialect of Lisp called MacLisp, which originated at MIT. It is a subset and has some differences and extensions. The best thing about this document is that it references the book (included in the package) entitled *LISP* by Winston and Horn (see bibliography). It even has an appendix in the back that indicates the differences between the book and implementation. The location of the differences in the book are noted, and additional examples are given which show how the same thing can be done using the Stiff Upper Lisp package. In fact, there is a cross-reference between all the MacLisp functions and the corresponding Stiff Upper Lisp functions.

This is not to say that the Stiff Upper Lisp documentation is perfect. Even though it references one book, it does lack a bibliography. It, too, does not tell you how to start or exit the program. Initial use of the system is by trial and error. You have to guess that you start Lisp by typing UPLISP, and that you return to CP/M by entering a control-C in response to any Lisp prompt (very bad usage). This omission is unforgivable, since a sample session is one of the first sections in the manual. The sample is poorly placed because it comes before any information about Lisp itself. The order of the other sections is also poor, with extremely technical information presented before any kind of general information or overview.

The presence of examples is a good feature, although they are not included with every function definition. The most useful one is in the area of the "auto-loading" of functions from disk. An autoloaded function actually has a small resident definition responsible for loading the main function from the disk the first time the function is called. Once the main function has been loaded, the small resident

Someday software developers will get it into their heads that documentation is as important as the software.

function becomes garbage, because all references to that function are now made to the main function definition.

The advantage comes in the form of space savings, since unused functions are disk resident. Only those functions that are actually used need be loaded into memory. Although most Lisp implementations support this type of loading, the operation is not always described. This is too bad, since auto-loading is one of the most useful features of Lisp. It is especially powerful in that you are able to write the auto-load function to your own specifications. Pascal and Fortran sometimes have overlay operations available, but they usually cannot be modified by the user.

In all three packages, the bottom line in the documentation area is that a separate book on Lisp is required for all but the expert Lisp user. Stiff Upper Lisp wins this round, with muLisp-80 coming up second. I just hope that the next edition of software documentation improves significantly.

List compatibility and enhancement

Unfortunately there is no Lisp standard, although projects are underway to develop "Standard Lisp" and "Common Lisp." In the meantime, the major dialects of Lisp include MacLisp, UCI/Rutgers Lisp, Franz Lisp, and InterLisp. Stiff Upper Lisp does the best in the area of compatibility, since it is a pretty good subset of MacLisp. SuperSoft Lisp is a subset of the original Lisp 1.5 specification, but the set is too small to be useful. A discussion of the enhancements in each package is the best way to get an idea of the differences between the packages.

The **muLisp-80** designers did a few things that prevent this implementation from being a true subset of any Lisp, which makes it difficult to transport programs between muLisp-80 and other Lisp systems. First of all, the PROG, GO, and RETURN functions do not exist in muLisp-80 and are replaced by an unconventional form of the LOOP function. The PROG function provides a conventional statement type execution mode that differs from the normal Lisp functional execution mode. Its absence simply makes it difficult to move Lisp programs from other systems to muLisp-80. As a general rule, however, the PROG function should be avoided, since it uses the nonstructured GOTO statement (GO).

"Auto-quoting" is another muLisp-80 feature that can cause confusion. This feature causes the initial value of an atom to be itself instead of NIL, the conventional value. This would be like having a Basic string variable initialized to the name of the variable instead of to a zero-length string. The feature can lead to confusion on the user's part.

For example, a typical application of the auto-quote option would be using the atom MYFILE as a parameter to the function that reads a file. If MYFILE has not been used before, its value would be MYFILE instead of NIL because of the auto-quote option. The function would use the value of MYFILE (which is MYFILE) as the file name. This would work well; however, if the value of MYFILE were changed to something else (e.g., NIL), then a subsequent use of MYFILE with the same function would give different results because the system would look for a file named NIL.

muLisp-80 also has the bad habit of never generating an error, even if the function is undefined. Evaluation of a list whose function is not defined causes the list to be returned as the value of the evaluation. Thus, the evaluation of the list (+ 1 2) returns (+ 1 2), since "+" is not a defined function. The proper muLisp-80 function would be (PLUS 1 2). Debugging functions under these conditions is extremely difficult, especially for the first time. Typographical errors are very hard to find.

The way in which muLisp-80 binds function arguments to parameters can also cause problems. First, any extra arguments given to a function are ignored; second, parameters are bound to NIL if there is an insufficient number of arguments. The first instance causes problems if you forget the proper number of arguments, because the functions will always work. The second was required since the PROG function was eliminated, and there is no easy way of creating local variables that are not parameters. This mode of operation means you can call a function with just about anything, and get some sort of result back.

Mulisp-80 has one last feature that makes the programs shorter, but harder to understand and difficult to migrate. The body of a function definition is really evaluated as if it were the argument of the COND function. This implied COND function saves space, which is important in a small version of Lisp; but it also leads to a confusing Lisp syntax (see Figure 1). The addition of these muLisp-80 features was done with good intentions, but they lead to problems when debugging and running programs.

SuperSoft Lisp provides the FUNCTION function, which can be used to create a functional argument known as a FUNARG. A number of papers have been written on the subject of the FUNARG, both for and against it, but it is a useful device. The muLisp-80 implementation does not provide this function; Stiff Upper Lisp provides a more powerful operation called a "closure" that will be discussed in more detail later.

SuperSoft Lisp also implements a special "property" called APVAL, which seems to mean "apparent value." It is a feature not found in any other Lisp that I know of. The existence of APVAL in the property list of an atom causes the system to bypass the normal evaluation mechanism of Lisp, so that the value of an atom will always be the property value associated with APVAL if this exists, even if the atom has been assigned another value. Figure 2 shows how APVAL can be used to implement a lock function that cannot be undone by simply changing the value

There is no Lisp standard, although projects are under way to develop "Standard Lisp" and "Common Lisp." Major dialects include MacLisp, UCI/Rutgers Lisp, and Franz Lisp.

of the atom using the normal functions. This seems to be a useful example, but it is confusing the use. The value of this feature is dubious.

Stiff Upper Lisp has a number of features not found in the other Lisps reviewed here, but some of them are found in MacLisp, which is very convenient. This is not to say that all the features are useful, but most are. First let us look at the useful items.

The user interface to Lisp consists of a simple loop that prompts the user for input, reads the input, evaluates the input using Lisp's normal function evaluation mode, and prints the result of the evaluation. This is sometimes called the Lisp top level, or the READ-EVAL-PRINT loop. It is simple and to the point. The value of an atom can be found by simply typing the name. Functions are started by entering an "s-expression."

Stiff Upper Lisp uses a different prompt each time, for a special reason. The prompt consists of an asterisk followed by a number that is incremented after each input. The input is bound to the atom that is the prompt, and the result is bound to an atom with the same number but prefixed by an equal sign. The following is a sample of this interface.

```
*1  (+  1  2)
3
*2  (+  3  4)
7
*3  (+  =1  =2)
1Ø
*4  *3
(+  =1  =2)
```

In this case, *1, *2, and *3 are bound to (+ 1 2), (+ 3 4), and (+ =1 =2) respectively; and the results =1, =2, and =3 are bound to 3, 7, and 10 respectively. The input to prompt *4 returns the input from prompt *3.

The advantages of keeping a record of previous inputs and results should be obvious. Not only can you use results generated previously, but you can also use the resident editor to change any inputs or the results, thereby allowing you to correct mistakes. For example, (EDIT *3) would allow you to edit a previous entry. Only the last eight sets are saved, so as not to use up too much space.

The user interface is further enhanced by the inclusion of a HELP function. The main HELP function resides on disk and is brought in using the "autoload" procedure. It gets the bulk of the text for the help function from the disk. This makes things slower, but it is acceptable even on a floppy disk based system. It provides an invaluable aid to the user, novice or expert. The only gripe is that it tries to match on exact atom names, so you need to know what you are looking for.

Stiff Upper Lisp also has "read macros." These are single characters that perform a special function if encountered in the input stream. You make a character into a read macro by adding a function definition that has no parameters to the property list of the respective character. This function is called whenever the character is read using the normal input routines. The function can then read items from the input stream and return a result. If the result is NIL, then nothing is added to the input stream; otherwise, the atom or list is added to the input stream. This is one function compatible with MacLisp and extremely useful. The QUOTE function is a standard use for "read macros." For example, the following inputs would be equivalent if the quote character (') has the standard "read macro" definition, which reads the next item in the input stream and returns a list in which the first element is the atom QUOTE. More complex macros can be created.

```
Normal Input          Read Macro Equivalent
============          =====================
(QUOTE 1)              '1
(QUOTE (1 2 3 4))      '(1 2 3 4)
```

Stiff Upper Lisp also includes a "pretty printer" that can be used for displaying information or saving definitions on disk. A "pretty print" function really needs to be part of any Lisp system, just to ensure readability.

Stiff Upper Lisp supports a CLOSURE that is a more powerful version of the FUNARG. A CLOSURE is a special type of Lisp function: it does not use the normal "dynamic scope" access method, but instead has an "association list" (a-list) containing all variable bindings that are not parameters to a CLOSURE function. An a-list is a list of elements. Each element is a list whose CAR is an atom and whose CDR is the associated value. The following is the a-list where X is 1, Y is 2, and Z is the list (A B C).

```
((X . 1)
 (Y . 2)
 (Z A B C))
```

A CLOSURE retains its a-list and the associated values between evaluations. The example given in the documentation is for a random-number generator function that has the random number seed as a local CLOSURE variable. In this example, the seed value is changed each time the CLOSURE is called.

CLOSUREs have one major advantage over normal Lisp lambda functions. If the last function evaluated in a CLOSURE function is to another CLOSURE, then the result of the last evaluation would be the result for the initial CLOSURE. Normally Lisp, like Pascal, would keep the information about the initial function call on the internal evaluation stack. However, since the last function call is a CLOSURE, it does not need this information because CLOSUREs use the a-list to access variables other than parameters. The Stiff Upper Lisp system is very smart: it discards the state of the initial CLOSURE and evaluates the last CLOSURE, returning the result directly to the function that called the initial

The user interface to Stiff Upper Lisp consists of a simple loop that prompts the user for input, evaluates it, then prints the results.

☐ APPLE ☐ EAGLE ☐ IBM, PC ☐ NORTHSTAR ☐ TRS-80 II 8" ☐ OSBORNE ☐ FRANKLIN ☐ STANDARD CP/M 8" ☐

☐ ACCOUNTS PAYABLE ☐ MAILING ☐ SPELLING ☐ COMMUNICATIONS ☐ REAL ESTATE ☐ TAXES ☐ STOCK MARKET ☐ INVENTORIES ☐ HEATH/ZENITH 89 8"

☐ XEROX 820 ☐ GAMES ☐ COMPILERS ☐ UTILITIES ☐ BUSINESS ☐ ACCOUNTING ☐ SPREAD SHEETS ☐ MEDICAL ☐ DENTAL ☐ PAYROLL ☐ GENERAL LEDGER

RENT SOFTWARE BEFORE YOU BUY!

from our

SOFTWARE RENTAL LIBRARY

You can now RENT the most popular software available for just 20-25%* of Manufacturers' Retail Price

- Eliminate the risk—rent first!
- 100% of rental fee applies toward purchase
- All purchases are 20% Off of Manufacturer's Suggested List
- Rentals are for 7-days (plus 3 days grace for return shipping)

SPECIAL INTRODUCTORY OFFER

There are now 2 different plans to choose from:

Join the **Game Group,** $25.00 per year and rent as many games as you like for only 20-25% of Mfrs. Sugg. Retail Price.* Minimum order, 3 game rentals	Join the **Business Group,** $50.00 per year and rent as many business application programs as you like for only 20-25% of Mfrs. Sugg. Retail Price.*

REMEMBER, THESE ARE NOT DEMOS, BUT ORIGINAL UNRESTRICTED SOFTWARE PROGRAMS

(complete with manuals in original manufacturers' packages)

To Immediately Order, or for more information:

UNITED COMPUTER CORP.
Software Rental Library
Culver City, California

Toll Free CALL 1-800 992-7777
In California CALL 1-800 992-8888
In L.A. County CALL 1-213 823-4400

Canadian Orders Welcome

MasterCard AMERICAN EXPRESS VISA

*Plus postage and handling.

☐ EDUCATIONAL ☐ ACCOUNTS RECEIVABLE ☐ WORD PROCESSORS ☐ DATA BASES ☐ LANGUAGES ☐ GRAPHICS ☐

CIRCLE 4 ON READER SERVICE CARD

CLOSURE.

This may seem a bit confusing, but it means that a CLOSURE can call another CLOSURE any number of times, and the internal stack will not grow one bit. CLOSUREs can replace the LOOP and PROG functions and provide a better description of operations being performed; they can also be as efficient as the archaic functions they replace.

Numbers, strings, and I/O

None of the Lisps reviewed here has support for floating point numbers. This makes it difficult to use them for general-purpose programs, although many AI programs use only integers. SuperSoft and Stiff Upper Lisp have signed 16-bit integers, but muLisp-80 has a unique form of integer that can be one to 254 bytes in length (see Figure 3) muLisp-80 makes programs possible that cannot even be attempted in conventional languages such as Fortran, which have only double-precision numbers. Imagine, more than 600 decimal digits of accuracy!

String handling for all three versions is extremely poor (Figure 4). Atom names are strings that can be converted to and from lists where each element is a single character. This is extremely cumbersome and time consuming. The addition of string functions would greatly enhance any of these versions of Lisp. Control and special characters can be included in atom names. Note, however, that muLisp-80 does not accept the ESC character (27 decimal) because it is also used to interrupt processing by the user. This makes it extremely difficult to generate terminal escape sequences for cursor positioning.

I/O support and resident Lisp editors

Traditionally, Lisp has had bad I/O support, and these three systems adhere to that tradition. The **muLisp-80** implementation provides the user with the usual single I/O device and, using the ReaD Select (RDS) and WRite Select (WRS) functions, allows you to select whether the device shall be the console or a disk file. All information must be written to or read from a disk file once it is opened, at which point I/O is from the console. This allows functions that can be processed by the normal CP/M text editors to be loaded and saved in ASCII text form. There are no functions to rename or erase a file.

There is also a muLisp-80 screen editor, written in Lisp, that can be modified by the user. Some documentation is provided for just this purpose, but a good knowledge of Lisp is required to make changes. The resident screen editor assumes a very dumb terminal, so screen updates often cause an entire line or page to be rewritten. Nevertheless, the usefulness of a screen editor greatly outweighs this annoyance. The editor also has a menu mode that lets you save and load functions, set and reset the trace mode on a function, and has the ability to edit a function, a variable, or a property.

There is also a single-key entry that will put muLisp-80 back into CP/M. The CONTINUE.COM program must then be used to get back into muLisp-80; otherwise you lose everything. Note that if you do not have CONTINUE.COM available when you exit, and you do not want to lose all your work, you can save the entire session as a restarted by entering the file name. You will need to know the size of the CP/M Transient Program Area (TPA) in order to use the CP/M SAVE command successfully. Check your CP/M documentation for more information on the CP/M SAVE command.

MuLisp-80 also provides the ability to save and restore the entire state of the system on disk, but in a more orderly fashion. The SAVE and LOAD functions operate *much* faster than the RDS or WRS functions; however, the SAVE/LOAD file format is binary. This means that it cannot be processed by the CP/M text editors, but luckily you can always LOAD a file and then save the functions as an ASCII text file using the WRS function.

SuperSoft Lisp is extremely poor in both the I/O support and resident editor area. It does have an erase function, but only with files of type LSP. The LOAD function operates like the muLisp-80 RDS function, but the corresponding output function writes only ONE list per file. Even worse, the list is written as a mass of characters, thus making it almost impossible to process the file with a CP/M text editor. About the only saving grace (excuse the pun!) is the SAVEFNS function, which takes a list of functions to be saved as its argument. Unfortunately, SAVEFNS saves each function in a separate file, thereby wasting quite a bit of disk space on a typical CP/M system.

The SuperSoft resident editor operates only on function definitions. It is difficult to use and will abort if you enter bad information. This alone makes it a pain to use, because it does not save a copy of the changes. Using a CP/M text editor and loading the functions is the only way to use SuperSoft Lisp, and this process is tedious at best. The documentation for the editor is just as bad.

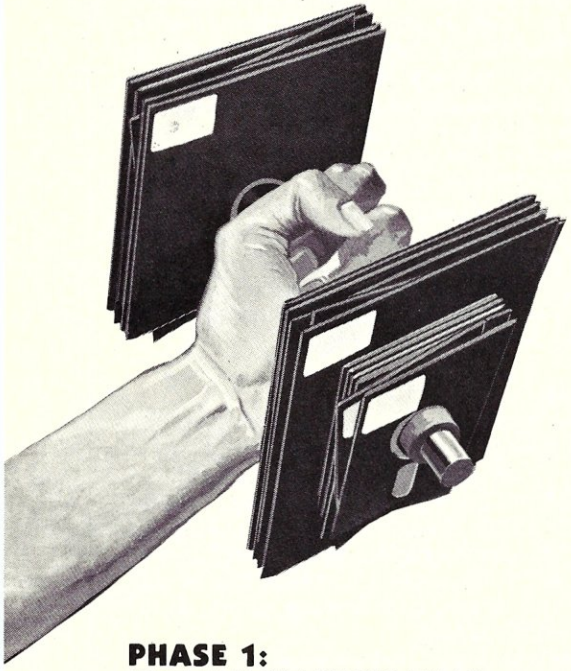
Stiff Upper Lisp I/O support has both good and bad points. You can save and restore the state of the system in binary form, but the method is hard to use and the documentation on it is very poor. The good points are that it does have full-fledged rename and erase functions, along with fair ASCII text file manipulation.

Stiff Upper Lisp uses the same method of ASCII text file access as muLisp-80; however, it can also append text to files. A "pretty printer" is used to generate files, and there are a number of user-alterable variables to control formatting parameters such as line width. Comments can appear in a file but, to save space, they are not loaded. Saving multiple functions in a file is reasonable easy.

The Stiff Upper Lisp resident editor is similar to

None of the Lisps reviewed here has support for floating-point numbers. String handling for all versions is extremely poor.

WHY YOU NEED MICROSHELL™: NEW POWER FOR CP/M® SYSTEMS



PHASE 1: "LITTLE" FEATURES THAT ADD UP TO BIG TIME-SAVERS

You appreciate MicroShell from the moment you run it. Suddenly your CP/M computer seems much smarter, more powerful, more tolerant of your errors. That's because MicroShell adds features to CP/M normally found only on more powerful operating systems.

You spend a lot of time with your computer. MicroShell makes your time more productive. Take the warm-start after you run a program or when you change disks. MicroShell eliminates it — you can run another program immediately. Change a disk — even in drive A — and a quick command logs it in. No more time lost booting up — no more juggling "system" disks with CP/M on them!

Or how about finding files? Under CP/M you have to know what disk the file is on and then change disks or preface the file name with the disk drive. And if you have a hard disk and are using CP/M user numbers, you also need the user number the file is in. MicroShell automatically searches for files on all disks, or selected disks — your option. And it works even when a program calls for a file. So you can run WordStar, for instance, from any disk or user number and MicroShell finds the necessary WordStar files.

Running several programs in a row? MicroShell lets you put a string of CP/M commands on one line. You end the distraction of entering commands, waiting for execution, then entering the next command.

Make a mistake? With CP/M you have to start over. Add MicroShell and you can edit the last command line (with simple WordStar commands). MicroShell saves the last command string, so you can run through a series several times — or change commands slightly and run again — without retyping.

MicroShell doesn't affect how your CP/M runs. It helps CP/M, but it doesn't get in the way. And your system probably never misses the 8K bytes MicroShell uses because most all popular programs run with 48K. So in a 64K CP/M system, there's plenty of room. For the rare, big program, it's easy to exit MicroShell, run the program and re-enter MicroShell.

You appreciate MicroShell from the start. It has powerful features so easy-to-use that the MicroShell manual tells you "How to Run MicroShell Without Reading the Manual".

PHASE II: NEVER RE-ENTER DATA AGAIN

But please — read all 75 pages of the manual and get to know MicroShell in depth. You appreciate MicroShell more and more when you do the big projects you bought your micro for.

Take the time you turn a spreadsheet analysis into a table for a report. Without MicroShell you have to re-enter data in a file for your word processing software. Why the duplication of effort? MicroShell lets you *redirect* the output of any program to a file — no duplicated keystrokes.

MicroShell captures what appears on your CRT and sends it anywhere. Send data to your printer for hard-copy or save it on disk. Or capture that report your accounting program sends to the printer. MicroShell tames your data-handling jobs — it even "slows down" to feed programs that process input slower than MicroShell sends it.

When you review your work, MicroShell has a "TYP" command that displays 23 lines (an average CRT screen) then stops until you hit a key. If you've ever scanned a long document whipping by at 10 or 20 lines per second, you know you need this feature. Or have you tried to hit control-S in time to stop the display and missed?

For more information about MicroShell, see the following reviews:
Christopher Kern, BYTE, December, 1982
Alan R. Miller, Interface Age, July, 1982
David Fiedler, Microsystems, January, 1983

PHASE III: THE SYSTEM EVOLVES WITH YOU

If you're in computers for the long haul, MicroShell will grow on you. MicroShell helps you teach new users. And it lets you develop new capabilities in a clear and logical way.

MicroShell lets you set up "menus" to make your system "friendly" for non-technical users. And fast — no slow CP/M "submit" files used. As you develop your own software, MicroShell's "pipe" feature lets you grow into ever more sophisticated applications. It links simple programs together by directing the output of one program to the input of another.

Just as MicroShell lets you combine commands on one line, it lets you save any command series in a file. MicroShell reads and executes commands directly from the file — no slow, intermediate steps. For example, you can compile, link and run a program with one command, "compile filename". MicroShell executes all the commands from a file you've built.

And you can call other command files from within a command file to enlarge on what you've built. You can set conditional commands with an "IF" statement. Or you can set command files to call for input from a user and act on it. Before you know it, you have a system much more powerful and friendly than bare CP/M.

SO DON'T WASTE ANY MORE TIME WITH YOUR CP/M SYSTEM — ADD MICROSHELL TO IT TODAY.

ORDER LINE 800-368-3359

Available for all popular 8080, 8085 or Z80, CP/M 2.2 systems on 8" or 5¼" diskette.

Price: $150.00 ($25.00 manual only). Foreign orders add $20.00 for Air Mail shipment.

NEW GENERATION SYSTEMS, inc.

2153 Golf Course Drive
Reston, VA 22091
(703) 476-9143

CP/M®, MP/M™ — Trademarks of Digital Research.
WordStar® — Trademark of MicroPro.

CIRCLE 19 ON READER SERVICE CARD

the SuperSoft Lisp editor, except that it is more powerful and traps some, but not all, errors. The editor source is available and can be modified to trap all errors. You can edit functions, values, and lists in general. It is not as easy to use as the muLisp-80 editor, but it is adequate for the job.

Error handling

Error handling in **muLisp-80** is very poor. Evaluation of an undefined function causes the list to be returned as the result, with no indication of an error. For this reason, there are no error-trapping facilities, and this makes debugging extremely difficult. During function evaluation, the system periodically checks the console for an escape character; if one is present, the system issues a prompt to the user, at which point the user has the choice of continuing the computation or exiting to the top level of CP/M or muLisp-80.

SuperSoft Lisp has no error-trapping facilities, and all error messages consist of an error number. The error numbers are described, but the corresponding documentation is so poor that a novice would be thoroughly confused in many instances.

Stiff Upper Lisp error handling is a joy to use—far superior to the error handling of the other two implementations. It has a BREAK package that allows you to trace and set break points on any function. It also has a single-step mode and provides access to the evaluation stack. There are also SPELL and HELP functions that can be accessed at any time.

Errors can be trapped in the Stiff Upper Lisp system using the MacLisp-compatible functions CATCH and THROW. These provide a very powerful general-purpose tool that also supports error trapping. The parameters to CATCH and THROW are a "label" and an "expression." The result from a CATCH function will be the value of the expression if a THROW function is not executed; otherwise, the result is the expression in a THROW function. The CATCH and THROW functions are matched by the label, and a NIL label matches anything. A THROW will skip over labels that do not match.

The system signals an error by using a THROW label of ERROR, and the top level of the Stiff Upper Lisp uses a CATCH with a NIL label to ensure that all errors are caught. You can use CATCH to write functions that handle all errors, too. One AI application for the CATCH and THROW functions would be a search function that would return the result via THROW directly to the corresponding CATCH function. The functions can also be quite useful in providing a user-friendly interface.

Garbage collectors

The **muLisp-80** garbage collector is automatic and compacts the garbage into the top of memory. The compacting garbage collector speeds up execution, because new CONS cells are simply taken from the top of memory. The collector can be initiated manually, and it may optionally print statistics on the amount of free space.

The presence of the **SuperSoft Lisp** garbage collector is hidden in the machine representation area. The description is very bad. It can be started manually, but this has to be done by using the POKE function to change absolute memory locations. Luckily, the garbage collector works automatically; this should be good enough for most applications.

The **Stiff Upper Lisp** garbage collector can be started manually through a function. It is a typical implementation with no special features.

The garbage collection time for all the systems is less than one second and usually does not interfere with the speed of operation. Manual initiation is possible; however, the automatic nature of the garbage collector usually makes this unnecessary.

Performance

Flexibility is the area where Lisp excels with its list data structures. Execution speed of loops and arithmetic compares favorably with that of Basic interpreters, although compilers will run rings around Lisp in numeric calculations. In general, muLisp-80 runs about as fast as MicroSoft Basic; Stiff Upper Lisp and SuperSoft Lisp run at about half that speed. None of the Lisp systems had floating point, so all tests were run with integers.

I/O to the terminal was just as good as Basic or Pascal in all cases, but all the Lisps reviewed are very deficient in the area of formatted output. Functions for printing lists without parentheses normally have to be written by the user, but this is not a big problem in an extensible language like Lisp.

All the Lisp systems compared very poorly with Basic when doing string or array-oriented operations. This was to be expected because the string-handling functions in Lisp are very poor. Arrays have to be simulated using lists; however, the lists must be accessed in a sequential fashion, which is slower than indexing an array. Some implementations of Lisp do provide strings and arrays implemented in the normal fashion.

A big problem with all the packages is space for the Lisp lists and atoms. They all have around 6 to 8 thousand "CONS" cells; this might sound like a lot, but they disappear quite fast. A typical function definition can consume around 50 cells. It is sufficient to do quite a bit of work and is more than acceptable for anyone wishing to learn Lisp. Also, the garbage collection overhead does become larger as the number of free cells becomes smaller. This is because the garbage collector is called more often as you use up a smaller number of "CONS" cells in a shorter amount of time.

Lisp performed much better than Basic on list-oriented problems, which seems only natural. Lisp can also perform functions that I would not even attempt to do in Basic or Pascal, because they would be too complex or impossible in those languages. For example, a program to do symbolic differentiation can

Flexibility is where Lisp excels with its list data structures. Execution speed is good for loops.

be written easily in Lisp, but would be difficult in most other languages.

Summary

Stiff Upper Lisp seems like the most powerful package of the three. It matches MacLisp very well, and has good user support and a good set of basic functions. It is the one package that I can recommend.

The **muLisp-80** error-handling facilities greatly detract from an otherwise good implementation of Lisp. Although it does not match any existing dialect of Lisp well, it does have a reasonably complete set of functions. The resident screen editor provides a good user environment. The large integer capability of over 600 decimal digits is unmatched even in other languages. It is used as the basis for a companion product called muMath-80, which provides symbolic math manipulation functions such as symbolic differentiation. muLisp-80 is also the fastest of the three packages, and is good if you would like to get a taste of Lisp.

SuperSoft Lisp cannot be recommended for even a casual user. It will be a disappointment to a novice and frustration to an expert. Bad documentation just adds to the user's frustration. Luckily, the other alternatives are better.

It is nice to see Lisp available under CP/M because it is unlike most other languages in the marketplace. It can do jobs that are impossible or extremely difficult in procedure-oriented languages such as Basic and Pascal. Its functionality and extensibility makes jobs easier to do in a small amount of time. Although Lisp was intended for the AI community, it has much to offer all users. I think you will have fun using Lisp as a tool.

Lisp is becoming more popular on microcomputers, and these three implementations are only the tip of the iceberg. A review of three more Lisp systems for microcomputers is in the works, and will appear in *Microsystems* soon.

References

Allen, J. R., *Anatomy of LISP*, McGraw-Hill Book Co., New York, NY, 1978.

McCarthy, J., et al., *LISP 1.5 Programmer's Manual*, M.I.T. Press, Cambridge, MA, 1963.

Tennant, Harry, *The Stiff Upper Lisp User's Manual and Study Guide*, Tennant & Tennant Computing, Garland, TX, 1983.

Siklossy, L., *Let's Talk LISP*, Prentice-Hall, Englewood Cliffs, NJ, 1976.

Soft Warehouse, The, *muLisp/muStar-80 Artificial Intelligence Development System Reference Manual, CP/M Version*, MicroSoft Corp., Bellevue, WA, 1981.

SuperSoft, *LISP for CP/M*, SuperSoft Inc., Champaign, IL, 1981.

Winston, P. H., and Horn, B. K., *LISP*, Addison-Wesley Publishing Co., Reading, MA, 1981.

Submit file to patch Stiff Upper Lisp

```
XSUB
DDT UPLISP.COM
A100
CALL 3C80
.
A3C80
LDA A,4
ANI 0F
STA 2559
RET
.
G0
SAVE 60 UPLISP.COM
```

Above is a listing of a CP/M submit file that will make the distribution version of Stiff Upper Lisp read the file INIT.LSP from the CP/M default drive instead of only from drive A, so you do not need a disk in drive A. The commands can also be entered manually, in which case the XSUB command can be ignored.

Stiff Upper Lisp CLOSURE example

```
Stiff Upper Lisp CLOSURE Factorial Definition Example

Recursive Definition:

(DE FACTORIAL (N)
    (COND ((< N 2) 1)
          (T (* N (FACTORIAL (- N 1))))) )


CLOSURE Definition:

(CDE () FACTORIAL (N) (FACTORIAL1 1 N))
(CDE () FACTORIAL1 (RESULT N)
        (COND ((< N 2) RESULT)
              (T (FACTORIAL1 (* RESULT N) (- N 1))) ) )


LOOP Definition:

(DE FACTORIAL (N)
     (LET ((RESULT 1))
          (LOOP EXIT ((< N 2) RESULT)
                (SETQ RESULT (* RESULT N))
                (SETQ N (- N 1)) ) ) )
```

Execution of (FACTORIAL 25) would require 25 stack levels using the recursive definition and one for the CLOSURE and LOOP definitions. The recursive definition can be written like the CLOSURE definition; however, it would still not use one stack level. Note that the LOOP definition requires a local variable supplied by LET and uses assignment statements to change the state of the computation.

Bill Wong is currently doing program development for Rising Star. He received his M.S. in Computer Sciences from Rutgers University and is currently working on his Ph.D. His B.E.E. is from Georgia Tech. Bill is interested in microcomputer systems, parallel processing, and artificial intelligence.

A program to do symbolic differentiation can be written easily in Lisp, but would be difficult in most other languages.

Figure 1. muLisp-80 Implied COND Example.

```
Normal Lisp COND                     muLisp-8Ø Implied COND
================                     ======================

(DEFUN TEST                          (DEFUN TEST
  (LAMBDA (X Y)                            (LAMBDA (X Y)
    (COND ((EQ X Y) X)                           ((EQ X Y) X)
          ((NULL X) Y)))                         ((NULL X) Y)))


The muLisp-8Ø savings is one Lisp cell per implied COND.
```

Figure 2. SuperSoft Lisp APVAL Example.

```
> (SETQ X 2)          ;; Set X to 2
2                     ;; Returns value set
> X                    ;;Display value of X
2                     ;; Returns value of X just set
> (SETQ X 3)          ;; Change value of X to 3
3                     ;; Returns value set
> X                   ;; Display value of X
3                     ;; Returns value of X just set
> (PUTPROP (QUOTE X) (QUOTE Ø) (QUOTE APVAL))
X                     ;; Put APVAL on property list of X
> X                   ;; Display value of X
Ø                     ;; Returns APVAL value !!!!
> (SETQ X 3)          ;; Change value of X to 3
3                     ;; Returns value set
> X                   ;; Display value of X
Ø                     ;; Returns APVAL not value just set !!!
```

Figure 3. Big Numbers under muLisp-80.

```
$ (TIMES 2 65536)
13ØØ72
$ (PLUS 1234 987654321Ø98765432 1Ø)
98765432 1Ø98765 44444
```

Figure 4. String Manipulation Functions for Stiff Upper Lisp.

```
*1 (EXPLODE (QUOTE ALPHA))
(A L P H A)
*2 (COMPRESS (QUOTE (A L P H A)))
ALPHA
*3 (COMPRESS (QUOTE (A 12 GO 56))
A12GO56
*4 (FLATSIZE (QUOTE ALPHA))
5
```

Comparison of String Functions

Stiff Upper Lisp	muLisp-8Ø	SuperSoft Lisp
COMPRESS	PACK	IMPLODE
EXPLODE	UNPACK	EXPLODE
FLATSIZE	LENGTH	-----

LOWER PRICES! FASTER SERVICE! GUARANTEED!

5¼ inch Disk Drives

Tandon TM 100-1 *single sided, double density 48 TPI*
MSM-551001 _____ $219.95 ea 2 for **$199.95 ea**

Shugart SA 400L *Sigle sided, double density 40 track*
MSM-104000 _____ $234.95 ea 2 for **$224.95 ea**

Tandon TM 100-2 *Double sided, double density 48 TPI*
MSM-551002 _____ $294.95 ea 2 for **$269.95 ea**

TEAC FD55A *Single sided, double density 40 track*
MSM-660551 _____ $299.95 ea 2 for **$239.95 ea**

5¼ inch Cabinets with Power Supply

END-000216 *Single cab w/power supply* _____ **$69.95**
END-000226 *Dual cab w/power supply* _____ **$94.95**

8 inch Disk Drives

Siemens FDD 100-8 *Single sided, double density*
MSF-201120 _____ $274.95 ea 2 for **$249.95 ea**

Shugart SA 801R *Single sided, double density*
MSF-10801R _____ $394.95 ea 2 for **$389.95 ea**

Shugart SA851R *Double sided, double density*
MSF-10851R _____ $554.95 2 for **$529.95 ea**

Qume DT-8 *Double sided, double density*
MSF-750080 _____ $554.95 2 for **$529.95 ea**

Tandon TM848-1 *Single sided, double density thin line*
MSF-558481 _____ $379.95 ea 2 for **$369.95 ea**

Tandon TM848-2 *Double sided, double density thin line*
MSF-558482 _____ $494.95 ea 2 for **$484.95 ea**

NEC FD1165 *Double sided, double density half high*
MSF-558482 _____ $485.00 ea 2 for **$449.95 ea**

Dual Disk Sub Systems

Disk Sub-Systems - JADE

Handsome metal cabinet with proportionally balanced air flow system, rugged dual drive power supply, power cable kit, power switch, line cord, fuse holder, cooling fan, never mar rubber feet, all necessary hardware to mount 2-8 inch disk drives, power supply, and fan, does not include signal cable

Dual 8-inch Sub-Assembly Cabinet

END-000420 *Bare cabinet* _____ **$49.95**
END-000421 *Cabinet kit* _____ **$199.95**
END-000431 *A & T* _____ **$249.95**

8-inch Sub-Systems - Single Sided, Double Density

END-000423 *Kit w/2 Siemens FD100-8Ds* ____ **$650.00**
END-000424 *A & T w/2 Siemens FD100-8Ds* ___ **$695.00**
END-000433 *Kit w/2 Shugart SA-801Rs* ____ **$999.95**
END-000434 *A & T w/2 Shugart SA-801Rs* ___ **$1195.00**

8-inch Sub-Systems - Double-Sided Double Density

END-000426 *Kit w/2 Qume DT-8s* _____ **$1274.95**
END-000427 *A & T w/2 Qume D-8s* _____ **$1474.95**
END-000436 *Kit w/2 Shugart SA-851Rs* ____ **$1274.95**
END-000437 *A & T w/2 Shugart SA-851Rs* ___ **$1474.95**

S-100 Disk Controllers

DISK 1 - CompuPro

8 inch or 5¼ inch DMA disk controller, single or double density, single or double sided, 10MHz

IOD-1810A *A & T* _____ **$449.95**
IOD-1810C *CSC* _____ **$554.95**

Versafloppy II - SD Systems

Double density disk controller for any combination of 5¼ inch and 8 inch single or double sided, analog phase locked loop data seperator, vectored interrupts. CP/M 2.2 Oasis compatible, control/diagnostic software PROM included

IOD-1160A *A & T with PROM* _____ **$359.95**
SFC-55009047F *CP/M 3.0 with VF-11* _____ **$139.95**

2422 Disk Controller - C.C.S.

5¼ inch or 8 inch double density disk controller with on board boot loader ROM, FREE CP/M 2.2 & manual set

IOD-1300A *A & T with CP/M 2.2* _____ **$399.95**

Double D - JADE

High reliability double density disk controller with on board Z-80A, auxiliary printer port. IEEE S-100 can function in multi-user interrupt driven bus

IOD-1200B *Bare board & hdwr man* _____ **$59.95**
IOD-1200K *Kit w/hdwr & sftwr man* _____ **$299.95**
IOD-1200A *A & T w/hdwr & sftwr man* _____ **$325.95**
SFC-59002001F *CP/M 2.2 with Double D* ____ **$99.95**

S-100 Package

10 Megabyte S-100 Package

If your data storage needs have outgrown your floppy-based S-100 system, Jade now gives you the opportunity to add a fast reliable hard disk package to your system at a bargain price. Shugart SA1004 8-inch 10MB Winchester, X Comp S-100 controller for SA1004, your choice of X Comp CP/M auto-attach software (for most S-100 systems) or "WD BOOT" PROM (for SD Systems board sets) Cables and manuals included, limited quantity

MSH-991000 *with SD Boot Prom* _____ **$1295.00**
MSH-991002 *with Auto attach software* ____ **$1295.00**

Disk Tubs

Diskette Storage Boxes

Clear Plexiglass storage boxes for up to 75 Diskettes

MMA-505 *Holds 75 5¼" Diskettes* _____ **$19.95**
MMA-508 *Holds 50 8" Diskettes* _____ **$24.95**

Modems

Smart Buy in MODEMS - Signalman

1200 and/or 300 baud, direct connect, automatic answer or originate selection, auto-answer/auto-dial on deluxe models. 9v battery allows total portability, full one year warranty

IOM-5600A *300 baud direct connect* _____ **$89.95**
IOM-5610A *300 baud Deluxe* _____ **$149.95**
IOM-5620A *1200/300 baud Deluxe* _____ **$369.95**
IOM-5650A *300 baud for Osborne* _____ **$119.95**
IOM-5630A *300 baud card for IBM* _____ **$199.95**

Smartmodem - HAYES

Sophisticated direct-connect auto answer/auto dial modem, touch tone or pulse dialing. RS-232C interface, programmable

IOM-5500A *Smartmodem 1200* _____ **$599.95**
IOM-5400A *Smartmodem 300* _____ **$224.95**
IOK-1500A *Hayes Chronograph* _____ **$218.95**
IOM-1100A *Micromodem 100* _____ **$368.95**
IOM-2010A *Micromodem II w/term prgm* ____ **$329.95**
IOM-2012A *Terminal program for MMII* _____ **$89.95**

1200 Baud Smart Cat - NOVATION

103/212 Smart Cat & 103 Smart Cat, 1200 & 300 baud, built-in dialer, auto redial if busy, auto answer/disconnect, direct connect, LED readout displays mode, analog/digital loopback self tests, usable with multi line phones

IOM-5241A *300 baud 103 Smart Cat* _____ **$229.95**
IOM-5251A *1200 baud 212/103 Smart Cat* ____ **$549.95**

J-CAT Modem - NOVATION

1/5 the size of ordinary modems, Bell 103, manual or auto-answer, automatic answer/originate, direct connect, built in self test, two LEDs and audio beeps provide status information

IOM-5261A *Novation* _____ **$149.95**

NEW! CP/M plus 3.0

CP/M 3.0 is Digital Research's latest version of the industry standard disk operating system. It features many performance improvements, such as intelligent record buffering, improved diretory handling, "HELP" facility, time date stamping of files and many more improvements. AND A TREMENDOUS INCREASE IN SPEED!! it is fully CP/M 2.2 compatible and requires no changes to your existing application software. Available only to Versafloppy II owners with SBC-200 CPUs

SFC-55009059F *Unbanked,* _____ **$250.00**
SFC-55009057F *Banked,* _____ **$250.00**
SFC-55009057D *Manual* _____ **$50.00**

CP/M Plus On EPROM

SFC-55009059E *Unbanked RS232* _____ **$400.00**
SFC-55009057E *Banked RS232* _____ **$400.00**

PLACE ORDERS TOLL FREE

Continental U.S.
800-421-5500

Inside California
800-262-1710

For Technical Inquires or Customer Service call:
213-973-7707

We accept cash, checks, credit cards, or Purchase Orders from qualified firms and institutions. **Minimum prepaid order $15.00** California residents add 6½% tax. Export customers outside the U.S. or Canada please add 10% to all prices. **Prices and availibility subject to change without notice.** Shipping and handling charges via UPS Ground 50¢/lb. UPS Air $1.00/lb. minimum charge $3.00

S-100 Memory Boards

64K STATIC RAM - Jade

Uses new 2K x 8 static RAMs, fully supports IEEE 696 24 bit extended addressing, 200ns RAMs, lower 32K or entire board phantomable, 2716 EPROMs may be subbed for RAMs, any 2K segment of upper 8K may be disabled, low power typically less than 500ma.

MEM-99152B *Bare board* — $49.95
MEM-99152K *Kit less RAM* — $99.95
MEM-32152K *32K kit* — $199.95
MEM-56152K *56K kit* — $289.95
MEM-64152K *64K kit* — $299.95
Assembled & Tested — add $50.00

EXPANDORAM III

SD Systems new ExpandoRAM III is a high density S—100 memory board utilizing the new 64K x 1 dynamic RAM chips. It allows memory sizes of 64K, 128K or 256K all on a single S-100 board

MEM-65064A *64K* — $495.00
MEM-65128A *128K* — $595.00
MEM-65192A *192K* — $675.00
MEM-65256A *256K* — $755.00

RAM 17 - CompuPro

Meets or exceeds all IEEE 696/S-100 specifications including timing - works up to and including 10 MHz with 8088/86 CPUs Guaranteed to perform flawlessly with any IEEE 696/S-100 extended addressing specification

MEM-64180A *RAM 17, 64K A & T* — $459.95
MEM-64180C *RAM 17, 64K CSC* — $549.95

RAM 16 - CompuPro

There is an alternative to quirk-prone dynamic memories: thanks to low power consumption, high speed operation, fully static technology, and suitability for 8 or 16 bit systems, RAM 16 represents a new high in performance for sophisticated S-100 systems

MEM-32180A *RAM 16, A & T* — $495.00
MEM-32180C *RAM 16, CSC* — $595.00

RAM 21 - CompuPro

Operates at 12 MHz with 8088, 8086, 68000, 80286, and 16032 type CPUs, extremely low power consumption, meets or exceeds all IEEE 696/S-100 specifications, including timing

MEM-12810A *RAM 21 A & T* — $995.00
MEM-12810C *RAM 21 CSC* — $1125.00

S-100 Motherboards

ISO-BUS - Jade

Silent, simple and on sale — a better motherboard

6 Slot (5¼" x 8⅝")

MBS-061B *Bare board* — $22.95
MBS-061K *Kit* — $39.95
MBS-061A *A & T* — $69.95

12 Slot (9¾" x 8⅝")

MBS-121B *Bare board* — $34.95
MBS-121K *Kit* — $69.95
MBS-121A *A & T* — $109.95

18 Slot (14½" x 8⅝")

MBS-181B *Bare board* — $54.95
MBS-181K *Kit* — $99.95
MBS-181A *A & T* — $149.95

EPROM Erasers

Ultra-Violet EPROM ERASERS

Inexpensive erasers for industry or home

XME-3100A *Spectronics w/o timer* — $69.50
XME-3101A *Spectronics with timer* — $94.95
XME-3200A *Logical Devices* — $49.95

S-100 I/O Boards

INTERFACER 4 - CompuPro

3 serial, 1 parallel, 1 Centronics parallel.

IOI-1840A *A & T* — $399.00
IOI-1830C *CSC* — $499.00

I/O-4 - SSM Microcomputer

2 serial I/O ports plus 2 parallel I/O ports.

IOI-1010A *A & T* — $249.95

I/O-5 - SSM Microcomputer

Two serial & 3 parallel ports, 110-19.2K Baud

IOI-1015A *A & T* — $289.95

INTERFACER 2 - CompuPro

3 parallel, 1 serial and interrupt timer

IOI-1820A *Interfacer 2, A & T* — $289.95
IOI-1820C *Interfacer 2, CSC* — $359.00

INTERFACER 3 - CompuPro

5 or 8 channel serial I/O board for interrupt driven multi-user systems up to 250K baud.

IOI-1830A *Interfacer 3/5 A & T* — $558.00
IOI-1831A *Interfacer 3/8 A & T* — $628.00
IOI-1830C *Interfacer 3/5 CSC* — $628.00
IOI-1831C *Interfacer 3/8 CSC* — $748.00

MPX-1 - CompuPro

Multi-user I/O multiplexer and interrupt controller with on board 8085A-2 CPU and 4K or 16K RAM

IOI-1880A *MPX-1, 16K RAM, A & T* — $584.00
IOI-1880C *MPX-1, 16K RAM, CSC* — $674.00

S-100 CPU Boards

SBC-200 - SD Systems

4 MHz Z-80A CPU with serial & parallel I/O, 1K RAM, 8K ROM space, monitor PROM included.

CPU-30200A *A & T* — $329.95

THE BIG Z - Jade

2 or 4 MHz switachable Z-80 CPU board with serial I/O, accommodates 2708, 2716, or 2732 EPROM, baud rates from75 to 9600.

CPU-30201B *Bare board w/manual* — $35.00
CPU-30201K *Kit with Manual* — $149.95
CPU-30201A *A & T with Manual* — $199.95

CPU 8086/87 - CompuPro

CPU 8086/87, based around Intel's highest performance 8086 sixteen bit processor, also includes sockets for their 8086 sixteen bit processor, also includes sockets for their 8087 math co-processor, and 80130 Operating System Firmware componet

CPU-70520A *CPU 8086/87 A & T* — $699.00
CPU-70520C *CPU 8086/87* — $764.00

CPU-Z CompuPro

2 or 4 MHz Z-80A CPU, 24 bit addressing

CPU-30500A *2/4 MHz A & T* — $279.95
CPU-30500C *3/6 MHz CSC* — $374.95

8085/8088 - CompuPro

Both8 & 16 bit CPUs, standard 8 bit S-100 bus, up to 8 MHz, accesses 16 Megabytes of memory

CPU-20510A *MHz A & T* — $398.95
CPU-20510C *6/8 MHz CSC* — $479.95

Letter Quality Printers

Lowest Price Daisywheel Printer - JUKI

Full featured daisywheel printer with graphics mode and built-in word processing functions. 18 CPS print speed, 13-inch platen, 10, 12, or 15 pitch plus proportioned spacing. Uses standard IBM ribbons. This is an extremely reliable letter quality printer, at an unheard of low price!

PRD-61001 *Parallel* — $629.95
PRD-61002 *RS232 serial board* — $59.95
PRA-61000 *Tractor option* — $139.95

380Z by Data Terminals & Communications

Based on the same quality mechanism as the Comrex printer, the 380Z contains electronic enhancements that allow it to print at speeds up to 32 CPS. Other features include a 48K buffer, proportional spacing and Diablo 1640/1650/630 compatible protocol. Comes with printwheel, ribbon and users manual. Serial, parallel, and IEEE 488 interfaces standard

PRD-11300 *380Z printer* — $1295.00
PRA-11000 *Tractor option* — $169.95
PRA-12000 *Cut Sheet Feeder* — $699.95

Printers From JADE

SEIKOSHA *10" Graphics, 30 CPS* — $229.95
OKIDATA 82 *10" 120 CPS* — CALL
OKIDATA 92 *10" 160 CPS* — CALL
OKIDATA 93 *15" 160 CPS* — CALL
OKIDATA 83 *15" 120 CPS* — CALL
OKIDATA 84 *15" 200 CPS* — CALL
OKIDATA 2350 *15" 350 CPS* — CALL
OKIDATA 2410 *15" 350 CPS* — CALL
DAISYWRITER 2000 *48K Daisywheel* — $1395.00
GEMINI 10 *100 CPS with Graphics* — $349.95
GEMINI 15 *15" with Graphics* — $499.95
COMREX CR1 — $849.95
SILVER REED — $699.95
STARWRITER — $1249.95

S-100 EPROM Boards

PB-1 - SSM Microcomputer

2708, 2716 EPROM board with on-board programmer.

MEM-99510K *Kit with manual* — $154.95
MEM-99510A *A & T With Manual* — $219.95

PROM-100 - SD Systems

2708, 2716, 2732,EPROM programmer with software.

MEM-99520K *Kit with software* — $189.95
MEM-99520A *A & T with software* — $249.95

32K PROM/RAM Board

The 32K S-100 PROM/RAM board can hold up to 16 each 2716 style EPROMs, 6116 style RAMs, or 8 each style EPROMs. This board was designed to fit into holder S-100 systems as well as the newer IEEE-696 S-100 proposed standard, addressable as two 16K blocks on any 64K page, supports Cromemco as well as Northstar bank select, perfect for MP/M systems

MEM-99153B *Bare board & manual* — $49.95
MEM-99153K *Kit with No RAM* — $89.95
MEM-99153A *A & T with No RAM* — $139.95
MEM-16153K *Kit with 16K RAM* — $129.95
MEM-16153A *A & T with 16K RAM* — $179.95
MEM-32153K *Kit with 32K RAM* — $179.95
MEM-32153A *A & T with 32K RAM* — $229.95
Call Us For Lowest Prices On EPROMs-2732s — $4.90

JADE
Computer Products

4901 West Rosecrans Ave. Hawthorne, California 90250

CIRCLE 16 ON READER SERVICE CARD

MICROSYSTEMS TESTS THE

Cromemco C10 System

by Bill Machrone

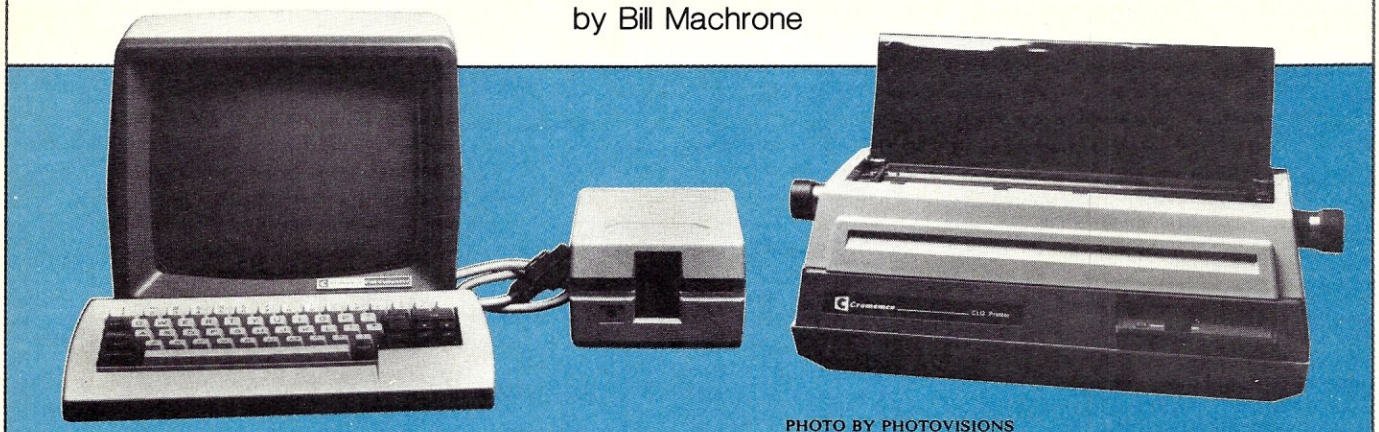


PHOTO BY PHOTOVISIONS

It all started as a social experiment: Take one computer, specifically designed, packaged and outfitted for the first-time user. Take one potential user with no computer experience but a good application for a computer. The computer is Cromemco's new C10 with 64K and a single disk drive. The user is my neighbor, Crosby, who has been nosing around the discount electronics stores and reading ads for Quick Brown Fox.

"Hello?"

"Hey, Crosby, it's Bill. How would you like a free computer?"

"Yeah? What's the catch?"

"Two things. You've got to give it back after a month or two and you have to take notes and tell me your experiences, good and bad, in learning how to use it. In exchange, I'll give you about the same amount of help and support you'd expect from a discount electronics store."

"How much is that?"

"Almost none."

"OK, you've got a deal."

I brought the C10 over, hooked all the cables together, booted it, made a few work disks and ran through all the programs. Everything seemed to be in working order, so I concentrated on the word processing package. Crosby's job is to inspect buildings for a fire insurance company prior to their being insured. He travels around a lot, takes a lot of Polaroids and writes a lot of memos. He's not a bad hand on his electric typewriter, but he hasn't been blind to the information age, either.

To get an idea of how beginner-oriented this machine is, consider that it doesn't offer DDT or any kind of debugging program. There are two operating systems, a ROM-resident extended disk monitor and a version of CDOS. The average user will never know that the ROM monitor is there. CDOS is Cromemco's CP/M-compatible operating system. In this case, it is a special version for the C10. It has a menu-driven "shell" program that calls the other programs. The utilities are well developed, transcending many of the equivalent CP/M programs. For instance, the file listing program pauses at the end of each screenful and the directory program includes file sizes.

The C10's word processing software, WriteMaster, is a Cromemco-developed package. It includes all the popular features, including cut and paste, search and replace, mail merge and full screen editing. It uses two operating modes, editing and command, to keep things simple. It uses the arrow keys rather than control codes to move the cursor. Command mode features automatic command completion. That is, if you type "ED" the system supplies "IT", signifying that you want to edit something. If you type a question mark, WriteMaster displays all the options available to you at that point in time. There is a significant amount of on-line help available, a boon to the new user. Features include paragraph reform and text alignment, but you have to indicate the end of a paragraph with a special character, control-5. This displays as a tilde (~) on the screen, but does not show when the document is printed. Similarly, the end of a page must be indicated with a control-shift-5. These and other control codes are made easy to use by a sticker that defines them, attached to the keyboard just above the numbers row.

WriteMaster's performance is very good. Insertion and deletion are quick; the only thing that slows it down is disk access. I'm not sure how he did it, but every time Crosby used the print screen function instead of the print from file function, he got two copies of the screen on the printer.

"Hello, Bill?"

"Hi, Crosby, what's up?"

"Just wanted to give you a progress report on the

Bill Machrone, 121 N. Ave., Fanwood, NJ 07023

HARD DISKS

For Systems Integrators

Whether you're an OEM, system integrator, or end user, when the time comes to add a hard disk unit to your computer you want a building block that offers high performance, quality, and cost effectiveness. The Pragmatic Designs PD-10M, PD-20M, and PD-40M all provide these features and more.

All Pragmatic Designs hard disk sub-systems are designed for use in systems equipped with the CompuPro® Disk II hard disk controller. They can also be used with other OEM controllers which support the popular SA-4000 hard disk interface. Standard features include:

- 10, 20, and 40 Megabyte formatted storage
- 11.7, 23.4, or 47.5 Megabyte unformatted storage
- Fully compatible with CompuPro Disk II controller
- Heavy duty power supply with 110/220V capability
- 19" rack mount configuration available
- 1 Year limited warranty
- Full hard disk system including controller, cables, and software available

Hard disks . . . easy solution. If you're ready to add a full capability industrial grade hard disk sub-system to your computer system then call Jerry Hall at Pragmatic today.

pragmatic designs INC.

Pragmatic Designs, Inc., 950 Benicia Ave., Sunnyvale, CA 94086 408/736-8670 TLX: 171627

™ *CompuPro is a registered trademark of Godbout Electronics*

CIRCLE 186 ON READER SERVICE CARD

computer. It's pretty nice, and I haven't wasted nearly so much typing paper since I began using it. I wish, though, that it would let me type letters more than 23 lines long."

"Huh? What do you mean?"

"Well, when you've got more than 23 lines, the bottom one pushes the top one off the screen and it doesn't come out on the printer or anything, so you lose it."

"Umm, Crosby, it's supposed to do that. The top line really doesn't disappear, it's still in the computer's memory. The whole letter will come out when you tell it to print."

"Oh. Gotcha. Thanks. See you."

The documentation for the C10's word processor is excellent (if you read it). It is well organized, with good examples, color illustrations, and is convenient in size. It demonstrates that a traditionally hardware-oriented company like Cromemco is ready to respond to the latest swings in the market and meet the marketing-oriented firms on their own ground. The manual is tutorial in orientation and is one of the package's nicest features.

There is nothing earth-shattering in the C10's specifications: A 4 MHz Z80 with 64K, two serial ports and an integral monitor. The second serial port can be used as a link to another computer for uploading or downloading files, or to control a modem. There is a simple communications program that permits uploading and downloading of files and operation as a remote terminal. The power switch is as inconveniently located as any I have seen. Why can't manufacturers just put them on the front? The monitor sports a green phosphor screen, but I found the character matrix to be a little coarse for my taste. The graphics mode, however, provided a very nice character fill and good definition.

The C10 that Crosby and I tested came with the optional letter-quality printer, a Smith-Corona TP-1 dressed up to match the C10. It is slow compared to the Diablo and NEC printers I am used to. The impression is clean, if somewhat light. Crosby thought the printer should have produced better carbon copies.

"Hey Bill, I've got a problem."

"Really? What happened?"

"The printer won't work. I've tried everything, including your trick of turning everything off and back on again. And I need the memo that's in there."

"Try looking behind the printer to see if the white wire from the computer is firmly plugged in."

"Okay, hold on a moment. Hey, the thing's just sitting here on the table, not plugged into anything. Hang on while I try it. Ah, there it goes. Thanks, Bill."

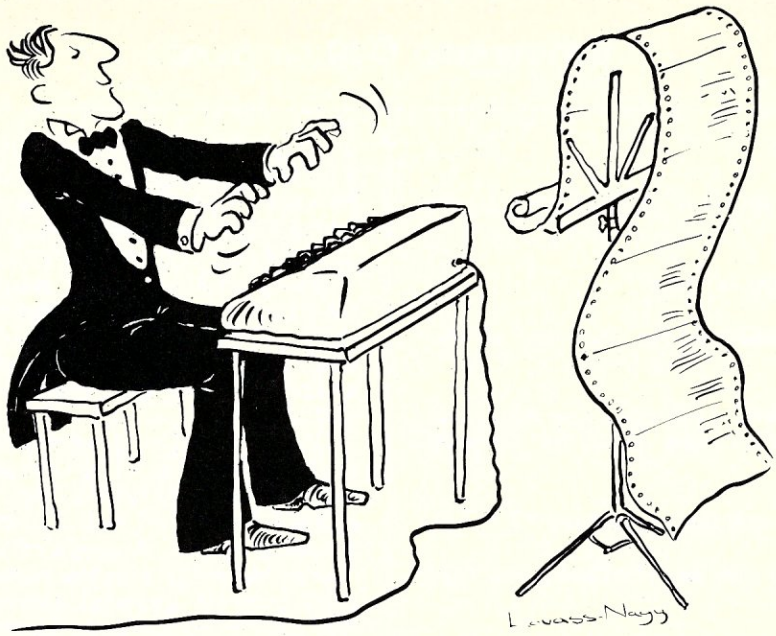
The C10's mechanicals are a little on the light side, an effect of keeping the manufacturing costs down, typical of the machines at the low end of the market. After Crosby got rolling, his main gripe with the C10 was the keyboard enclosure. The molded plastic shell comes too close to the keys and binds a few of them. We disassembled the keyboard twice and tried to realign it, to no avail. I took a file to it and the problem was solved. The video display unit, which also contains the processor, is attractively styled. Its clean, beige lines elicit compliments from everyone. My wife, who has seen computers come and go by the dozens, remarked, "Now, that's the least threatening-looking computer I've ever seen." And so it is. The keyboard has nothing more than you need to get the job done. It's exactly what you'd find on a typewriter with the exception of a control key, four cursor keys and a couple of keys to complete the ASCII character set. No arcane function keys. There is an optional "ergonomic" stand available for the monitor unit. I'd recommend it to anyone, because it tilts and swivels and has a much smaller footprint than the monitor itself. There are threaded feet to adjust the angle of the keyboard and the display.

The innards of the C10 are utterly uninspiring. All you can really see is the video board. The processor board is nestled beneath it, mostly out of sight. It contains a 4-MHz Z80, 64K of dynamic RAM, a couple of software-controllable UARTS, and not much else beside the disk controller. In keeping with the nontechnical nature of the machine, there is nary a mention of the insides in the documentation.

"Bill?"

Close-up of the Cromemco C-10 keyboard shows its simplified design. Photo by PhotoVisions.

The utilities are well developed, transcending many of the equivalent CP/M programs . . . the documentation for the C10's word processor is excellent.



The Well-Tempered Cross-Assembler

Before Johann Sebastian Bach developed a new method of tuning, you had to change instruments practically every time you wanted to change keys. Very difficult.

Before Avocet introduced its family of cross-assemblers, developing micro-processor software was much the same. You needed a separate development system for practically every type of processor. Very difficult and very expensive.

But with Avocet's cross-assemblers, a single computer can develop software for virtually any microprocessor! Does that put us in a league with Bach? You decide

Development Tools That Work

Avocet cross-assemblers are fast, reliable and user-proven in over 3 years of actual use. Ask NASA, IBM, XEROX or the hundreds of other organizations that use them. Every time you see a new microprocessor-based product, there's a good chance it was developed with Avocet cross-assemblers.

Avocet cross-assemblers are easy to use. They run on any computer with CP/M* and process assembly language for the most popular microprocessor families.

XASM05	6805	$200 each
XASM09	6809	
XASM18	1802	
XASM48	8048/8041	
XASM51	8051	
XASM65	6502	
XASM68	6800/01	
XASMF8	F8/3870	
XASMZ8	Z8	
XASM400....	COP400	
XASM75	NEC 7500	$500

(Coming soon: XASM68K .... 68000)

Turn Your Computer Into A Complete Development System

Of course, there's more. Avocet has the tools you need from start to finish to enter, assemble and test your software and finally cast it in EPROM:

Text Editor VEDIT -- full-screen text editor by CompuView. Makes source code entry a snap. Full-screen text editing, plus TECO-like macro facility for repetitive tasks. Pre-configured for over 40 terminals and personal computers as well as in user-configurable form.

CP/M-80 version	$150
CP/M-86 or MDOS version	$195

(when ordered with any Avocet product)

ROM Simulator -- ROMSIM by Inner Access eliminates need to erase and reprogram EPROM. Installed in an S-100 host, ROMSIM substitutes RAM for EPROM in external target system. 16K memory can be configured to simulate the 2708, 2758, 2716, 2516, 2732, 2532, 2764, 2564 in either byte or word organization. Avocet's configurable driver makes loading of HEX or COM files fast and easy.

From $495 depending on cabling and RAM installed.

EPROM Programmer -- Model 7128 EPROM Programmer by GTek programs most EPROMS without the need for personality modules. Self-contained power supply ... accepts ASCII commands and data from any computer through RS 232 serial interface. Cross-assembler hex object files can be down-loaded directly. Commands include verify and read, as well as partial programming.

PROM types supported: 2508, 2758, 2516, 2716, 2532, 2732, 2732A, 27C32, MCM8766, 2564, 2764, 27C64, 27128, 8748, 8741, 8749, 8742, 8751, 8755, plus Seeq and Xicor EEPROMS.

(Upgrade kits will be available for new PROM types as they are introduced.)

Programmer	$389
Options include:	
Software Driver Package	$ 30
RS 232 Cable	$ 30
8748 family socket adaptor	$ 98
8751 family socket adaptor	$174

Call Us

If you're thinking about development systems, call us for some straight talk. If we don't have what you need, we'll help you find out who does. If you like, we'll even talk about Bach.

VISA and Mastercard accepted. All popular disc formats now available -- please specify. Prices do not include shipping and handling -- call for exact quotes. OEM INQUIRIES INVITED.

*Trademark of Digital Research.

AVOCET SYSTEMS INC.
DEPT. 883-M
804 SOUTH STATE STREET
DOVER, DELAWARE 19901
302-734-0151 TLX 467210

"Hi. What's up?"

"You're going to think this is silly, but I can't get the chess game to work."

"What disk do you have in there?"

"The work disk you made for me, the one I always use."

"Well, that's the problem. I removed everything but the word processor and a couple of utilities to make room for your memos."

"Oh. Well, how do I work the chess game?"

"Put in one of the disks that has the Cromemco logo on the top."

"All right, bye."

When I removed from disk all the things Crosby wasn't going to be using, I didn't bother to revise the menu. You can, however, customize it to whatever you want. In this way, each disk that you put into the machine can be tailored to a specific application.

I had initially expected the C10's single disk drive to be a bottleneck, but it isn't. The typical C10 user will not be doing a lot of disk-to-disk copying. The dual-sided 40-track drive yields 360K of storage, fine for most applications. There are provisions for single-drive copying of individual files and entire disks. It's a bit slow, having to swap the disk every minute or so, but not impossible. You can add a second drive, which daisy-chains onto the modular cable that plugs into the rear of the monitor/computer.

Speaking of that chess game, the C10 has character-based graphics. The shapes available go beyond the usual line-drawing set into a group obviously intended for games. The chess characters look as good as any I've seen, and the game itself is challenging and fast.

The spreadsheet program, PlanMaster, is typical of the genre. You set up the usual grid and fill it with numbers and cell calculations. I found its disk access to be a little slow, but the actual calculations were agreeably quick. A group of Basic programs, called MoneyMaster, perform a variety of useful home and light business financial calculations. These include loan and mortgage amortization and stock and bond analysis. Most home computerists get programs like these by buying a book and keying them in. Their inclusion here is thoughtful.

The C10 also comes with Cromemco's Structured Basic, a highly evolved Basic interpreter. It was once a contender for the "best Basic" contest, but Cromemco's reluctance to move it outside the CDOS environment limited its acceptance. As the name implies it provides control structures similar to Pascal or PL/I. Your code can be quite elegant. You would expect an interpreter like this to be slow, given the advanced control structures it offers. The design,

Super Prices - Quick Delivery

MINIMUM SHIPPING $3.00 - NO SURCHARGE ON VISA/MASTERCHARGE

1-800-545-2633 - Continental U.S.

DISK DRIVES

SHUGART

5¼" SA400 (35) 150.00
5¼" SA400L (40) 175.00
5¼" SA410L (40) 200.00
5¼" SA455L (40) 235.00
8" SA801R (SS/DD) 355.00

MITSUBISHI

5¼" M-4851 ½ size
DS/DD 40Tr 295.00
5¼" M-4853 ½ size 410.00
96 TPI same as TM100-4
5¼" M-4854 ½ size 439.00
1.5 MG w/8" electronics formats same as IBM 1 & 2D 8" std
8" M-2894-63 ..... 409.00
Std 8" DS/DD 1.2 MG
8" M-2996-63 ..... 439.00
Thinline 8" DS/DD 1.2 MG

SHUGART 800/801R RECONDITIONED

90 Day Warranty 160.00

TANDON

5¼" TM100-1 SS/DD 195.00
5¼" TM100-2 (IBM) DS/DD 239.00

PERTEC

5¼" FD200-S *SS/DD 40TR* 139.00
5¼" FD250-5 *DS/DD 40TR* 195.00

MPI

5¼" B-51 *40TR SS/DD* ... 230.00
5¼" B-52 (IBM) *80TR DS/DD* 250.00

All above drives SS/DD unless otherwise specified

DISK DRIVE CABINETS

**NOTE - Power supply will accommodate hard disk*

8" *DDC88V28 w/PS vertical - for 2-8" drives 300.00
8" DDC88T-1 w/PS vertical - For 2 or 4-8" thinline drives 300.00
8" DDC88T-2 w/PS vertical - inexpensive version for 2-8" thinline drives 225.00
8" *DDC8V w/PS vertical - for 1-8" drive 249.00
8" *DDC88H w/PS horizontal - for 2-8" drives 300.00
5¼" DDC5H w/PS horizontal - for 1-5¼" drive ... 59.00
5¼" DDC55H w/PS vertical - for 2-5¼" drives ... 89.00

APPLE DISK DRIVES

"Thinline II" - ½ size 163K 40TR 235.00
TEAC "Thinline" - ½ size 163K 40TR ... 275.00
Micro-SCI 239.00
Controller 79.00
Rana Elite I 275.00
Rana Elite II 439.00
Rana Elite III 569.00

DISKETTES

All diskettes have 1 yr. min. to lifetime warranty. Prices are for package of 10. Call for qty. pricing.

5¼" Sgl side/sgl den 19.00/10
5¼" Sgl side/dbl den 20.00/10
5¼" Dbl side/dbl den 26.00/10
5¼" 10 sector 22.00/10
5¼" 16 sector 22.00/10
8" Sgl side/dbl den 29.00/10
8" Dbl side/dbl den 36.00/10

MONITORS/TERMINALS

Sanyo 2112 (15 MHZ) 80.00
BMC 12A (15MHZ) 12" Grn 80.00
BMC 9191 color 12" 269.00
Amdek Color III 395.00
USI 9" Grn (20 MHZ) 139.00
USI 12" Grn (20 MHZ) 139.00
USI 12" Amber (20 MHZ) 149.00
Televideo 910 ..... 560.00
Televideo 925 ..... 715.00
Televideo 950 ..... 899.00
Televideo 970 ..... 1095.00

PRINTERS

C. ITOH

Pro-Writer I (8510A) 120 CPS 380.00
Star-Writer F10 (Ltr quality) 1225.00

STAR MICRO-TRONICS

Gemini 10 (100 CPS) 80 Col 310.00
Gemini 15 (100 CPS) 132 Col 445.00

OKI-DATA

82A Par + Ser 80 Col 399.00
83A Par + Ser 132 Col 650.00
d4B Ber 200 CPS 132 Col 950.00
92A 160 CPS - Par 80 Col 519.00
93A 160 CPS - Par 132 Col 899.00

NEC

NEC 8023 A 449.00

TOSHIBA

P-1350 Ltr quality - 120 CPS 1500.00
Draft mode 192 CPS Par or Serial
P-350 (100 CPS) Centronics Par-15" 429.00

MPI

MPI 99G (9") 499.00
MPI 150A1 (15") 935.00

EPSON

MX80 w/Graphtrax 399.00
M80FT w/Graphtrax 449.00
FX80 ..... 549.00
MX100 ..... 629.00
MX100FX ..... CALL

AC SURGE ELIMINATORS

Lemon (6AC outlets - 3 prong) 4600
Lime (5' - 3 prong pwr cord w/on-off switch 69.00
Orange - AC surge + EMI filter (6 outlets) 119.00
Plum - EMI-RF filter 39.00

MODEMS

Hayes Smart Modem 300
209.00

PARTS

TR1602B	$2.50 ea.	TMS2716(5+12V)	$3.95ea.
2114L2	$1.50 ea.	TMS2732	$4.95 ea.
TMS2532	$5.95 ea.	TMS4164(200NS)	5.95 ea.
2716 (5V)	$3.95 ea.	TMS6116-4(200NS)	$4.75 ea.

MANY OTHERS IN STOCK

DATA CABLES

8" DSC 88-2SKT - for 2-8" drvs w/skt. conn. 16.95
5¼" DSC55-2SKT - for 1-5¼" drvs w/skt conn. $15
RS232MM-5' (male to male) 16.00

E-PROM ERASERS

QUV-T8/1 (hobby) 49.95
QUV-T8/2 (Industrial version) 68.95
QUV-T8/2 (w/timer & safety switch) 97.50

CONNECTORS

RS232 Connectors

SOLDER TYPE		FLAT RIBBON TYPE	
DB25P	2.50	IDC25P	6.25
DB25S	3.00	IDC25S	6.60
DB25 Hood	1.00	IDC25 Hood	1.60
S-100 Connectors			10 for 25.00

S-100 PRODUCTS

GIGANTIC REDUCTIONS IN PRICE!

CARD CAGES/MOTHER BOARDS

**IEEE - 696 - No termination required*

Slots	Bare Bd	A+T	w/card cage
4	15.00	40.00	60.00
6	20.00	48.00	70.00
8	25.00	69.00	100.00
12	30.00	99.00	140.00
18	45.00	150.00	200.00
22	60.00	185.00	———

All card cages will accommodate a 4" fan
Add $20.00 for 1 fan - Add $30.00 for 2 fans

MAINFRAMES

For 2 Standard 8" Drives

MF+DD6 (6 slot M/B) 699.00
MF+DD8 (8 slot M/B) 749.00
MF+DD12 (12 slot M/B) 799.00

For 2 Thinline 8" Drives

IMF+DD6C (Cadillac version) 500.00
IMF+DD6F (Ford version) 350.00

For 2 - 5¼" Disk Drives

MF+MD12 (12 slot M/B) 575.00

Standard Plain Front

MF+12 (12 slot M/B) 500.00
MF+22 (22 slot M/B) 600.00

All mainframes except IMF+DD6F have EMI filter, 2 AC outlets, 15 ea. DB25, 2 ea. 50 pin, 2 ea. 34 pin, 1 ea. Centronic cutouts, power supply for 8" MF (−5V1A/+5V6A/+8V16A/±16V3A/+24V6A)

SIERRA DATA

SDS-SBC-100-Z80(4mhz) master 2 serial 2 par/floppy controller/64k ram ..$655.00
SDS-SBC-100S-4mhz slave/2 serial 2 par/64k ram $585.00
SDS-ZSIO/4 - 4 serial port I/O bd $235.00
SDS-MUX - RS232 multiplexer bd $235.00
SDS-HDI-M - Hard disk bd for micropolis $129.00
SDS-CPM/B105--cp/m for SBC100 w/BIOS $150.00
SDS-Turbodos - Multi-user for master & slaves $845.00

BOARD SETS

Best Bare Board Set Available
QTC-SBC 2/4 CPU
QTC-EXP+III 256K Memory bd.
QTC-FDC 5/8 Floppy disk controller
Bare Board Set $150.00
1) Includes manuals & assembly instructions
2) Parts available 3) Monitor & B10S available Add $30.00

QT CLOCK/CALENDAR

S-100 or Apple • Time in hrs, min, sec. • AM/PM or Military Format • Date in Mo., Day, Yr, Day of Week & Leap Year recognition • 4 hard interrupts (1024 Hz, 1Hz, 1 min, 1 hr) • On board battery (will last 14 mos. w/no power on)

QTC-CCS-BB (S-100) $45.00
QTC-CCS-K (kit) for S-100 ..... $100.00
QTC-CCS-A (A+T) for S-100 ..... $125.00
QTC-CCA-BB (for Apple) $33.00
QTC-CCA-A (for Apple) $100.00

COMPUPRO

S-100 - all assembled and tested

System 816A 4000.00
20 Slot MB 210.00
CPU 8085/88 ..... 389.00
Ram 17-64K 410.00
Ram 21-128K 900.00
Disk II 610.00
System Support I 310.00
Interfacer 4 ..... 310.00
Active Teammates 50.00
Enclosure 2 (disk) 675.00
Call for others

CPU / MEM / I/O

QTC-SBC 2/4 BB 1 ser 1 par $50.00
QTC-SBC 2/4 A A+T $265.00
QTC-Z+80 BB 1 serial $28.00

Teletek FDC-I A+T $595.00
Teletek Systemaster A+T $995.00

QTC-EXP+III Bare Bd. (dynamic) $65.00
QTC-EXP+III 64K A+T (64K/256K or 1 MEG) ... $450.00

S-RAM 64K BB (static) $60.00
S-RAM 64 A+T $349.00
QTC-ADA *ADA Converter* A+T $375.00
QTC-Dual GPIB-488 *IEEE 488 Interface bd.* A+T $475.00

Retail Sales
123 East 200 South
Salt Lake City, Utah 84111

The Great Salt Lake Computer Company, Inc.
1-801-363-3314

Mail Orders
P.O. Box 3150
Salt Lake City, Utah 84110

CIRCLE 39 ON READER SERVICE CARD

SOFTWARE DESCRIPTIONS

TPM (TPM I) - $80 A Z80 only operating system which is capable of running CP/M programs. Includes many features not found in CP/M such as independent disk directory partitioning for up to 255 user partitions, space, time and version commands, date and time, create FCB, chain program, direct disk I/O, abbreviated commands and more! Available for North Star (either single or double density), TRS—80 Model I (offset 4200H) or II, Versafloppy I, or Tarbell I.

TPM-II - $125 An expanded version of TPM which is fully CP/M 2.2 compatible but still retains the extra features our customers have come to depend on. This version is super FAST. Extended density capability allows over 600K per side on an 8" disk. Available preconfigured for Versafloppy II (8" or 5"), Epson QX-10, Osborne II or TRS-80 Model II.

CONFIGURATOR I

This package provides all the necessary programs for customizing TPM for a floppy controller which we do not support. We suggest ordering this on single density (8SD).
Includes: TPM-II ($125), Sample PIOS (BIOS) SOURCE ($FREE), MACRO II ($100), LINKER ($80), DEBUG I ($80), QED ($150), ZEDIT ($50), TOP I ($80), BASIC I ($50) and BASIC II ($100)
$815 Value **NOW $250**

CONFIGURATOR II

Includes: TPM-II ($125), Sample PIOS (BIOS) SOURCE ($FREE), MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), QSAL ($200), QED ($150), ZTEL ($80), TOP II ($100), BUSINESS BASIC ($200) and MODEM SOURCE ($40) and DISASSEMBLER ($80)
$1485 Value **NOW $400**

MODEL I PROGRAMMER

This package is only for the TRS-80 Model I. Note: These are the ONLY CDL programs available for the Model I. It includes: TPM I ($80), BUSINESS BASIC ($200), MACRO I ($80), DEBUG I ($80), ZDDT ($40), ZTEL ($80), TOP I ($80) and MODEM ($40)
$680 Value **NOW $175**

MODEL II PROGRAMMER

This package is only for the TRS-80 Model II. It includes: TPM-II ($125), BUSINESS BASIC ($200), MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), QED ($150), ZTEL ($80), TOP II ($100), ZDDT ($40), ZAPPLE SOURCE ($80), MODEM ($40), MODEM SOURCE ($40) and DISASSEMBLER ($80)
$1445 Value **NOW $375**

BASIC I - $50, a 12K+ basic interpreter with 7 digit precision.

BASIC II - $100. A 12 digit precision version of Basic I.

BUSINESS BASIC - $200. A full disk extended basic with random or sequential disk file handling and 12 digit precision (even for TRIG functions). Also includes PRIVACY command to protect source code, fixed and variable record lengths, simultaneous access to multiple disk files, global editing, and more!

ACCOUNTING PACKAGE - $300. Written in Business Basic. Includes General Ledger, Accounts Receivable/Payable, and Payroll. Set up for Hazeltine 1500 terminal. Minor modifications needed for other terminals. Provided in unprotected source form.

MACRO I - $80. A Z80/8080 assembler which uses CDL/TDL mnemonics. Handles MACROs and generates relocateable code. Includes 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical ops, local and global symbols, linkable module generation, and more!

MACRO II - $100. An improved version of Macro I with expanded linking capabilities and more listing options. Also internal code has been greatly improved for faster more reliable operation.

MACRO III - $150. An enchanced version of Macro II. Internal buffers have been increased to achieve a significant improvement in speed of assembly. Additional features include line numbers, cross reference, compressed PRN files, form feeds, page parity, additional pseudo-ops, internal setting of time and date, and expanded assembly-time data entry.

DEVELOPER I

Includes: MACRO I ($80), DEBUG I ($80), ZEDIT ($50), TOP I ($80), BASIC I ($50) and BASIC II ($100)
$440 Value **NOW $150**

DEVELOPER II

Includes: MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), BUSINESS BASIC ($200), QED ($150), TOP II ($100), ZDDT ($40), ZAPPLE SOURCE ($80), MODEM SOURCE ($40), ZTEL ($80), and DISASSEMBLER ($80).
$1280 Value **NOW $350**

DEVELOPER III

Includes: QSAL ($200), QED ($150), BUSINESS BASIC ($200), ZTEL ($80) and TOP II ($100)
$730 Value **NOW $300**

COMBO

Includes: DEVELOPER II ($1280), ACCOUNTING PACKAGE ($300), QSAL ($200) and 6502X ($150)
$1930 Value **NOW $500**

LINKER - $80. A linking loader for handling the linkable modules created by the above assemblers.

DEBUG I - $80. A tool for debugging Z80 or 8080 code. Disassembles to CDL/TDL mnemonics compatible with above assemblers. Traces code even through ROM. Commands include Calculate, Display, Examine, Fill, Goto, List, Mode, Open File, Put, Set Wait, Trace, and Search.

DEBUG II - $100. A superset of Debug I. Adds Instruction Interpreter, Radix change, Set Trap/Conditional display, Trace options, and Zap FCB.

6502X - $150. A 6502 cross assembler. Runs on the Z80 but assembles 6502 instructions into 6502 object code! Similar features as our Macro assemblers.

QSAL - $200. A SUPER FAST Z80 assembler. Up to 10 times faster than conventional assemblers. Directly generates code into memory in one pass but also to offset for execution in its own memory space. Pascal like structures: repeat...until, if...then...else, while...do, begin...end, case...of. Multiple statements per line, special register handling expressions, long symbol names, auto and modular assembly, and more! This one uses ZILOG Mnemonics.

QED - $150. A screen editor which is both FAST and easy to learn. Commands include block delete, copy, and move to a named file or within text, repeat previous command, change, locate, find at start of line, and numerous cursor and window movement functions. Works with any CRT having clear screen, addressable cursor, clear to end of line, clear to end of screen, and 80X24.

ZTEL - $80. An extensive text editing language and editor modelled after DEC's TECO.

ZEDIT - $50. A mini text editor. Character/line oriented. Works well with hardcopy terminals and is easy to use. Includes macro command capability.

TOP I - $80. A Text Output Processor for formatting manuals, documents, etc. Interprets commands which are entered into the text by an editor. Commands include justify, page number, heading, subheading, centering, and more.

TOP II - $100. A superset of TOP I. Adds: embedded control characters in the file, page at a time printing, selected portion printing, include/merge files, form feed/CRLF option for paging, instant start up, and final page ejection.

ZDDT - $40. This is the disk version of our famous Zapple monitor. It will also load hex and relocatable files.

ZAPPLE SOURCE - $80. This is the source to the SMB ROM version of our famous Zapple monitor. It can be used to create your own custom version or as an example of the features of our assemblers. Must be assembled using one of our assemblers.

MODEM - A communication program for file transfer between systems or using a system as a terminal. Based on the user group version but modified to work with our SMB board or TRS-80 Models I or II. You must specify which version you want.

MODEM SOURCE - $40. For making your own custom version. Requires one of our Macro Assemblers.

DISASSEMBLER - $80. Does bulk disassembly of object files creating source files which can be assembled by one of our assemblers.

HARDWARE

S-100 — SMB II Bare Board $50. "System Monitor Board" for S-100 systems. 2 serial ports, 2 parallel ports, cassette interface, 4K memory (ROM, 2708 EPROM, 2114 RAM), and power on jump. When used with Zapple ROM below, it makes putting a S-100 system together a snap.
Zapple ROM $35. Properly initializes SMB I/II hardware, provides a powerful debug monitor.

IBM PC — Big Blue Z80 board $595. Add Z80 capability to your IBM Personal Computer. Runs CP/M programs but does not require CP/M or TPM. Complete with Z80 CPU, 64K add on memory, serial port, parallel port, time and date clock with battery backup, hard disk interface, and software to attach to PC DOS and transfer programs. Mfr'd by QCS.
50% Discount on all CDL software ordered at the same time as a Big Blue (and for the Big Blue).

APPLE II — Chairman Z80 $345. Add Z80 capability to your Apple II/II Plus computer. Runs CP/M programs with our more powerful TPM. Includes 64K memory add on (unlike the competition this is also useable by the 6502/DOS as well as the Z80), TPM, QSAL assembler, QED Screen Editor, and Business Basic. Mfr'd by AMT Research.
Apple Special $175. Buy the Apple Z80 Developer at the same time as the "Chairman" and pay only $175 instead of $325.

APPLE Z80 DEVELOPER

Includes: 6502X ($150), MACRO II ($100), MACRO III ($150), QSAL ($200), QED ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), ZDDT ($40) and BUSINESS BASIC ($200)
VALUE: $1250 **NOW $325**
$175 when purchased with AMT "Chairman" Board

DISK FORMATS

When ordering software specify which disk format you would like.

CODE	DESCRIPTION
8SD	8" IBM 3740 Single Density (128 bytes/26 sectors/77 tracks)
8DD	8" Double Density (256 bytes/26 sectors/77 tracks)
8XD	8" CDL Extended Density (1024 bytes/8 sector/77 traceks 616K)
5SD	5.25" Single Density (TRS80 Model I, Versafloppy I, Tarbell I)
5EP	5.25" Epson Double Density
5PC	5.25" IBM PC Double Density
5XE	5.25" Xerox 820 Single Density
5OS	5.25" Osborne Single Density
5ZA	5.25" Z80 Apple (Softcard compatible)

TPM INFO When ordering TPM I or II, in addition to Disk Format, please specify one of the following codes:

	CODE	DESCRIPTION
TPM I:	NSSD/H	North Star Single Density for Horizon I/O
	NSSD/Z	North Star Single Density for Zapple I/O
	NSDD/H	North Star Double Density for Horizon I/O
	NSDD/Z	North Star Double Density for Zapple I/O
	TRS80-I	TRS-80 Model I (4200H Offset)
	TRS80II	TRS-80 Model II
	VI8	Versafloppy I 8"
	VI5	Versafloppy I 5.25"
TPM-II:	VII8	Versafloppy II 8" (XD)
	VII5	Versafloppy II 5.25"
	TRS80II	TRS-80 Model II (XD)

ORDERING INFORMATION:

VISA/MasterCard/C.O.D.
Call or Write With Ordering Information....

MasterCard VISA

OEMS:

Many CDL products are available for licensing to OEM's. Write to Carl Galletti with your requirements.

Dealer Inquiries Invited.

For Phone Orders ONLY Call Toll Free...

1-(800) 458-3491 (Except Pa.)

Ask For Extension #15

For information and Tech Queries call
(609) 599-2146

Prices and Specifications subject to change without notice.
TPM, Z80, CP/M, TRS80 are trademarks of CDL, Zilog, DRI and Tandy respectively.

Computer Design Labs

342 Columbus Avenue/Trenton, NJ 08629

CIRCLE 84 ON READER SERVICE CARD

Cromemco C10 continued . . .

however is that of a semicompiler or incremental compiler. Line references are stored in internal tables and expressions are tokenized, improving execution speed. It has a strong following in the Cromemco community, though, and it's inconceivable that they would have introduced this machine without it. If you are a Structured Basic fan, you are nodding and smiling, glad of Cromemco's strategy of giving it away on a low-end machine. If you have never used it you are probably wondering what all the fuss is about.

In the course of the review, Cromemco sent a new set of disks to all C10 owners. They contained improvements to just about everything on the system: CDOS, the word processor, and some of the utilities. Such action on the part of system manufacturers is commendable and all too rare.

"Hey, Crosby, how's the machine treating you?"

"I'm saving so much money on White-out that I think I can afford a new car. Really, though, it's a timesaver. Now if I could just learn to move paragraphs around. . . . Is that in the manual?"

For further information on the C10, contact:

Cromemco
280 Bernardo Ave.
P.O. Box 7400
Mt. View, CA 94039
(415) 964-7400

XERA—An Extended Erase Utility for CP/M

by Ian Ashdown

Of all the inconveniences that CP/M offers the programmer and end-user who is not protected by an application program, the built-in command "ERA" is probably the worst. With a single stroke of the carriage return key, you can set in motion a seemingly innocent command like "ERA B:*.LIB" to delete two macro files on an otherwise blank disk—until you remember that those files were on drive "A", and that the disk in drive "B" holds (or rather held) your entire library of assembly language macros.

Digital Research long ago recognized the problem, and wrote the code for a command called "ERAQ", which displays the name of each file found using the ambiguous file references "*" and "?", then asks the operator if it is to be erased. They thoughtfully included it in their multitasking operating system MP/M, but left CP/M users to fend for themselves.

Having recently made a late-night programming error that cost me six hours of file reconstruction the next morning (see above hypothetical situation), I wrote "XERA", an 8080 assembly language utility that emulates MP/M's "ERAQ"—with the usual (and inevitable) "added conveniences." The use of "XERA" is fully explained in the comments included with the program listing accompanying this article, and those wanting to add their own bells and whistles should have little trouble—the program is fully structured and documented, with subroutines performing one simple function each. Also, all I/O is done through calls to CP/M's BDOS, so that "XERA" should run without modification on any standard version of CP/M.

What more can I say, except perhaps that I hope this utility spares you the headaches that drove me to write it!

Ian Ashdown, byHeart Software, 2-2016 West First Avenue, Vancouver, BC, V6J 1G8, Canada

```
                        XERA - AN EXTENDED ERASE UTILITY FOR CP/M

                        Copyright:  Ian Ashdown
                                    byHeart Software
                                    2 - 2016 West 1st Avenue
                                    Vancouver, B.C. V6J 1G8
                                    August 13th, 1982

                        Version:    1.1

                        Use:  Same as CP/M's "ERA" built-in command, except
                              that "XERA" queries the operator whether each
                              file found using wild-card characters such as
                              "?" and "*" are to be erased or not. It
                              performs a function similar to MP/M's "ERAQ"
                              command.

                              As an added convenience, "XERA" asks the operator
                              if it is O.K. to erase the indicated files before
                              actually doing so. Typing anything other than "Y"
                              or "y" in response or typing ^C anytime prior to
                              this question will terminate the routine without
                              erasing any files.

                              "System" and "Read-Only" files are displayed but
                              not erased. Normally, the "System" and "Read-Only"
                              file attributes have been assigned for a good
                              reason, and thus these files should not be easily
                              erasable along with the "Directory" and "Read-Write"
                              files. In order to erase them it is necessary to
                              either use CP/M's built-in command "ERA" or else
                              set the files attributes to "DIR" and "R/W" using
                              CP/M's "STAT" utility and then rerun "XERA".

                        Example:    A> XERA PRO*.???

                                    A: PROG     .TXT ?    <Y> Yes
                                    A: PROG     .COM ?    READ/ONLY and SYSTEM
                                    A: PROG     .ASM ?    <N> No
                                    A: PROG     .001      READ/ONLY
                                    A: PROGRESS.BAS ?     <Y> Yes
                                    A: PROGRESS.INT ?     <Y> Yes

                                    O.K. to erase?        <Y> Yes

                        ;
                        ;***** DEFINITIONS *****
                        ;
0000 =                  BOOT    EQU     0000H       ;Warm start vector
0005 =                  BDOS    EQU     0005H       ;FDOS entry vector
                        ;
0002 =                  CONOUT  EQU     02H         ;BDOS console output function
0006 =                  DIRCIO  EQU     06H         ;BDOS direct console I/O function
0009 =                  PSTRNG  EQU     09H         ;BDOS print string function
0011 =                  SRCHF   EQU     11H         ;BDOS search for first function
0012 =                  SRCHN   EQU     12H         ;BDOS search for next function
0013 =                  DELF    EQU     13H         ;BDOS delete file function
0019 =                  DSKNO   EQU     19H         ;BDOS return current disk function
                        ;
0007 =                  BELL    EQU     07H         ;ASCII bell
000A =                  LF      EQU     0AH         ;ASCII line feed
000D =                  CR      EQU     0DH         ;ASCII carriage return
                        ;
000F =                  FNCNT   EQU     0FH         ;File name character count
005C =                  FCB     EQU     05CH        ;Default file control block address
0080 =                  DEFDMA  EQU     0080H       ;Default DMA address
                        ;
                        ;***** MAIN PROGRAM *****
                        ;
0100                    XERA    ORG     0100H       ;Start of main program
0100 31960B             LXI     SP,STACK;Set stack pointer to local stack
0103 CD6901             CALL    INIT    ;Initialize variables
                        ;
0106 CD7D01             CALL    SETFTPRS;Set file name buffer & erase vector pointers
0109 CD8A01             CALL    SFIRST  ;Search for first match
010C CD9C01             CALL    QFFND   ;Check to see if a file match was found
010F C21B01             JNZ     FFND    ;Jump to FFND if a file match was found
0112 113403             LXI     D,NFMSG ;Print "NO FILE" if no file match
0115 CDE202             CALL    PRINT
0118 C30000             JMP     BOOT    ;Return to CP/M
                        ;
011B CDA401     FFND    CALL    MOVNAME ;Move current file name to file name buffer
011E CDC101             CALL    INCBUFF ;Increment file name buffer pointer
0121 CD9301             CALL    SNEXT   ;Search for next file match
0124 CD9C01             CALL    QFFND   ;Check to see if a file match was found
0127 C21B01             JNZ     FFND    ;Loop to FFND if a file was found
                        ;
012A 210E04             LXI     H,NAME  ;Reset file name buffer pointer
012D 228503             SHLD    BUFF
0130 CDCC01             CALL    SAVECNT ;Save copy of directory file count
0133 CDD301             CALL    QDSK    ;Determine name of specified disk
0136 111803             LXI     D,CRLF  ;Print carriage return & line feed
0139 CDE202             CALL    PRINT
013C CD8901     NEXTFN  CALL    TFATT   ;Test file attributes (R/O and SYS check)
```

```
013F CD0002             CALL    PRTFN   ;Print current file name
0142 CD4C02             CALL    PFATT   ;Print file attributes if R/O and/or SYS
0145 CC6302             CZ      QERA    ;If DIR and R/W ask if file to be erased
0148 CD9302             CALL    INCERA  ;Increment erase vector pointer
014B CDC101             CALL    INCBUFF ;Increment file name buffer pointer
014E 218703             LXI     H,DFCNT ;Decrement directory file count register
0151 35                 DCR     M
0152 C23C01             JNZ     NEXTFN  ;Loop to NEXTFN if more files

0155 3A8D03             LDA     QFLAG   ;Test query flag to see if any files
0158 FEFF               CPI     0FFH    ;are to be erased
015A C20000             JNZ     BOOT    ;Return to CP/M if no files to be erased
015D CD7D01             CALL    SETPTRS ;Reset file name buffer & erase vector pointers
0160 CD9B02             CALL    QOK     ;Ask if indicated files should be erased
0163 DCBF02             CC      DOERA   ;Erase indicated files if O.K.
0166 C30000             JMP     BOOT    ;Return to CP/M

                ;       *** SUBROUTINES ***

                ;       INITIALIZE VARIABLES

0169 3E00       INIT    MVI     A,0     ;Clear directory file count register
016B 328703             STA     DFCNT
016E 328D03             STA     QFLAG   ;Clear query flag
0171 218E03             LXI     H,ERASE ;Clear erase vector
0174 0680               MVI     B,128
0176 77         NEXTEV  MOV     M,A
0177 23                 INX     H
0178 05                 DCR     B
0179 C27601             JNZ     NEXTEV
017C C9                 RET

                ;       SET FILE NAME BUFFER AND ERASE VECTOR POINTERS

017D 210E04     SETPTRS LXI     H,NAME  ;Store address of start of file name buffer
0180 228503             SHLD    BUFF    ;in pointer BUFF
0183 218E03             LXI     H,ERASE ;Store address of start of erase vector in
0186 228A03             SHLD    EVCTR   ;pointer EVCTR
0189 C9                 RET

                ;       SEARCH FOR FIRST FILE

018A 115C00     SFIRST  LXI     D,DFCB  ;Search for first file match
018D 0E11               MVI     C,SRCHF
018F CD0500             CALL    BDOS
0192 C9                 RET

                ;       SEARCH FOR NEXT FILE

0193 115C00     SNEXT   LXI     D,DFCB  ;Search for next file match
0196 0E12               MVI     C,SRCHN
0198 CD0500             CALL    BDOS
019B C9                 RET

                ;       DETERMINE IF FILE FOUND

019C FEFF       QFFND   CPI     0FFH    ;Did search operation return -1 in ACC?
019E C8                 RZ              ;If ACC = -1 (file not found) then return
019F 218703             LXI     H,DFCNT ;Increment directory file counter (file found)
01A2 34                 INR     M
01A3 C9                 RET

                ;       MOVE FILE NAME FROM DEFAULT DMA ADDRESS TO FILE NAME BUFFER

01A4 2A8503     MOVNAME LHLD    BUFF    ;Set file name buffer pointer in DE
01A7 EB                 XCHG
01A8 210000             LXI     H,DFDMA ;Set default DMA address file name ptr in HL
01AB 87                 ADD     A       ;(Search functions return four FCB's in 128-
01AC 87                 ADD     A       ;byte default buffer, with returned value
01AD 87                 ADD     A       ;in A indicating which is the correct FCB.
01AE 87                 ADD     A       ;Multiplying A by 32 and adding the result
01AF 87                 ADD     A       ;to HL makes HL point to the correct FCB.)
01B0 85                 ADD     L
01B1 6F                 MOV     L,A
01B2 3A5C00             LDA     DFCB    ;Get specified disk number from DFCB
01B5 77                 MOV     M,A     ;Store it in current file control block
01B6 060F               MVI     B,FNCNT ;Set file name character counter
01B8 7E         NEXTCH  MOV     A,M     ;Move file name characters
01B9 12                 STAX    D
01BA 23                 INX     H
01BB 13                 INX     D
01BC 05                 DCR     B
01BD C2B801             JNZ     NEXTCH
01C0 C9                 RET

                ;       INCREMENT FILE NAME POINTER

01C1 2A8503     INCBUFF LHLD    BUFF    ;Get current value of file name buffer pointer
01C4 110F00             LXI     D,FNCNT ;Increment by 15 bytes
01C7 19                 DAD     D
01C8 228503             SHLD    BUFF    ;Store new value
01CB C9                 RET

                ;       SAVE COPY OF DIRECTORY FILE COUNT

01CC 3A8703     SAVECNT LDA     DFCNT   ;Get number of directory files
01CF 328803             STA     DFCNT1  ;Store in aux. directory file count register
01D2 C9                 RET

                ;       DETERMINE NAME OF SPECIFIED DISK

01D3 3A5C00     QDSK    LDA     DFCB    ;Get disk number from default FCB
01D6 FE00               CPI     0       ;If none specified then get current disk number
01D8 CCE101             CZ      CURRENT
01DB C640               ADI     40H     ;Convert number to ASCII disk name
01DD 328903             STA     DSK     ;Store it in DSK
01E0 C9                 RET

01E1 0E19       CURRENT MVI     C,DSKNO ;Get current disk number
01E3 CD0500             CALL    BDOS
01E6 C601               ADI     1
01E8 C9                 RET

                ;       TEST FILE ATTRIBUTES FOR "READ-ONLY" AND "SYSTEM"

01E9 2A8503     TFATT   LHLD    BUFF    ;Set HL to point to current file name
01EC 110900             LXI     D,9     ;Add 9-byte offset to point to R/O file
01EF 19                 DAD     D       ;attribute byte
01F0 7E                 MOV     A,M     ;Get R/O file attribute byte
01F1 E680               ANI     80H     ;Mask off R/O attribute bit
01F3 07                 RLC             ;Save result as bit 1 of file attribute byte
01F4 47                 MOV     B,A     ;Store it temporarily in register B
01F5 23                 INX     H       ;Increment pointer to SYSTEM attribute byte
01F6 7E                 MOV     A,M     ;Get SYS file attribute byte
01F7 E680               ANI     80H     ;Mask off SYS attribute bit
01F9 07                 RLC             ;Save result as bit 2 of file attribute byte
01FA 07                 RLC
01FB 80                 ADD     B       ;Add bit 1 from register B
01FC 32C003             STA     FATT    ;Store result in file attribute register
01FF C9                 RET

                PRINT FILE NAME

0200 3A8903     PRTFN   LDA     DSK     ;Print disk name
0203 5F                 MOV     E,A
0204 0E02               MVI     C,CONOUT
0206 CD0500             CALL    BDOS
0209 111503             LXI     D,COLON ;Print ": "
020C CDE202             CALL    PRINT

020F 2A8503             LHLD    BUFF    ;Set HL to point to start of file name buffer
0212 23                 INX     H       ;Set pointer to first file name character
0213 0608               MVI     B,8     ;Set counter to number of file name characters
0215 E5         NEXTN2  PUSH    H       ;Save pointer and counter
0216 C5                 PUSH    B
0217 7E                 MOV     A,M     ;Get file name character
0218 E67F               ANI     7FH     ;Strip off top bit (if any)
021A 5F                 MOV     E,A     ;Print ASCII character
021B 0E02               MVI     C,CONOUT
021D CD0500             CALL    BDOS
0220 C1                 POP     B       ;Recall counter and pointer
0221 E1                 POP     H
0222 23                 INX     H       ;Increment pointer
0223 05                 DCR     B       ;Decrement character counter
0224 C21502             JNZ     NEXTN2  ;Loop to NEXTN2 if more file name characters

0227 E5                 PUSH    H       ;Save pointer
0228 1E2E               MVI     E,'.'   ;Print '.'
022A 0E02               MVI     C,CONOUT
022C CD0500             CALL    BDOS
022F E1                 POP     H       ;Recall pointer

0230 0603               MVI     B,3     ;Set counter to number of file type characters
0232 E5         NEXTT2  PUSH    H       ;Save pointer and counter
0233 C5                 PUSH    B
0234 7E                 MOV     A,M     ;Get file type character
0235 E67F               ANI     7FH     ;Strip off top bit (if any)
0237 5F                 MOV     E,A     ;Print ASCII character
0238 0E02               MVI     C,CONOUT
023A CD0500             CALL    BDOS
023D C1                 POP     B       ;Recall counter and pointer
023E E1                 POP     H
023F 23                 INX     H       ;Increment pointer
0240 05                 DCR     B       ;Decrement character counter
0241 C23202             JNZ     NEXTT2  ;Loop to NEXTT2 if more file type characters

0244 1E20               MVI     E,' '   ;Print ' '
0246 0E02               MVI     C,CONOUT
0248 CD0500             CALL    BDOS
024B C9                 RET
```

XERA continued . . .

```
                ;       PRINT FILE ATTRIBUTES IF "READ/ONLY" AND/OR "SYSTEM"
024C 3A8C03     PPATT   LDA     FATT    ;Get file attribute register contents
024F 4F         MOV     C,A     ;Store it as a 16-bit value in BC
0250 0600               MVI     B,0
0252 217D03             LXI     H,ATTMSG ;Set HL to point to base of msg. ptr. table
0255 09                 DAD     B       ;Add appropriate message pointer offset as
0256 09                 DAD     B       ;indicated by file attribute register contents
0257 5E                 MOV     E,M     ;Move address of message indicated by pointer
0258 23                 INX     H       ;to DE
0259 56                 MOV     D,M
025A CDE202             CALL    PRINT   ;Print the appropriate message (will print
                                        ;one of RDWR, RDON, SYST or BOTH)
025D 3A8C03             LDA     FATT    ;Get file attribute register contents
0260 FE00               CPI     0       ;Test for R/W and DIR File attributes
0262 C9                 RET

                ;       ASK IF FILE IS TO BE ERASED

0263 11F502     QERA    LXI     D,ASK   ;Print "? "
0266 CDE202             CALL    PRINT
0269 CD8802             CALL    CHARIN  ;Get operator's response
026C FE03               CPI     03      ;Return to CP/M if 'C'
026E CA0000             JZ      BOOT
0271 FE59               CPI     'Y'     ;Jump to SETERA if it is "Y" or "y"
0273 CA8202             JZ      SETERA
0276 FE79               CPI     'y'
0278 CA8202             JZ      SETERA
027B 113D03             LXI     D,NO    ;Print "No"
027E CDE202             CALL    PRINT
0281 C9                 RET

0282 2A8A03     SETERA  LHLD    EVCTR   ;Set Erase File flag in erase vector
0285 36FF               MVI     M,0FFH
0287 3EFF               MVI     A,0FFH  ;Set query flag QFLAG to indicate that there
0289 328703             STA     QFLAG   ;are files to be erased
028C CDE203             CALL    PRINT   ;Print "Yes"
028F C9                 RET

                ;       INCREMENT ERASE VECTOR POINTER

0293 2A8A03     INCERA  LHLD    EVCTR   ;Get current value of erase vector pointer
0296 23                 INX     H       ;Increment it
0297 228A03             SHLD    EVCTR   ;Store new value
029A C9                 RET

                ;       ASK IF INDICATED FILES SHOULD BE ERASED

029B 114203     QOK     LXI     D,OKERA ;Print "O.K. to erase?"
029E CDE202             CALL    PRINT
02A1 CD8802             CALL    CHARIN  ;Get operator's response
02A4 FE59               CPI     'Y'     ;Jump to ERAYES if it is "Y" or "y"
02A6 CAB702             JZ      ERAYES
02A9 FE79               CPI     'y'
02AB CAB702             JZ      ERAYES
02AE 11B803             LXI     D,NOAB  ;Print "No - FILES NOT ERASED"
02B1 CDE202             CALL    PRINT
02B4 37                 STC             ;Set Carry Status to 0 as "NOT ERASED" flag
02B5 3F                 CMC
02B6 C9                 RET

02B7 117703     ERAYES  LXI     D,YES   ;Print "Yes"
02BA CDE202             CALL    PRINT
02BD 37                 STC             ;Set Carry Status to 1 as "ERASED" flag
02BE C9                 RET

                ;       ERASE INDICATED FILES

02BF 3A8803     DOERA   LDA     DFCNT1  ;Set counter to number of directory files
02C2 47                 MOV     B,A
02C3 2A8A03             LHLD    EVCTR   ;Is Erase File flag in erase vector set?
02C6 7E                 MOV     A,M
02C7 FEFF               CPI     0FFH
02C9 C2D702             JNZ     NOERA   ;Jump to NOERA if not set
02CC C5                 PUSH    B       ;Save file counter
02CD 2A8503             LHLD    BUFF    ;Delete file
02D0 EB                 XCHG
02D1 0E13               MVI     C,DELF
02D3 CD0500             CALL    BDOS
02D6 C1                 POP     B       ;Recall counter
02D7 CD9302             CALL    INCERA  ;Increment erase vector pointer
02DA CDC101             CALL    INCBUFF ;Increment file name buffer pointer
02DD 05                 DCR     B       ;Decrement file counter
02DE C2C302             JNZ     NEXTFC  ;Loop to NEXTFC if more files
02E1 C9                 RET

                ;       GENERAL PRINT STRING ROUTINE

02E2 0E09       PRINT   MVI     C,PSTRNG
02E4 CD0500             CALL    BDOS
02E7 C9                 RET
```

XERA continued . . .

```
                ;       GENERAL DIRECT INPUT ROUTINE
02E8 1EFF       CHARIN  MVI     E,0FFH  ;Wait for operator's response
02EA 0E06               MVI     C,DIRCIO
02EC CD0500             CALL    BDOS
02EF FE00               CPI     0
02F1 CAE802             JZ      CHARIN  ;Loop to CHARIN if no response yet
02F4 C9                 RET
                ;       *** DATA STATEMENTS ***
02F5 3F20202024ASK      DB      '?   $'         READ/ONLY and SYSTEM',CR,LF,'$'
02FA 20202020520TH      DB      '    R'
0315 3A2024     COLON   DB      ': $'
0318 0D0A24     CRLF    DB      CR,LF,'$'
031B 07456F2020NOAB     DB      BELL,'No - FILES NOT ERASED',CR,LF,'$'
0334 4E6F0D0A24NO       DB      'No',CR,LF,'$'
033D 0D4E4F2046NFMSG    DB      CR,LF,'NO FILES$'
0342 0D0A4F2E4BOKERA    DB      CR,LF,'O.K. to erase? $'
0359 2020202052RDON     DB      '   READ-ONLY',CR,LF,'$'
0369 24         RDWR    DB      '$'
036A 2020202053SYST     DB      '   SYSTEM',CR,LF,'$'
0377 596573 0DOAYES     DB      'Yes',CR,LF,'$'
                ;       *** MESSAGE POINTER TABLE ***
037D 6903       ATTMSG  DW      RDWR
037F 5903               DW      RDON
0381 6A03               DW      SYST
0383 FA02               DW      BOTH
                ;       *** RESERVED DATA AREA ***
0385            BUFF    DS      128     ;File name buffer pointer
0387            DFCNT   DS      2       ;Directory file count register
0388            DFCNT1  DS      1       ;Auxiliary directory file count register
0389            DSK     DS      1       ;Specified disk name register
038A            EVCTR   DS      2       ;Erase vector pointer
038C            FATT    DS      1       ;File attribute register
038D            QFLAG   DS      1       ;Query flag
038E            ERASE   DS      128     ;Beginning of 128 byte File Erase vector
040E            NAME    DS      128*FNCNT ;Beginning of 128 15-byte File name buffers
0B8E                    DS      8       ;4-level local stack
0B96            STACK   END
```

EASI STRUCTURAL ANALYSIS

EASI ($300.00) is a data based oriented finite element program which solves plane frames, floor grids, space trusses, and space frames. The program computes member forces, node displacements, and restraint reactions. The semi-banded matrix solution allows large problem sizes in 48 K - 64 K microcomputers. The program is coded in Microsoft BASIC and is available on TRS-80 & CP/M computers.

EASI "4+" ($995.00) all of the above plus the following features:

RELEASED MOMENTS - end moments can be released on individual members.

IMPOSED DISPLACEMENTS - nodes can be set to a predetermined displacement prior to the analysis.

THERMAL LOADS — accounts for temperature changes in the structure. Typical uses of this feature are steam pipes, uneven solar heating, and the effects of welding.

3D PLOTTING ON DOT GRAPHICS PRINTER (graphics screen) — When a mode is three dimensional, the program prompts the user for rotations (degrees) about the x, y, & z axis, then plots (with amplified displacements) that view. A low resolution graphics option is also available for 80 column screen display or standard printers.

WEIGHT REPORTING — tabulates structural weights by material type.

ANISOTROPIC MATERIALS — can model orthotropic or anisotropic materials (i.e. fiber-glass, plywood, etc.). It permits elements to have different material stiffness in the local x and y directions of the element.

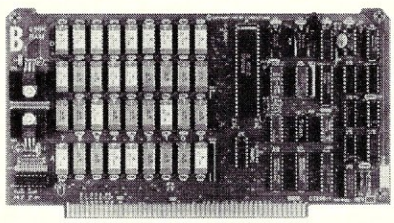
MEMBRANE ELEMENT (triangular and quadrilateral) - the membrane element can be incorporated into the space (plane) frame elements to ANALYZE SHEAR WALLS.

The **EASI "4+"** program is coded in baZic and requires CP/M. The baZic (included with program) is upward compatable with North Star Basic and executes programs very fast. TRS-80 computers will require CP/M.

EASI Software Inc.

CIRCLE 36 ON READER SERVICE CARD

SMARTS DISK



the RAMDISK™ Alternative
A&T $550 Kit $500
HOBBYISTS:

Computer in a Bag!

Consists of 4Mhz Z-80,
UFDC-I Floppy Controller,
256K DRAM Memory and
Monitor/BIOS Disk w/SMARTS
Disk Software (all Bare Boards) $200

SYSTEMS UPGRADE

4Mhz Z-80 CPU, UFDC-I
256K memory, CP/M 2.2
fully configured $1050

For sales or additional info:

GSR Computers
60-10 69th St. or
Maspeth, NY 11378
(212) 476-2091

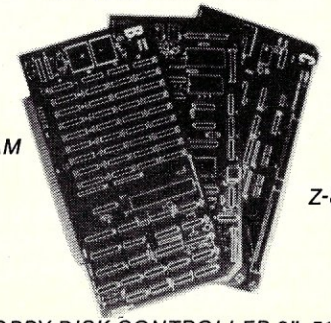
Source Mail
STA 767
Compuserve
EMAIL
70265,500

CIRCLE 22 ON READER SERVICE CARD

S-100 BUS
BARE BOARDS
COMPUTER IN A BAG

ADD PARTS AND LABOR

256K-DRAM



Z-80 CPU

FLOPPY DISK CONTROLLER 8"x5¼"

$150 - This Three Board Set

OTHER BAREBOARDS AVAILABLE

	Prod. Code	*Price*
256K - DYNAMIC RAM	CT256K-1	$70.00
64K STATIC RAM	S-64K	$60.00
Z80 PROCESSOR	SBC-880	$45.00
FLOPPY DISK CONTROLLER	UFDC-1	$60.00
CLOCK/CALENDAR - WITH BATTERY BACKUP	S100-880	$45.00

ALL MANUALS INCLUDED

CA Residents - Add Sales Tax
$3 - min. S & H

(714) 536-5000

COMPUTIME
8614 HAMILTON AVENUE
HUNTINGTON BEACH, CA 92646

CIRCLE 210 ON READER SERVICE CARD

SOFTWARE REVIEW

Access Manager

by Bruce H. Hunter

Access Manager is a utility for the creation of database-managed systems. A product of Digital Research, this utility is a B+ tree (hierarchical) file system for the building and accessing of data and index files. Access Manager (or AM-80 for short) is designed to interface with Digital Languages such as the CBasic Compiler, Pascal/MT+, and PL/I-80. It is presently furnished in an 8-bit version and will be furnished in a 16-bit version. It is both CP/M and MP/M compatible and has both file and record locking.

The mark of any good database system is its storage capability and its accessing speed. In an MP/M environment, AM-80 will allow file sizes of 32 Mb each. The maximum number of open data files is 40. Also the maximum number of index files is 40. Sixty simultaneously open buffers are permissible. The result is well over a gigabyte of open storage. The CP/M version is limited to 20 open data files of 8 Mb each and 10 open index files. Because of the B+ tree organization, a substantial number of indices are stored at each node, and the number of passes through the index to find a particular record is unbelievably small. It is possible to search nearly 200,000 records in no more than four passes. Compare this to a binary tree that would require 18 passes for the same accomplishment. Data entered to index files is sorted sequentially. Once entered, data is never moved. The index files do all the housekeeping.

AM-80 is a series of routines (program files) that are linked to the applications programs(s) at the final linkage. For a single user system, a subroutine library and a buffer are all that need be linked to the program. For multiuser systems, three components are required: the subroutine interface, the shared multiple-user background server, and multiple-user resident system processes. The codes generated by these files are reasonably small, requiring less than 12K for all the functions. Multiuser object code will be increased by no more than 23K.

Performance

In practice, Access Manager is relatively easy to use. A header file is included into the object code at runtime by way of a preprocessor command. For example, in PL/I the command

```
%include 'am80extr.pli';
```

will cause the header file to be read into the compiler as part of the object code. It saves about half a page of coding by declaring all the files and variables external to the program.

There are only four functions necessary to call for system initialization. Data and index files are opened or closed with a single function call. The most intriguing part of Access Manager is the set of index search functions. The functions available are:

- getkey—searches for an exact match
- serkey—searches for a match that will be equal or greater
- befkey—finds the key before the target key
- aftkey—finds the key following the target key
- frskey—returns the first key in the index file
- laskey—finds the last key in the index file
- prvkey—finds the key immediately preceding the the last entry found*
- nxtkey—finds the key after the last key found*

**not recommended under MP/M*

The distinction between getkey and serkey is that getkey expects an exact match, as in looking for an I.D. number, while serkey is less exact, as in searching for a name that could be subtly different, like Alfred Newman vs. Alfred E. Newman.

The system not only allows for the deletion of data records, but also allows for recovery of the space taken up by the record by returning that space back to the system for re-use. Statistical functions are available to show the number of records in use versus the actual number of records. Functions are also available for the setting and unlocking of file locks.

Keys are stored as unsigned two-byte integers for most applications, but for unusually large files, a four-byte integer can be used with some additional housekeeping to bring capacity to the gigabyte range. Data files may be of any reasonably length as long as they are at least four bytes in length. The CP/M restriction of 128-byte logical records has been nicely overcome by the very sophisticated housekeeping performed by AM-80.

In practice, Access Manager is a pleasure to use. Once the preprocessor inclusion command has been put into the code, the variable names that are to be used in the code are declared (this not being necessary in Basic, of course). The number of buffers, keys and index files are defined, and the lock request is or is not made. Then the system is initialized. In all, the last few tasks will account for less than a dozen lines of code. Had this been done in BT-80 instead, a couple of pages would have to be ground out before seeing an executable line of code (something like programming in Cobal). In AM-80 you enter executable code at the bottom of the first page of code. Opening a couple of data files and a couple of index files takes a bit less than half a page more of code.

In my code, index and data records are written simultaneously at a cost of a half dozen lines per write. Error codes are returned on nearly every AM-80 routine, and error processing functions can be created with reasonable ease by the programmer to keep the program from "falling down." The search routines are straightforward and easily implemented. On the average, they take about four arguments each. For example:

```
record-number = nxtkey
        (act-key, fil-no, lock, idxval);
```

Bruce H. Hunter, 1020 S. Jenifer, Glendora, CA 91740

Access Manager continued . . .

A mailing list or a chart of accounts can be written with Access Manager in about 10 pages of code, using almost all of AM-80's facilities, including a procedure to rebuild the index files should they become "corrupted" by a premature closing.

The documentation for Access Manager is a well-written 178-page manual. It tends to get just a bit obscure on the handling of numeric key values and on the index value of the output argument, IDXVAL, but other than that it reads easily. There is a new manual coming out for the 16-bit version, AM-86 that is supposed to be even better, and it will also be the new manual for AM-80. DRI is making a distinct effort to have common manuals for both 8-and 16-bit versions of just about everything coming out of the Language Division to demonstrate the near-total compatability of the versions. Like most of DRI's recent documentation, an index is provided. There are a number of programming examples included. The appendix has a program to create a database. It is written in Basic, Pascal, and PL/I.

Unfortunately, the Pascal and PL/I versions are direct translations of the Basic program. I say "unfortunately," because Basic and Pascal of necessity must be programmed bottom up as opposed to the top down programming that PL/I thrives on. The result is that the code is a bit difficult to follow. An entire re-creation program is furnished as well, to rebuild an index that has been inadvertently destroyed. It is educational and downright handy.

I have written a number of business programs with AM-80, and to date have found no bugs. Correspondence with a beta test site of AM-80 has revealed no bugs either. It speaks well for DRI's beta testing.

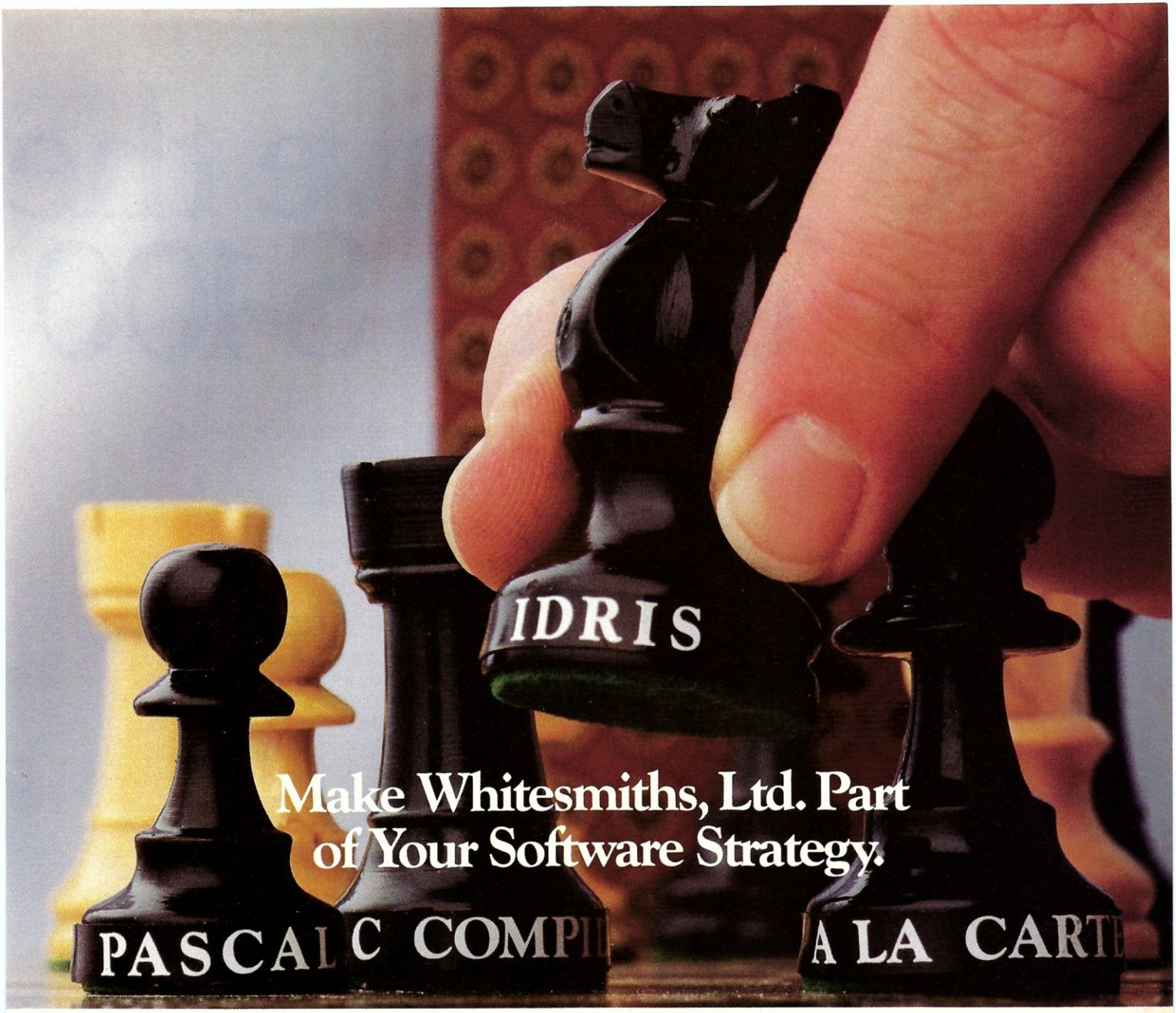
There are few utilities comparable to AM-80. I have been informed that there is a system called Micro B+ written by Bill Fairman of FairCom that is reputed to be a close match. The closest that I have worked with personally is Digital's BT-80. I brought up BT-80 about a year earlier and found that the amount of pointer handling involved and the enormous amount of initialization required made the utility difficult to use. In addition, BT-80 is compatable only with PL/I-80 (which was no hardship, for me as PL/I is my language of choice, but it is not everyone's choice). AM-80 is compatable with a minimum of three languages. There are a few DBMSs available that provide a language interface. Among them are MDBMS, TIM, CCA, and IDM-M2. Of these, the closest to AM-80 would be MDBMS (at $900). The majority of DBMSs, like Ashton Tate's dBase II, are not programmatically accessable. Access Manager is priced at $300 and is available from both Digital Research directly and through the normal distribution channels. From a consumer's point of view, I should mention that software purchased directly from DRI, although not discounted, does have the advantage of being refundable if returned within a reasonable pe-

Access Manager continued . . .

riod of time (for a nominal restocking charge).

Programs written using AM-80 to create a database-managed system could and should rival mainframe systems. Retrieval speeds are phenomenal. A common database built by these programs is accessable from any other program using AM-80, even if it is written in another language. Additional benefits are extremely compact data and index files. AM-80 allows the programmer to assign only as little space as required by the records. In spite of the fact that it still has fixed-length records, the removal of the restriction of multiples of the logical record length allows fantastic economy of storage. Couple this with gigabyte capacities, and you have the capability of writing an application program to run under CP/M that would otherwise need a $10,000 computer running MP/M. In performance it may rival programs run on a much larger machine.

Most of the programs I have written and linked with AM-80 routines produced under 30K of machine code, including high-speed storage allocation. The need for using large amounts of storage for sorting data is eliminated because AM-80 has kept the keys in ascending sort order automatically by its built-in housekeeping routines. With these routines, the writing of search and sort routines becomes academic. For any programmer involved in the writing of database systems in one of Digital's 3 languages (CBasic, Pascal MT+, or PL/I-80), AM-80 will be a welcome aid.



Make Whitesmiths, Ltd. Part of Your Software Strategy.

Selecting software is really a game of strategy: positioning the right product at the right time for the most effective results.

At Whitesmiths, Ltd., our full line of software products is designed to do just that and more. Our time-tested compilers are available for five architectures, complete with Pascal translators and cross-support for a wide range of environments. From the MC68000, to the IBM PC, to the DEC Professional, our compilers and cross compilers cut software development time and reduce dependence on operating systems and instruction sets.

Idris, our exclusive operating system, is powerful, small and extremely portable. Written almost entirely in C, it runs where other operating systems can't even fit – on an MC68000 with no memory management hardware, for example, or on a bank-switched 8080 or Z80. And it outperforms UNIX on comparable systems – an MC68000 with memory management hardware, for instance, or any PDP-11 or LSI-11. It even makes good use of a VAX, so you know you won't outgrow it!

In addition, our full line of Software A La Carte lets you select just the pieces you need to tailor our products for your unusual requirements. From portable libraries to special purpose tools, all A La Carte items are tested, easy to use, and very affordable.

To make things even simpler, Whitesmiths, Ltd. has introduced a new marketing tool: the Authorization Seal. Instead of hassling with complex license agreements and other legal stumbling blocks, you simply affix a serially numbered Authorization Seal, issued with each sale, to any machine that retains a copy of our software. End of hassle. All our customers – end-users, OEMs, and sublicensors – benefit from the elimination of multi-tier licensing agreements and detailed record keeping.

Compilers, Idris, Software A La Carte, and the Authorization Seal: together they can position you for today's competitive market – and tomorrow's. Before you make your next move, call or write to Whitesmiths, Ltd. today. We'll be glad to help you play to win.

Whitesmiths, Ltd.,
97 Lowell Road, Concord,
MA 01742, (617) 369-8499
TLX 951708 SOFTWARE CNCM.

Whitesmiths, Ltd.
Software Craftsmen

Do you have the the entire S-100

We do.

And now you can too: The most powerful multi-processing CPU board you can buy. The world's first 256K static RAM memory board. And the most sophisticated hard disk controller board available.

Run 8-bit CP/M 80* and 16-bit CP/M 86* simultaneously in a single or multi-user environment. $895.

Now you can upgrade to 16-bit without forsaking your investment in 8-bit software. Octagon's plug-and-play CPU Board 8/16™ is a complete computer-on-a-board with all this:

- Dual processors: 4MHz NSC-800 (executes full Z80* instruction set) and 8MHz 8088.
- Intel 8272 floppy disk controller with 24-bit DMA. Runs up to four 5¼" or 8" floppies in any combination at the same time.
- 8K PROM monitor. Boots operating system and also contains all the disk interface software you need for almost any operating system. Plus many useful debugging features, such as memory test, memory dump and memory search.
- Two serial ports (baud rate software selectable up to 19.2 Kbauds.)
- Interrupt controller with eight vectored interrupts.
- Fixed frequency real-time clock interrupt for multi-user dispatching.
- IEEE-696 S-100 compatible.

Add the optional 8087 math co-processor for just $300.

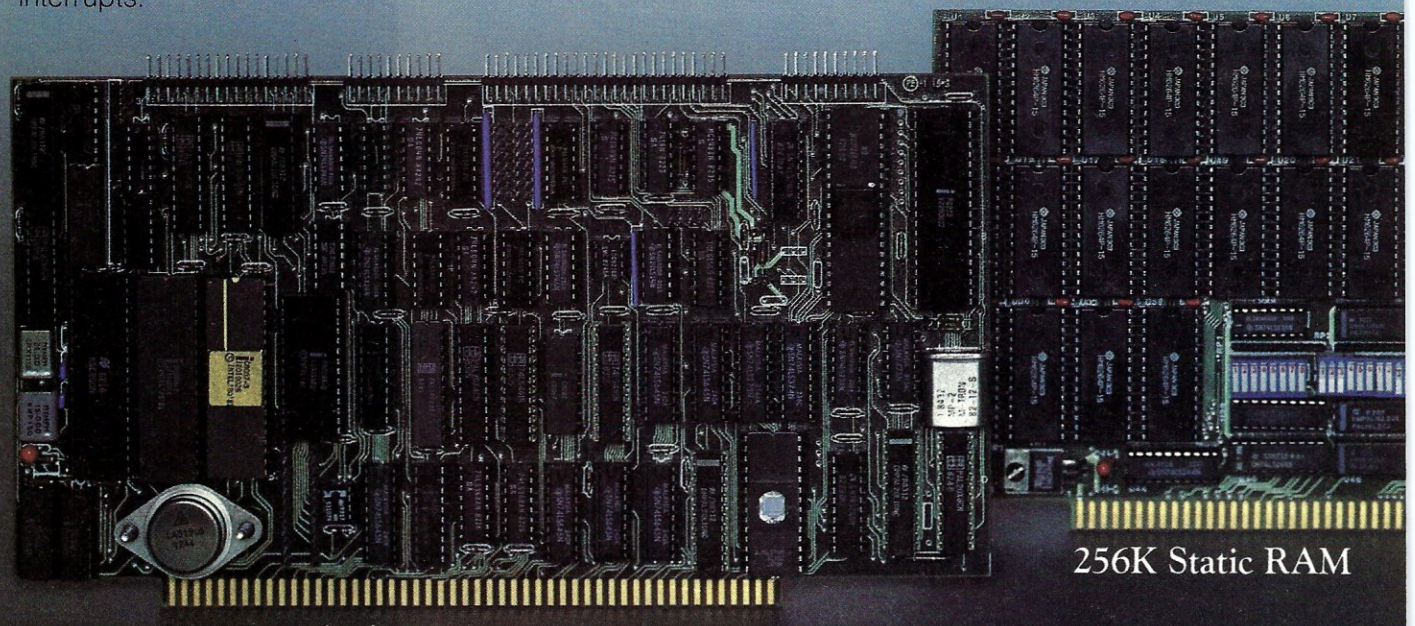
And save $155 when you buy the operating system from Octagon, whether you take Concurrent CP/M 86 at $195, CP/M 86 at $150 or MP/M 86* at $495.

CPU Board 8/16 is completely ready to run with no additional software. And both operating systems automatically include your CP/M 80 emulator, so you can run any 8-bit CP/M program—as well as auto-select bios that let you transfer files back and forth between 5¼" and 8" floppies.

World's first 256K Static RAM Memory Board. $1850

Memory may be configured as either four totally independent 64K software-selectable blocks or as a single 256K block which responds to IEEE-696 extended addressing lines. Will accept either 8-bit or 16-bit bus requests.

And each 64K block can be one of 256 addresses—so theoretically you could address



CPU Board 8/16

256K Static RAM

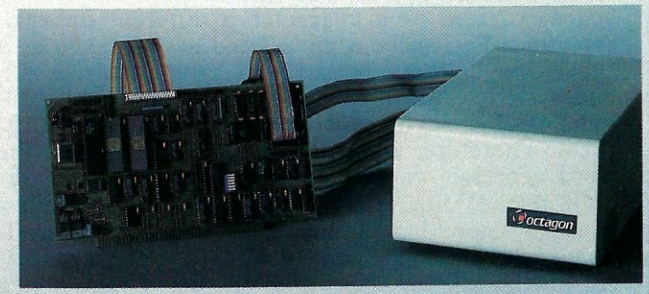
guts to take on micro industry?

up to 16MB of these boards in your system.

A partially populated 128K board with the same characteristics is just $1095.

Sophisticated, reliable Hard Disk Controller. $550.

Designed around LSI technology for greater reliability. With its sophisticated Western Digital



1010 hard disk controller chip to control up to four 5¼" drives, this unique board gives you automatic seek, automatic retries after error; comes with both CP/M 80 and CP/M 86 bios; and includes two serial ports and one Centronics-compatible parallel port.

Using the WD 1014, it detects and corrects single-bit errors and detects double-bit errors. ECC generation and correction on data fields is user selectable.

And its phase lock loop lets you forget about critical timing adjustments. In fact, you never need to adjust it at all. There's no pot.

Octagon's Hard Disk Controller Board also gives you data rates up to 5MBits/sec., and CRC generation and verification on ID fields. Complete subsystem including board and 19.2 MB hard disk packaged with power supply: $2195.

1-408-262-7777

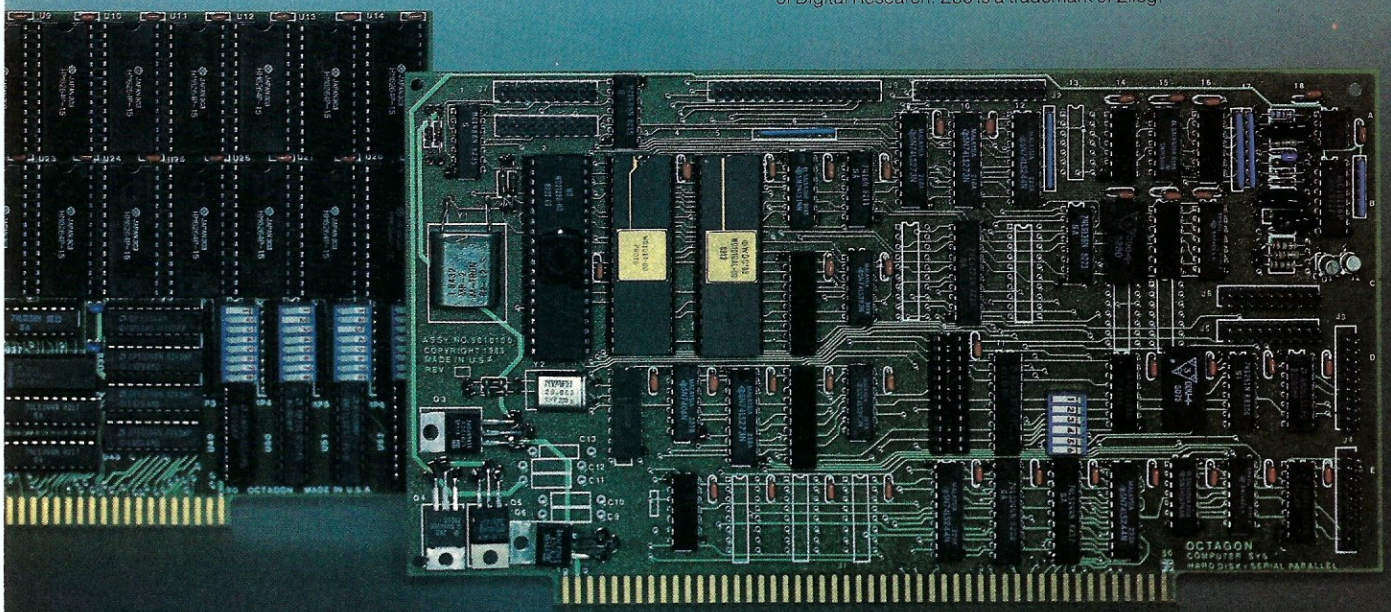
Call us now for more details. And take advantage of these introductory offers: Buy CPU Board 8/16 plus the Memory Board and get Concurrent CP/M 86 or CP/M 86 free. (Includes CP/M 80 emulator.) Or buy all three boards and get MP/M *plus* new Personal BASIC from Digital Research free.

You'll never need any other S-100 boards. It takes guts to say that, but try us out and you'll know: Octagon has what it takes.

octagon

2960 North First St., San Jose, CA 95134 408-262-7777

*CP/M 80, CP/M 86, MP/M 86 and Personal BASIC are trademarks of Digital Research. Z80 is a trademark of Zilog.



Hard Disk Controller

CIRCLE 189 ON READER SERVICE CARD

SOFTWARE REVIEW

Fancy Font

A new software package that marries CP/M and the MX series of Epson printers

by Bob Kowitt

With Fancy Font, you can (1) set print using all of the standard Epson type styles, (2) use a great many type styles supplied by the authors, or—and this is what gives the package some of its greatest strength—(3) create your own type font. The fonts supplied or created by Fancy Font use the Epson bit graphics capabilities, and are therefore very slow printing compared to the standard speed of the Epson line. The results, however, are worth the time whenever special appearance is important.

Perhaps the expression, "create your own type font" is a bit of hyperbole. As anyone familiar with type knows, designing a type font is a major job requiring many months of work. You can, however, take an existing type font and, with the help of a typographic specification book, modify it to resemble another font already in existence. For example, take the simple Sans Serif type font supplied, extend it, thicken it where required, and create a copy of Venus Bold extended.

Fancy Font is more than a system that uses the bit graphics of the Epson printers—it is also a text formatter. You can set margins, justify, print headings and footings, number pages, print selected pages, underline, modify line width, substitute strings within the copy, and do a few other operations from within the text. Copy can be created via any CP/M word processor with word wrap. You must, however, be certain that the word processor's justification is off to allow Fancy Font's justification to take over. Incidentally, the WordStar soft hyphen is recognized and translated to a hard hyphen for printing.

The first lesson should consist simply of producing a document using standard type fonts. SoftCraft, the distributor of Fancy Font, includes 27 type fonts for your use, among them a Roman style in regular, bold, and italic in 8-,10-,11-,12-, and 18-point sizes, a Sans Serif style in the same sizes (with one size in italic) a script font in 12-, 14-, and 18-point, and an Old English font in 18- and 20-point, as well as several other inclusions (see Figure 1).

Your text must include, in addition to the formatting commands mentioned above, directives to Fancy Font to change type fonts. This is done by entering a change character which, as supplied, is the backslash '/' character. Should you use the backslash in your text, it is a simple matter to substitute any other character you want as a change character by calling for it when invoking the program. By including the directive '/f2', Fancy Font is directed to use the second type font in the list of type fonts specified when the program is invoked. This type font will be used until another is called for. Should you not specify any type fonts in the program invocation, Fancy Font will select the standard Epson MX fonts from a table included in the manual: Font 0 is normal, Font 1 is compressed, Font 2 is expanded, etc.

Establishing the correct line width takes a bit of experience. The default setting for the widest line is set at 7.5 inches. This may be modified by altering the FANCFONT.PRO file to read MX100 instead of MX80. Fancy Font recognizes the fact that the space required by a character is different for each type font. This space is not related to that taken by a typewriter or the normal Epson printer font. A 50-character line will print differently using Roman 12-point than it will using Roman 8-point or Roman 18-point. Therefore, when you set your margins on the word processor to 50 and try to print Roman 18-point on a 5" line, you will get an error message letting you know that the line is too long for the available space. If you arbitrarily reduce the word processor line width, you might find out that Fancy Font's right justification puts too much space between words.

I resolved the problem by making a typeface sample sheet. This consisted of 30-character wide sentences of each type font. I selected 30 and not 65, since the larger type fonts would have printed wider than my sheet. Using this sample sheet, I could set my word processor margins to the width that would produce the size desired in the finished product (see Figure 1).

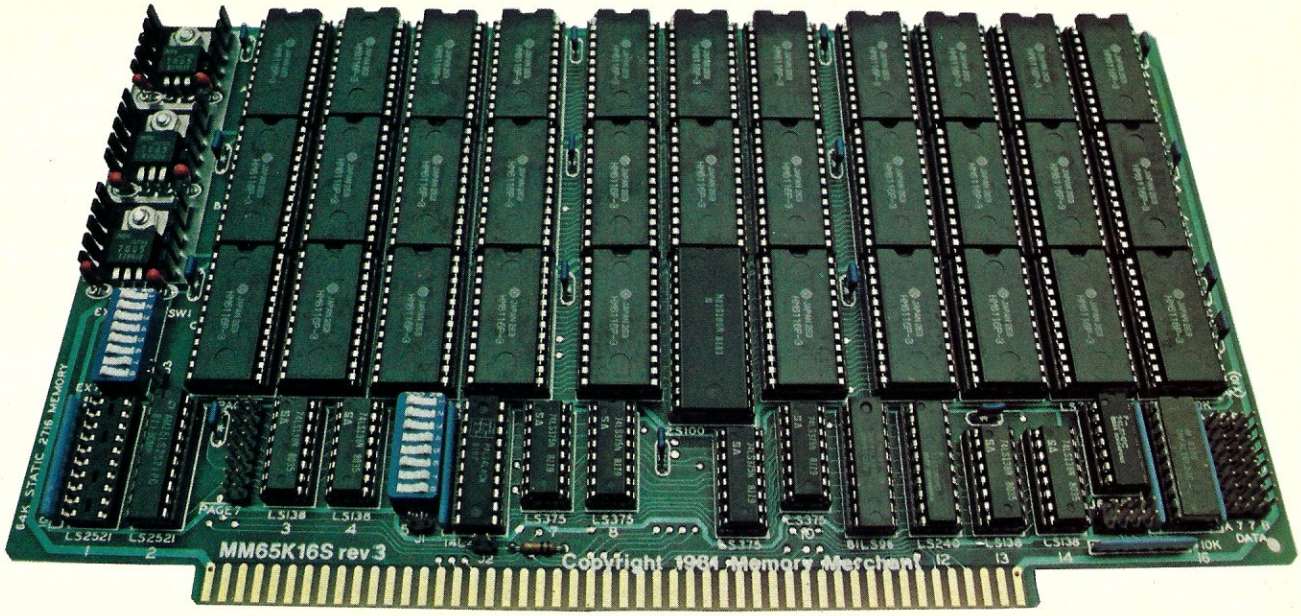
The manual recommends running the PFONT printing program in 'SD' mode during text preparation. This mode does not print the text but does display all lines on the terminal. In this mode you can catch any overlong lines or any other errors you might have made in specifications. Naturally, the lines are not displayed in the special fonts. I would also suggest that the 'RD 2' mode be invoked on the first attempt to print. RD permits printing in a choice of three degrees of roughness. RD 0 is the default mode and is the finest quality available. However, it is slow, since it takes nine one-way passes to produce each line of type in a 12-point size when there are descenders, and six without descenders. RD 2 is rough and certainly not presentable quality. It is, however, a fast presentation of how the text will lie on the page and how the formatting is working.

Once the text is prepared, Fancy Font can be invoked in three ways:

1. within the CP/M command line
2. interactively, with PFONT prompting for input

Bob Kowitt, 1727 North Jerusalem Rd., East Meadow, NY 11554

64K STATIC RAM MEMORY



S-100 STATIC MEMORY BREAKTHROUGH

Finally, you can buy state-of-the-art S-100/IEEE 696 static memory for your computer at an unprecedented savings.

Memory Merchant's memory boards provide the advanced features, quality and reliability you need for the kind of operational performance demanded by new high-speed processors.

Completely Assembled.

These memory boards are not kits, nor skeletons — but top-quality, high-performance memories that are shipped to you completely assembled, burned-in, socketed, tested and insured with one of the industry's best warranties.

Superior Design & Quality.

Memory Merchant's boards are created by a designer, well known for his proven ability in advanced, cost-efficient memory design. Innovative circuitry provides you with highly desired features and incredible versatility.

Only first-quality components are used throughout, and each board is rigorously tested to assure perfect and dependable performance.

No Risk Trial.

We are so convinced that you will be absolutely delighted with our boards that we extend a no-risk trial offer. After purchasing one of our boards, you may return it (intact) for any reason within 15 days after shipment and we will refund the purchase price (less shipping).

NEW S-100 PRODUCTS COMING SOON:

* DUAL 8/16 BIT CPU BOARD
* 128K 8/16 BIT STATIC RAM
* 256K 8/16 BIT DYNAMIC RAM

$629.

48K PARTIALLY POPULATED $519.
32K PARTIALLY POPULATED $409.

64K RAM, MODEL MM65K16S

- 64K × 8-bit
- Speed in excess of 6 MHz
- Uses 150ns 16K (2K × 8) static RAMS
- Ultra-low power (435 Ma. max. — loaded with 64K)
- Bank Select **and** Extended Addressing
- A 2K window which can be placed anywhere in the 64K memory map
- Four independently addressable 16K blocks organized as:
  — Two independent 32K banks **or**
  — One 64K Extended Address Page **or**
  — One 48K and one 16K bank for use in MP/M[1] (option)
- Each 32K bank responds independently to phantom
- 2716 (5V) EPROMS may replace any or all of the RAM
- Field-proven operation in CROMEMCO CROMIX* and CDOS*.
- Compatible with latest IEEE 696 systems such as Northstar, CompuPro, Morrow, IMS, IMSAI front panel, Altair and many others.

OEM and DEALER inquiries invited.

Memory Merchant™
14666 Doolittle Drive
San Leandro, CA 94577
(415) 483-1008

CIRCLE 98 ON READER SERVICE CARD

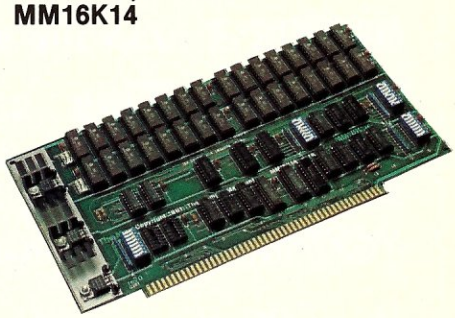
FULL TWO-YEAR WARRANTY.

The reliability of our boards, through quality-controlled production and proven performance, has enabled us to extend our warranty to a full two years. That's standard with us, not an option. This includes a 6-month exchange program for defective units.

Shipped direct from stock.

All Memory Merchant's boards are shipped direct from stock, normally within 48 hours of receipt of your order. Call us at (415) 483-1008 and we may be able to ship the same day.

16K RAM, Model MM16K14



16K STATIC RAM $169.

16K × 8 Bit
Bank Select & Extended Addressing
Four independently addressable 4K blocks
One 4K segment equipped with 1K windows
Uses field-proven 2114 (1K × 4) RAMS
Low Power (less than 1.2 Amps)
Runs on any S-100 8080, 4 MHz Z-80 or 5 MHz 8085 system.

Prices, terms, specifications subject to change without notice.

*Cromix and CDOS are trademarks of CROMEMCO.
[1] MP/M is a trademark of Digital Research

3. from a command file called for in the command line.

The command line for the sample in Figure 1 is:

```
A>PFONT DEMO.FF +F0 ROMN12 SCRP12 :
        ROMNB12 SANSI12 +LW 5.5
```

This is translated as:

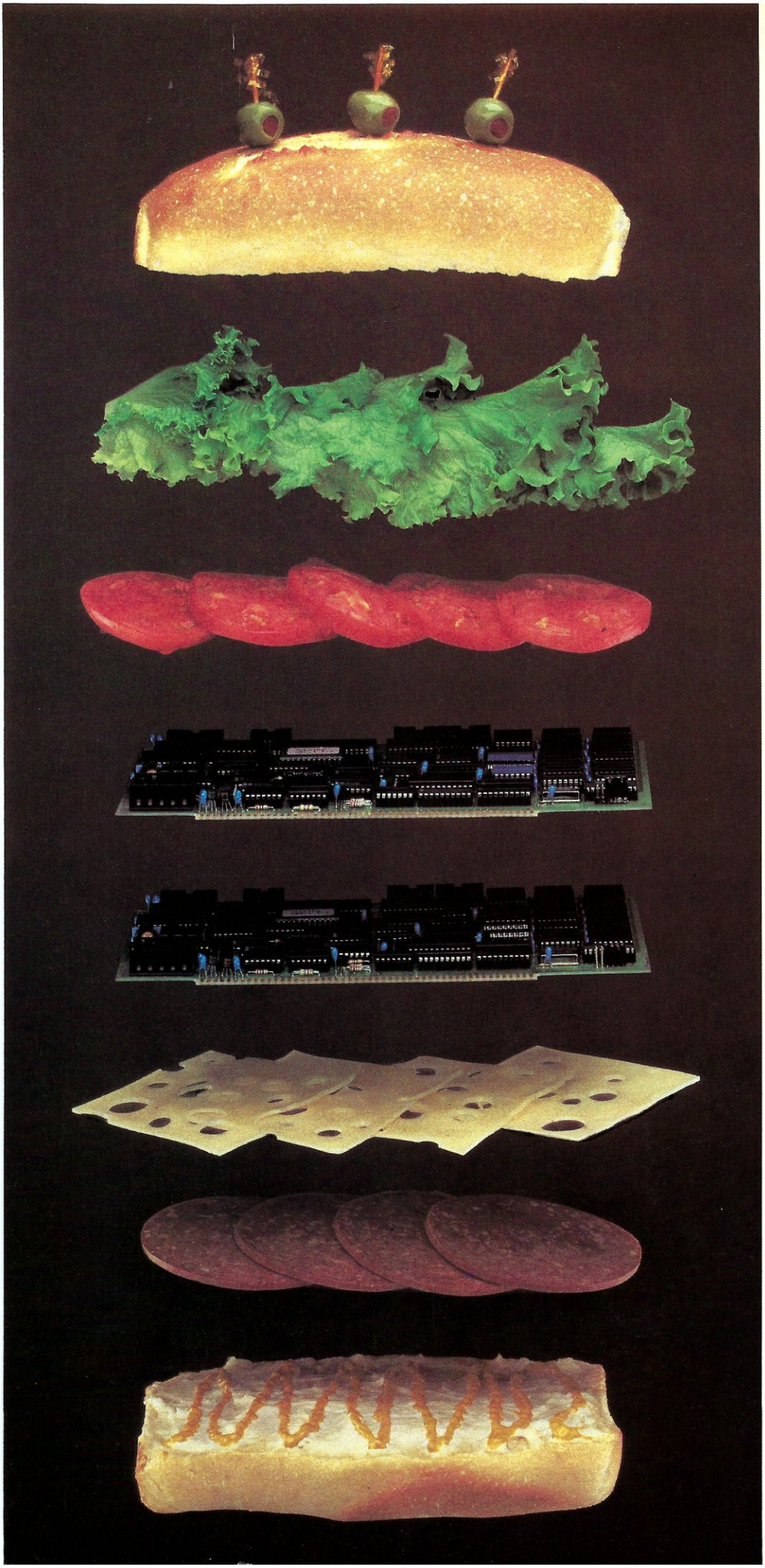
PFONT: Use the file DEMO.FF on the default drive. At all points in the file that say '/f0' use Roman 12 point, for '/f1' use Script 12 point, for '/f2' use Roman Bold 12 point and for '/f3' use Sans Serif Italic 12 point. Also, make the line width 5.5 inches. Within the text the '/j' directs Fancy Font to right justify starting at this line and continuing until the '/k' turns off the justification for the line in which it is located and thereafter. To turn off justification, temporarily, just for the one line, use /b.

These directives can take either upper or lower case.

Editing a font

Fancy Font stores each font in a file that contains

Figure 1. *Samples of fonts provided.*



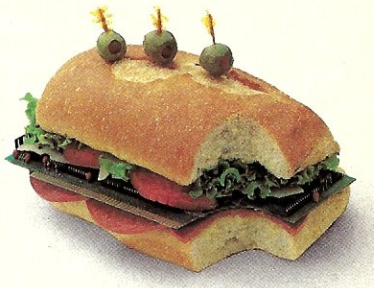
Teletek's New Combo Could Make You A Hero!

The SBC-II could be just the right ingredient for your latest concoction. The newest member of Teletek's family of multi-user, multi-processing S-100 products, the SBC-II essentially combines, or "sandwiches" two Teletek SBC-I's into one board. The SBC-II provides the capability to support two users from one standard size IEEE-696/S-100 slave board.

The SBC-II maintains full performance for each user with an independent CPU (Z80A or Z80B), 64K RAM, Serial I/O, and FIFO communications port to the system master. The system integrator benefits by getting complete support for two users for the price of one board.

TurboDOS and MDZ operating systems will support combinations of SBC-I's and SBC-II's offering system design efficiency and flexibility never before possible.

If you're hungry for value and efficiency, order an SBC-II from Teletek. You'll love every byte.



TELETEK

4600 Pell Drive
Sacramento, CA 95838
(916) 361-1777
Telex 4991834 TELETEK
Dealer inquiries invited.

CIRCLE 174 ON READER SERVICE CARD

It took hundreds of years to build the world's most advanced spreadsheet.

Way back in 1534, a Venetian scholar wrote a text that standardized the techniques of modern bookkeeping. And up till 1979, there weren't a lot of real improvements in the way people did spreadsheets. You still had to enter all your figures by hand. Run all the totals yourself. And when you had to make a few changes, you had to run the totals all over again.

Then along came a program that let you create spreadsheets on a personal computer. Without touching a pencil. Or adding anything by hand. Best of all, this new program gave you the freedom to look at your balance sheet a dozen different ways if you felt like it. Because all you had to do was change one figure, and the computer would refigure everything else for you. Automatically.

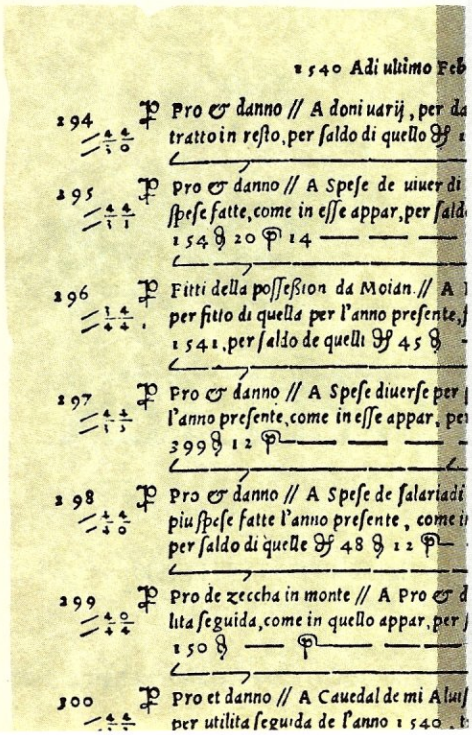
Like most breakthrough ideas though, this original spreadsheet program had a few things that definitely needed improvement. For instance, all the columns had to be the same width. And the entire system took about 20 hours to learn.

So we developed SuperCalc,® which was such a giant step in the right direction that it became a best-seller. Over 250,000 copies in only 18 months.

Today, we've gone even farther. After talking with thousands of customers, we've found a lot of important new features you'd like to have. And we've built them into the world's most advanced spreadsheet.

SuperCalc².

1534



1979

STATEMENT

	JAN	FEB	MAR
SALES	1000	1100	121
COST	300	330	363
GROSS	700	770	847
R & D	160	176	194
MARKETING	200	224	251
ADMIN.	140	151	163
TOTAL	500	551	608
INCOME	200	219	239
TAXES	80	88	96
NET	120	131	143

1981

INCOME STATEMENT

	JAN
NET SALES	1000
COST OF GOODS SOLD	300
GROSS PROFIT	700
RESEARCH & DEVELOPMENT	160
MARKETING	200
ADMINISTRATIVE	140
TOTAL OPERATING EXPENSES	500
INCOME BEFORE TAXES	200
INCOME TAXES	80
NET INCOME	120

It'll take you 10 minutes to learn it.

10 minutes to SuperCalc².

Create your first spreadsheet model in 10 minutes flat.

With SuperCalc², you get all kinds of features that have never been possible before. Like custom formatting capabilities. Automatic consolidation. Sorting. And more. Take a look at the spreadsheet below, and you'll see just how much we're giving you.

Best of all, we made it just as easy to use as the original SuperCalc. There's a self-paced starter guide that shows you exactly what to do, step by step. And an AnswerKey™ that gives you instant help right on the screen. Because of these features, you can actually create your first spreadsheet in 10 minutes flat. Even if you've never used a computer before.

SuperCalc² is available for the IBM PC and XT. Where it can run in full color. With full memory. Even on hard disk. It's also available for CP/M, CP/M-86, MS DOS and other operating systems, so you can use it with most popular personal computers.

But with all its advanced features, SuperCalc² is the same price as the original SuperCalc. Only $295, suggested retail. We're also offering SuperCalc for only $195. And an upgrade to SuperCalc² whenever you're ready.

So. If you've always thought there had to be a better way to do your spreadsheets, stop by your closest computer store. And take a good look at SuperCalc² After just 10 minutes, you won't want to do one more spreadsheet the old way.

1983

Combines any number of spreadsheets —automatically!

Calendar/date capabilities.

Percentages.

User-set decimal places.

```
        COMPANY CONSOLIDATED MONTHLY PAYROLL

Today's Date                        4/15/83        Deduction Percentages
Payroll Start Date                  4/ 1/83        Fica              6.700%
Days this period                        15         SDI                .8%
Recalculate YTD Y/N ?  N

                                      Gross        Total          Net          YTD
 Emp# Employee name      Status      Salary       Deduct          Pay        Gross
==========================================================================================
  34   Adams              M       $1,100.00  ( $82.50)   $1,017.50    $6,200.00
  49   Beckette           S         $750.00  ( $56.25)     $693.75    $5,250.00
  84   Johnson            S         $850.00  ( $63.75)     $786.25    $5,950.00
  92   Jones              M         $900.00  ( $67.50)     $832.50    $6,300.00
  12   Samson             S         $560.00  ( $42.00)     $518.00    $3,920.00
  19   Santos             M         $650.00  ( $48.75)     $601.25    $4,550.00
  45   Smith              S         $700.00  ( $52.50)     $647.50    $4,900.00
==========================================================================================
Total # employees          7

Total Gross Salaries (100's)        $55.1
Total Deductions (100's)            $-4.1
Total Net Pay (100's)               $51.0
```

Floating $ signs.

Embedded commas.

Negative numbers in ( ).

Arrange reports numerically or alphabetically, like this

Numbers in 100s or 1000s.

SuperCalc²™

SORCIM®

2310 Lundy Avenue
San Jose, CA 95131
(408) 942-1727

directives to the Epson printer. Within EFONT, the editing program, lies the means to create an ASCII file for each character in an existing font. This ASCII file consists of an asterisk for each dot on the Epson dot matrix head. This file can be loaded into any text editor or word processor and modified into any shape desired. Of course, the idea is not to take a Roman type font and turn it into script. But it may be desirable to take this Roman type font and make it **bold** or take a Sans Serif font and create a condensed version.

Moreover, anyone familiar with the printing industry is aware of the hundreds of specialized type styles available. Using a standard typeface, it is quite easy to modify it into any of these standard fonts. Also, it is possible to create a small typeface and set it within the space allowed for a larger size—for example, 6-point in a 12-point space. The smaller typeface can then be moved to the top of the space provided where, when used, it becomes a superscript. As you can see, the system is very flexible.

Creating a font

Creating a new font should be as simple as looking at a font within the Hershey file supplied. This file is a set of characters developed for the National Bureau of Standards. The Hershey set is used to create your own fonts by assigning any of the characters within it to a set of ASCII values. For a normal alphabet creation, this would mean assigning the letter 'A' to 65, 'B' to 66, etc. During the creation process, the standard characters may be enlarged or modified in any way desired.

There can be more interesting things available, however. The Hershey file contains a Russian, Greek, and German alphabet, as well as many special symbols for music, playing cards, zodiac, math, bells, trees, triangles, stars, and many, many more. The new font procedure would be to assign an ASCII value to any of these characters; at least, it should be that easy but, unfortunately, until you learn the procedure, it isn't so simple.

That is the one fault I could find in the documentation. There is no tutorial on creating and editing a font, and the process, while not difficult once learned, is difficult to figure out from the procedure as outlined in the manual.

When CFONT is called up, several options are offered. To create a new font, each character in the ASCII set must be (M)apped to a corresponding character in the Hershey set. This mapping is continued until all the characters wanted are mapped. I thought at first that each character had to be generated as soon as it was mapped, before going on to the next one; this would require a very long creation process. However, mapping is best done as a complete operation. Once all the characters are mapped, the size factors should be selected for modifying the shape. Then, and only then, should characters be (G)enerated. They are then saved to an output file. I suggest that this output file not be named by the final name you will use for the font.

Now invoke EFONT. With EFONT, the left and right margins are set. This determines the amount of space each character will take up on the page, and can be uniform or changed for each character. All characters are supplied with a margin the size of one dot on each side. Therefore, the 'I' will take up less space than the 'M'. This can be modified during a font design, and time should be taken to do it right. After all, this font will be used many times, and if it is worth creating at all, it should be fine-tuned as closely as possible. Mapping, margins, and resulting output are shown in Figure 2.

I enjoy bridge, so I designed a four-character font called CARDS. This font assigns the symbol for the club suit to the letter 'C', the symbol for diamonds 'D', the symbol for hearts 'H', and the symbol for spades to 'S'. After creating this card font with the Hershey set, I invoked EFONT, the Fancy Font edit module. Using the (L)oad command, I brought in the created font and, with the (E)dit command, saved each character in its own ASCII file. The (P)rint function with EFONT printed each character, its parameters, and five examples of the character as it would appear in use.

Since I prefer to have the card symbols filled in, I edited each character's ASCII file with WordStar to fill in the open spaces with asterisks. This produced the final file as shown in Figure 3.

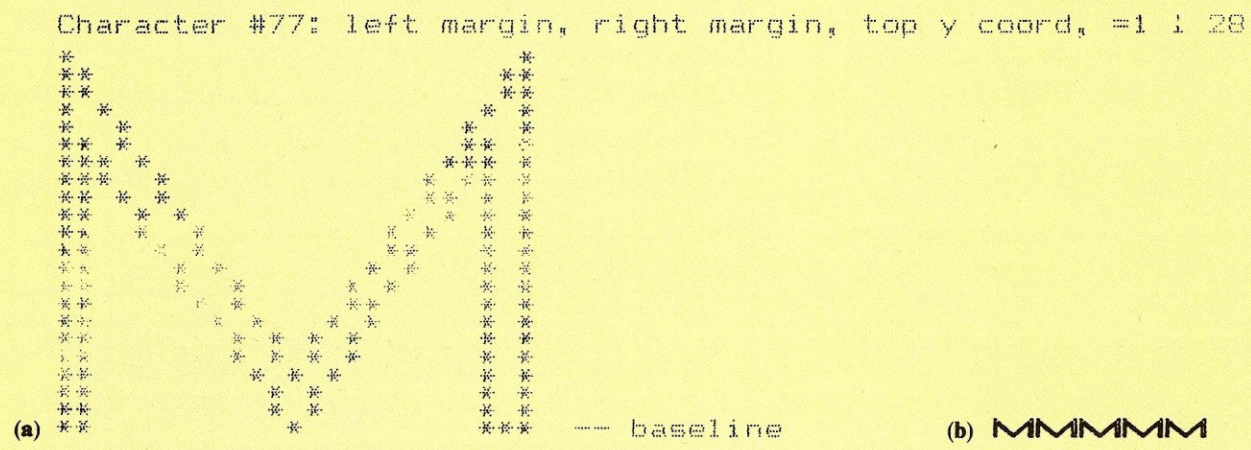
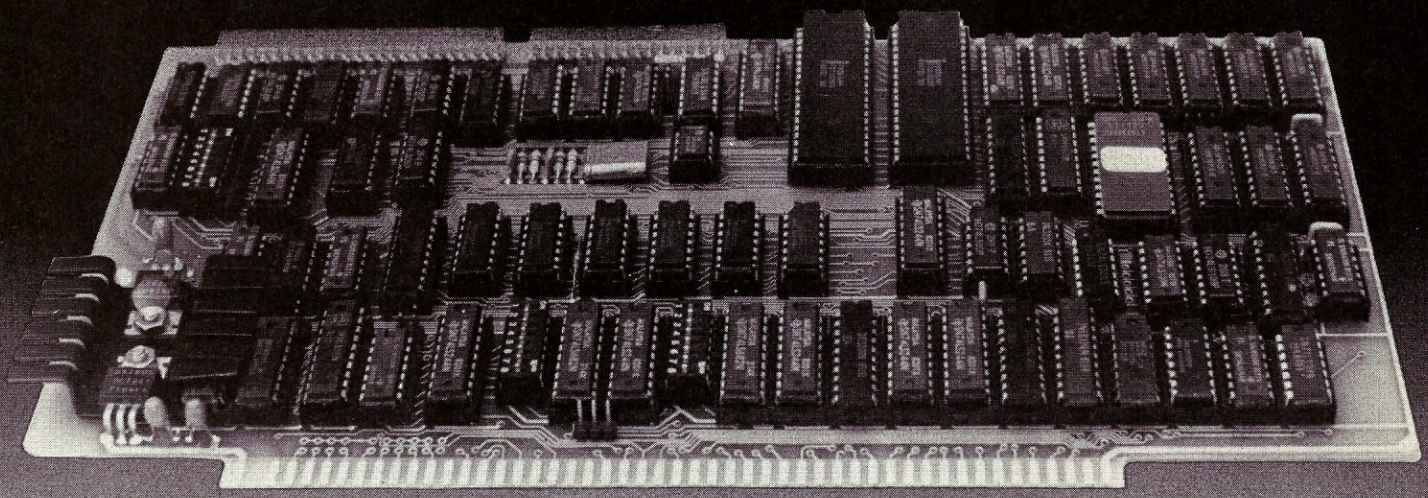


Figure 2. *(a) Mapping of a character: (b) resulting printed output.*

Does Yours Compare with OMNIDISK?



Introducing tomorrow's disk controller . . . OMNIDISK offers S-100 users a unique combination of compatibility and technological innovation that together produce features not found in any conventional disk controllers. See for yourself what tomorrow looks like:

- ✔ **Simultaneous support of both 5¼" and 8" floppy disks allows software transfer between disks.**
- ✔ **Our custom on board firm wear permits addressing of memory beyond 64K as RAMDISK.**
- ✔ **24 bit DMA allows CPU by-pass.**
- ✔ **Power-on boot PROM gets you running fast.**
- ✔ **On-board de-blocking conserves valuable RAM space above bios.**
- ✔ **Interfaces with the WD 1001® hard disk controller. No need to buy a host adapter.**
- ✔ **Full 16 bit port addressing.**
- ✔ **Full track buffer allows controller to recall entire track, not just sectors. Results in a speed increase 3-to-7-times greater than conventional controllers for both read and write operations.**

OMNIDISK'S features reflect our committment to designing S-100 products with an eye on the future. OMNIDISK'S price reflects our committment to offer products with an eye on the needs of today's user. You can begin using tomorrow's disk controller today for only . . . **$399***

So why wait, order now.

***CP/M* 2.2 configured for OMNIDISK, only $25 with purchase.**

FULCRUM™ COMPUTER PRODUCTS

distributed by:
W.W. COMPONENT SUPPLY INC.
1771 Junction Avenue
San Jose, CA 95112
(408) 295-7171

FREE U.P.S. ground shipping on pre-paid orders. Shipping will be added to C.O.D., VISA and M/C orders. CA residents please add sales tax.
*CP/M is a registered trademark of Digital Research.

Call today for information regarding our new 8 MHz CPU.

CIRCLE 58 ON READER SERVICE CARD

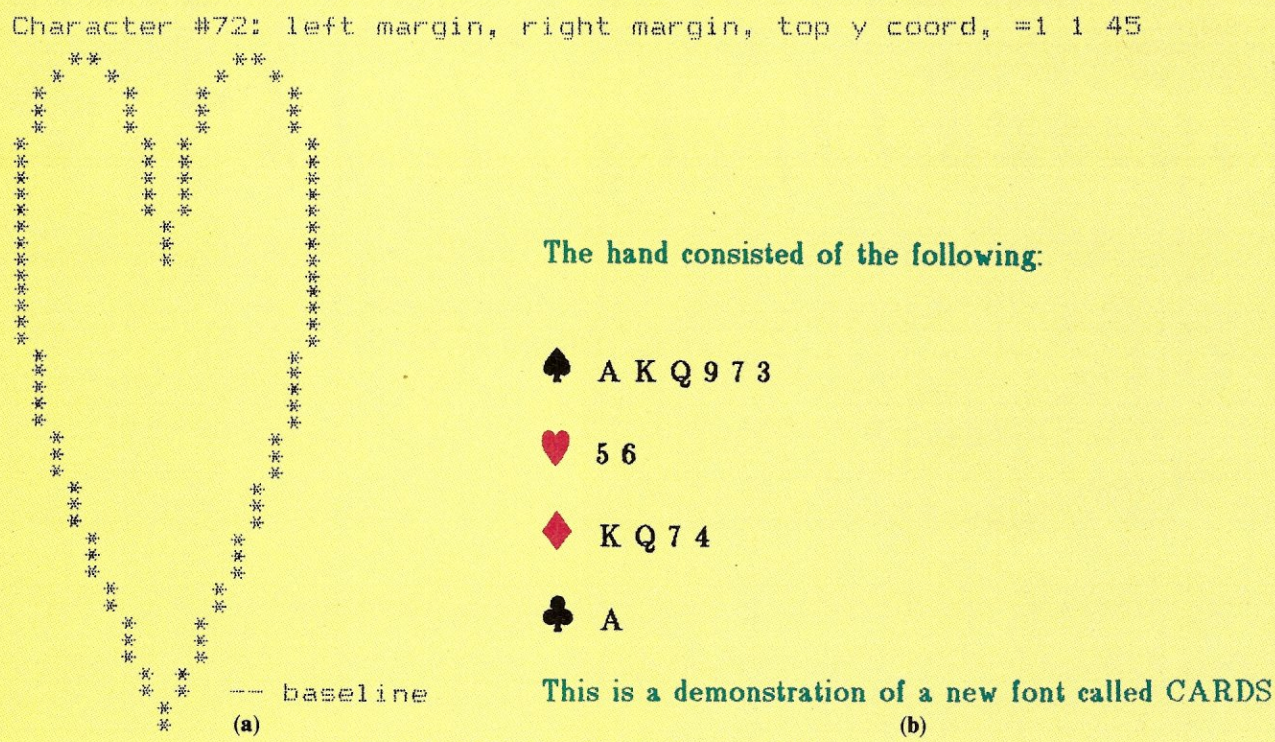


Figure 3. *(a) Mapping of Hearts character; (b) demonstration of output of complete CARDS font.*

The original text file for this demonstration is:

```
The hand consists of the following:
\f1S\f0  A K Q 9 7 3
\f1H\f0  5 6
\f1D\f0  K Q 7 4
\f1C\f0  A
This is a demonstration of a new font called CARDS.
```

The print produced in Figure 3a was produced by invoking Fancy Font in the command mode:

```
A>PFONT CARDS.FF +F0 ROMN12 CARDS
```

The CARDS font consists of four characters in which the club symbol is ASCII 67, 'C'; the diamond is ASCII 68, 'D'; the heart is ASCII 72, 'H'; and the spade is ASCII 83, 'S'. Whenever PFONT reads an 'S' in font 1, it produces the symbol for the spade suit. It would be possible to modify a boldface letter in a similar manner, for example, by providing internal space to produce a standard type font called US.

New release suggestions

I believe the manual should include directions for using a file which would be supplied by Softcraft. This file would have several lines of measured text. The user could print these lines using the various fonts he wanted. These printed lines would demonstrate the amount of space taken by a measured number of characters, and would allow formatting the original text input to produce the line width desired with a minimum of errors.

I would also like to see some means of modifying the line width required from within the text. At present, including a narrower line width in the text causes the spreadout of words on the line, while the justification continues to work at the old line width.

Conclusion

If you want to produce fast letters and text, you should not use Fancy Font. But for really unusual presentations, or where the product will be reproduced, it is less expensive to use this process than send your material to a typesetter. Although the typeset material would be superior, that superiority comes at a price, and there are times when the cost is not justified. The Fancy Font manual was printed entirely from copy prepared on an Epsom using Fancy Font, so the idea certainly is not impractical.

The evaluation table below should have one more category: evaluation of the results. I did not include this item because I feel the results should be evaluated on two levels: compared to typeset material and compared to dot matrix or daisy wheel output. When compared to typeset copy, Fancy Font is not as good, but, unless you want to settle for the size and fonts available with the daisy wheel or the roughness associated with dot matrix, Fancy Font is well worth the price. I think it deserves a place among the fine products available for microcomputers such as Wordstar, dBase II, CBasic and Microsoft Basic. ■

Fancy Font is more than a system that uses the bit graphics of the Epson printers—it is also a text formatter.

Fancy Font continued . . .

Overview

Available from:

SoftCraft
8726 S. Sepulveda Blvd., Suite 1641
Los Angeles, CA 90045

Price: $180.00

Currently available only for use on the Epson MX, FX and RX series of printers either under V/M or for the IBM PC under (MSDOS).

Characters allowed per font:	128
Max. characters per line input:	depends upon font and character
Max. size of a character:	approx. 1″ x 1″

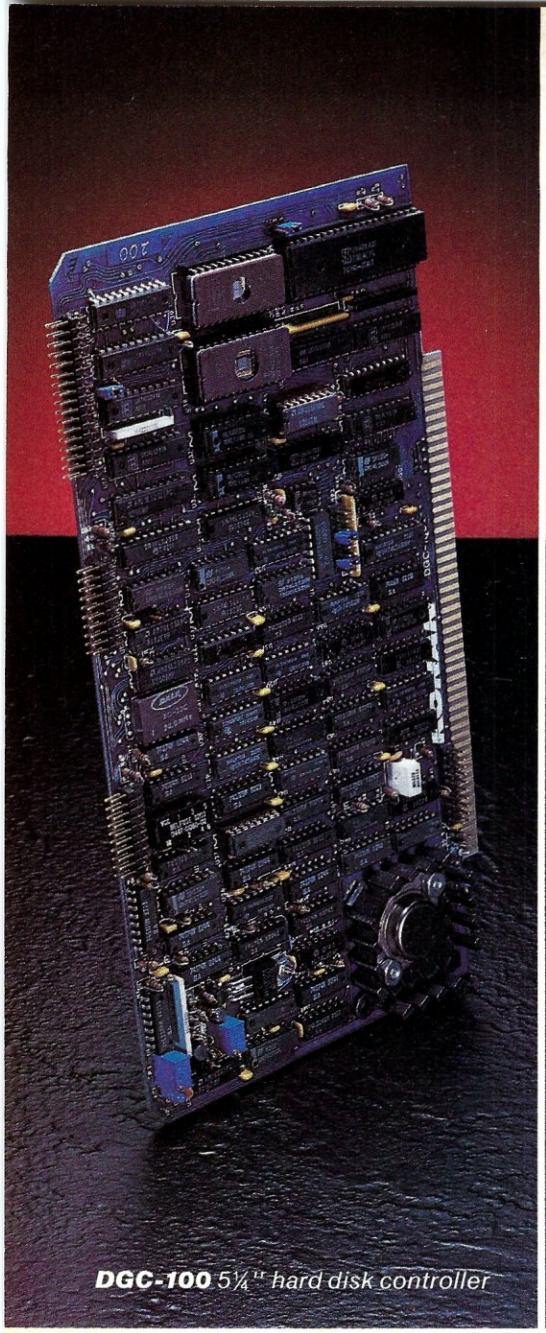
Evaluation on a scale of 1 to 10:

Documentation:

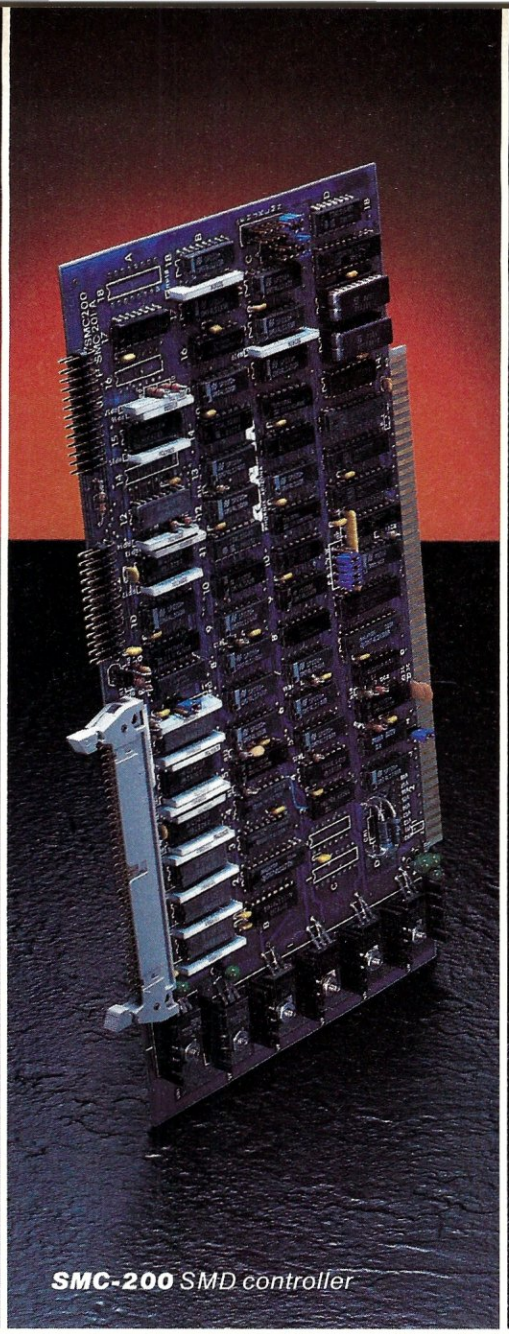
Organization for learning	2
Organization for reference	8
Readability	7
Includes all needed information	7

Ease of use:

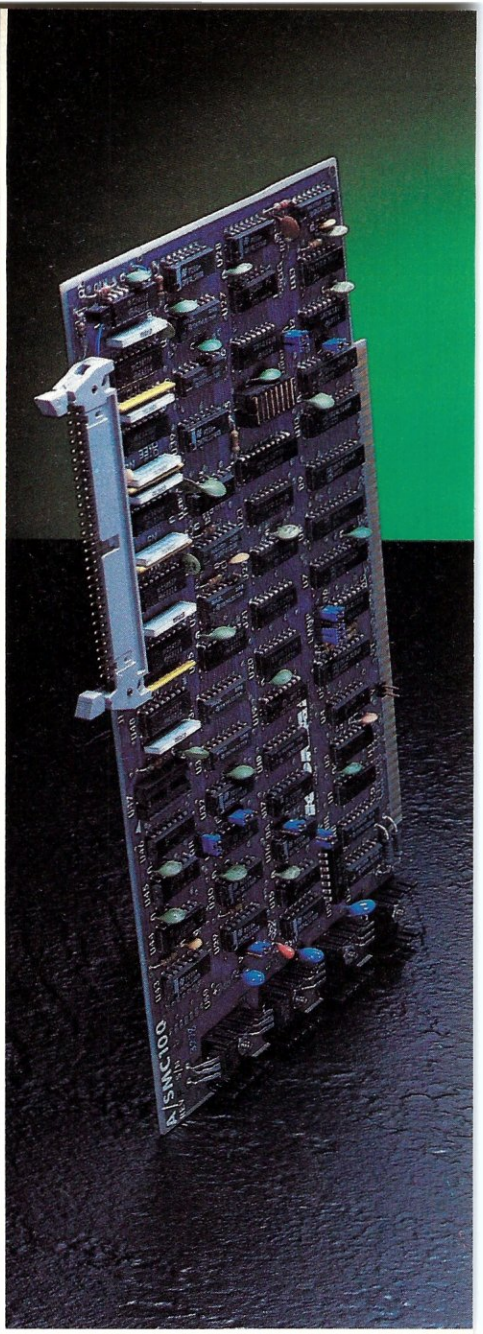
Initial start up	9
Setting up your own application	5



DGC-100 5¼" hard disk controller



SMC-200 SMD controller



A SPECTRUM OF THAT PUT YOU

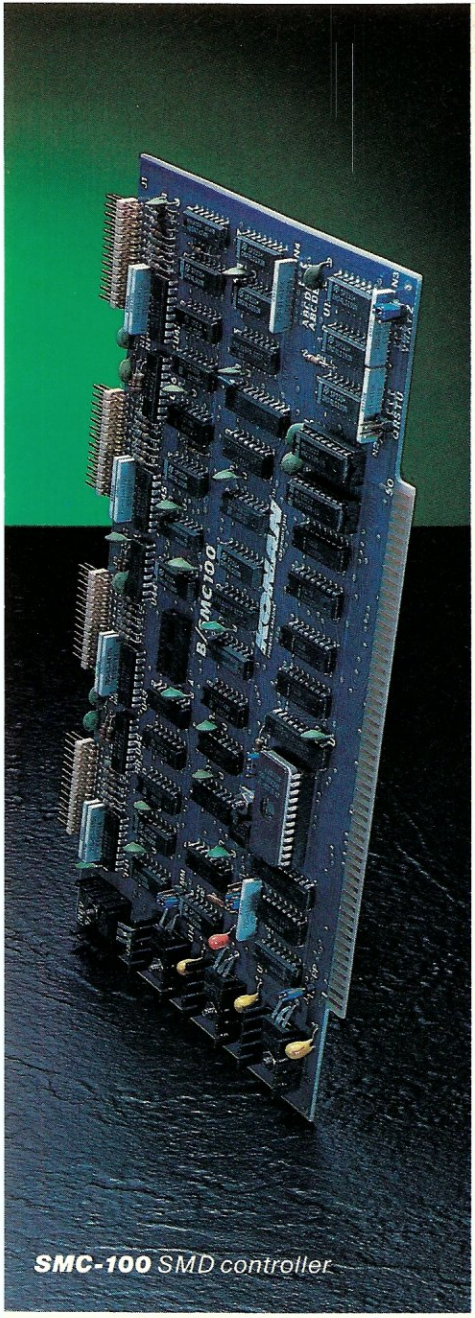
Konan's board array of feature-packed, versatile controllers makes your design decision simple and cost-effective. The Konan S-100 controller line provides you with the assurance that you have designed-in the high performance and quality you demand. And highly qualified, technical support and prompt deliveries go a long way to ensure you'll meet your production schedule... on time! The spectrum includes:

The DGC-100 *A high performance 5¼" hard disk controller...with full sector buffering, error correction, automatic flaw mapping...all standard features. And the versatile DGC-100 supports the known spectrum of Seagate ST-506 compatible drives.*

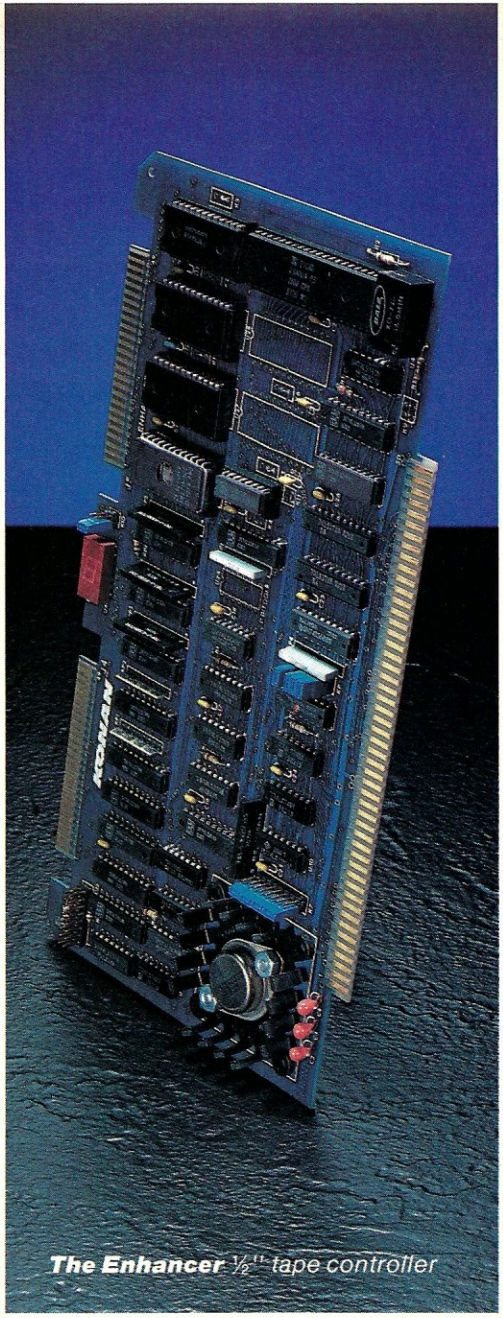
The SMC-200 *A versatile disk controller that interfaces S-100 bus microcomputers with all fixed or removable media disk drives with SMD interfaces. And each SMC-200 will control up to 2 drives and up to 1200 Megabytes...using only one bus slot.*

The SMC-100 *A cost-efficient, versatile controller that interfaces S-100 bus microcomputers with all fixed or removable media disk drives with SMD interfaces. Each SMC-100 will control up to 4 drives and up to 2400 Megabytes...using only 2 bus slots.*

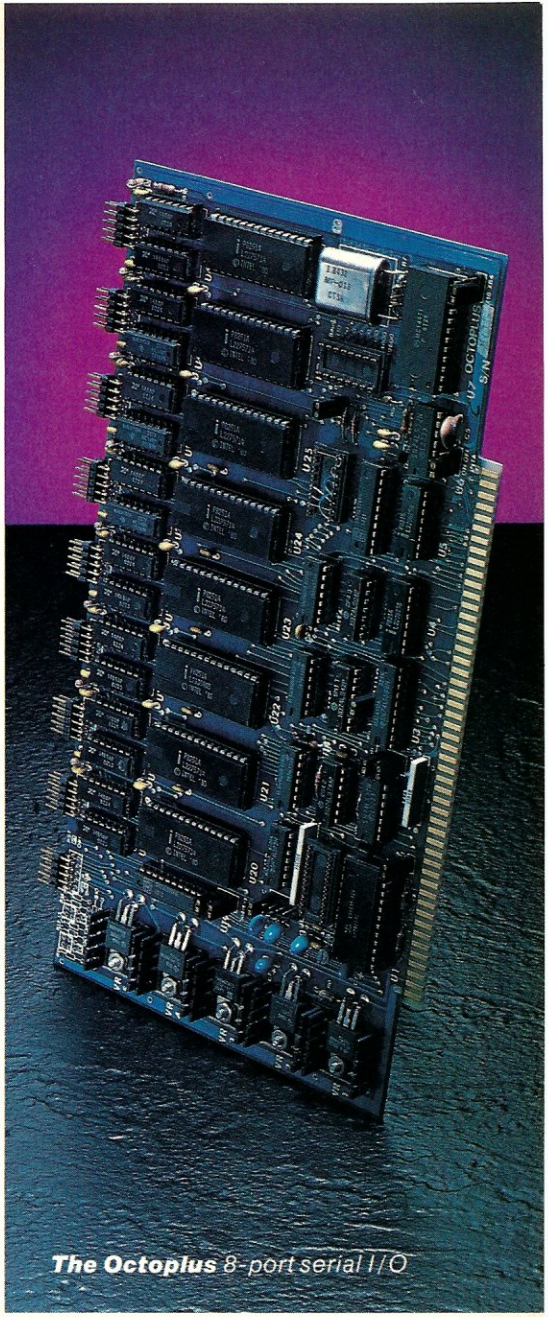
The Enhancer *An intelligent, 9-track IBM compatible, ½" tape and high speed printer and reel-to-reel*



SMC-100 *SMD controller*



The Enhancer *½" tape controller*



The Octoplus *8-port serial I/O*

S-100 SOLUTIONS IN CONTROL.

tape controller. The Enhancer interfaces to both Pertec Micro Formatter and CDC 92144 compatible tape drives. And the Enhancer features an 8-bit parallel, high speed printer port.

The Octoplus *A versatile, low cost, quality solution for multi-user or data acquisition systems. The Octoplus features selectable serial data rates from 110 to 19.2 K Baud, selectable real-time interrupts from 3.125 ms to 100 ms, and a general purpose, 8-bit parallel input port.*

For more information on how Konan's spectrum of S-100 solutions can put you in control... call or write:

KONAN T.M.
in control

1448 North 27th Avenue/Phoenix, Arizona 85009
(602) 257-1355
TWX/TELEX 9109511552

DGC-100, SMC-200, SMC-100, Enhancer and Octoplus are tradenames of Konan Corporation.

CIRCLE 242 ON READER SERVICE CARD

MICROSYSTEMS REVIEWS THE

Computime SBC-880 Single Board Computer

Another in the *Microsystems* series of reviews on S-100 single-board computers

by Dave Hardy and Ken Jackson

Like most Z80-based "single-board computers," the Computime SBC-880 has a CPU, some on-board RAM and EPROM, and serial and parallel I/O. Unlike most SBC's, the SBC-880 does not have a built-in disk controller. It uses another board (Computime's UFDC-1 floppy disk controller) for disk I/O. In fact, the SBC-880 has only only 2K of RAM and EPROM, instead of the usual 64K of RAM and 2K or more of EPROM.

Instead of building all of the functions on one board, Computime has chosen to build its SBC on two. While this may seem like a step backward (and not at all like a *single*-board computer), it is actually a rather unique approach to solving some of the problems found in most S-100 computer systems.

A unique characteristic of the SBC-880 is the ability of its on-board RAM and EPROM to exist in the same memory space as external (S-100) memory. The SBC-880's on-board memory is given a higher priority than external memory, so that whenever the processor looks at memory that exists both on-board and externally, the on-board memory is the only one enabled. This allows the system builder to use virtually any memory boards available, without regard to what memory on the SBC might overlap external memory. This feature also allows the SBC-880 to work in a "dead" S-100 system, that is, in a system that may have a bad memory or I/O board, or even a bus problem.

The SCB-880 is meant to work in conjunction with Computime's UFDC-1 floppy disk controller and a 64K RAM board. Since the Computime "single-board computer" is actually both the SBC-880 and the UFDC-1, this review includes both boards.

Common features

Both the SBC-880 and the UFDC-1 are high-quality glass-epoxy PC boards, solder masked and silk-screened. Both have gold-plated edge connectors and are fully socketed, which increases reliability and makes IC replacement fast and easy.

The SBC-880 single-board computer

The SBC-880 has the following features:

- Z80 CPU
- 8253 timer
- 8251 USART
- A parallel port (uses 74LS374 octal D flip-flops)
- 1K on-board 2114 RAM
- Up to 2K on-board EPROM (2704, 2708, 2716, 2516)
- IEEE-696 compatible (8-bit data, 16-bit address)
- All memory and I/O can be re-addressed
- Allows external DMA (but none to on-board RAM)
- Selectable phantom/power-on jump circuitry
- Selectable MWRT signal

The SBC-880 is based on a Z80 processor, and can be strapped to run at either 2 or 4 MHz.

The SBC-880 uses one third of its 8253 timer for baud rate generation. The remaining two timers are available to the user, and may be used under program control.

The on-board serial I/O is provided by an 8251 USART. To simplify handshaking, Computime chose to tie the 8251's Clear-to-Send line to ground. While this makes it easier to connect a simple RS-232 device to the USART, it also makes it impossible to perform any serial operations that require CTS/RTS handshaking (including BiSync, etc.). The 8251 can be run at any standard rate from 110 baud to 9600 baud. Unfortunately, 19,200 baud is not available.

The on-board parallel port is provided by a pair of 74LS374 octal D-type flip-flops, and has simple handshake capability (strobe and acknowledge).

Depending on which EPROM you want to use, you may have to add another voltage regulator. This is not necessary, however, to use the 2716 EPROM supplied.

As mentioned above, the SBC-880 has 1K of on-board 2114 RAM and up to 2K of EPROM that can safely overlay system (external S-100) RAM because of the on-board memory decoding. This feature makes it possible to run the SBC-880 with 64K RAM boards that do not have provision for "windows" or addresses where RAM isn't allowed. Also, because of the on-board decoding scheme used, the on-board EPROM and RAM cannot be accessed by any external devices (including Temporary Bus Masters). DIP switches are provided to address the RAM and I/O, but trace cutting may be required to re-address the EPROM.

In addition to its memory overlaying ability, the SBC-880 also will allow its EPROM to be "shadowed" (i.e., turned OFF). This comes in handy when you want to run in a 64K RAM environment. Unfortunately, the EPROM can only be "un-shadowed" by

Dave Hardy, 736 Notre Dame, Grosse Pointe, MI 48230

C compilers and Cross compilers

TELECON'S C COMPILERS OFFER YOU

- FULL C
- UNIX* Ver. 7 COMPATABILITY
- NO ROYALTIES ON GENERATED CODE
- GENERATED CODE IS REENTRANT
- C AND ASSEMBLY SOURCE MAY BE INTERMIXED
- UPGRADES & SUPPORT FOR 1 YEAR

IN THESE CONFIGURATIONS:

HOST	6809 TARGET	PDP-11*/LSI-11* TARGET	8080/(Z80) TARGET	8088/8086 TARGET
FLEX*/UNIFLEX* 6809	$200.00 WITHOUT FLOAT $350.00 WITH FLOAT	500.00	500.00	500.00
RT-11*/RSX-11* PDP-11*	500.00	200.00 WITHOUT FLOAT 350.00 WITH FLOAT	500.00	500.00
CP/M* 8080/(Z80)	500.00	500.00	200.00 WITHOUT FLOAT 350.00 WITH FLOAT	500.00
PCDOS*/MSDOS* 8088/8086	500.00	500.00	500.00	200.00 WITHOUT FLOAT 350.00 WITH FLOAT

Others Pending

C SOURCE AVAILABLE FOR $2,500^{00}$

SO . . . IF YOU'RE READY TO MOVE UP TO C ...

CALL

408-275-1659

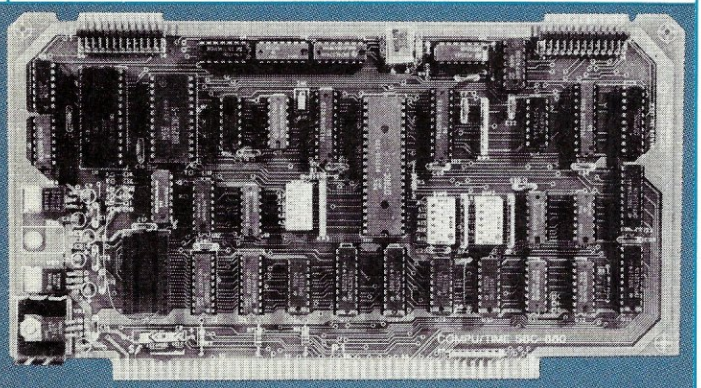
TELECON SYSTEMS
1155 Meridian Avenue, Suite 218
San Jose, CA 95125

a system reset (or POR).

Although unable to use the IEEE-696 16-bit data path and 24-bit (extended) addressing, the SBC-880 is otherwise IEEE-696 compatible. The SBC-880 should be able to work with any IEEE-696 compatible memory board and, in our tests, performed well even with some of the older (non-IEEE-696) dynamic RAM boards, including the SD Sales' 64K and Morrow's 16K boards.

Unlike many other SBC's, the SBC-880 will allow external DMA operations to take place; however, its memory decoding scheme will not allow DMA with any of its on-board memory.

Several other options are available to the user when configuring the SBC-880, including power-on jump with or without phantom, no power-on jump, no EPROM, and MWRT generation by CPU or externally (or both).



The Computime SBC-880 board.

The UFDC-1 floppy disk controller

The UFDC-1 has the following features:

- 1795 Floppy disk controller IC
- 9216 Floppy disk data separator
- Allows any combination of up to four 8″ and 5¼″ drives
- Strap-selectable 3-, 6-, 10-, or 15-ms step rate
- I/O port-mapped
- IEEE-696 compatible

The UFDC-1 is based on the Western Digital 1795 Floppy Disk Controller IC and uses the Standard Microsystems Corp. 9216 Data Separator for both read data separation and write precompensation. It supports both single- and double-density and single- and double-sided floppy disk drives. Disk transfers are performed by a "pseudo-DMA" technique using CPU wait states. Write precompensation (for tracks greater than 43) is fixed at 250 ns (early and late).

The most outstanding feature of this board is its ability to handle both 5¼″ and 8″ floppy disk drives simultaneously. Up to four drives (in any combination of densities, size, or number of sides) can be connected at any time to the UFDC-1 by simply selecting the proper strap settings on the board. In addition, the board will also handle the newer 77-TPI and 96-TPI 5¼″ floppies.

The UFDC-1 also allows the step rate to be individually selected for each of its four drives. By setting the appropriate jumper, the default step rate for each drive can be set at system boot time.

All controlling I/O is port-mapped. The ports used are hard-coded at addresses 98H to 9FH. Changing the port addressing of this board would require cutting traces and possibly adding an IC or two. Interestingly, the UFDC-1 contains three unused 16-pin DIP sockets labeled "SPARE," which might come in handy for this sort of thing.

Software provided

Software provided with these boards includes the source for the system monitor (which contains the BIOS) and source for the disk formatting program. The software is written in TDL/Xitan format, so it will have to be converted, unless you have a compatible assembler (like TDLASM, PASM, etc.).

The source files are well written and fairly well commented, and can be used to better your understanding of the operation of the BIOS and monitor.

The monitor program provided on a 2716 EPROM is very much like a debugger program. It boots automatically at 9600 baud and offers the standard memory examination and modification commands. It also has commands to read and write to a port, test and compare memory, and perform elementary disk operations like select drive, move head, read track, etc.

CP/M is supplied as a more or less standard 64K system. Although both the cold-boot and the BIOS are contained in EPROM, the BIOS jump table is kept in RAM, which means that there are generally no ill effects due to the ROM'ed BIOS. Of course, BIOS modifications would usually require burning a new EPROM.

The BIOS, as supplied, can read and write 128-, 256-, 512- and 1024-byte sectors, on 5¼″ or 8″ disks, single- or double-sided, etc. The BIOS is very versatile in this respect, due to the way that it determines a disk's characteristics. All in all, it can read at least a dozen of the different disk formats currently available.

In the Computime BIOS, instead of reading a disk and trying to dynamically determine its density, the BIOS reads a disk parameter table (that is incredibly similar to CP/M's Disk Parameter Block) on track 0, sector 1 of each disk before it "logs" it in. This makes it very easy to read and write a plethora of different disk formats and densities.

The bad part of this method is that it requires that a disk *absolutely* has to have a disk parameter table in track 0, sector 1. In other words, there is no way to plug in a standard, run-of-the-mill, single-density,

A unique characteristic of the SBC-880 is the ability of its on-board RAM and EPROM to exist in the same memory space as external (S-100) memory.

CP/M diskette and read it without either first writing a disk parameter table onto it at track 0, sector 1, or logging in a similar disk WITH a disk parameter table and then fooling CP/M by swapping disks. And if you use the "swap disks" method, you can't write to the disk because CP/M will mark the disk as Read Only. This is by far the worst flaw in the Computime BIOS. In other respects, the BIOS works very well.

Documentation

Documentation is provided in the form of two manuals, one for each board. The UFDC-1 manual also contains the CP/M documentation, including information on how to bring up your own CP/M, in case you bought only the BIOS (and not the CP/M) from Computime.

In general, both manuals are well written and complete, and include concise set-up instructions for their respective boards. Both manuals include complete schematics, and quick-reference pages are provided for DIP switch and jumper options.

Each manual includes a general description of its board, a functional description, a list of board options, and a detailed description that includes the theory of operation.

In addition, appendices are furnished that cover, in depth, the programming of the 8251 USART, the 8253 timer, and the 1795 FDC, board addressing, and parallel I/O. Also included are sections on using non-IEEE-696 dynamic memory and installing a printer, as well as how to set up specific disk drives to work with the UFDC-1.

The UFDC-1 manual also has a short trouble-shooting section that explains how to use the monitor's disk commands to diagnose system and/or disk problems.

Listings of the monitor and BIOS source are appended to each manual, in addition to being furnished on a floppy disk.

Bringing up the SBC-800 and UFDC-1

Installing these boards is fairly easy. Read the manuals, set up the jumpers on the SBC-880, set up the UFDC-1 for your particular types of drives, plug the terminal and floppy drive cables into their respective sockets, and power up. Of course, a RAM board (preferably 64K) is also required. Both the terminal and floppy drive cables are standard, and can be readily purchased at a local computer or parts store.

Once the power is on, insert the system disk and type "B" to boot, and you are up and running. The UFDC-1 manual gives a complete procedure for system start-up and also provides a list of things to check if the system won't boot. The start-up procedure also lets you generate a back-up disk, so that you can tuck away the master disk in a safe place.

If you don't buy your CP/M from Computime, they will provide you with a master disk that will install your CP/M on their machine. The procedure, as documented, is simple, and can be done with just a single drive; however, we were unable to successfully install our own copy of CP/M, and finally just used another system to patch the Computime jump table into our CP/M. After that, all was well.

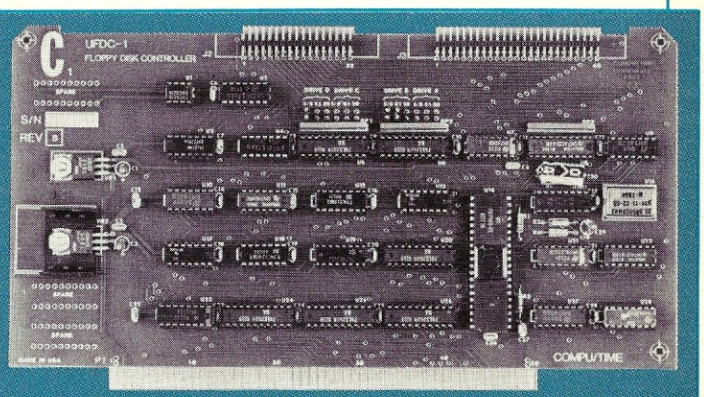
The monitor comes in very handy for checking various system functions when things don't work as planned. Using monitor commands, it is possible to check out all memory and I/O, including the floppy disk controller board and drives. Because of the peculiar strapping arrangements of some floppy drives, this type of monitor can save hours of trouble-shooting.

Using the boards with Computime's CP/M

In actual use, the Computime boards were dependable and trouble-free. Although only a lightweight heatsink is used for the 5-volt power supply on each board, we experienced none of the failures or mysterious "crashes" that usually indicate overheating components. In fact, convection cooling (i.e., no forced air) seemed to be adequate for both the SBC-880 and the UFDC-1, in spite of the fact that many of the RAM boards used produced a great deal of heat.

Computime's implementation of CP/M worked well and seemed reasonably fast (although no speed performance measurements were made). BDOS errors are trapped in the BIOS and give the user the option of aborting, retrying, or returning to CP/M (which is the normal procedure). The ROM'ed BIOS could conceivably cause trouble for some programs that try modify it, but the jump table is kept in RAM, and we had no problems running various programs that like to modify the BIOS jump table.

The only real problem with the BIOS (as was mentioned previously) is that it requires disk parameter tables on track 0, sector 1 of every disk. This can be a real pain, especially if you want to do something like copy 50 standard single-density CPMUG disks, or



The Computime UFDC-1 floppy disk controller.

The most outstanding feature of this board is its ability to handle both 5¼" and 8" floppy disk drives simultaneously.

Computime SBC-880 continued . . .

read any disks not generated with the SBC-880 and UFDC-1. This problem could be cured by making a few simple changes to the BIOS, but unfortunately, the BIOS is in EPROM, so changing it might require burning a new EPROM.

Conclusion

Although actually built on three boards instead of one (CPU+I/O, FDC, RAM), the Computime SBC-880 and FDC-1 performed well, and offered the additional advantage of being able to run both 5¼" and 8" drives simultaneously. The "overlayed" on-board RAM and EPROM make troubleshooting easy, even in a "dead" S-100 machine. Except for the inconvenience of not being able to directly read and write disks written on other machines, the software provided is very good.

The SBC-880 and UFDC-1 boards used in this review were provided by the east coast distributor of Computime boards, GSR Computers Maspeth, NY, and are available from them for $325 each (assembled and tested).

Our tests were performed using only 8" single- and double-sided drives, and 5¼" single-sided drives.

Prices: SBC-880: $375 A&T, $245 kit, $60 bare board; FDC UFDC-1: $275 A&T, $245 kit, $45 bare board. For further information contact:

Computime
8614 Hamilton Ave.
Huntington Beach, CA 92646
(714) 536-5000

"After 12 years work on mainframes and micros, I must say this machine is the finest piece of end-user engineering I have seen."

"We waited several years for a system integrating the 68000 with UNIX.™ Several tempting alternatives appeared along the way, particularly the Z-8000, the 8086 and ersatz UNIX™ Operating Systems.

The quality of construction, S-100 bus architecture and ease of service were the factors that led us to prefer your system.

It was worth the wait.

The part I enjoy the most about the Dual System 83 is demonstrating it to academics who are used to comparable DEC systems, costing three times as much."

Dr. Leland Wilkinson
University of Illinois, Chicago
October 20, 1982

DUAL

DUAL SYSTEMS CORPORATION
2530 San Pablo Avenue, Berkeley, CA 94702 (415) 549-3854 172029 SPX

SYSTEM 83/20 MC68000, full UNIX,™ 20-slot motherboard, IEEE 696/S-100, 10 MHz, from 516 KB to 3.25 MB of parity RAM, 20 MB storage, 4 to 6 users.

SYSTEM 83/80 All system 83/20 features except 80 MB to 168 MB storage, expandable to 1200 MB, 8 users. Includes DUAL's proprietary SMD controller producing superior speed over other UNIX™ systems.

CIRCLE 27 ON READER SERVICE CARD

UNIX is a trademark of Bell Laboratories, Inc.

ARE YOU DROWNING IN A SEA OF SOLUTIONS?

Recognize the symptoms?

Applications backlog? Technology blur? Mounting user pressures?

You suspect that suppliers may have answers to your pressing problems, but you don't have enough time to sort through fragmented information to determine what will really work for you.

Like the shoemaker's kid who goes barefoot, you (of all people) need a system. An information system that displays all your current software, hardware and communications options for instant reference.

That system is DATA SOURCES, and you're invited to make it work for you. Simply fill in the coupon and we'll send you DATA SOURCES at 50% off its regular cover price.

What DATA SOURCES does is organize your options. It guarantees that you've considered all relevant products. And makes the most of your time for product evaluations by guiding you to those with the best possibility of success.

DATA SOURCES is a conveniently sized, extremely well-organized and skillfully indexed reference...maintained on a database, constantly updated and published quarterly.

READ WHAT USERS SAY...

"...Current and comprehensive listings are becoming indispensable."

Gary Yost
Marketing Services Director
ASK Computer Systems, Inc.

"Literally use it daily...dramatically shortens research time...provides more alternatives than would have been known or considered."

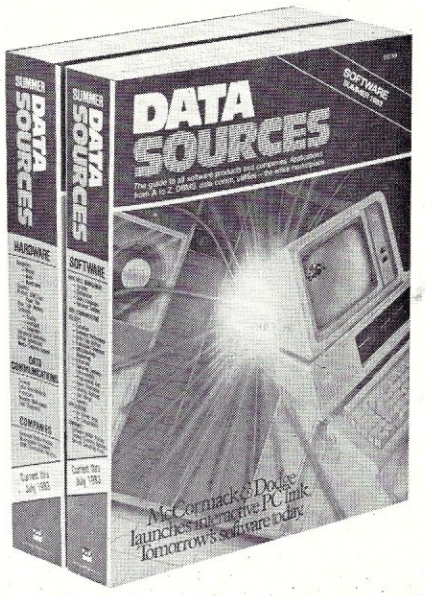
Curt Flatness, V.P.
Logic, Inc.

"Answer to a Systems Integrator's prayer...very effective for locating the equipment we need ...using it for everything from single-user systems to systems interfacing with mainframes."

Richard Fletcher, President
Micro Computer Adaptation Procedure

"Dependable source of information otherwise unobtainable or costly to obtain..."

Barry Kukes, President
RSI Repair Service, Inc.



NEW FEATURES...

- ✓ Quick-reference product comparison charts.
- ✓ Market-tracking trends of new DP products.
- ✓ Before-you-buy selection criteria and checklist.
- ✓ Geographical listing of vendors offering on-site and depot maintenance.

YOU GET...

Hardware—Over 12,000 products from Micros to Mainframes including peripherals—terminals, printers, memory devices, etc...organized by systems compatibility.

Software—The largest available inventory of business packages for micros, minis, and mainframes including operating systems, utilities, and application packages.

Communications Equipment—from modems and multiplexors to local networks, carriers and services. Quick reference comparison charts, plus all diagnostic and test equipment.

Services, Suppliers, Support—Most complete organized listing of suppliers from installation design to maintenance and data center operations.

ORDER NOW AND SAVE 50%

Yes! Rush me the latest two-volume edition of DATA SOURCES and enter my subscription at $60.00 for 4 quarterly editions. I save a full 50% off the regular $120.00 cover price.

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone _____

Nature of Company's Business _____

Please check appropriate box: ☐ END USER ☐ VAR/OEM ☐ DISTRIBUTOR
☐ DP MFG ☐ OTHER _____

An information product of Ziff-Davis Publishing Company. Satisfaction guaranteed.

Return coupon to:

DATA SOURCES

P.O. Box 5845. Cherry Hill, N.J. 08034.

Residents of Ca., Co., Ct., D.C., Fl., Il., Ma., Mi., Mo., N.J., N.Y., Vt., please add applicable state taxes.

T034

CIRCLE 21 ON READER SERVICE CARD

Logging-On CP/M

by Ralph J. Jannelli

This simple program allows a degree of security to multiuser CP/M systems. It operates in a manner that simulates log-on/log-off procedures found in minicomputer operating systems, and is intended to be used in conjunction with the CP/M enhancements described in my article "Enchancing CP/M 2.2" (*Microsystems*, March/April 1982).

On a cold start, CP/M signs on in a normal manner (i.e., disk A: user 0). This user directory (directory 0) should be designated to the system operator. All common user files reside in this directory. When the system operator is finished with any required system start-up procedures, he simply types 'BYE' followed by a carriage return. Once the BYE program is loaded, it searches user 0 directory (system directory) of the currently logged disk for the file PASSWORD.DAT.

If the password file is not found, an error message is sent to the console and control is returned to CP/M. If the password file is found, it is loaded into memory so the user name/password pair entered by the user can be matched against the contents of the PASSWORD.DAT file. Now any other system users, including the system operator, must log on.

Ralph J. Jannelli, 101 Cottonwood Dr., Jamestown, NC 27282

The BYE program sends the following prompt to the console when it is waiting for a user to log on:

```
"ENTER ANY KEY TO BEGIN LOGON
PROCEDURE"
```

When a user comes up to the console, he must enter a key from the keyboard. The program now knows someone is ready to log on. The next step is to enter the user name previously assigned to the user by the system operator. The program prompts with:

```
"ENTER USER NAME :"
```

The user then enters his user name. This can be any letter or number keys or a combination of letter and number keys. The maximum length of the user name is 10 characters. If a typing error occurs while entering the user name, it can be corrected by using the normal CP/M command line edit keys (i.e., backspace, rubout, control U, etc.). The user name entry is terminated by a carriage return. The program then asks for the password:

```
"ENTER PASSWORD :"
```

The password is a four-character field. Typographical errors are not allowed in entering the password. As each character is entered, a question mark is

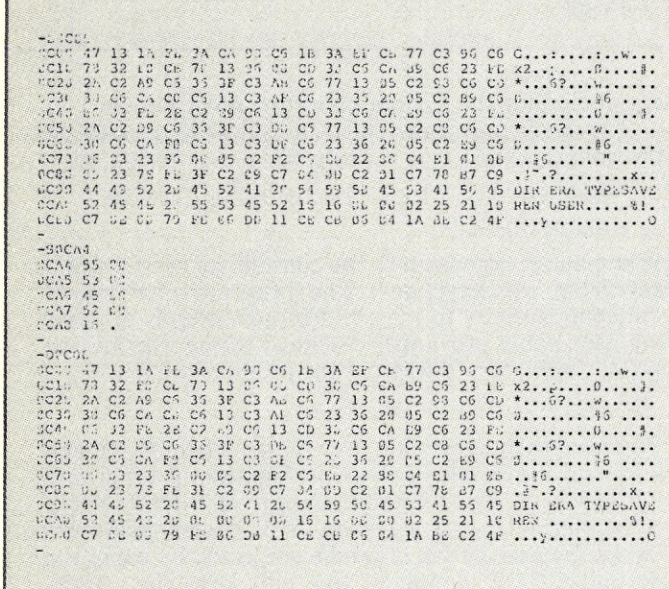


Figure 1

echoed to the terminal. This prevents any on-lookers from gaining access to the user's password. If the user name/password pair does not match any of the entries loaded from the PASSWORD.DAT file, this message is displayed on the console:

```
"INVALID USER NAME - PASSWORD
PAIR"
```

After three consecutive invalid log-on attempts, the following message is displayed:

```
"TOO MANY TRIES--CONSOLE LOCKING
UP FOR 30 SECONDS"
```

At this point the program enters a software loop and times out approximately 30 seconds. The loop count is set up to run on a 2.5 MHz Z80 processor. The loop can be modified to accommodate for faster or slower clock rates as shown in the program listing. The purpose of the console lockout is to frustrate attempts by unauthorized users to break the system security through trial and error. A successful log-on results in control being returned to CP/M, with the user number corresponding to the positional location of the user name/password pair in the PASSWORD.DAT file (i.e., if the fourth entry in the file is RALPH—1234, then the user number will be 4). See Figure 2 for examples.

When the user is done, he types BYE followed by a carriage return, and the process is repeated for the next user.

As added security, your copy of CP/M should be further modified to remove the USER command. This can be accomplished quite easily. Execute MOVCPM for your memory size as outlined in the

```
0A>BYE

ENTER ANY KEY TO BEGIN LOGON PROCEDURE
ENTER USER NAME : ONE
ENTER PASSWORD : ????
1A>BYE

ENTER ANY KEY TO BEGIN LOGON PROCEDURE
ENTER USER NAME : FIFTEEN
ENTER PASSWORD : ????
15A>BYE

ENTER ANY KEY TO BEGIN LOGON PROCEDURE
ENTER USER NAME : JOE
ENTER PASSWORD : ????
INVALID USER NAME - PASSWORD PAIR
ENTER USER NAME : PETE
ENTER PASSWORD : ????
INVALID USER NAME - PASSWORD PAIR
ENTER USER NAME : SAM
ENTER PASSWORD : ????
INVALID USER NAME - PASSWORD PAIR
TOO MANY TRIES--CONSOLE LOCKING UP FOR 30 SECONDS
ENTER ANY KEY TO BEGIN LOGON PROCEDURE
ENTER USER NAME : OP
ENTER PASSWORD : ????
0A>
```

Figure 2

Digital Research manuals. Then SAVE it to the disk. Next load the file under DDT and display memory at 0C00 using the D(isplay) command: It should look like Figure 1. To remove the USER command use S(et) to write zeros at locations 0CA4 thru 0CA7, as also shown in Figure 1. Now exit DDT and SAVE the modified CP/M image back to the disk. The modified version can now be SYSGENed. Any attempt at invoking USER when the system has been booted from this disk will be in vain. Note this change only removes the ability of the CCP to change user directories. BDOS calls to function 32 by application programs will still change user directories as before.

```
0A>TYPE PASSWORD.DAT
OP        1234
ONE       0001
TWO       0002
THREE     0003
FOUR      0004
FIVE      0005
SIX       0006
SEVEN     0007
EIGHT     0008
NINE      0009
TEN       0010
ELEVEN    0011
TWELVE    0012
THIRTEEN  0013
FOURTEEN  0014
FIFTEEN   0015

0A>DDT PASSWORD.DAT
DDT VERS 2.2
NEXT  PC
0238 0100
-D100
0100 4F 50 20 20 20 20 20 20 20 20 31 32 33 34 0D 0A OP        1234..
0110 4F 4E 45 20 20 20 20 20 20 20 30 30 30 31 0D 0A ONE       0001..
0120 54 57 4F 20 20 20 20 20 20 20 30 30 30 32 0D 0A TWO       0002..
0130 54 48 52 45 45 20 20 20 20 20 30 30 30 33 0D 0A THREE     0003..
0140 46 4F 55 52 20 20 20 20 20 20 30 30 30 34 0D 0A FOUR      0004..
0150 46 49 56 45 20 20 20 20 20 20 30 30 30 35 0D 0A FIVE      0005..
0160 53 49 53 20 20 20 20 20 20 20 30 30 30 36 0D 0A SIX       0006..
0170 53 45 56 45 4E 20 20 20 20 20 30 30 30 37 0D 0A SEVEN     0007..
0180 45 49 47 48 54 20 20 20 20 20 30 30 30 38 0D 0A EIGHT     0008..
0190 4E 49 4E 45 20 20 20 20 20 20 30 30 30 39 0D 0A NINE      0009..
01A0 54 45 4E 20 20 20 20 20 20 20 30 30 31 30 0D 0A TEN       0010..
01B0 45 4C 45 56 45 4E 20 20 20 20 30 30 31 31 0D 0A ELEVEN    0011..
-
```

Figure 3

Preparing the PASSWORD.DAT file

The system operator should assign a user directory to each user of the system and give a user name/password pair to the user. The PASSWORD.DAT file can be prepared using any standard text editor or word processor. Each entry is terminated by a carriage return and linefeed and consists of 14 characters. The first 10 characters are the user name; this field must be padded with spaces to make up 10 characters. The last four characters are the password; see the example in Figure 3. The first entry corresponds to user directory 0, the second to user directory 1, and so on up to directory 15.

As a precautionary measure both the BYE.COM and PASSWORD.DAT files should be given "read only" and "system" attributes using the STAT command as described in the standard CP/M documentation. This will keep users from knowing of their existence, since system files will not show up in a DIR display, even if a user does manage to gain access to the system directory.

I hope some of you will find this program useful. I have used it on a system supported by a 10MB Winchester disk with excellent results. ❑

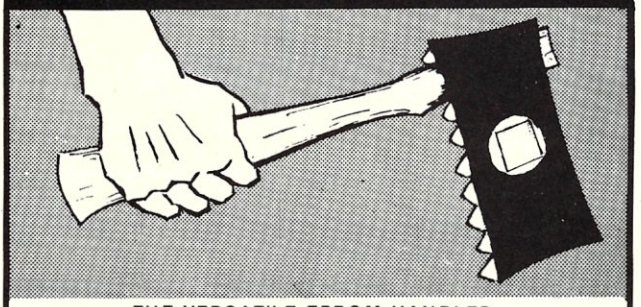
CP/M USER PASSWORD PROGRAM

```
;
CR      EQU     0DH             ;ASCII CARRIAGE RETURN
LF      EQU     0AH             ;ASCII LINE FEED
BDOS    EQU     5               ;ENTRY LOCATION INTO BDOS
CPM     EQU     0               ;WARM START LOCATION
TOUT    EQU     50              ;VALUE FOR CONSOLE LOCKOUT TIME
;                                50 = APPROX. 30 SECONDS @2.5 MHZ
RETRY   EQU     3               ;NUMBER OF RETRIES ALLOWED BEFORE
;                                CONSOLE LOCK-UP OCCURS
;
;
;
;
        ORG     100H            ;BEGINNING OF TPA
;
;
BYE:    LXI     H,0             ;SAVE CP/M STACK POINTER
        DAD     SP
        SHLD    SPSAVE          ;IN LOCATION SPSAVE
        LXI     SP,STACK        ;SET-UP LOCAL STACK
        MVI     C,32            ;SET USER NUMBER TO 0
        MVI     E,0
        CALL    BDOS
        CALL    GTPASS          ;READ IN PASSWORD.DAT FILE
;                                PASSWORD.DAT CONTAINS THE
;                                USER NAME PASSWORD PAIRS
BYE1:   MVI     A,RETRY         ;SET UP THE RETRY COUNTER
        STA     TRYCTR
        LXI     D,PROMPT        ;SEND THE LOGON PROMPT TO
;                                THE CONSOLE
        MVI     C,9
        CALL    BDOS
        MVI     C,1             ;INPUT A KEY FROM THE CONSOLE
;                                BEFORE CONTINUING
        CALL    BDOS
LOGON:  LXI     D,LOGON1        ;SEND USER NAME PROMPT
;                                TO CONSOLE
        MVI     C,9
        CALL    BDOS
        LXI     D,NAMBUF        ;INPUT USER NAME TO NAMEBUF
        MVI     C,10
        CALL    BDOS
        LXI     H,NAMBUF
        CALL    FILBUF          ;FILL REMAINING PORTION OF
;                                NAMBUF  WITH ASCII SPACES
        LXI     D,LOGON2        ;SEND PASSWORD PROMPT TO
;                                CONSOLE
        MVI     C,9
        CALL    BDOS
        LXI     H,PSSBUF        ;GET FOUR CHARACTERS TO PSSBUF
        MVI     B,4
PASSLP: PUSH    B
        PUSH    H
        CALL    GETCHR          ;GETCHR GETS ONE CHARACTER
        POP     H
        POP     B
        MOV     M,A             ;SAVE CHARACTER IN PSSBUF
        INX     H               ;INCREMENT H TO NEXT POSTION
;                                IN PSSBUF
        DCR     B
        JNZ     PASSLP          ;B REG HOLDS THE CHASACTER
;                                COUNT
        LXI     H,PSSTBL        ;POINT TO TABLE OF USER
;                                NAME AND  PASSWORD PAIRS
        MVI     B,16            ;LOAD B WITH NUMBER OF
;                                TABLE ENTRIES
SEARCH: MVI     C,14            ;LOAD C WITH LENGTH OF
;                                TABLE ENTRY
        LXI     D,NAMBUF+2      ;POINT D TO PAIR ENTERED
;                                BY USER
        PUSH    H
        CALL    TEST            ;TEST COMPARES USER ENTRY TO
;                                TABLE ENTRY POINTED TO BY
;                                DE REG
        JZ      MATCH           ;IF ENTRIES MATCHED JUMP
        POP     H
        LXI     D,16            ;ELSE POINT TO NEXT ENTRY
        DAD     D
        DCR     B               ;AND DECREMENT NUMBER OF
;                                ENTRIES
;                                LEFT TO SEARCH
        JNZ     SEARCH          ;GO BACK AND TRY NEXT ENTRY
        LXI     D,LOGON3        ;IF NO MATCH FOUND SEND
;                                INVALID MESSAGE TO CONSOLE
        MVI     C,9
        CALL    BDOS
        LDA     TRYCTR          ;DECREMENT RETRY COUNTER
        DCR     A
```

```
        STA     TRYCTR
        JNZ     LOGON           ;IF LESS THEN THREE TRIES
;                                LET HIM TRY AGAIN
        LXI     D,TMSG          ;ELSE SEND LOCK-UP MESSAGE
        MVI     C,9
        CALL    BDOS
        LXI     H,0             ;H=0
        MVI     B,TOUT          ;LOAD B WITH TIME OUT VALUE
LOOP1:  DCX     H               ;DECREMENT H UNTIL ZERO
        MOV     A,L             ;THIS LOOP DELAYS APPROX.
;                                .63 MS. @2.5 MHZ TO FIND
;                                DELAY FOR OTHER CLOCK FREQ.
;                                MULTIPLY 1.573 X (CLOCK RATE
;                                IN MICRO-SECONDS) = DELAY
;                                TIME IN MILLI-SECONDS i.e.
;                                1.573 x .4 = .6292 MILLI-
;                                SECONDS
        ORA     H
        JNZ     LOOP1
        DCR     B               ;IF TOUT=50 TOTAL LOOP TIME
;                                APPROX. 31.5 SECONDS
        JNZ     LOOP1
        JMP     BYE1            ;AFTER LOCK-UP LET USER TRY
;                                AGAIN
MATCH:  MOV     A,B             ;IF MATCH OCCURS B REG HOLDS
;                                15 MINUS TABLE ENTRY OF MATCH
        CMA                     ;COMPLIMENT THAT NUMBER
        INR     A               ;INCREMENT IT
        ANI     0FH             ;GET RID OF HIGH FOUR BITS
        MOV     E,A             ;IT EQUALS USER NUMBER OF
;                                MATCHED ENTRY
        MVI     C,20H           ;SET USER NUMBER
        CALL    BDOS
        LHLD    SPSAVE          ;RESTORE CP/M STACK POINTER
        SPHL
        RET                     ;RETURN TO CP/M
ERROR:  MVI     C,9             ;IF PASSWORD.DAT FILE NOT FOUND
        LXI     D,ERRMSG        ;SEND ERROR MESSAGE TO CONSOLE
        CALL    BDOS
        JMP     CPM
FILBUF: MOV     A,M             ;GET MAX BUFF LENGTH
        INR     A               ;PLUS ONE
        MOV     B,A             ;SAVE IN B REG.
        INX     H
        MOV     A,M             ;GET ACTUAL LENGTH
        INR     A               ;PLUS ONE
        MOV     E,A             ;SAVE IN E REG.
        MVI     D,0
        DAD     D               ;ADD ACTUAL LENGTH PLUS ONE
;                                TO HL. SO HL NOW POINTS TO
;                                FIRST ONE USED LOCATION IN
;                                BUFFER
FILLP:  MOV     A,E
        CMP     B
        RZ                      ;IF CURRENT POS. IS MAX LENGTH
;                                PLUS
;                                ONE THEN RETURN
        MVI     M,20H           ;ELSE FILL WITH ASCII SPACE
        INX     H               ;BUMP POINTERS
        INR     E
        JMP     FILLP           ;CONTINUE
GETCHR: MVI     C,6             ;INPUT A CHARACTER
        MVI     E,0FFH          ;USING DIRECT CONSOLE I/O
        CALL    BDOS
        ORA     A               ;IF ZERO RETURNED THEN CHAR
;                                NOT READY
        JZ      GETCHR
        PUSH    PSW             ;SAVE CHAR ON STACK
        MVI     C,2             ;SEND A QUESTION MARK FOR EACH
        MVI     E,'?'           ;CHARACTER INPUT
        CALL    BDOS
        POP     PSW             ;GET CHAR BACK TO A REG
        RET                     ;RETURN
TEST:   LDAX    D               ;GET USER ENTRY CHARACTER  TO
;                                A REG
        CMP     M               ;COMPARE IT TO TABLE ENTRY
        RNZ                     ;IF NO MATCH THEN RETURN
        INX     D               ;IF MATCH THEN BUMP POINTERS
        INX     H
        DCR     C               ;C REG HAS LENGTH OF FIELD
        JNZ     TEST            ;IF C NO ZERO THEN THERE IS
;                                MORE TO DO
        XRA     A               ;IF ALL MATCHED RETURN WITH
;                                A REG = 0
        RET
GTPASS: LXI     D,FCB1          ;OPEN FILE
        MVI     C,15            ;PASSWORD.DAT
        CALL    BDOS
        CPI     0FFH            ;CHECK FOR SUCESSFUL OPEN
        JZ      ERROR           ;IF NOT THEN DISPLAY ERROR
;                                MESSAGE
        LXI     D,PSSTBL        ;SET DMA TO PSSTBL
        MVI     C,26
        CALL    BDOS
        LXI     D,FCB1          ;READ PASSWORD.DAT INTO PSSTBL
        MVI     C,20
        CALL    BDOS
        LXI     D,PSSTBL+128    ;TABLE IS 256 BYTES LONG SO
;                                READ
        MVI     C,26            ;TWO SECTORS
        CALL    BDOS
        LXI     D,FCB1
        MVI     C,20
        JMP     BDOS            ;JUMP HERE CAUSES BDOS TO
;                                RETURN TO ROUTINE THAT
;                                CALLED GTPASS
;
; DATA AREA
;
;FILE CONTROL BLOCK FOR PASSWORD.DAT
;
FCB1:   DB      0               ;IF ZERO HERE SEARCH CURRENTLY
;                                LOGGED DRIVE, IF A ONE PLACED
;                                HERE WILL SEARCH DRIVE A:
        DB      'PASSWORD','DAT'
        DW      0,0,0,0,0,0
        DW      0,0,0,0,0,0
;
;
```

```
;BUFFER FOR USER NAME FIRST BYTE=MAX BUFFER LENGTH
;SECOND BYTE FILLED IN BY CP/M AS TO NUMBER OF BYTES ACTUALLY
;READ LAST 10 BYTES ACTUAL DATA STORAGE
;
;
NAMBUF: DB      10              ;MAX LENGTH OF NAMBUF
        DS      11              ;RESERVE STORAGE FOR USER NAME
                                 BUFFER
;
;
;BUFFER FOR PASSWORD MUST FOLLOW NAMEBUF IN MEMORY
;FOR SEARCH TO WORK CORRECTLY
;
;
PSSBUF: DS      4               ;RESERVE STORAGE FOR PASSWORD
;
;
;STORED MESSAGE AREA
;
;
PROMPT: DB      CR,LF,'ENTER ANY KEY TO BEGIN LOGON PROCEDURE'
        DB      '$'
LOGON1: DB      CR,LF,'ENTER USER NAME : '
        DB      '$'
LOGON2: DB      CR,LF,'ENTER PASSWORD : '
        DB      '$'
LOGON3: DB      CR,LF,'INVALID USER NAME - PASSWORD PAIR'
        DB      '$'
TMSG:   DB      CR,LF,'TOO MANY TRIES--CONSOLE LOCKING'
        DB      ' UP FOR 30 SECONDS'
        DB      '$'
ERRMSG: DB      CR,LF,'FILE PASSWORD.DAT NOT FOUND','$'
;
;
;STORAGE AREA FOR CP/M STACK POINTER
;
;
SPSAVE: DS      2
;
;
;STORAGE FOR TRY COUNTER
;
;
TRYCTR: DS      1
;
;
;STORAGE FOR USER NAME - PASSWORD PAIRS READ IN FROM
;FILE PASSWORD.DAT
;
;
PSSTBL: DS      256
;
;
;THIS IS JUST TO FORCE A LABEL TO THE END OF MEMORY
;USED BY THE PROGRAM SO WE CAN CALCULATE WHERE TO
;PUT THE LOCAL STACK
;
;
LAST:   DB      0
STACK   EQU     LAST + 128      ;RESERVE ROOM FOR LOCAL STACK
;
;
        END
```

UNIPROM



THE VERSATILE EPROM HANDLER

- Reads programs 2704, 2708, 2758, 2508, 2516, 2716 (1supply) 2532, 2732, INTEL'S 2732A and the 8755A (INTEL/NEC).
- Reads/erases/programs Hitachi 48016 EEPROM'S
- No personality modules required.
- All signals are S-100 compatible (can adapt to most other buses)
- Port mapping occupies NO memory space.
- Bus clock rates exceeding 6mhz.
- All software is 8080/8085/Z80 compatible.
- Software "user" friendly.
- All software is fully CP/M® & CDOS compatible.

Board (A&T) with extensive documentation .. $199.00
Disk software (8" or NORTHSTAR 5.25") $38.00
EPROM-based software with source listing ... $55.00
Expansion console, 24 pin $50.00

CDOS is a registered trademark of CROMENCO
CP/M® is a registered trademark of DIGITAL RESEARCH INC.
NORTHSTAR is a registered trademark of NORTHSTAR COMPUTER, INC.

Featured in July, August '82 Microsystems

CER-TEK

6020 Doniphan (915) 581-6697 El Paso, TX 79932

ELECTRONICS MANUFACTURING SINCE 1975

CIRCLE 76 ON READER SERVICE CARD

A> INFORMATION MANAGEMENT

DBPACK - Data Base Management: Indexing, Sort & Search, Tabulation, Address Labels, Word Processor Interfaces, and lots more!

FILER - Files Compression, Archiving and Cataloging: Substantial Disk Space Savings; Date-stamping & Descriptive Comments ...

A> COMMUNICATION SOFTWARE

COMCOM - Communication Program: Communicate with Time-sharing systems and other computers; Advanced Features; Easy to Use.

CPMCPM - File Transfer Program: Transfer Files (any type) between different CP/M-based systems.

A> For CP/M® Computers

CP/M is a registered trademark of Digital Research

Configured for a wide variety of systems.
Disk formats include 8-inch, Osborne, Xerox ...

Call or write for information

COMPU-DRAW
1227 Goler House
Rochester, NY 14620
Phone: (716)-454-3188

Dealer inquiries invited

MasterCard, Visa & American Express cards welcome.
Separately ordered documentation may be returned for full refund within 10 days!

It's the writing on the wall

CIRCLE 207 ON READER SERVICE CARD

BDS C

The *fastest* CP/M-80 C compiler you can get

Version 1.5 contains some nifty improvements:

The unscrambled, *comprehensive* new User's Guide comes complete with tutorials, hints, error message explanations and an index.

The *CDB* symbolic debugger is a valuable new tool, written in C and included in *source form*. Debug with it, and *learn* from it.

Hard disk users: You can finally organize your file directories sensibly. During compilation, take advantage of the new path searching ability for all compiler/linker system files. And at run-time, the enhanced file I/O mechanism recognized user numbers as part of simple filenames, so you can manipulate files located *anywhere* on your system.

BDS C's powerful original features include dynamic overlays, full library and run-time package source code (to allow customized run-time environments, such as for execution in ROM), plenty of both utilitarian and recreational sample programs, *and speed*. BDS C takes less time to compile and link programs than any other C compiler around. And the execution speed of that compiled code is typically lightning fast, as the Sieve of Eratosthenes benchmark illustrates. (See the January 1980 BYTE, pg. 303).

BD Software
P.O. Box 9
Brighton, MA 02135
(617) 782-0836

8" SSSD format, $150
Free shipping on pre-paid orders
Call or write for availability on other disk formats

CIRCLE 202 ON READER SERVICE CARD

THEY SAY IT ALL... WE DO IT ALL!

C 86™ C COMPILER

ANNOUNCING THE C86™ C COMPILER —THE COMPILER THAT SPEAKS THE LANGUAGE OF THE FUTURE!

Kernighan and Ritchie's book, *The C-Programming Language,* is the key source for C. Just as fundamental is the C86 ™ C Compiler.

The C86 ™ C Compiler is especially designed for the IBM® Personal, IBM® Display Writer, CP/M-86® and MS-DOS®.

For further information on the C-programming language and the C86 ™ C Compiler, please contact:

C86 is a trademark of Computer Innovations, Inc.; CP/M-86 is a trademark of Digital Research; IBM and MS-DOS are registered trademarks of International Business Machines, Inc.

Computer Innovations, Inc.
75 Pine Street
Lincroft, New Jersey 07738
Telephone: (201) 530-0995

CIRCLE 209 ON READER SERVICE CARD

WordMaster Patches for WordStar Compatibility and Date/Time

by Jim Gilbreath

MicroPro International's WordMaster is a popular screen editor for the CP/M-80 environment. It is small, fast, easily tailored to a wide variety of equipment, and is basically just what you want for editing program source code. It's not a word processor, but there is the very popular WordStar (also from MicroPro) for that job.

WordStar is so popular, in fact, that it seems everyone is familiar with the keystrokes it uses for moving the cursor around and for basic editing tasks. These keystrokes quickly become second nature, and are done without thinking, just as operating the clutch and brake in driving a car.

But after a few hours of working with WordStar on a document, these second-nature habitual keystrokes get you in trouble when you go back to WordMaster to do some source code editing. Why not just stay with WordStar in the nondocument mode for programming? Several reasons:

1. It is bigger, and takes longer to load into memory.
2. It takes up a lot more disk space that I always seem to need for programs.
3. It doesn't normally come up in nondocument mode, so it takes much longer to get a file opened and get to work.
4. It has to drag in an overlay file for such things as delete to end of line, and this takes extra time.

For documents, I like WordStar very much, but for programming, I'll take WordMaster, thank you.

Release 1.06 and earlier of WordMaster used the ADM-3A default keys of control H, J, K, and L to move the cursor around, while WordStar uses control S, E, D, and X because the diamond shape they describe supposedly makes it easier to remember which does what. I disagree with this idea, but never mind, the deed is done, and the whole world uses S E D X now.

Release 1.07 of WordMaster "fixed" part of the problem by making control S E D X the cursor keys, but didn't go quite far enough in the right direction. Many of the other basic editing functions that could have been made the same as WordStar's were not. A particularly hazardous conflict is the use in WordStar of Control K for a variety of functions such as closing a file, while in WordMaster 1.07 it means DELETE TO END OF LINE. So without thinking, you do a control K, and oops!—there went most of a line of complex code into the bit bucket.

After two years of frustration from the inability of my fingers to remember which editor they were driving, I spent a little time to see if I could find a way to patch WordMaster 1.07 to make it a lot *more* like WordStar. Specific goals were:

1. Control K, L, and O "disconnected," so hitting them would not do any damage.
2. Make Control V toggle the insert mode, as in WordStar.
3. Make Control QS and Control QD position the cursor to left and right ends of the line, as in WordStar.
4. Make Control QE and Control QX position the cursor to top and bottom of the screen, as in WordStar.
5. Make Control QY delete to end of line, as in WordStar.
6. Disconnect Control B, since there is no line justification in WordMaster.
7. Make Control QI move cursor to next tab stop.
8. Make Control U "interrupt" to Command Mode (very useful when you've done *<Nsearchword$V and want to get out of it before hitting the end of file).

Several hours of poking around with the debugger and experimenting led to the successful development of a patch program (Listing 1), which, when loaded on top of WordMaster with DDT, will provide all the above features. And the frustration is gone! I can now switch back and forth between WordMaster and WordStar and find that all the commands are the same or harmless.

Date and time

An important feature not found in the CP/M-80 file system is the ability to keep track of when a program was last changed, and therefore which version is the latest and greatest. Based on some earlier work by Dick Mason and Mike Lehman, another WordMaster patch was developed that allows you to keep the date and time as a comment on the first line of a file. It's a simple kludge, consisting of four "anything" characters on the first line followed by the date and time, as in "DATE 06sep82 1735".

The "anything" characters depend on the language's requirements for a comment line: for example, REM in Basic, /* in C, ; in Assembly, C in Fortran, (* in Pascal, etc.).

The mechanism is activated the first time you enter Command Mode, *if* the fifth through eighth characters of the file are DATE. If you have a clock/calendar provision in your computer, you can modify the patch code to automatically update the date line. Otherwise, the patch will print the current DATE line at the bottom of the screen and encourage you to update it manually. If you don't want to bother with it at the moment and respond with only a carriage return, the time field will be incremented by one. (It doesn't know about seconds and minutes, so 1459 will

Jim Gilbreath, 7266 Courtney Drive, San Diego, CA 92111

be incremented to 1460, 1499 to 1500, etc.).

Installing the patch

First produce a .HEX file, either by assembling the program of Listing 1, or by typing the .HEX output in Listing 2 into a file manually. Although the program is written using Zilog mnemonics, there are no Z80-only instructions, and it will run on an 8080 or 8085, as well as a Z80. First load your WORDMASTER.COM file into memory with DDT, overlay the .HEX file on top of it, type Control-C, and SAVE 42 blocks. Example:

```
A>DDT WMSTR.COM
DDT VERS 2.2
NEXT  PC
2680 0100
-IWMPATCH.HEX
-R
NEXT  PC
2AB6 0000
-^C
A>SAVE 42 NEWED.COM
```

Now practice using your new tool by editing WORDMASTER.HLP to make it tell the truth about the new features! ■

I can switch between WordMaster and WordStar, and find that all the commands are the same or harmless.

LISTING 1.

```
;   DATE 20nov82 2324

; Patch for WORDMASTER 1.07 to provide WORDSTAR keystroke
; compatibility and dating of file changes.
; Beginning of files must be like the beginning of this one
;   (four anything characters, followed by DATE ddmmmyy tttt)

; History
;           ca 1980   dick mason customized terminal mods
;           16 mar 81 mike lehman added VERSION numbers
;           28 mar 81 jim gilbreath adapted to 1.07
;           12 jan 82 jim gilbreath changed VERSION to DATE and added time
;           18 aug 82 jim gilbreath for wordstar compatibility

bdos     equ      5

; These are version sensitive equates.  They MUST all be correct.

bufptr   equ      1b7h              ;adr of patch area
probe    equ      04e0h             ;adr to place jump to patch
ret1     equ      04e3h             ;place to return if no action
ret2     equ      04deh             ;place to go if we did anything
patarea  equ      29b8h             ;location of patch area (old text buffer)
chout    equ      24fch             ;routine called where probe was placed
chinp    equ      23b4h             ;get a character from keyboard

pritbl   equ      10d6h             ;location of key priority table
keyvec   equ      11c5h             ;location of key dispatch table
ctlvpat  equ      2346h             ;location of control v patch

DUMMY    equ      1345h
CURLTWD  equ      126Dh
EOLTOGL  equ      1289h
WINDOWN  equ      142Ch
CURRT    equ      1216h
CURUP    equ      1303h
CURRTWD  equ      123Fh
GOBBLE   equ      14ECh
CURLT    equ      122Ah
TAB      equ      153Ah
HELP     equ      1954h
CARRET   equ      15ABh
INSLF    equ      1594h
CTLCHR   equ      151Ah
CURTAB   equ      12B6h
WINDUP   equ      1414h
DELWD    equ      1476h
INSTOGL  equ      150Bh
WINDUP1  equ      1409h
CURDN    equ      12C1h
LINEDEL  equ      14CEh
WINDN1   equ      137Ch
ESC      equ      1205h
DELWDLT  equ      1460h
UPDOWN   equ      1346h
GOTOP    equ      134ch
GOBOTM   equ      1362h
GOLTEND  equ      128fh
GORTEND  equ      12ach
DELEOL   equ      14ddh
LEFTDEL  equ      1448h
ERMSG    equ      0534h
EPAT     equ      053eh
CMDMODE  equ      04b9h

         org      bufptr            ;address of pointer to text buffer
         defw     next              ;change start of edit buffer

         org      pritbl

         defb     00       ;^@
         defb     40h      ;^A
         defb     00       ;^B
         defb     80H      ;^C
         defb     00h      ;^D
         defb     80h      ;^E
         defb     00       ;^F
         defb     20h      ;^G
         defb     00       ;^H
         defb     08h      ;^I
         defb     01h      ;^J
         defb     40h      ;^K
         defb     00       ;^L      formerly 01
         defb     05h      ;^M
         defb     30h      ;^N
         defb     00       ;^O      formerly 20
         defb     00       ;^P
         defb     00       ;^Q
         defb     80h      ;^R
         defb     00       ;^S
         defb     00       ;^T
         defb     00       ;^U
         defb     20h      ;^V      formerly 00
         defb     00       ;^W
         defb     80h      ;^X      formerly 0
         defb     30h      ;^Y
         defb     01h      ;^Z
         defb     80h      ;^[
         defb     20h      ;^\
         defb     00       ;^]
         defb     00       ;^^
         defb     20h      ;^_

         org      keyvec

;KEY JUMP VECTOR, BEGINNING WITH ^@

         defw     DUMMY   ; ^@ do next key 4 times
         defw     CURLTWD ; ^A
```

WordMaster Patches continued . . .

```
;
;   init looks at edit buffer - if the first text is
;   'DATE ' then user is asked for new date and time.
;   if user answers with only c/r, then existing time
;   is incremented.  if text does not match, no change
;   is made.
;
vmsg:   defb    'DATE '
vmlen:  equ     $-vmsg

initit:         push    bc
                push    de
                push    hl
                call    init
                pop     hl
                pop     de
                pop     bc
                pop     af
                jp      ret2            ;go back to type * again (VERSION sensitive)

init:   ld      a,01h                   ;set been here before flag
        ld      (bhb),a
        ld      hl,vertxt               ;point to edit buff vers no
        ld      de,vmsg                 ;point to std text
        ld      b,vmlen                 ;length to compare
        call    compar                  ;same?
        ret     nz                      ;n- do no more here

        ld      c,12
        ld      hl,dateloc
        ld      de,curdat               ;place to move existing date
pikup:  ld      a,(hl)
        ld      (de),a
        inc     hl
        inc     de
        dec     c
        jp      nz,pikup
        ld      c,9                     ;com print buffer call
        ld      de,datmsg
        call    bdos                    ;type out exiting date and time
        ld      c,10                    ;read console buffer
        ld      de,msgin                ;where to put the line
        call    bdos

        ld      c,9
        ld      de,crlf                 ;put out cr and lf
        call    bdos

        ld      hl,msgin+1              ;length byte
        ld      a,(hl)                  ;get length byte
        cp      0                       ;see if just c/r
        jp      z,noinp                 ;default only
        inc     hl                      ;point to first byte
        ld      a,(hl)                  ;get first char

        cp      '0'                     ;see if number 0-3
        jp      c,noinp                 ;don't change if less than 0
        cp      '4'
        jp      nc,noinp                ;or more than 3

        ld      c,12                    ;chars to move
        ld      de,dateloc              ;destination (source is in hl)
pikupl: ld      a,(hl)
        ld      (de),a
        inc     hl
        inc     de
        dec     c
        jp      nz,pikupl
        ret

noinp:
```

WordMaster Patches continued . . .

```
        defw    DUMMY       ; ^B formerly end of line toggle
        defw    WINDOWN     ; ^C
        defw    CURRT       ; ^D
        defw    CURUP       ; ^E
        defw    CURRTWD     ; ^F
        defw    GOBBLE      ; ^G
        defw    CURLT       ; ^H
        defw    TAB         ; ^I
        defw    HELP        ; ^J formerly delete to end line
        defw    DUMMY       ; ^K formerly cursor right
        defw    CARRET      ; ^M
        defw    INSLF       ; ^N
        defw    DUMMY       ; ^O formerly insert mode toggle
        defw    CTLCHR      ; ^P
        defw    qpatch      ; ^Q formerly CURTAB, now multi-use
        defw    WINDUP      ; ^R
        defw    CURLT       ; ^S
        defw    DELWD       ; ^T
        defw    NTRUPT      ; ^U formerly delete line left
        defw    INSTOGL     ; ^V new insert toggle, formerly vio control
        defw    WINDUP1     ; ^W
        defw    CURDN       ; ^X
        defw    LINEDEL     ; ^Y
        defw    WINDMl      ; ^Z
        defw    ESC         ; ^[
        defw    DELWDLT     ; ^\
        defw    CTLCHR      ; ^]
        defw    UPDOWN      ; ^^
        defw    LEFTDEL     ; del

        org     ctlvpat
        jp      ctlvpat + 6;    jump around test for vio control

        org     epat
        jp      cmdmode

        org     probe       ;patch jump location (VERSION sensitive)
        jp      dapatch     ;overlays call to print char routine

        ora     patarea     ;place to put our code

qpatch:                     ;got a control q multicode
        call    chinp       ;get next char from keyboard
        and     01fh        ;strip off to bare control code bits
        cp      04h         ;control d
        jp      z,GOMTEND   ;go to end of line
        cp      05h         ;control e
        jp      z,GOMTOP    ;go to top of screen
        cp      09h         ;control i
        jp      z,CURTAB    ;cursor tab
        cp      13h         ;control s
        jp      z,GOLTEND   ;go to left end of line
        cp      18h         ;control x

        jp      z,GOBOTM    ;go to bottom of screen
        cp      19h         ;control y
        jp      z,DELEOL    ;delete to end of line
        jp      DUMMY       ;none of the above

NTRUPT: call    ERMSG       ;return to command mode
        defb    'COMMAND MODE ',0

dapatch:
        call    chout       ;type * (VERSION dependent)
        push    af          ;save char
        ld      a,(bhb)     ;first time through?
        or      a           ;set flag
        jp      z,initit    ;yes
        pop     af          ;no, continue
        jp      retl        ;jp back where left off (VERSION sensitive)
```

WordMaster Patches continued . . .

```
        ld      hl,num+digits-1 ;point to last digit
        ld      b,digits        ;increment ascii number
        call    bmpasc
        ret

msgin:  defs    15              ;actually needs only 12

datmsg: defb    'DATE is '
curdat: defs    12              ;gets stored into by whats in file now
        defb    '; $'
crlf:   defb    0dh,0ah,'$'

; compar compares string pointed by hl to that pointed
;       by de for chars in b.  returns with zero set
;       if a match.
;
compar: ld      a,(de)
        cp      (hl)
        ret     nz              ;return if mismatch
        dec     b
        ret     z               ;z is set if match to end
        inc     de
        inc     hl
        jp      compar          ;keep checking

; bmpasc increments ascii number whose last digit is
;       pointed by hl.  b has number of digits.
;
bmpasc: inc     (hl)            ;increment ascii digit
        ld      a,(hl)
        cp      '9'+1
        ret     nz              ;overflow?
        ld      (hl),'0'        ;n- were done
        dec     hl              ;y- reset to 0 and do
        dec     b               ; next digit
        jp      nz,bmpasc       ;last digit?
        ret                     ;n- do next
                                ;y- done

bhb:    defb    0               ;been here before flag

next    equ     $               ;start of edit buffer
vertxt  equ     next+1+4        ;4 don't cares, then date
dateloc equ     vertxt + vmlen  ;where 11jan82 or equiv goes
num     equ     vertxt+13       ;start of ascii number
digits  equ     4               ;number of digits in time no
        end
```

LISTING 2. HEX File

```
:0201B700B62A66
:1C10D6000400080008000200008014000053000000080000000200080300180 4F
:0410F200200000220BA
:1C11C500451360124513 2C1416120313 3F12EC142A123A1554194513451 3AB15A8
:1C11E1009415451131A15B82914142A127614DE290B150914C112CE147C13051254
:0811FD0060141A154613481492
:0323460 0C34C2362
:03053E00C3B9043A
:0304E000C3EF293E
:1C29B800CDB423B61FFE04CAAC12FE05CA4C13FE09CAB612FE13CA8F12FE18CAAF
:1C29D4006213FE19CADD14C34513CD3405434F4D4D414E44204D4F4445200CD4E
:1C29F000FC24F53AB52AB7CA032AF1C3E3044441544520C5D5E5CD102AE1D1C11D
:1C2A0C00F1C3DE043E0132B52A21BB2A11FE290605CD9E2AC000E0C21C02A118C68
:1C2A28002A7E1223130DC22292A0E0911842ACD05000E0A11752ACD05000E091116
:1C2A44009B2ACD0500217 62A7EFE00CA6C2A237EFE30DA6C2AFE34D26C2A0E0C4F
:152A60011C02A7E1223130DC2632AC921CB2A0604CDA82AC9F3
:082A84004441544206973010
:1C2A98003A20240D0A241ABEC005C81323C39E2A347EFE3AC036302B05C2A82A6F
:022AB400C90057
:000000000
```

LOWEST SOFTWARE PRICES GUARANTEED

We hereby certify that your purchase from Discount Software represents the lowest price sold anywhere. If you find a lower price on what you purchased within 30 days, send the ad and we'll refund the difference.

Discount Price

CP/M

ARTIFICIAL INTELLIGENCE
Medical (PAS-3) $849
Dental (PAS-3) $849
ASHTON-TATE
$4?? dBASE II... call for price
dBASE User's Guide $29
Friday! $265
Financial Planner $595
Bottom Line Strategist... $349
ASYST DESIGN/FRONTIER
Prof Time Accounting.... $549
General Subroutine $269
Application Utilities $439
DIGITAL RESEARCH
CP/M 2.2
Intel MDS..... $135
$149 Northstar
$159 TRS-80 Model II (P&T)
Micropolis $175
$98 CBasic-2
Display Manager..... $319
Access Manager..... $239
$449 PL/1-80
BT-80 ..... $179
MAC $85
RMAC $179
Sid $65
$90 Z-Sid
DeSpool..... $49
CB-80..... $459
Link-80..... $90
FOX & GELLER
Quickscreen $135
Quickcode..... $265
$65 DUtil
DGraph $265
MICRO DATA BASE SYSTEMS
Knowledge Man $429
HDBS..... $595
MDBS $1099
DRS or QRS or RTL..... $319
MDBS PKG..... $1999
MICROPRO
$279 WordStar
$179 Mail Merge
WordStar/Mailmerge $399
WS/MM/SpellStar $549
$199 SpellStar
DataStar..... $249
InfoStar $349
ReportStar..... $254
Wordmaster..... $119
Supersort I $199
Calc Star $129
MICROSOFT
$249 Basic-80
$329 Basic Compiler
$349 Fortran-80
$549 Cobol-80
Multiplan $219
M-Sort $175
$159 Macro-80
MuSimp/MuMath $224
MuLisp-80 ..... $174
ORGANIC SOFTWARE
Textwriter III $111
Datebook II $269
Milestone..... $269
OSBORNE (McGraw/Hill)
G/L, or AR & AP, or PAY... $59
All 3 ..... $129
All 3 + CBASIC-2..... $199
Enhanced Osborne $299
PEACHTREE
G/1,A/R,A/P,PAY,INV(each) $399
P8 Version ..... Add $234
Peachcalc $249
Peachpak (G/L, A/R, A/P)..... $349
STAR COMPUTER SYSTEMS
G/L, A/R, A/P, Pay(each).. $349
All 4 ..... $1129
Legal or Property Mgt.... $849
STRUCTURED SYSTEMS
Business Packages (call)
SORCIM
$179 SuperCalc I
SuperCalc II $265
Act $157
SUPERSOFT
Ada..... $270
Diagnostic II $89
Disk Doctor $89
Forth (8080 or z80) $149
Fortran..... $319
Ratfor..... $79
C Compiler $225
Scratch Pad..... $266
StatsGraph $174
Analiza II $45
Disk Edit $89
Term II $179
Utilities I or II..... $54
SOFTWARE DIMENSIONS/ ACCOUNTING PLUS
1 Module $399
4 Modules..... $1499
All 8..... $2799
UNICORN
Mince or Scribble (each).. $149
Both..... $249
The Final Word $270
WHITESMITHS
"C" Compiler..... $600
Pascal (incl "C") $850
"PASCAL"
Pascal/MT+ Pkg..... $429
Compiler $315
SP Prog $175
Pascal Z..... $349
Pascal/UCSD 4.0 ..... $349
DATA BASE
dBASE II..... Call 4??
Friday..... $265
FMS-80 ..... $799
FMS-80-1..... $399
Condor I & III..... Call
Selector..... $689
Superfile $159
"WORD PROCESSING"
Perfect Writer $389
WordSearch $114
SpellGuard $139
Peachtext $289
Spell Binder $349
Select..... $495
The Word $65
$145 The Word Plus
Palantier-1 (WP) $385
"COMMUNICATIONS"
Ascom $149
BSTAM or BSTMS $149
$139 Crosstalk
$89 Move-it
"OTHER GOODIES"
Micro Plan $419
Plan 80..... $264
Target PlannerCalc $79
Target Financial Modeling.. $299
Target Task $299
MicroStat..... $224
Vedit..... $130
MiniModel $449
StatPak $449
Micro B+..... $229
String/80 ..... $84
String/80 (source) $279
ISIS CP/M Utility $199
Lynx..... $199
Supervyz..... $95
ATI Power (tutorial)..... $75
Mathe Magic $95
CIS Cobol $765
Forms II $179
Basic..... $149
Zip MBasic, CBasic (each)..... $129

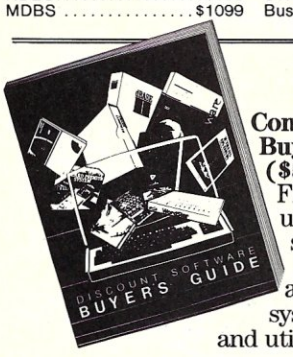
APPLE II

ASHTON-TATE
(See CP/M Ashton-Tate)
BRODERBUND
G/L (with A/P)..... $444
Payroll $355
INFO UNLIMITED
EasyWriter (Prof) $155
EasyMailer (Prof) $129
Datadex $134
MICROSOFT
Softcard (Z-80 CP/M).... $249
Fortran..... $179
Cobol $549
Tasc $139
Premium Package $549
RAM Card $89
MICROPRO
(See CP/M Micropro)
Z-80 Card..... $299
VISICORP
Visicalc 3.3 ..... $189
Desktop/Plan II..... $219
Visiterm $90
Visidex $219
Visitrend/Visiplot $259
Visifile $219
Visischedule $259
PEACHTREE
G/L, A/R, A/P, PAY, (each).. $224
PeachPack P40..... $349
ACCOUNTING PLUS
G/L, AR, AP, INV, (each).. $385
"OTHER GOODIES"
Dow Jones..... $314
Incredible Jack..... $129
Super-Text II $127
Data Factory $269
Mini Factory $139
DB Master..... $184
Versaform VS1 ..... $350

IBM PC, 16 BIT & DISPLAYWRITER

"WORD PROCESSING"
Wordstar $289
Spellstar $199
Mailmerge..... $179
Easywriter $299
Easyspeller $159
Select/Superspell $535
Write On $115
Spellguard..... $189
Spellbinder $349
Final Word..... $264
Wordex..... $159
Edix $159
"LANGUAGES & UTILITIES"
Crosstalk $139
Move-it..... $129
BSTAM or BSTMS $149
Pascal MT+ /86, SPP.... $679
CBasic 86 ..... $294
Act 86..... $157
Trans 86 ..... $115
XLT 86 ..... $135
MBasic (MSDOS)..... $329
MBasic Compiler (MSDOS).. $329
Cobol (MSDOS) $649
Pascal (MSDOS)..... $429
Fortran (MSDOS) $429
"C" (MSDOS) $429
"OTHER GOODIES"
SuperCalc II $265
VisiCalc $219
Visiplot/trend $259
Visidex..... $219
Easyfiler..... $359
Mathemagic..... $95
dBase II ..... Call 4??
Friday!..... $265
Statpak..... $449
Optimizer..... $174
Desktop Plan..... $259

FREE WITH PURCHASE:



Complete Software Buyer's Guide ($5.00 value)
Filled with facts and usable advice about scores and scores of software programs from accounting and business systems to word processing and utilities.

Exclusive "Hotline"
Our reputation for courteous and knowledgeable service has resulted in calls from people who never purchased our products. Now a separate "hotline" is available to customers only.

Confidential Software BargainGrams
Regular notices of insider's bargains not available to the general public.

ORDER TOLL-FREE VIA VISA OR MASTERCARD:

1 800 421-4003
or 1 213-837-5141
Calif: 1 800 252-4092
6520 Selma Avenue, Los Angeles, CA 90028

DISCOUNT SOFTWARE

Outside Continental U.S.—add $10 plus Air Parcel Post. Add $3.50 postage and handling per each item. California residents add 6½% sales tax. Allow 2 weeks on checks. C.O.D. $3.00 extra. Prices subject to change without notice. All items subject to availability. ®Mfr. trademark. Blue Label $3.00 additional per item. CP/M is a registered trademark of DIGITAL RESEARCH, INC.

M883

DISCOUNT SOFTWARE

Your DISCOUNT Sour

dBASE II™ /$398

The most widely sold DBMS for micros, dBASE II defines the state of the art. An inexperienced user can create a dBASE file, begin data entry, and print out a report in minutes; the experienced programmer can create complex program systems using dBASE's unique programming language. Both can use automatic program generators and other programming tools to create applications quickly and easily.

QUICKCODE: Automatically creates menus and "intelligent" data entry screens which check input for correctness. Eliminates 80–100% of dBASE programming time! **$185.00**

dUTIL: Formats programs to make them easier to read, detects some errors and combines program files to make them run fast! **$59.00**

dGRAPH: Allows you to see your dBASE file in graphic form—a real aid to understanding. **$195.00**

ABSTAT: Allows you to analyze your data using a wide variety of statistical tests. Can be used with dBASE files or by itself. Also does bar graphs and plots. **$359.00**

More Summer Specials

MULTIPLAN™ /$189

The world's hottest-selling spreadsheet, Multiplan™ is now setting the standard for ease of use and quality of documentation. Each of the Multiplan™ series comes complete with sophisticated training programs.

Multi-Tool Budget 'expert system': used with Multiplan™, enables new users to get budgets out in minutes **$69**

Multi-Tool Financial Statement: Expert help in getting out financials. **$109**

EDIX & WORDIX/$279

EDIX is a full-screen editor specifically designed to take advantage of the advanced features of the IBM PC; WORDIX is the best text formatter on the market. Together, they are an unbeatable combination, the finest word processor available on a micro—at an unbeatable price!

EDIX alone **$149**

WORDIX **$149**

SMARTMODEM 300 $209
1200 $509

These Hayes Stack Modems link your computer's RS232 serial port directly to a modular phone jack. They will dial the phone for you, and answer it as well. The Smartmodem 300 runs at 0–300 baud; for those needing higher speed, the Smartmodem 1200 can run at 1200 baud.

NEW! NEW! NEW!
Smartmodem 1200B: for the IBM PC only **$469**

Orders must be PAID by AUGUST 31 to qualify for these special prices

PROGRAMMING INTERNATIONAL

TERMS: All prices subject to change without notice and availability. Cashier's check/MO/bank transfer. Allow time for company or personal checks to clear. Prices reflect cash prepaid discount. VISA/MASTERCARD/COD/PO's + 3%. CA residents add sales tax. All sales final for games & special orders.

SHIPPING: $3 per item for UPS surface ($6 for Blue Label) within Continental USA, except where shipping cost is specified in square brackets. UPS does not go to Canada, Alaska, APO's, FPO's; call for ship charge or add 15%—we will refund/credit difference.

RETURNS: Must have authorization number, obtained at 415-324-0305. Unauthorized returns will be refused; damaged goods will be refused. All returns subject to 15% restocking fee. No return after 30 days.

(800)222-8811
Inside California:
(415)324-3730

Your *CP/M & HARDWARE* Source

PROGRAMMING INTERNATIONAL

More summer specials!

CP/M-80

8" standard SSSD and Apple 5¼". Please call to confirm price & availability for other formats, available as special orders.

	LIST	SALE
Mark of The Unicorn		
Final Word	300	223
Mince	175	139
Adventure International		
Adventures 1-12 (Req Z80)	129	97
ATI		
Power for CP/M	75	54
Anderson-Bell		
ABSTAT		359
Artificial Intelligence		
Dental (PAS-3)	995	749
Medical (PAS-3)	995	749
Ashton-Tate		
Bottom Line Strategist	400	279
dBASE II [5]	700	398
dBASE II w/User's Guide [5]	729	419
Aspen Software		
Grammatik	75	56
Proofreader	50	38
Balcones		
BOSS Fin. Acctg System	2495	1496
Byrom Software		
BSTAM	200	149
BSTMS	200	149
Caxton		
Cardbox	245	177
Comm soft		
Roots/M	195	142
Computer Control		
FABS-BTree File Access	195	146
Ultrasort II	195	146
Computer development		
Image	495	356
Comshare		
Planner Calc	99	74
Target Application Guide	50	38
Combo Pak	125	94
Condor		
Condor 1	295	189
Condor 3	650	398
Designer Software		
Palantir	425	319
Dictronics		
Random House Thesaurus	150	113
Digital Research		
Access Manager	300	225
CB80	500	395
CBasic	150	94
CIS Cobol	800	595
Despool	50	38
Display Manager	400	295
MAC	90	75
Pascal MT +	350	262
Pascal MT + with SPP	500	395
PL/1 80	500	375
SID	75	63
SPP	200	150
ZSID	100	88
DJR Associates, Inc.		
FMS 80	995	625
FMS 80-1	495	312
FMS 80-2	495	312
Eagle		
Citation	250	169
Ecosoft		
Microstat	295	248
Microstat 3.0	395	320
Epic		
Supervyz	150	94
Faircom		
Micro B + (CBasic)	260	211
Micro B + (Other)	260	211
Fox & Geller		
dGRAPH	295	195
dUTIL	99	59
Quickcode	295	195
Quickscreen/ CBasic	149	125
Quickscreen/ dBASE II	149	125
Quickscreen/ FMS-80	149	125
Quickscreen/ MBasic	149	125
FYI, Inc.		
Superfile	195	129
Human Soft		
dBPlus	125	95
Infocom		
Deadline	60	43
Starcross	50	38
Suspended	50	38
Zork I	50	38
Zork II	50	38
Zork III	50	38
ISM		
Mathemagic	100	72
Ithaca Intersystems		
Pascal BZ	450	375
Pascal Z	450	375
Pascal Z & BZ	650	542
Lexisoft		
Spellbinder	495	281
Metasoft		
Benchmark Mail List	250	184
Benchmark Wordprocessor	500	367
Calcstar	145	96
Datastar	295	195
Infostar	495	327
Mailmerge	250	165
Reportsrar	350	231
Spellstar	250	165
Supersort	250	165
Word/Mail [5]	695	426
Word/Spell [5]	695	426
Word/Spell/Mail [5]	845	558
Wordmaster	150	99
Wordstar [5]	495	327
Microsoft		
BASIC 80™	350	252
BASIC Compiler™	395	296
COBOL 80™	750	562
FORTRAN 80™	500	360
MACRO 80™	200	150
μLips/μStar™	200	156
Multiplan™	275	198
μMath/μSimp™	250	194
Sorting Facility (MSORT)™	195	151
Text Editor™	120	94
Microstuf		
Crosstalk/ Smartmodem	195	135
Northwest Analytical		
Statpak	495	369
Oasis		
The Word Plus	150	112
Organic Software		
Datebook	295	229
Milestone	295	299
Peachtree		
Series 4 Pak (GL,AR,AP) [5]	595	398
• AP	600	398
• AR	600	398
• GL	600	398
• Inventory	600	398
Series 8		
• GL	750	515
• AP	750	515
• AR	750	515
• Inventory	750	515
• Payroll	750	515
• Sales Inv.	750	515
Peachpak Series 9	500	345
• Mailing List	375	258
• Peachcalc	375	258
• Peachtext	500	345
• Proofreader	300	206
• Telecom	375	258
Series 40 Pak (GL,AR,AP)† [5]	595	398
• AP†	400	275
• AR†	400	275
• GL†	400	275
Series 80 Pak (GL,AR,AP)† [5]	595	398
• AP†	400	275
• AR†	400	275
• GL†	400	275
• Inventory†	400	275
Pearl Soft		
Personal Pearl	295	184
Quality		
GBS—DB Appl Dev Prog	650	488
Select		
Select Word Processor	595	356
Sensible		
Sensible Speller†	125	94
Software Dimensions		
Accounting + AP	645	398
• AR	645	398
• GL	645	398
• Inventory	645	398
• Payroll	645	398
• POE	645	398
• POS	645	398
• SOE	645	398
Sorcim		
ACT 65	175	126
ACT 68	175	126
ACT 69	175	126
ACT 80	175	126
ACT 86/88	175	126
Pascal/M-80	395	284
Pascal/M-86/88	495	356
Supercalc	195	129
Supercalc 2	295	179
Superspellguard	195	129
Superwriter	295	185
Trans 86	175	126
Southwestern		
Z-Term†	100	75
Z-Term Professional†	150	109
Supersoft		
Ada Compiler	300	225
Analiza II	50	38
C Compiler	250	188
Disk Doctor	100	75
Dungeon Master	40	30
Nemesis	45	34
Scratchpad	295	221
T/Maker Co.		
T/Maker III	275	215
Woolf Software Systems		
Move It	125	83

† Apple CP/M only

CP/M-86

	LIST	SALE
Ashton-Tate		
dBASE II [5]	700	398
dBASE II w/User's Guide [5]	729	419
Digital Research		
Pascal MT + 86 w/SPP-86	800	600
XLT 86	150	132
Woolf		
Move It	150	99

HARDWARE (not machine specific)

	LIST	SALE
Amdek		
300 (12" Green) [10]	200	157
300A (12" Amber) [10]	210	167
Color II (RGB) [10]	899	645
Corvus Systems		
6MB Hard Disk [10]	2295	1895
11MB Hard Disk [10]	3295	2695
20MB Hard Disk [10]	4295	3495
Daisywriter		
Printer [10]	1395	1235
Hayes		
Chronograph [5]	249	199
Smartmodem 1200 [5]	699	509
Smartmodem 300 [5]	279	209
NEC		
3530 Spinwriter [10]	2190	1735
7710 Spinwriter [10]	3085	2295
8023 Dot Matrix [10]	695	495
Monitor 12" Green Hi. Res [10]	285	167
Novation		
J Cat	149	119
Smart Cat 103 [4]	259	207
Smart Cat 103/ 212 [4]	595	476
Quadram		
Microfazer	189	160
• In-line serial	330	270
• In-line parallel	225	185

TM = Manufacturer's trademark

TERMS: All prices subject to change without notice and availability. Cashier's check/MO/bank transfer. Allow time for company or personal checks to clear. Prices reflect cash prepaid discount. VISA/MASTERCARD/COD/PO's + 3%. CA residents add sales tax. All sales final for games & special orders.

SHIPPING: $3 per item for UPS surface ($6 for Blue Label) within Continental USA, except where shipping cost is specified in square brackets. UPS does not go to Canada, Alaska, APO's, FPO's; call for ship charge or add 15%—we will refund/credit difference.

RETURNS: Must have authorization number, obtained at 415-324-0305. Unauthorized returns will be refused; damaged goods will be refused. All returns subject to 15% restocking fee. No return after 30 days.

PROGRAMMING INTERNATIONAL
505 HAMILTON AVE. • SUITE 301
PALO ALTO, CA 94301
(800) 222-8811
Inside California
(415) 324-3730

TELEPHONE ORDERS

Outside Calif.: (800) 222-8811
Inside Calif.: (415) 324-3730

Order lines are manned 6:30–6 Monday thru Friday and 9–5 Saturday. Other lines are open 9–5 Monday thru Friday

Technical Support (415) 324-0311
Order Status (415) 324-0306
Sales Manager (415) 324-0305

More CP/M®, APPLE®, IBM PC®, UNIX™, UCSD p-System™ software, hardware, etc: call for quote.

Orders must be **PAID** by **AUGUST 31** to qualify for these special prices

Finding the Location of a Variable in North Star Basic

by Doug J. Anderson and John C. Nash

Programs in Basic may call machine language subroutines to perform special operations not normally provided by the language itself or performed only inefficiently. North Star Basic, which will be considered here, uses the following syntax:

```
line number  LET  variable = CALL(address,number)

e.g.   100 LET T5=CALL(10748,4096)
```

which puts the (decimal) value 4096 in the 8080/Z80 DE register pair and executes a call to the (decimal) address 10748. The value of the HL register pair at the end of the routine is returned in the variable T5.

For many applications, however, we would like to pass the location or address of a particular variable, array or string so that this data could be processed by a machine language subroutine. In our own case, the requirement was for a rapid sort of an integer array. Before the sort could be called, we needed the location of this array in memory. This can only be found if the storage mechanism for data is understood. (This task is accomplished in Microsoft Basic 80 by the VARPTR function, which North Star Basic does not provide to date.)

North Star Basic (version 4.1 of N* DOS, Basic, and FPBasic are considered here) uses memory as outlined in Diagram 1. Some key addresses are:

SBAS—start of BASIC = 02A00H.
ENDBAS—end of BASIC; the pointer EBASP at 02A06H gives this.
ENDRAM—the last memory location in contiguous RAM that Basic may use; the pointer EMEMP at 02A09H gives this.
ENDPROG—the last byte of the program, which is always 01H to conform to N* program storage conventions; the pointer EPRGP contains this address.

Of the above, only EPRGP is awkward to find, since North Star does not list it in their description of Basic or FPBasic. However, it can be found by means of the simple program given in Listing 1. This uses the mechanism by which a program is stored in memory, which is to precede each program statement by its length in bytes. The count includes the length byte, so the last "statement" has just one byte—the length byte—so is stored as 01H. The program in Listing 1 finds the length of the first statement in the program itself in the byte at ENDBAS+1. This length is added to that location to get the start of the next line, and so on until 01H is encountered. The program then searches Basic until this address is encountered.

Doug J. Anderson and John C. Nash, University of Ottawa, Ottawa, Ontario, Canada

Note that different precisions of Basic or the use of FPBasic will change the location of EPRGP.

In North Star Basic, the symbol table follows the program. Data is identified by a letter (upper case) followed, optionally, by a digit (0-9). Arrays are distinguished by the opening parenthesis "(", while strings are explicitly denoted by the $ symbol. The following examples are presented to clarify this:

```
Simple variable      A, T, Z9, S0 (S-zero)

Array                T(, T2(

String               B$, Q1$
```

Simple variables require space to store just one number. In N-digit precision (N is always even in North Star systems), binary-coded decimal storage is used to pack the mantissa in N/2 bytes, with the sign and exponent being placed together in a further byte. The total is thus (N/2)+1 bytes for each number stored, exclusive of any overhead for identification (see below). The exponent is stored in excess 40H notation, i.e., 40H is added to the true exponent, with the (implicit) decimal point preceding the first significant digit of the mantissa.

An array dimensioned by the statement DIM B(3,4,5) requires space for 4*5*6 = 120 such numbers, in addition to administrative overheads. The dimensions must be incremented by one in this calculation because North Star Basic allows a zero index for array elements, i.e., the element B(0,0,0) is permitted, as are B(0,1,1), B(2,0,2), and B(3,3,0).

Strings require one byte to store each character space reserved by the appropriate DIM statement. Thus DIM T$(22) reserves 22 bytes in addition to overhead.

Internally, the mechanism used to access data is the linked list. Each identifier must begin with a letter of the alphabet, and North Star Basic has a table of 26 addresses (pointers) immediately following the program terminator 01H. The first of these pointers gives the location of the first member of the A list, the second pointer that of the first member of the B list, etc. If there are no variables, arrays, or strings beginning with the corresponding letter, the pointer is set to 0000H. Addresses, including these pointers, are stored in the usual 8080/Z80 reverse fashion, i.e., low-order byte, high-order byte.

Suppose the A pointer is non-zero. At the designated address, North Star Basic stores a one-byte code identifying the type of data that follows. This code is split into two hexadecimal digits. The first specifies the type of label: it has value 0 for a simple variable, 2 for an array and 8 for a string. The second hexadecimal digit has value 0-9 for numbered identifiers and value 0FH for un-numbered ones, e.g., Z, A$.

The one-byte code is followed by another two-byte pointer to the next member of the sequence belonging to a given letter of the alphabet. Once again, the address is in 8080 reverse order, and a zero value indicates the end of the linked list.

Following the pointer in the case of a simple variable will be the numerical value itself occupying (N/2)+1 bytes.

In the case of an array, the pointer is followed by a two-byte count, low-order byte first, of the number of bytes of storage for the actual data elements. Call this value C.

After this count a set of two-byte values are stored, which have the dimensions of the array plus one. The extra one is to allow for the possibility of zero indices of the array subscripts. There are (p+1) of these two-byte counts for a p-dimensional array, with the last value being zero (i.e., 00 00H). This allows the interpreter to know how many subscripts are permitted for a given array. The data elements then follow in the order where the last index varies most rapidly. Thus, for a two-dimensional array dimensioned 2 by 3, the elements are stored in the order

```
0,0   0,1   0,2   0,3   1,0   1,1   1,2   1,3   2,0   2,1   2,2   2,3
```

For a string, the pointer to the next member of a letter sequence is followed by two counts of two bytes each (low-order byte first as usual) giving the maximum (or dimensioned) number of characters and the actual number of characters currently stored in the string. The string data follows these counts in ASCII.

In diagram form we have the following byte maps:

—simple variable (total of N/2 + 4 bytes):

```
code  pointerLO pointerHI (N/2 bytes mantissa) (sign/exponent)
```

—p=dimensional array (total of C + 2*(p+3) + 1 bytes to store array and overhead information, where

```
C = (N/2+1) * DIM1 * DIM2 * ... * DIMp
```

is the space for the data itself). Note that here DIMx is the declared x dimension plus one for the zero index:

```
   code  pointerLO pointerHI bytecountLO bytecountHI DIM1LO DIM1HI
DIM2LO DIM2HI ... DIMpLO DIMpHI 00 00 (element 0,0...,0 etc. for a total
of C bytes)
```

—q-character string (DIMensioned for q characters, default is 10 characters if no explicit dimension is given). A total of q+7 bytes are used.

```
   code pointerLO pointerHI DIMLO DIMHI lengthLO lengthHI (q character
spaces)
```

To see how this appears in memory, one can load and run the program given in listing 2. Note that the symbol table and data are created as the program runs. Remember that the assembled symbol finding subroutine must already be in memory before the STORTEST program is RUN, or unpredictable and probably disastrous results are likely to ensue. The program shows how data corresponding to each variable name is stored in memory. A sample output is given in Listing 4.

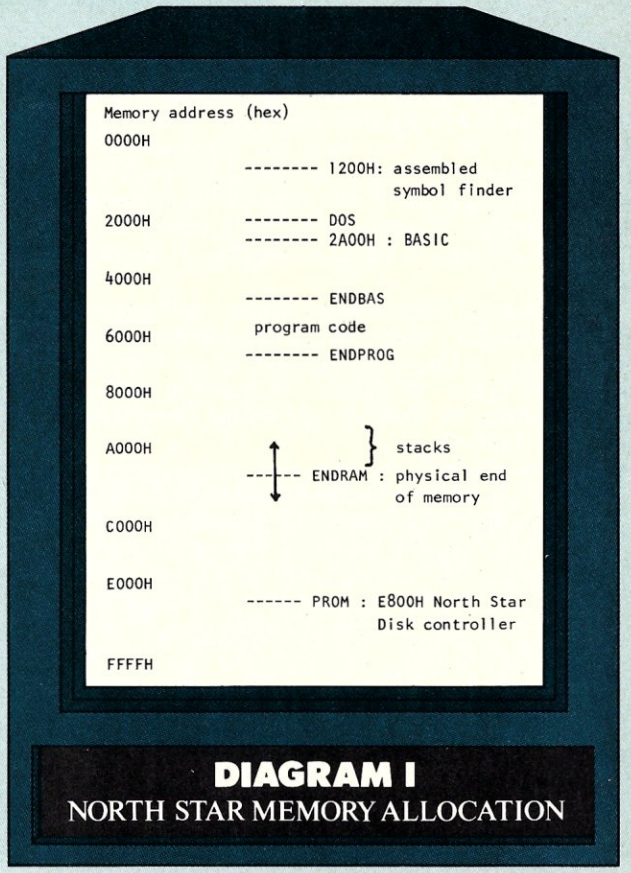


DIAGRAM I
NORTH STAR MEMORY ALLOCATION

Listing 3 gives an 8080 Assembler subroutine which, given the name of a variable, array or string, returns the address of the code byte which begins the storage area for that label. This routine is provided with the name by FILLing (poking) three bytes of information into bytes 4, 5, and 6 of the assembly routine memory area. If the address of the start of the assembly routine is called ALOAD, these bytes and their functions are:

```
ALOAD+3  :  letter of alphabet of name (A - Z)

ALOAD+$  :  blank or 0-9 (secondary part of name)

ALOAD+%  :  blank for simple variable

            (  for array

            $  for string
```

For the most effective use of calls to machine language subroutines, we need to pass not just the value processed by the CALL function, but rather the address of the data string or array processed by the subroutine.

Because the symbol finding routine returns the address of the code byte that starts the data area rather than the data itself, we must add to the returned value the number of bytes of overhead for each variable name. This depends on the type of the variable, and the number of dimensions in the case of an array. This is illustrated in Listing 2 in lines numbered 500 onwards.

To give some idea of the value of calling assembly level code to carry out specific functions, consider the problem of sorting a numerical array (one dimension). The figures in Table 1 compare a Shell-Metzner sort coded in Basic to one called from Basic but coded in Assembler. ■

Table 1. Time in seconds to sort n floating point numbers (randomly generated)

n	Heapsort (Basic)	Shell-Metzner (Assembler)
501	127.6	2.0
1001	283.0	4.4
1501	450.9	7.2
2001	622.3	11.0

Note: The timing ratios are not constant because the algorithms are different in addition to overheads incurred by Basic.

Listing 1

```
LOAD FINDPROG
READY

LIST

10 PRINT "FIND ENDPROG POINTER"
20 REM 810708
30 LET B1=10752 \ REM START BASIC - SBAS = 2A00H
40 LET E1 = B1+6 \ REM EBASP END BASIC POINTER
50 LET E9=EXAM(E1)+256*EXAM(E1+1) \ REM NOTE 8080 INVERSE ORDER
60 LET M1=B1+9 \ REM EMEMP
70 LET E8=EXAM(M1)+256*EXAM(M1+1)
80 PRINT "ENDBAS=",E9,"  ENDRAM=",E8
90 REM WANT ENDPROG, SO SEARCH FOR IT
100 REM SEARCH FOR 01H AS LINE LENGTH
110 LET I=E9+1 \ REM FIRST LINE LENGTH POSITION
120 LET C=EXAM(I)
130 IF C=1 THEN 160 \ REM FOUND IT?
140 LET I=I+C \ REM NEXT  POSITION
150 GOTO 120
160 PRINT "FOUND 01 AT ",I
170 LET I1=INT(I/256)
180 LET I2=I-256*I1 \ REM TWO PARTS OF ADDRESS
190 REM NOW SEARCH BASIC FOR POINTER
200 FOR J=B1 TO E9 \ REM ONLY THIS PART OF MEMORY
210 IF EXAM(J)<>I2 THEN 230 \ REM LOW ORDER BYTE FIRST
220 IF EXAM(J+1)=I1 THEN EXIT 260 \ REM GOTO I1
230 NEXT J
240 PRINT "POINTER NOT FOUND"
250 STOP
260 PRINT "POINTER AT LOCATIONS:",J,J+1
270 DIM A$(16)
280 LET A$="0123456789ABCDEF"
290 LET K=INT(J/(256*16))
300 LET J1=J-256*16*K
310 PRINT "HEX ADDRESS=",A$(K+1,K+1),
320 LET K=INT(J1/256)
330 LET J1=J1-256*K
340 PRINT A$(K+1,K+1),
350 LET K=INT(J1/16)
360 LET J1=J1-16*K
370 PRINT A$(K+1,K+1),A$(J1+1,J1+1)
380 STOP
READY
BYE
*GO FPBASICF,2
READY
LOAD FINDPROG
READY
RUN

FIND ENDPROG POINTER
ENDBAS= 22769  ENDRAM= 57087
FOUND 01 AT  23771
POINTER AT LOCATIONS: 21771 21772
HEX ADDRESS=550B
READY
```

Listing 2

```
LOAD STORTEST
READY

LIST

10 PRINT "STORTEST MARCH 1982"
20 DIM Q(2),Z9(2,3),H$(9),H5$(2),X$(3),S$(16)
30 REM ONLY S$ AND X$ ARE FOR THIS PROGRAM WORK
40 LET S$="0123456789ABCDEF" \ REM HEX CHARACTERS
50 REM REST OF DECLARATIONS ARE PART OF THE TEST
60 LET N=8 \ REM PRECISION
70 LET Y0=0 \ LET Y1=0 \ LET Y2=0 \ LET Y3=0 \ LET Y4=0
80 LET Y5=0 \ LET Y6=0 \ LET Y7=0 \ LET Y8=0 \ LET Y9=0
90 REM DEFINE WORKING STORAGE NOW TO FIX SYMBOL TABLE
100 REM NOW SET SOME DATA IN STORAGE FOR TEST
110 A=23.4
120 A4=.333
130 A$="A$"
140 A3$="A3$"
150 FOR I=0 TO 2
160 Q(I)=I
170 FOR J=1 TO 3
180 Z9(I,J)=I*J
190 IF I=1 THEN Z9(I,J)=-I*J
200 NEXT J
210 NEXT I
220 H$="123456"
230 H5$="H5"
240 REM NOW READY TO FIND DATA AND DISPLAY IT
250 REM AFTER THIS POINT MUST NOT CHANGE ANY VARIABLES
260 REM EXCEPT THOSE BEGINNING WITH LETTER Y
270 REM ELSE SYMBOL TABLE IS ALTERED
280  PRINT "LABEL=",
290 INPUT X$
300 IF LEN(X$)=0 THEN STOP
310 IF LEN(X$)=3 THEN 340
320 LET X$=X$+" "
330 GOTO 310
340 IF X$(1,1)<>"Y" THEN 370
350 PRINT "CANNOT USE NAMES BEGINNING WITH 'Y' HERE"
360 GOTO 280
370 IF X$(2,2)="(" THEN 400
380 IF X$(2,2)="$" THEN 400
390 GOTO 420
400 LET X$(3,3)=X$(2,2)
410 LET X$(2,2)=" "
420 FILL 4611,ASC(X$(1,1))
430 FILL 4612,ASC(X$(2,2))
440 FILL 4613,ASC(X$(3,3))
450 PRINT "CHECK X$:",
460 LET Y=EXAM(4611) \ GOSUB 1410
470 LET Y=EXAM(4612) \ GOSUB 1410
480 LET Y=EXAM(4613) \ GOSUB 1410 \ PRINT
490 Y=CALL(4608) \ REM 1200H IS LOCATION USED
500 PRINT "CODE BYTE FOR ",X$," IN ADDRESS (DEC) ",Y," (HEX) ",
510 LET Y1=Y \ LET Y=INT(Y1/256) \ GOSUB 1410
520 LET Y=Y1-256*Y \ GOSUB 1410 \ PRINT
530 REM NOW LOOK AT THE DATA
540 LET Y=EXAM(Y1) \ REM CODE BYTE
550 PRINT "CONTENTS OF CODE BYTE:",
560 GOSUB 1410 \ PRINT \ REM PRINT IN HEX
570 IF INT(Y/16)=0 THEN 620 \ REM SIMPLE VARIABLE
580 IF INT(Y/16)=2 THEN 900 \ REM ARRAY
590 IF INT(Y/16)=8 THEN 1120 \ REM STRING
```

```
600 PRINT "ERROR - CODE BYTE HAS INVALID CODE!!!"
610 STOP
620 PRINT "SIMPLE VARIABLE  ---"
630 GOSUB 1460
640 LET Y1=Y1+2 \ REM POINTER TO DATA
650 PRINT "DATA AT CODE BYTE ADDR + 2"
660 GOSUB 670 \ GOTO 280
670 PRINT "MANTISSA : ",
680 LET Y3=0
690 FOR Y4=1 TO N/2
700 LET Y=EXAM(Y1+Y4) \ GOSUB 1410
710 LET Y3=Y3*10+INT(Y/16) \ REM FIRST DIGIT
720 LET Y3=Y3*10+Y-16*INT(Y/16) \ REM SECOND DIGIT
730 NEXT Y4
740 PRINT "   SIGN/EXPONENT :",\ LET Y=EXAM(Y1+N/2+1)
750 IF Y<128 THEN 780 \ REM POSITIVE?
760 LET Y3=-Y3 \ REM MAKE VALUE NEGATIVE
770 LET Y=Y-128 \ REM AND ADJUST EXPONENT/SIGN TO LEAVE EXPONENT
780 LET Y5=Y-N-64 \ REM EXPONENT ITSELF ADJUSTED FOR 10^N
790 IF Y5<0 THEN 850
800 IF Y5=0 THEN 880
810 FOR Y6=1 TO Y5
820 LET Y3=Y3*10
830 NEXT Y6
840 GOTO 880
850 FOR Y6=1 TO -Y5
860 LET Y3=Y3*0.1
870 NEXT Y6
880 GOSUB 1410 \ PRINT "  VALUE=",Y3
890 RETURN
900 REM ARRAY
910 GOSUB 1460 \ REM LOOK AT POINTER
920 LET Y1=Y1+2 \ REM BYTECOUNT START
930 LET Y0=EXAM(Y1+1)+256*EXAM(Y1+2)
940 PRINT "BYTE COUNT IN ARRAY=",Y0,"  ",
950 GOSUB 1480
960 LET Y9=0 \ REM DIMENSION COUNT
970 LET Y1=Y1+2 \ REM GET NEXT START FOR DIMENSIONS
980 LET Y9=Y9+1 \ REM ONE MORE DIMENSION
990 LET Y2=EXAM(Y1+1)+256*EXAM(Y1+2)
1000 PRINT "DIMENSION(",Y9,")+1 =",Y2,"  ",
1010 GOSUB 1480
1020 IF Y2>0 THEN 970
1030 PRINT Y9-1," DIMENSIONAL ARRAY"
1040 PRINT "DATA ELEMENTS - LAST DIMENSION VARIES MOST RAPIDLY"
1050 LET Y1=Y1+2 \ REM INITIAL DATA POSN
1060 LET Y0=Y0/(N/2+1) \ REM NUMBER OF ARRAY ELEMENTS
1070 FOR Y9=1 TO Y0
1080 GOSUB 670
1090 LET Y1=Y1+(N/2+1)
1100 NEXT Y9
1110 GOTO 280
1120 REM STRING
1130 GOSUB 1460 \ REM POINTER
1140 LET Y1=Y1+2
1150 LET Y0=EXAM(Y1+1)+256*EXAM(Y1+2)
1160 PRINT "STRING DIMENSION=",Y0,", HEX:",
1170 GOSUB 1480
1180 LET Y1=Y1+2 \ REM LENGTH START - 1
1190 LET Y3=EXAM(Y1+1)+256*EXAM(Y1+2)
1200 PRINT "LENGTH=",Y3,", HEX:",
1210 GOSUB 1480
1220 PRINT "STRING CONTENTS ; '/' MARKS END OF ACTUAL LENGTH"
1230 PRINT "IN ASCII - ONE SPACE BEFORE AND AFTER EACH CHARACTER"
1240 LET Y1=Y1+2 \ REM FIRST CHARACTER ADDRESS MINUS 1
1250 FOR Y9=1 TO Y0
1260 LET Y=EXAM(Y1+Y9)
1270 PRINT " ",CHR$(Y)," ",
1280 IF Y9=Y3 THEN PRINT "/",
1290 IF 25*INT(Y9/25)=Y9 THEN PRINT
1300 NEXT Y9
1310 PRINT
1320 PRINT "IN HEX"
1330 FOR Y9=1 TO Y0
1340 LET Y=EXAM(Y9+Y1)
1350 GOSUB 1410
1360 IF Y9=Y3 THEN PRINT "/",
1370 IF 25*INT(Y9/25)=Y9 THEN PRINT
1380 NEXT Y9
1390 PRINT
1400 GOTO 280
1410 REM PRINT BYTE IN HEX
1420 LET Y7=INT(Y/16)
1430 LET Y8=Y-16*Y7
1440 PRINT " ",S$(Y7+1,Y7+1),S$(Y8+1,Y8+1),
1450 RETURN
1460 REM POINTER TO NEXT SYMBOL TABLE ENTRY
1470 PRINT "POINTER TO NEXT ENTRY IN TABLE :",
1480 LET Y=EXAM(Y1+1)
1490 PRINT "LO BYTE =",\GOSUB 1410
1500 LET Y=EXAM(Y1+2)
1510 PRINT "   HI BYTE =",\GOSUB 1410 \ PRINT
1520 RETURN
READY
```

Listing 3

```
ASSM 1200

1200                    1000 * FIND SYMBOL IN BASIC SYMBOL TABLE
1200                    1010 * FSYM MAR 4 82
1200                    1020 * FOR USE BY BASIC 'CALL'
1200 C3 08 12           1030 FSYM   JMP  GOPT     JUMP OVER STORAGE AREA
1203                    1040 STORE  ASC  'A3$'
1206                    1050 * STORE HAS A-Z CHARACTER
1206                    1060 * STORE + 1 IS BLANK OR HAS DIGIT 0-9
1206                    1070 * STORE + 2 HAS BLANK OR '$' OR '('
1206                    1080 * TO INDICATE SCALAR OR STRING OR ARRAY
1206                    1090 SBAS   EQU  02A00H   START OF BASIC
1206                    1100 EBASP  EQU  02A06H   END OF BASIC POINTER
1206                    1110 EMEMP  EQU  2A09H    END OF MEMORY POINTER
1206                    1120 EPRGP  EQU  550BH    END OF PROG POINTER
1206                    1130 * ALL THE ABOVE FOR DOS 4.1 ORIGIN 2000H
1206                    1140 * CHANGE FOR OTHER DOS VERSIONS
1206                    1150 * CONTENTS OF EPRGP IS ADDRESS OF 01 BYTE AT PROG END
1206                    1160 DOSST  EQU  02000H    START OF DOS
1206                    1170 DOS    EQU  DOSST+028H HOT START POINT FOR DOS
1206                    1180 BLNK   EQU  20H
1206                    1190 LSTK   DS   2
1208 21 00 00           1200 GOPT   LXI  H,0
120B 39                 1210        DAD  SP
120C 22 06 12           1220        SHLD LSTK
120F 31 B6 12           1230        LXI  SP,STK
1212 3A 03 12           1240 START  LDA  STORE    FIRST CHAR
1215 FE 41              1250        CPI  'A'
1217 FA 28 20           1260        JM   DOS      ESCAPE
121A FE 5B              1270        CPI  'Z'+1
121C F2 28 20           1280        JP   DOS      ESCAPE
121F D6 41              1290        SUI  41H      SHIFT FOR ALPHABET
1221 87                 1300        ADD  A        DOUBLE
1222 5F                 1310        MOV  E,A      SAVE IN E
1223 16 00              1320        MVI  D,0      ZERO D
1225 21 0B 55           1330        LXI  H,EPRGP
1228 4E                 1340        MOV  C,M
1229 23                 1350        INX  H
122A 46                 1360        MOV  B,M      B,C NOW HAVE END PROG ADDRESS
122B 03                 1370        INX  B        PLUS 1 FOR TABLE START
122C EB                 1380        XCHG          GET SHIFT IN H,L
122D 09                 1390        DAD  B        ADD IN ADDRESS
122E 5E                 1400        MOV  E,M      GET LOW ORDER BYTE
122F 23                 1410        INX  H
1230 56                 1420        MOV  D,M      HI ORDER BYTE
1231 EB                 1430        XCHG          PUT IN H,L
1232                    1440 * H,L POINTS TO FIRST TABLE ENTRY
1232                    1450 * NOW GET CODE FOR MATCHING
1232 3A 05 12           1460 NEXTO  LDA  STORE+2
1235 FE 24              1470        CPI  '$'      STRING ?
1237 CA 47 12           1480        JZ   STRNG
123A FE 28              1490        CPI  '('      ARRAY ?
123C CA 4C 12           1500        JZ   ARRAY
123F FE 20              1510        CPI  BLNK     SCALAR ?
1241 C2 28 20           1520        JNZ  DOS
1244 C3 51 12           1530        JMP  CODER
1247 06 80              1540 STRNG  MVI  B,80H     8 IN LEFT NYBBLE
1249 C3 53 12           1550        JMP  CONT1
124C 06 20              1560 ARRAY  MVI  B,20H     2 IN LEFT NYBBLE
124E C3 53 12           1570        JMP  CONT1
1251 06 00              1580 CODER  MVI  B,0       0 IN LEFT NYBBLE
1253 3A 04 12           1590 CONT1  LDA  STORE+1   SECOND BYTE
1256 FE 20              1600        CPI  BLNK     BLANK ?
1258 CA 6A 12           1610        JZ   CONT2
125B FE 30              1620        CPI  '0'      TEST IN RANGE
125D FA 28 20           1630        JM   DOS
1260 FE 3A              1640        CPI  '9'+1
1262 F2 28 20           1650        JP   DOS
1265 D6 30              1660        SUI  30H      SHIFT FOR ASCII
1267 C3 6C 12           1670        JMP  CONT3
126A 3E 0F              1680 CONT2  MVI  A,0FH     SCALAR CODE
126C 80                 1690 CONT3  ADD  B
126D 32 96 12           1700        STA  CODE
1270                    1710 * NOW SEARCH
```

North Star Variables continued . . .

```
1270 3A 96 12            1720 SRCH   LDA   CODE       GET CODE
1273 BE                  1730        CMP   M          TEST FOR MATCH
1274 CA 8F 12            1740        JZ    DONE       FOUND IT?
1277 23                  1750        INX   H          NEXT BYTE
1278 5E                  1760        MOV   E,M        LOW ORDER BYTE
1279 23                  1770        INX   H
127A 56                  1780        MOV   D,M        HI ORDER BYTE
127B EB                  1790        XCHG             PUT NEW ADDRESS IN H,L
127C 7C                  1800        MOV   A,H
127D FE 00               1810        CPI   0          ZERO?
127F C2 88 12            1820        JNZ   SAFE
1282 7D                  1830        MOV   A,L
1283 FE 00               1840        CPI   0
1285 CA 28 20            1850        JZ    DOS
1288 0D                  1860 SAFE   DCR   C
1289 CA 28 20            1870        JZ    DOS
128C C3 70 12            1880        JMP   SRCH       TRY AGAIN
128F EB                  1890 DONE   XCHG              SAVE HL WHILE RESET!
1290 2A 06 12            1900        LHLD LSTK
1293 F9                  1910        SPHL             RESET STACK
1294 EB                  1920        XCHG             RECOVER FOR RETURN
1295 C9                  1930        RET
1296                     1940 STK    EQU   $+32       STACK FOR THIS PROG
1296                     1950 CODE   DS    1
PS
FSYM  1200     STORE 1203     SBAS  2A00     EBASP 2A06
EMEMP 2A09     EPRGP 550B     DOSST 2000     DOS   2028
BLNK  0020     LSTK  1206     GOPT  1208     START 1212
NEXT0 1232     STRNG 1247     ARRAY 124C     CODER 1251
CONT1 1253     CONT2 126A     CONT3 126C     SRCH  1270
SAFE  1288     DONE  128F     STK   12B6     CODE  1296
EXEC 2028
*GO FPBASIC,2
READY
```

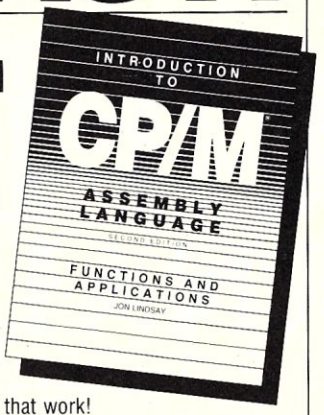
Listing 4

```
STORTEST MARCH 1982
LABEL=?H$
CHECK X$: 48 20 24
CODE BYTE FOR H $ IN ADDRESS (DEC)  26676 ,(HEX)  68 34
CONTENTS OF CODE BYTE: 8F
POINTER TO NEXT ENTRY IN TABLE :LO BYTE = 44   HI BYTE = 68
STRING DIMENSION= 9, HEX:LO BYTE = 09   HI BYTE = 00
LENGTH= 6, HEX:LO BYTE = 06   HI BYTE = 00
STRING CONTENTS ; '/' MARKS END OF ACTUAL LENGTH
IN ASCII - ONE SPACE BEFORE AND AFTER EACH CHARACTER
 1  2  3  4  5  6 /
IN HEX
 31 32 33 34 35 36/ 20 20 20
LABEL=?A
CHECK X$: 41 20 20
CODE BYTE FOR A   IN ADDRESS (DEC)  26822 ,(HEX)  68 C6
CONTENTS OF CODE BYTE: 0F
SIMPLE VARIABLE  ---
POINTER TO NEXT ENTRY IN TABLE :LO BYTE = CE   HI BYTE = 68
DATA AT CODE BYTE ADDR + 2
MANTISSA :  23 40 00 00   SIGN/EXPONENT : 42  VALUE= 23.4
LABEL=?Z9(
CHECK X$: 5A 39 28
CODE BYTE FOR Z9( IN ADDRESS (DEC)  26605 ,(HEX)  67 ED
CONTENTS OF CODE BYTE: 29
POINTER TO NEXT ENTRY IN TABLE :LO BYTE = 00   HI BYTE = 00
BYTE COUNT IN ARRAY= 60  LO BYTE = 3C   HI BYTE = 00
DIMENSION( 1)+1 = 3  LO BYTE = 03    HI BYTE = 00
DIMENSION( 2)+1 = 4  LO BYTE = 04    HI BYTE = 00
DIMENSION( 3)+1 = 0  LO BYTE = 00    HI BYTE = 00
 2 DIMENSIONAL ARRAY
DATA ELEMENTS - LAST DIMENSION VARIES MOST RAPIDLY
MANTISSA :  00 00 00 00   SIGN/EXPONENT : 00  VALUE= 0
MANTISSA :  00 00 00 00   SIGN/EXPONENT : 00  VALUE= 0
MANTISSA :  00 00 00 00   SIGN/EXPONENT : 00  VALUE= 0
MANTISSA :  00 00 00 00   SIGN/EXPONENT : 00  VALUE= 0
MANTISSA :  00 00 00 00   SIGN/EXPONENT : 00  VALUE= 0
MANTISSA :  10 00 00 00   SIGN/EXPONENT : 41  VALUE= -1
MANTISSA :  20 00 00 00   SIGN/EXPONENT : 41  VALUE= -2
MANTISSA :  30 00 00 00   SIGN/EXPONENT : 41  VALUE= -3
MANTISSA :  00 00 00 00   SIGN/EXPONENT : 00  VALUE= 0
MANTISSA :  20 00 00 00   SIGN/EXPONENT : 41  VALUE= 2
MANTISSA :  40 00 00 00   SIGN/EXPONENT : 41  VALUE= 4
MANTISSA :  60 00 00 00   SIGN/EXPONENT : 41  VALUE= 6
LABEL=?Y
CANNOT USE NAMES BEGINNING WITH 'Y' HERE
LABEL=?H0
CHECK X$: 48 30 20
*
```

AT LAST!

A NEW BOOK DEALING WITH **ASSEMBLY LANGUAGE FOR CP/M®** SYSTEM USERS

CP/M is a registered trademark of Digital Research, Inc.



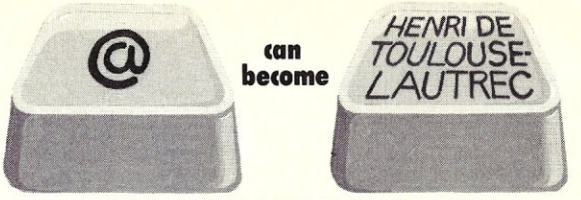
Introduction to CP/M Assembly Language is a step-by-step instruction manual on how to construct simple programs operating in CP/M that work! The full size (8½x11) perfect-bound 180 pages present the reader with various ways of inputting and outputting data to a terminal, as well as sending data to a line printer. Since the I/O methods used are based on CP/M function constructions, the programs are meant to be interchangeable with "standard" CP/M systems. The reader can immediately "talk" with his computer in assembly language. **Only minimal prior exposure to assembly language is required.** Two useful programs are constructed to demonstrate CP/M techniques: **Single-drive copy program** *(sequential filing)*; **Data base program** *(random access filing)*. Also included is a simple game program demonstrating some basic game programming techniques and a short section on program troubleshooting.

TO ORDER SEND: $15.95 PLUS $1.25 FOR POSTAGE AND HANDLING (EUROPE—ADD $7.00 FOR POSTAGE) (CALIFORNIA RES. ADD 6% SALES TAX) TO:

EXECUTIVE COMPUTER

DEPT. G, P.O. BOX 222178, CARMEL, CA 93922, (408) 272-4800

CIRCLE 92 ON READER SERVICE CARD



IMPOSSIBLE? NOT WITH **SMARTKEY!™**

SMARTKEY™ is a unique utility that can redefine any ASCII character or function key to become anything you want. For example, "#" becomes "pip b:=a:*.pas[v]". With a single stroke, a key can represent a chosen character or string at the system level or within a program. Instantly. Without rewiring or soldering.

SMARTKEY™ is completely transparent to the user. It resides on the top of memory and intercepts calls to the BIOS, translating system input to whatever you desire. You can even change a key definition while another program, such as WordStar™, is in operation . . . without interruption! It's perfect for programming, data entry or word processing.

"EXCELLENT" InfoWorld
"VERSATILE AND RELIABLE" Lifelines
"WORKS LIKE A CHARM" Microsystems

Think of the acceleration in productivity. Think of the versatility in keyboard layouts. Think of the possibilities. And, best of all, **SMARTKEY™** is only $60.

Ask about **SMARTPRINT™, SMARTSCREEN™, SPOOL™** and other programs.

To order or obtain more information, call or write to:

HERITAGE SOFTWARE, INC.

2130 S. Vermont Ave., Los Angeles, CA 90007/(213) 737-7252

SMARTKEY™ is compatible with all standard versions of CP/M.™
Programs copyrighted by FBN Software.
WordStar™ is a registered trademark of MicroPro, Inc.
CP/M™ is a trademark of Digital Research.

CIRCLE 95 ON READER SERVICE CARD

TECHTYPE

You need to write по-русски? ?עברית ¿En español?
Or worse yet --

$$P_{nm}(\cos\theta) = \frac{\sin^m\theta}{2^n\, n!}\,\frac{d^{n+m}\,(n^2-1)^n}{dn^{n+m}} \quad ?$$

Your present word processing system isn't exactly a polyglot and flunks algebra? What's the solution?

TECHTYPE™ is a text-formatting system designed especially for *scientific, engineering, mathematical,* and *multi-lingual* document preparation. **TECHTYPE** runs under CP/M® and is adaptable to most hardware. By using your present editor and its three programs

- DISPLAY - Preview on CRT screen
- DRAFT - High-speed dot-matrix printout
- PRINT - High-quality daisywheel printout

you can spend more of your time *solving* equations instead of typing them.

TECHTYPE's capabilities include:

- Multiple type fonts
- Multipass printing
- Unlimited sub/superscript levels
- Control of format, font, pitch, and emphasis.

Multipass printing allows the use of up to *ten* different fonts with only *one* printwheel change per page per font. Price $300.

GREEN MOUNTAIN RADIO RESEARCH COMPANY
240 Staniford Road
Burlington, Vermont 05401 U.S.A.
802-862-0997

CIRCLE 40 ON READER SERVICE CARD

MCP

MAKES HIGH PERFORMANCE S-100 AFFORDABLE

MCP 64KS (64K X 8) STATIC S-100 MEMORY
Sixty-four k bytes of low power high performance (2k x 8) 150ns static ram for systems requiring the ultimate in reliable performance. Extended address capability to one megabyte is provided.
ASSEMBLED AND TESTED LIST $390.00 **YOUR PRICE $238.00**

HIGH PERFORMANCE MCP/SCC-6 Z80B 6MHz S-100 CPU
Operates at 2,4, or 6MHz. Includes two serial ports using 8251 usarts with selectable baud rates from 50 to 19.2k, Three programmable 8 bit parallel ports, 1k onboard static ram, and 2716 Eprom with monitor. Onboard ram and eprom can be disabled on system boot.
ASSEMBLED AND TESTED LIST $350.00 **YOUR PRICE $200.00**

MCP/FDC DOUBLE DENSITY 8" S-100 FLOPPY DISK CONTROLLER
Utilizes the WD1793 double density F/D controller chip and the WD1691 floppy support logic chip. The MCP/FDC offers superb performance and reliability due to enhanced data recovery and write precomp support. The MCP/FDC will support up to four 8" single or double sided Shugart, Qume, Siemens or compatible drives. CP/M 2.2 Bios supplied on standard density 8" diskette. CP/M86 and MS-DOS operating systems with MCP/FDC support available. CP/M 2.2 Bios with MCP/FDC and Miniscribe 5.25 winchester using Xcomp hard disk controller is also available.
ASSEMBLED AND TESTED LIST $288.00 **YOUR PRICE $200.00**

CP/M 2.2 CONFIGURED FOR MCP S-100 BOARDS $130.00

COMPLETE MAINFRAMES SEVEN S-100 SLOTS CONFIGURED READY TO RUN.
Six MHz MCP/SCC-6 Z80B, with serial and parallel ports, MCP/FDC double density disk controller, MCP 64ks static memory and CP/M 2.2.

WITH TWO 8" SSDD FLOPPYS **$2495.00**
WITH TWO 8" DSDD FLOPPYS **$2695.00**
WITH 8" SSDD FLOPPY & 5.25 WINCHESTER (7.6MB FORMATED) **$3595.00**
WITH 8" DSDD FLOPPY & 5.25 WINCHESTER (15.7MB FORMATED) **$4795.00**

MCP
COMPUTER PRODUCTS

6992 El Camino Real - Suite 104-444
Carlsbad, CA 92008
Phone: (619) 438-3270

MASTER CHARGE AND VISA ACCEPTED
CP/M IS A REGISTERED TRADEMARK OF DIGITAL RESEARCH

CIRCLE 11 ON READER SERVICE CARD

If your thirst for technology is combined with an interest in the IBM Personal

A N N O U N C I N G

PC TECH JOURNAL

FOR IBM PERSONAL COMPUTER USERS

To The Limits... And Beyond

As you well know, the IBM Personal Computer is proving to be the success story of the 1980s. Only 18 months after its introduction, the IBM Personal Computer commands almost a fifth of all personal computer sales—a remarkable achievement. At present the number of IBM Personal Computers in users' hands exceeds a quarter-million, with perhaps as many as half a million on the way in the next 12 months.

But despite its widespread public acceptance and the wealth of products now available to support it, we are convinced that *vast areas of its potential remain unexplored,* not to mention those products waiting to be introduced for which the IBM Personal Computer is the precursor.

For those serious enough about microcomputers to accept the challenge, the IBM Personal Computer represents a host of untapped possibilities.

For those with a genuine appetite for technology, there are ways of stretching the IBM Personal Computer to the very limits of its specifications. Ways to willfully expand those specifications...to throw light into the dark recesses of its complex software systems. Ways to enlarge our understanding of the equipment and to *increase its usefulness.*

If this sounds appealing, then *PC Tech Journal* is for you. For it will be the definitive source of authoritative, technically-satisfying information about your IBM Personal Computer, its future transformations and the host of products developed to support it.

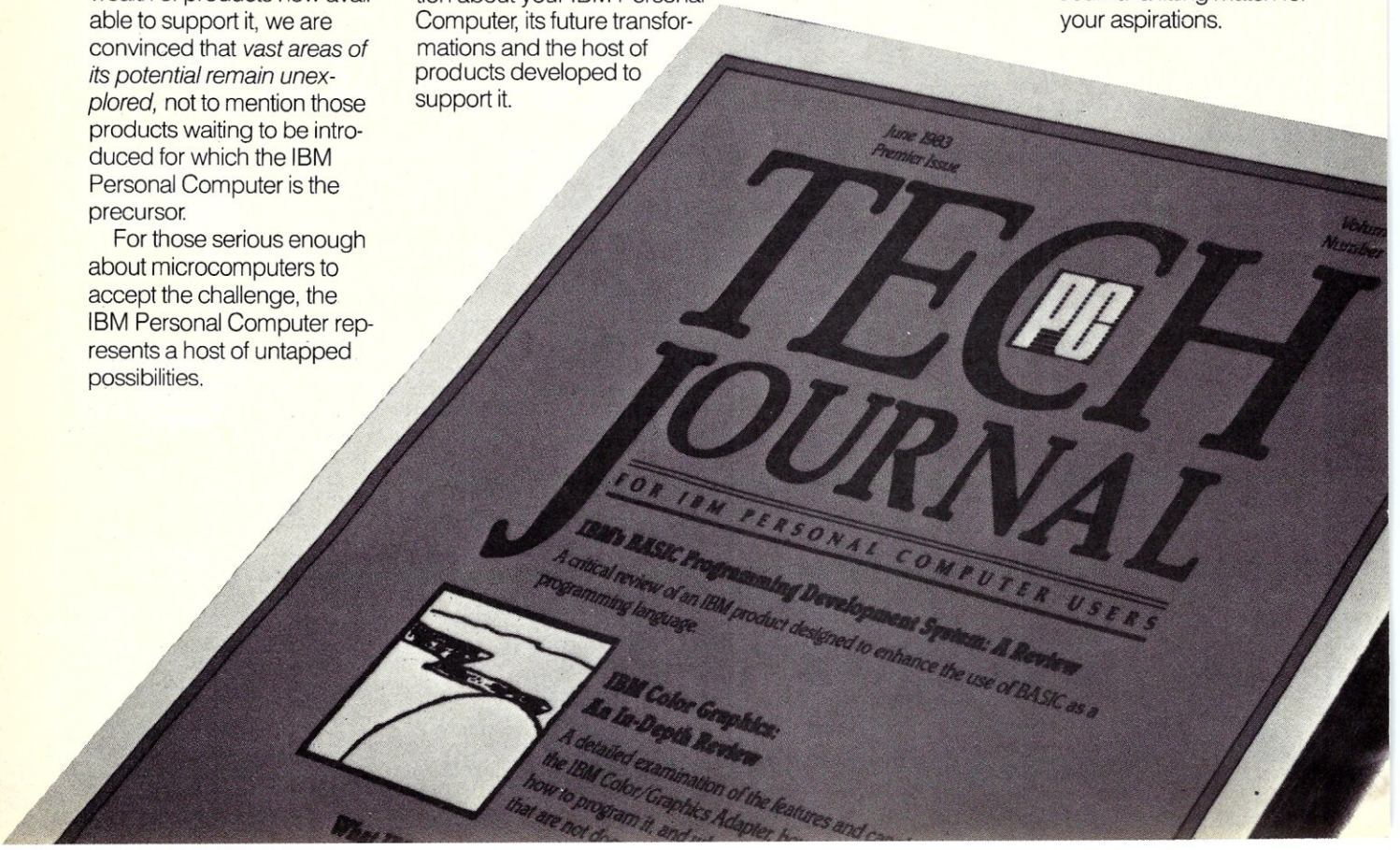
Why a Technical Journal?

The readership of *PC* Magazine has, in the last year, grown at a rate to match the success of the IBM Personal Computer itself. But as the number of committed readers has expanded, so has that portion of users who are not only interested in *output*—that is, using the IBM Personal Computer to better solve problems—but who care deeply about the *process* of the computer itself.

We consider you to be among that select group of readers. Your relationship with microcomputers, vocational or otherwise, has led you to levels of technical competence and curiosity which go far beyond problem-solving. In effect, you have a compelling interest not only in *what* your IBM Personal Computer does, but also *how* it does it and how it can do more.

For you, technological sophistication is a welcome challenge, perhaps even something of an adventure.

In both its depth and breadth of applicable sophistication, we are confident that you will find *PC Tech Journal* a fitting match for your aspirations.



Computer you will want to take part in exploring its vast technological landscape.

Enroll now as a Charter Subscriber at a special Charter rate ...and get your own copy of our Premiere Issue.

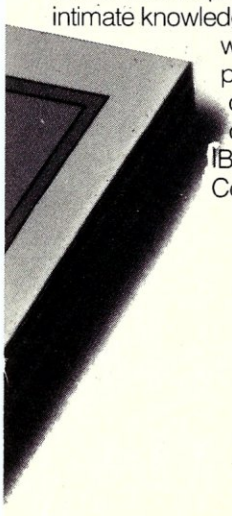
A Select Audience

Needless to say, *PC Tech Journal* is not for everyone. But among *PC* Magazine's current audience is a sizable segment where *PC Tech Journal* will find an active, committed readership:

Practicing professional developers who will want to read the *Tech Journal* for information they can't find anywhere else.

Serious programmers—both full-time professionals and non-professionals alike—with a clear interest in expanding the range and utility of software compatible with the IBM Personal Computer.

Management and staff serving in MIS and DP roles with a requirement for intimate knowledge of the working processes of their companies' IBM Personal Computers.



Independent businessmen and professionals who understand the increased benefits of tailoring their IBM Personal Computers to their own specific demands.

Members of the scientific and academic community whose ability and training allow them to enhance their use of their IBM Personal Computer.

And, of course, anyone with a genuine appetite for state-of-the-art information about the emerging possibilities presented by the IBM Personal Computer.

A Menu for Technical Gourmets

Under the leadership of Editor-in-Chief Will Fastie, author of the widely respected "IBM Images" column in Creative Computing Magazine, we've assembled a staff and a stable of regular contributors all with just the sort of demonstrated expertise you'd expect from *PC Tech Journal.*

Every bimonthly issue will be filled with useful articles on subjects such as...

- *How to fully exploit—or, if necessary, avoid—the complexities of BIOS, the IBM Personal Computer's installed firmware.*
- *Detailed examinations of component hardware available for the IBM Personal Computer, including multifunction cards, display devices, laboratory data acquisition equipment and voice recognition and synthesis devices.*
- *Elegant programming methodologies which exploit the IBM Personal Computer's various operating system environments, plus reviews of new operating environments.*
- *Evaluation of the IBM Personal Computer's communications protocols and standards, including a detailed discussion of computer-to-computer interconnection, terminal emulation and distributed data processing.*
- *Plus a range of articles on networks, mass storage and important user development tools.*

Moreover, as *PC Tech Journal* evolves, it is our explicit intent that you, our readers, play an important part in shaping the magazine. For we are convinced that it will be you who shape the future usefulness of the IBM Personal Computer itself.

To that end, we will make a special effort to involve you in the focus of the magazine, listening to your needs and together forging workable technical solutions to fully take advantage of IBM's revolutionary Personal Computer.

Charter Subscription Offer

Just fill in and mail the coupon or the attached card. We'll enroll you as a Charter Subscriber, send your Premiere Issue and bill you at the special introductory price—a discount of up to 33% (based on the full one-year subscription price.)

PC TECH JOURNAL
P.O. Box 598 Morris Plains, New Jersey 07950

8H07

☐ Yes. Please enroll me as a Charter Subscriber to *PC Tech Journal.*

☐ One year (6 issues) only $19.97—20% off!
☐ Two years only $36.97—26% off!
☐ Three years only $49.97—33% off!

Savings based on full one-year subscription price of $24.97.
CHECK ONE: ☐ Payment enclosed. ☐ Bill me later.

Mr./Mrs./Ms. _____
(please print full name)

Company _____

Address _____

City _____

State _____ Zip _____

Preventing System Crashes During Warm Boot

by Michael F. Sasso

You have a CP/M 2.2 system up and running, and you need to change the A disk. You remove the current disk, insert the new disk, and type control-C to perform the required warm boot. The disk indicator light comes on, then goes off, but no prompt is displayed. The new disk was not a system disk, and the system has crashed.

Don't be surprised if the above sounds familiar. It was happening to me more often than I care to admit. Then I realized that there are not that many causes of system crashes during warm boot, and that they can easily be detected in time to prevent a crash from occurring.

There are three principal reasons for warm boot failure. These are:

1. A disk error occurs while reading the system from the new disk;
2. The disk being booted is not a system disk; and
3. Although the disk being booted is a system disk, the system contained on the disk is not configured for the same memory size as the system that is running.

The first case is usually detected by checking disk status after each disk read operation; in the event of an error, the user is given the opportunity to remedy the situation before any harm is done to the system. In the other cases, however, the nonexistent or incompatible system is loaded and jumped to, causing a system crash. I will describe a procedure for detecting those other cases and requesting operator intervention before the system is destroyed.

My method is based on the fact that the CCP, which is the first part of the system to be reloaded during a warm boot, begins as follows:

```
CCP:  JMP    CCP+35CH
      JMP    CCP+358H
```

We can decide both if the new disk is a system disk and if the system configuration is correct during a warm boot by loading only the first sector of the CCP and examining the data that have been loaded. If the sector begins with two jump instructions, we can be fairly certain that the disk is a system disk. If the address contained in the first jump instruction equals the address of CCP+35CH for the current system, we can conclude that the configuration size is correct. If both conditions are satisfied, we can safely load the rest of the CCP and BDOS and branch to the CCP. Otherwise we will *remain in the BIOS* and inform the operator of the problem. The operator must then provide a new disk that will boot properly, and the boot can be retried.

The code presented in Listing 1 will do the loading and testing described above. Subroutines SETDMA, SELDSK, SETTRK, SETSEC, READ and CONIN are BIOS entry points that work as described in the *CP/M 2 Alteration Guide* from Digital Research. Subroutine PRMSG is assumed to be a routine to send a message whose starting address is contained in HL to the console. It is shown in Appendix B of the *Alteration Guide*. The example assumes that track 0 sector 1 of the disk is reserved for a boot loader, and the sector to be loaded and tested is track 0 sector 2. This will not be true for all systems; be sure to test the appropriate sector for your own system. In order to avoid cluttering up the example unnecessarily, the sector to be tested is assumed to read correctly; however, a test for a read error should be added (see the example in Listing 1).

I still insist on trying to warm-boot disks that aren't system disks. Now, however, I get an error message and a chance to correct the situation instead of a guaranteed system crash.

Michael F. Sasso, 6 Niles St., Brighton, MA 02135

Michael F. Sasso is a software engineer who is working as an independent consultant in the Boston area; his most recent projects include CP/M networking systems. Michael Sasso received a B.A. in Math from Case Western Reserve University and is working toward an M.A.S. in Information Science from Boston University.

```
; Place this code at or near the beginning of the warm boot routine
; in the BIOS, just before the point where the system is reloaded.
;
;
; Read the first sector of the CCP
TESTDISK:
        MVI     C,0
        CALL    SELDSK  ; Select the A-disk
        LXI     B,0
        CALL    SETTRK  ; Select track 0
        LXI     B,2
        CALL    SETSEC  ; Select sector 2
        LXI     B,CCP
```

```
        CALL    SETDMA  ; Set the DMA address to the start of the CCP
        CALL    READ    ; Read the first sector of the CCP
;       insert test for read error here
;
; Test for system disk
        LXI     H,NOSYS ; Get address of error message, just in case
        LDA     CCP
        CPI     0C3H    ; Does CCP start with a jump?
        JNZ     DSKERR  ; Branch to error routine if not
        LDA     CCP+3
        CPI     0C3H    ; Is there also a jump at CCP+3?
        JNZ     DSKERR  ; Branch to error routine if not
;
; Disk is system disk.  Test for configuration match.
        LHLD    CCP+1   ; Get address to which new system jumps
        XCHG            ; Move it to DE
        LXI     B,CCP+35CH ; Address to which it should jump
        LXI     H,SIZERR ; Get address of error message, just in case
        MOV     A,B
        CMP     D
        JNZ     DSKERR
        MOV     A,C
        CMP     E
        JZ      DISKOKAY ; If BC = DE we have configuration match
;
; Error routine. Display a message and wait for the operator to type
; a key on the keyboard. Then retry the warm boot. The address of
; the message is contained in HL.
DSKERR:
        CALL    PRMSG   ; Go display error message
        CALL    CONIN   ; Wait for operator to hit a key
        JMP     TESTDISK ; Go try again
;
; Error messages
NOSYS:
        DB      CR,LF
        DB      'Not a system disk'
        DB      CR,LF,0
;
SIZERR:
        DB      CR,LF
        DB      'Configuration size mismatch'
        DB      CR,LF,0
;
; If this point is reached it has been determined that the A-disk is
; a system disk that is configured for the same memory size as the
; running system.  The warm boot can be completed.
;
DISKOKAY:
        .
        .
        .
```

Enhancing Your Spreadsheet Print Files

by Tom Wiens

Far be it for me to criticize the spreadsheet programs, which have brought liberation to the number cruncher from the most tedious aspects of his routine, just as automation and robots have done for the factory worker. Well, almost: there is still a stage between the creation of a financial model and its presentation in finished form, where I must take to the word processor (or hand it to a secretary) to apply a bit of polish. Aside from adding headings and footings, lengthening labels, adding blank lines, and underlining, numbers must be aligned, zeroes must be added before decimal points, commas must be added to numbers greater than 999, and the number of significant digits must be adjusted so that the results appear no more precise than my assumptions justify. Perhaps someday new versions of spreadsheet programs will do more of this for me, but in the meantime some drudgery remains which ought to be done by a robot of modest intelligence. Being a lazy number cruncher, I couldn't wait, and so I developed a short program in MBasic (given in Listing 1) that may save you some tedium.

Program functions

Its primary purpose is really to round the numbers in tables to the extent warranted by their imprecision. It takes as input a disk file such as a spreadsheet .PRN file, or any other table for that matter, processes it according to the user's instructions, and returns to disk B: a file with the extension .RND. You can choose automatic or manual operation. If the first, you can relax over a cup of coffee while it flashes each line of the table at you, in original and rounded form. If the prospect of a mangled table makes you nervous, you can choose "manual." It will pause after the display of each line, awaiting your approval (a carriage return), a signal not to mess with all following lines (ESCAPE), and/or your editing. As the input file is not erased, nervousness is unwarranted.

As pattern recognition is not my computer's forte, the instructions it is given required some initial thought. Some of these are built into the program: It is told to ignore numbers between 1950 and 2000 with no decimal point or comma, for obvious reasons (statements 440–450). It will place a zero before a decimal point if one is omitted, will add one comma to numbers exceeding 1,000 (but ignore the millions place). It will cut out unnecessary zeroes to the right of the decimal place and drop the decimal point if unneeded. It ignores numbers that evaluate to zero (at least most of the time!), so that dimensional statements like '000s don't get mangled.

On the other hand, some decisions are better left to the user. Column and row headings often contain numbers that should not be rounded, so the user is asked how many lines to skip at the beginning of the file and how many columns to skip from the left of the table ("width of row label field"). The user may choose the rounding target: 5, 10, 50, or 100 are permissible alternatives. If T is the target, the program examines the rightmost digit (or two digits if T > 10) to decide whether to round up or down to the nearest target value. It does not care whether the digits are before or after the decimal point, so if T = 10, 156 will round to 160, 1.56 to 1.60, 156.00 to 156.00. This is a bit rigid, so the user may also choose a "minimum size in digits for rounding," so that if numbers like 1567 and 0.7 exist in the same table, the latter could be left as is.

Indeed, by making the minimum size greater than any number in the file, the rounding function can be skipped entirely (if, for instance, you just want to insert commas). Also, the user might choose to ignore numbers with decimal points, or to truncate all digits to the right of the decimal point after rounding. The program will fill in the extra spaces required to keep decimal points aligned where they were in the first place.

If the variety of options has you lost, study Listing 2 to see what the program does with a sample table. I should also warn that the program sometimes hits a case it can't handle properly; if so, run it again on "manual" and make the correction yourself.

How the program works

A word about how the program operates: A line of the input file is read into a string A$. From left to write, it searches for the beginning of a number and, when it finds one, identifies its end and all components, including signs, comma, decimal point, and numbers of digits on either side. The number itself, with all its paraphenalia removed, is converted from string D$ to an integer. The program tests for various exceptions, but otherwise rounds the integer appropriately, and finally puts Humpty Dumpty together again. The string D$ that contains the rounded number is concatenated onto the right of storage string B$, and the process continues until all of A$ has been processed. After displaying and allowing user changes (if not on auto), B$ is written to disk.

The section of the program beginning at line 790 is a line editor that employs the screen editing and positioning controls of the Osborne I or Televideo 920C terminal. The control codes corresponding to the arrow keys (lines 810–850) and the sequence

<ESCAPE> = yposition xposition

used to position the cursor (lines 760–770) may be adjusted for terminals with different screen control sequences. They should function as follows: when the original and rounded lines are displayed, the cursor is left positioned at the beginning of the line. Right and

Tom Wiens, 2025 Westwood Terrace, Vienna, VA 22180

Spreadsheets continued . . .

left arrows function to move the cursor, an up arrow to insert a space, and a down arrow to delete a character. A carriage return signals that the editing of the line has been completed, and the Escape key causes the remaining (up to 66) lines to be skipped (to prevent rounding of numbers in table footings).

```
1 REM ************************************
2 REM ***  NUMERICAL ROUNDING PROGRAM ***
3 REM ***       by Thomas B. Wiens     ***
4 REM ************************************
10 DEFDBL N
20 PRINT CHR$(26)
30 PRINT "TABLE ROUNDING PROGRAM"
40 PRINT "  by thomas b. wiens"
50 INPUT "FILENAME (DISK B: ASSUMED)";F$
60 OPEN "I",1,"B:"+F$
70 F$=LEFT$(F$,LEN(F$)-4)
80 INPUT "AUTOMATIC OPERATION (Y/N)";ATO$
90 INPUT "SKIP FIRST x LINES:  ENTER x";L1
100 INPUT "WIDTH OF ROW LABEL FIELD";L2
110 INPUT "ROUNDING TARGET (5,10,50, OR 100)";T
120 ROUNDLEN=1
130 IF T>10 THEN ROUNDLEN=2
140 INPUT "MINIMUM SIZE IN DIGITS FOR ROUNDING (0,1,2,3)";D
150 INPUT "IGNORE NUMBERS WITH DECIMALS (Y/N)";IGDEC$
160 IF IGDEC$="Y" THEN NODEC$="N":GOTO 190
170 PRINT "TRUNCATE ALL TO RIGHT OF DECIMAL"
180 INPUT "AFTER ROUNDING (Y/N)";NODEC$
190 OPEN "O",2,"B:"+F$+".RND"
200 IF EOF(1) THEN 920
210 FOR I=1 TO L1: LINE INPUT #1,A$:B$=A$:PRINT #2,B$:NEXT I
220 IF EOF(1) THEN 920
230 LINE INPUT #1,A$
240 B$=""
250 FOR I=1 TO LEN(A$)
260 REM FIND FIRST NUMBER
270 C$=MID$(A$,I,1)
280 IF I<L2 THEN B$=B$+C$:GOTO 710   :REM SKIP L2 COLUMNS
290 A=ASC(C$) : REM ASCII VALUE OF ITH CHARACTER
300 IF NUMBER THEN 330
310 IF A<48 OR A>57 THEN B$=B$+C$:GOTO 710    :REM TO RETURN
320 NUMBER=-1:D$=D$+C$:GOTO 710   :REM TO RETURN WITH SWITCH SET
330 IF A=44 THEN COMMA$=",":COMLEN=LEN(D$):GOTO 710
340 IF A>47 AND A<58 THEN D$=D$+C$:GOTO 710    :REM TO RETURN
350 IF A=46 THEN DEC$=".":LEFT=LEN(D$):GOTO 710    :REM TO RETURN
355 IF RIGHT$(B$,1)="-" THEN B$=LEFT$(B$,LEN(B$)-1):D$="-"+D$
360 IF RIGHT$(B$,1)<>"." THEN 380
370 B$=LEFT$(B$,LEN(B$)-2):D$="0"+D$:LEFT=1:DEC$="."
380 REM END OF NUMBER REACHED
390 NUMLEN=LEN(D$):NUM=VAL(D$)
400 IF LEFT=0 THEN LEFT=NUMLEN
410 RIGHTDIG=VAL(RIGHT$(D$,ROUNDLEN)): REM RIGHTMOST DIGITS
420 IF NUM=0 THEN 680: REM IGNORE DIMENSIONAL STATEMENTS LIKE '000
430 IF NUMLEN<D THEN 510      :REM NUMBER TOO SMALL FOR ROUNDING/REASSEMBLE IT
440 IF COMMA$="," OR DEC$="." THEN 460
450 IF NUM>1950 AND NUM<2000 THEN 680: REM DON'T ROUND YEARS
460 IF IGDEC$="Y" AND DEC$="." THEN 510
470 R=RIGHTDIG/T
480 IF R<.5 THEN NUM=NUM-RIGHTDIG:GOTO 510      :REM REASSEMBLE NUMBER
490 IF R>=.5 AND R<1.5 THEN NUM=NUM-RIGHTDIG+T: GOTO 510     :REM REASSEMBLE
500 IF R>=1.5 THEN NUM=NUM+2*T-RIGHTDIG
510 REM REASSEMBLE NUMBER
520 H$=STR$(NUM)
530 IF NUMLEN>=LEN(H$)-1 THEN 560
540 NUMLEN=NUMLEN+1:LEFT=LEFT+1
550 B$=LEFT$(B$,LEN(B$)-1)
560 H$=LEFT$(D$,NUMLEN-LEN(H$)+1)+RIGHT$(H$,LEN(H$)-1)
570 IF LEFT>3 AND COMMA$="" THEN COMLEN=LEFT-3:COMMA$=",":B$=LEFT$(B$,LEN(B$)-1)
580 IF LEFT=0 THEN H$="0"+H$:LEFT=1:NUMLEN=NUMLEN+1
590 DIFF=LEFT-COMLEN
600 IF DIFF<1 THEN STOP
610 RHS$=DEC$+RIGHT$(H$,NUMLEN-LEFT):RHS=INT(LEN(RHS$)/2)
620 IF NODEC$="Y" AND DEC$="." THEN B$=B$+SPACE$(RHS):RHS$=SPACE$(LEN(RHS$)-RHS)
630 IF RHS$="" OR NUMLEN<D OR T=5 THEN 670
640 IF T<100 THEN ZT=1 ELSE ZT=2
650 IF LEN(RHS$)=ZT+1 THEN ZT=ZT+1
660 RHS$=LEFT$(RHS$,LEN(RHS$)-ZT)+SPACE$(ZT)
670 D$=LEFT$(H$,COMLEN)+COMMA$+MID$(H$,COMLEN+1,DIFF)+RHS$
680 B$=B$+D$:COMMA$="":DEC$="":LEFT=0:NUMBER=0:COMLEN=0:D$="":H$=""
690 WIDTH 255   :REM INFINITE WIDTH SO NO AUTOMATIC CARRIAGE RETURNS
700 IF I<>LEN(A$) THEN B$=B$+C$
710 IF I=LEN(A$) AND NUMBER THEN 380
720 NEXT I
730 IF LEFT$(A$,1)=CHR$(13) THEN 890
740 PRINT CHR$(26):PRINT  CHR$(27)+"=+ ";
750 PRINT A$:STP=31:INDX=1
760 PRINT CHR$(27)+"=, ";B$:STP=STP+1
770 PRINT CHR$(27)+"=,"+CHR$(STP);
780 IF ATO$="Y" THEN FOR I=1 TO 50:X=12345^2:NEXT I:GOTO 890
790 STAT=ASC(INPUT$(1))
800 IF STAT=27 THEN 871  :REM ESCAPE
810 IF STAT=12 THEN 872 :REM RIGHT ARROW, OSBORNE ARROWS SET FOR CP/M
820 IF STAT=8 THEN 873 :REM LEFT ARROW
830 IF STAT=13 THEN 890 :REM CARRIAGE RETURN
840 IF STAT=11 THEN 874 :REM UP ARROW
850 IF STAT=10 THEN 875  :REM DOWN ARROW
860 B$=LEFT$(B$,INDX-1)+CHR$(STAT)+RIGHT$(B$,LEN(B$)-INDX)
870 INDX=INDX+1: GOTO 760
871 L1=66 : GOTO 200
872 INDX=INDX+1 : GOTO 760
873 INDX=INDX-1: STP=STP-2 : GOTO 760
874 B$=LEFT$(B$,INDX-1)+" "+RIGHT$(B$,LEN(B$)-INDX+1) : GOTO 880
875 B$=LEFT$(B$,INDX-1)+RIGHT$(B$,LEN(B$)-INDX) : GOTO 880
880 STP=STP-1: GOTO 760
890 PRINT #2,B$
900 PRINT
910 GOTO 220
920 END
```

While the program code is not particularly elegant, consisting mostly of logical tests and branches, it should be possible to combine this description with the comments in the listing and modify it to suit your own numerical formatting requirements. The program can be greatly speeded up by compiling it with the BASCOM compiler. It offers some good examples of string manipulation using Basic, and of use of screen control codes to build a simple line editor into the program. Most important, it eliminates some of the last remaining bits of drudgery faced by the professional number cruncher.

```
LISTING 2.  ROUNDING A SAMPLE FILE

A>MBASIC ROUND [CR]

TABLE ROUNDING PROGRAM
  by thomas b. wiens
FILENAME (DISK B: ASSUMED)? MODEL.PRN
AUTOMATIC OPERATION (Y/N)? Y
SKIP FIRST x LINES: ENTER x? 2
WIDTH OF ROW LABEL FIELD? 5
ROUNDING TARGET (5,10,50 OR 100)? 100
MINIMUM SIZE IN DIGITS FOR ROUNDING (0,1,2,3)? 2
IGNORE NUMBERS WITH DECIMALS (Y/N)? N
TRUNCATE ALL TO RIGHT OF DECIMAL
AFTER ROUNDING (Y/N)? N

Y1          55.78       55.780      5578      .550      0.55
Y1          56          55.8        5,600     0.6       1

Y2          0.06        0           5,578     .054      1.55
Y2          0           0           5,600     0.1       2

Footnote:  numbers such as 1.568 will not be rounded if
Footnote:  numbers such as 1.6   will not be rounded if

one is in manual mode and presses ESCAPE after the last
one is in manual mode and presses ESCAPE after the last

line which should be rounded.
line which should be rounded.

READY

File B:MODEL.PRN

A Sample Unrounded Table
                X1          X2          X3        X4        X5
Y1          55.78       55.780      5578      .550      0.55
Y2          0.06        0           5,578     .054      1.55

Footnote:  numbers such as 1.568 will not be rounded if
one is in manual mode and presses ESCAPE after the last
line which should be rounded.

File B:MODEL.RND

A Sample Unrounded Table
                X1          X2          X3        X4        X5
Y1          56          55.8        5,600     0.6       1
Y2          0           0           5,600     0.1       2

Footnote:  numbers such as 1.6   will not be rounded if
one is in manual mode as presses ESCAPE after the last
line which should be rounded.
```

A General-Purpose Graphic Plotting Package

Part 3: JRT Pascal and Pascal/Z listings

by David H. Freese, Jr.

The techniques for interfacing high-level languages to dot-matrix printers via machine-code drivers was described in Parts 1 and 2. The use of machine code drivers provides a common linkage to the printer's graphic capability for several different high-level languages. The extension of the drivers to include automatic scaling and translation was demonstrated for Basic programming techniques in Part 2.

Part 3 presents the same programs written in both JRT Pascal and Pascal/Z. While not presented in this article, I would like to add that I have also linked the machine-code drivers to Microsoft's Fortran-80 compiler and Digital Research's PL/I-80 compiler. In conclusion, it should be noted that Digital Research, Microsoft, Ithaca Intersystems, and JRT Inc. all provide well-documented information for parameter passing to external machine code subroutines. The Basic interpreters are the most difficult to work with, since care must be taken in moving the external subroutines to a protected area in memory. In addition, some special techniques must be used to force the interpreter into passing multiple parameters.

I have received many letters of appreciation prompted by Part 1 of this series of articles, and I would like to thank those who took the time to write. I hope this article will allow you to use your graphics printer to its fullest potential.

I have also developed a nearly identical set of function subroutines for interfacing to the HIPLOT DP-2 digital plotter. These also include alphanumeric lettering subroutines. Driving the DP-2 from Basic can be laboriously slow. These drivers allow the DP-2 to become a real racing machine.

David H. Freese, Jr., Clermont Computer Consultants, R.D. 1, Box 316, Cape May Court House, NJ 08210

Dave Freese is an electrical engineer, presently employed as a senior engineering analyst by Sanders & Thomas, Inc. He is the owner of Clermont Computer Consultants, a small company specializing in real-time control applications. He holds a B.S. and M.S. in electrical engineering. Dave recently retired from a 20-year career in the U.S. Coast Guard, where his duties included an assignment as head of the electrical engineering department at the Coast Guard Academy. He was also chief project engineer at the Coast Guard's electronics engineering center in Wildwood, NJ.

JRT Pascal Bode Plotting Program

```
(*
        BODE

        Example program showing use of graphics plotting package

        Version:  JRT PASCAL

        Date:     20 February 1983

        Programmer:
                  D. H. Freese Jr.
                  Clermont Computer Consultants
                  RD 1 Box 316
                  Cape May Court House, NJ  08210
                  (609) 263 7511
*)
program bode;

const   fourhd = 400.0;
        ten = 10.0;
        tpi = 6.283185;

type    complex  =  record r,i: real end;
        polr     =  record d,a: real end;

    (*  type plt defines the size of the plotting image in memory
        it is determined by the NLINES equate in the assembly code
        for the plotting package.  The array size is given by:

              N = NLINES * 7 * trunc(NLINES*7/8 + .5)

    *)
        plt      =  array [1..21476] of char;

var     image    :  plt;          (* image MUST be first var declared *)
        s,t      :  complex;
        f,f1,f2  :  real;
        delf,
        fscale   :  real;
        decibels :  real;
        dbmax,dbmin,
        dec0,deldec,
        dbscale,dbaxis,
        db0      :  real;
        i,j,k,npts,n,ndec,
        xiplt,yiplt,
        x2plt,y2plt :  integer;
        ans      :  char;

procedure  jplot(cmd: char; p1,p2,p3,p4,p5,p6,p7: integer); extern;

function   sin(x: real): real; extern;
function   cos(x: real): real; extern;
function   arctan(x: real): real; extern;
function   ln(x: real): real; extern;
function   exp(x: real): real; extern;
function   sqrt(x: real): real; extern;
procedure  clear_screen;
begin
   write(chr(27),'E')
end;

procedure  clear_line;
begin
   write(chr(27),'K')
end;

procedure  screen(line,column: integer);
begin
   write(chr(27),'Y',chr(32+line),chr(32+column))
end;

(* COMPLEX NUMBER FUNCTION/PROCEDURES *)
```

```
(* sum of two complex numbers *)
function   csum(c1,c2: complex): complex;
begin
  csum.r := c1.r + c2.r;
  csum.i := c1.i + c2.i
end;

(* difference of two complex numbers *)
function   cdiff(c1,c2: complex): complex;
begin
  cdiff.r := c1.r - c2.r;
  cdiff.i := c1.i - c2.i
end;

(* product of two complex numbers *)
function   cprod(c1,c2: complex): complex;
begin
  cprod.r := c1.r*c2.r - c1.i*c2.i;
  cprod.i := c1.r*c2.i + c1.i*c2.r
end;

(* complex number multiplied by its complex conjugate *)
function   cmag2(c: complex): real;
begin
  cmag2 := c.r*c.r + c.i*c.i
end;

(* magnitude of a complex number *)
function   cmag(c: complex): real;
begin
  cmag := sqrt(cmag2(c))
end;

(* quotient of two complex numbers *)
function   cdiv(c1,c2: complex): complex;
var  den   :  real;
begin
  den := cmag2(c2);
  if den < 1.0e-20
  then den := 1.0e-20;
  cdiv.r := (c1.r*c2.r + c1.i*c2.i)/den;
  cdiv.i := (c1.r*c1.i - c1.r*c2.i)/den
end;

(* angle associated with polar representation of a complex number *)
function   cang(c: complex): real;
const   pi = 3.141592654;
        hpi = 1.570796327;
        tpi = 6.283185308;
        zero = 0.0;
var     ang : real;
begin
  if c.i = zero
     then ang := hpi
     else ang := arctan(c.r/c.i);
  if (c.r >= zero) and (ang < zero)
     then ang := tpi + ang;
  if (c.r < zero)
     then ang := pi + ang;
  cang := ang
end;
(* polar to rectangular conversion of a complex number *)
function   rect(p: polr): complex;
begin
  rect.r := p.d*cos(p.a);
  rect.i := p.d*sin(p.a)
end;

(* rectangular to polar conversion of a complex number *)
function   polar(c: complex): polr;
begin
  polar.d := cmag(c);
  polar.a := cang(c)
end;
```

```
(* logarithm, base 10 of a real number *)
function   log(v: real): real;
const   k = 2.3025851;
begin
  log := ln(v)/k
end;

(* real number converted to decibels *)
function   db(v: real): real;
const   k = 20.0;
begin
  db := k*log(v)
end;

(* real number raised to a real power *)
function   pwr(a,b: real): real;
begin
  pwr := exp(b*ln(a))
end;

(* Transfer is the function representing the input/output
   relationship described in LaPlace Transform notation.
   The variable 's' represents the complex frequency: a + jw.
   The Transform is evaluated only on the jw axis, so 'a' is
   always zero.  'w' is equal to 2*pi*f, where f is frequency
   in Hertz.
*)
function   transfer(s: complex): complex;
var  a1,a2,a3,a4 : complex;
begin
  a1.r := 1.0; a1.i := 0.0;
  a2.r := 2.0; a2.i := 0.0;
  a3.r := 0.02; a3.i := 0.0;
  a4 := cprod(s,s);
  a4 := csum(a4,a2);
  a4 := csum(a4,cprod(s,a3));
  transfer := cdiv(csum(s,a1),a4)
(*
        transfer = (s + 1)/(s*s + .02*s + 2)
*)
end;
begin (* bode *)

repeat

clear_screen;
writeln('                          BODE DIAGRAM TEST PROGRAM');
writeln; writeln;
write('# of decades in frequency ..... '); read(ndec);
write('# of points/decade in plot .... '); read(n); npts := n*ndec;
write('minimum frequency to plot ..... '); read(f1);
write('maximum decibel value ......... '); read(dbmax);
write('minimum decibel value ......... '); read(dbmin);

dec0 := log(f1);
deldec := ndec/npts;
dbscale := fourhd/(dbmax - dbmin);
fscale := fourhd/ndec;
writeln; writeln('Initializing plot image.');
jplot('i', 0, 0, 0, 0, 0, 0, 0);
writeln; writeln('Drawing axis.');
jplot('a', 0, trunc(dbscale*(dbaxis - dbmin)),
           0, 0,
           0, trunc(dbscale*20.0),
           0);
writeln; writeln('Placing logarithmic grid.');
for i := 0 to ndec - 1 do
  begin
  for j := 2 to 10 do
     begin
     k := trunc((i + log(j))*fscale);
     jplot('l', k, 400, k, 0, 0, 0, 0);
     end;
  end;
```

```
f := pwr(ten, dec0);
s.r := 0.0;  s.i := tpi*f;
db0 := db(cmag(transfer(s)));

xlplt := round(fscale*(log(f) - dec0));
ylplt := round(dbscale*(db0 - dbmin));

clear_screen;
writeln('Bode Computations');

for i := 0 to npts do
  begin
  f := pwr(ten, i*deldec + dec0);
  s.r := 0.0;  s.i := tpi*f;
  t := transfer(s);
  decibels := db(cmag(t));
  screen(20,0); clear_line;
  write('Freq: ',f:8:2,'   dB: ',decibels:8:2);
  x2plt := round(fscale*i*deldec);
  y2plt := round(dbscale*(decibels - dbmin));
  jplot('l', xlplt, ylplt, x2plt, y2plt, 0, 0, 0);
  jplot('d', x2plt, y2plt, 0, 0, 0, 0, 0);
  xlplt := x2plt;
  ylplt := y2plt;
  end;

jplot('t', 0, 0, 0, 0, 0, 0, 0);

screen(23,0); clear_line;
write('To repeat the program enter "y"... ');
read(ans);

until not (ans in ['y','Y']);

end.
```

Pascal/Z Bode Plotting Program

```
(*      BODE

        Example program showing use of graphics plotting package

        Version: PASCAL/Z

        Date:    20 February 1983

        Programmer:
                 D. H. Freese Jr.
                 Clermont Computer Consultants
                 RD 1 Box 316
                 Cape May Court House, NJ  08210
                 (609) 263 7511

*)
program bode;

const   fourhd = 400.0;
        ten = 10.0;
        tpi = 6.283185;

type    complex  =  record r,i: real end;
        polr     =  record d,a: real end;


var     s,t      :  complex;
        f,f1,f2  :  real;
        delf,
        fscale   :  real;
        decibels :  real;
        dbmax,dbmin,
        dec0,deldec,
        dbscale,dbaxis,
        db0      :  real;
        i,j,k,npts,n,ndec,
        xlplt,ylplt,
        x2plt,y2plt :  integer;
        ans      :  char;

(* EXTERNAL PROCEDURES contained in file PASPLOT.REL *)

procedure  axis(x0,y0,xa,ya,xb,yb: integer; grid: boolean); external;
procedure  line(x1,y1,x2,y2: integer); external;
procedure  plot(x1,y1: integer); external;
procedure  point(x1,y1: integer); external;
procedure  circle(x1,y1,radius: integer); external;
procedure  init; external;
procedure  xfrplt; external;

(* H89 SCREEN HANDLING PROCEDURES *)

procedure  clear_screen;
begin
  write(chr(27),'E')
end;

procedure  clear_line;
begin
  write(chr(27),'K')
end;
procedure  screen(line,column: integer);
begin
  write(chr(27),'Y',chr(32+line),chr(32+column))
end;

(* COMPLEX NUMBER FUNCTION/PROCEDURES *)

(* sum of two complex numbers *)
function   csum(c1,c2: complex): complex;
begin
  csum.r := c1.r + c2.r;
  csum.i := c1.i + c2.i
end;

(* difference of two complex numbers *)
function   cdiff(c1,c2: complex): complex;
begin
  cdiff.r := c1.r - c2.r;
  cdiff.i := c1.i - c2.i
end;

(* product of two complex numbers *)
function   cprod(c1,c2: complex): complex;
begin
  cprod.r := c1.r*c2.r - c1.i*c2.i;
  cprod.i := c1.r*c2.i + c1.i*c2.r
end;

(* complex number multiplied by its complex conjugate *)
function   cmag2(c: complex): real;
begin
  cmag2 := c.r*c.r + c.i*c.i
end;

(* magnitude of a complex number *)
function   cmag(c: complex): real;
begin
  cmag := sqrt(cmag2(c))
end;

(* quotient of two complex numbers *)
function   cdiv(c1,c2: complex): complex;
var  den   :  real;
begin
  den := cmag2(c2);
  if den < 1.0e-20
  then den := 1.0e-20;
  cdiv.r := (c1.r*c2.r + c1.i*c2.i)/den;
end;
```

```
(* angle associated with polar representation of a complex number *)
function   cang(c: complex): real;
const   pi = 3.141592654;
        hpi = 1.570796327;
        tpi = 6.283185308;
        zero = 0.0;
var     ang : real;
begin
  if c.i = zero
     then ang := hpi
     else ang := arctan(c.r/c.i);
  if (c.r >= zero) and (ang < zero)
     then ang := tpi + ang;
  if (c.r < zero)
     then ang := pi + ang;
  cang := ang
end;

(* polar to rectangular conversion of a complex number *)
function   rect(p: polr): complex;
begin
  rect.r := p.d*cos(p.a);
  rect.i := p.d*sin(p.a)
end;

(* rectangular to polar conversion of a complex number *)
function   polar(c: complex): polr;
begin
  polar.d := cmag(c);
  polar.a := cang(c)
end;

(* logarithm, base 10 of a real number *)
function   log(v: real): real;
const   k = 2.3025851;
begin
  log := ln(v)/k
end;

(* real number converted to decibels *)
function   db(v: real): real;
const   k = 20.0;
begin
  db := k*log(v)
end;

(* real number raised to a real power *)
function   pwr(a,b: real): real;
begin
  pwr := exp(b*ln(a))
end;

(* Transfer is the function representing the input/output
   relationship described in LaPlace Transform notation.
   The variable 's' represents the complex frequency: a + jw.
   The Transform is evaluated only on the jw axis, so 'a' is
   always zero.  'w' is equal to 2*pi*f, where f is frequency
   in Hertz.
*)
function   transfer(s: complex): complex;
var  a1,a2,a3,a4 : complex;
begin
  a1.r := 1.0; a1.i := 0.0;
  a2.r := 2.0; a2.i := 0.0;
  a3.r := 0.02; a3.i := 0.0;
  a4 := cprod(s,s);
  a4 := csum(a4,a2);
  a4 := csum(a4,cprod(s,a3));
  transfer := cdiv(csum(s,a1),a4)
(*
        transfer = (s + 1)/(s*s + .02*s + 2)
*)
end;

begin (* bode *)

repeat
clear_screen;
writeln('                              BODE DIAGRAM TEST PROGRAM');
writeln; writeln;
write('# of decades in frequency ..... '); read(ndec);
write('# of points/decade in plot .... '); read(n); npts := n*ndec;
write('minimum frequency to plot ..... '); read(f1);
write('maximum decibel value ......... '); read(dbmax);
write('minimum decibel value ......... '); read(dbmin);

dec0 := log(f1);
deldec := ndec/npts;
dbscale := fourhd/(dbmax - dbmin);
fscale := fourhd/ndec;

writeln; writeln('Initializing plot image.');
init;
writeln; writeln('Drawing axis.');
axis(0, trunc(dbscale*(dbaxis - dbmin)),
     0, 0,
     0, trunc(dbscale*20.0),
     false);
writeln; writeln('Placing logarithmic grid.');
for i := 0 to ndec - 1 do
  begin
  for j := 2 to 10 do
     begin
     k := trunc((i + log(j))*fscale);
     line(k, 400, k, 0);
     end;
  end;

f := pwr(ten, dec0);
s.r := 0.0;  s.i := tpi*f;
db0 := db(cmag(transfer(s)));

xlplt := round(fscale*(log(f) - dec0));
ylplt := round(dbscale*(db0 - dbmin));

clear_screen;
writeln('Bode Computations');

for i := 0 to npts do
  begin
  f := pwr(ten, i*deldec + dec0);
  s.r := 0.0;  s.i := tpi*f;
  t := transfer(s);
  decibels := db(cmag(t));
  screen(20,0); clear_line;
  write('Freq: ',f:8:2,'   dB: ',decibels:8:2);
  x2plt := round(fscale*i*deldec);
  y2plt := round(dbscale*(decibels - dbmin));
  line(xlplt, ylplt, x2plt, y2plt);
  point(x2plt, y2plt);
  xlplt := x2plt;
  ylplt := y2plt;
  end;

xfrplt;

screen(23,0); clear_line;
write('To repeat the program enter "y"... ');
read(ans);

until not (ans in ['y','Y']);

end.
```

JRT Pascal Bullseye Test Plotting Program

```
(*      PLOT TEST PROGRAM

        Program plots a bullseye test pattern on the graphics
        printer.  Designed to demonstrate the calling sequence
```

Graphic Plotting Package continued . . .

```
    to the graphics plotting package.

    Version: JRT PASCAL

    Date:   20 February 1983

    Programmer:
            D. H. Freese Jr.
            Clermont Computer Consultants
            RD 1 Box 316
            Cape May Court House, NJ 08210
            (609) 263 7511
*)
program JPLTST;

type    pline = array [1..413] of char;
        plt = array [1..52] of pline;

var     image : plt;
        i : integer;

procedure jplot(func: char; p1,p2,p3,p4,p5,p6,p7: integer); extern;

begin
(* initialize the plot image in memory *)
jplot('I',0,0,0,0,0,0,0);

(* plot axis centered at 200,200,
      minor ticks at intervals of 5
      major ticks at intervals of 25
      superimpose a grid *)
jplot('A', 200, 200, 5, 25, 25, 1);

(* plot a crossed pair of lines *)
jplot('L', 10, 10, 390, 390, 0, 0);
jplot('L', 10, 390, 390, 10, 0, 0);

(* plot data point marks at ends of lines *)
jplot('D', 10, 10, 0, 0, 0, 0);
jplot('D', 10, 390,0, 0, 0, 0);
jplot('D', 390, 10,0, 0, 0, 0);
jplot('D', 390,390,0, 0, 0, 0);

(* plot 8 concentric circles of increasing radius *)
for i := 1 to 8 do
   jplot('C', 200, 200, 25*i, 0, 0, 0);

(* transfer image to the printer *)
jplot('T',0,0,0,0,0,0,0);

end.
```

Pascal/Z Bullseye Test Plotting Program

```
(*      PLOT TEST PROGRAM

        Program plots a bullseye test pattern on the graphics
        printer.  Designed to demonstrate the calling sequence
        to the graphics plotting package.

        Version: PASCAL/Z

        Date:   20 February 1983

        Programmer:
                D. H. Freese Jr.
                Clermont Computer Consultants
                RD 1 Box 316
                Cape May Court House, NJ 08210
                (609) 263 7511
*)
program ZPLTST;

var     i : integer;
```

Graphic Plotting Package

continued . . .

```
procedure axis(x0,y0,xa,ya,xb,yb: integer; grid: boolean); external;
procedure line(x1,y1,x2,y2: integer); external;
procedure plot(x1,y1: integer); external;
procedure point(x1,y1: integer); external;
procedure circle(x1,y1,radius: integer); external;
procedure init; external;
procedure xfrplt; external;

begin

(* initialize the plot image in memory *)
init;

(* plot axis centered at 200,200,
      minor ticks at intervals of 5
         major ticks at intervals of 25
            superimpose a grid *)
axis(200, 200, 5, 5, 25, 25, true);

(* plot a crossed pair of lines *)
line(10, 10, 390, 390);
line(10, 390, 390, 10);

(* plot data point marks at ends of lines *)
point(10, 10);
point(10, 390);
point(390, 10);
point(390,390);

(* plot 8 concentric circles of increasing radius *)
for i := 1 to 8 do
   circle(200, 200, 25*i);

(* transfer image to the printer *)
xfrplt;

end.
```

Running WordStar Under TP/M

by Peter V. Radatti

The TP/M operating system by Computer Design Labs, as implemented for the TDL/ZITAN SMB using ZAPPLE, traps the character Control-X for use as a command to execute the ZAPPLE ROM monitor.

Peter V. Radatti, 210 West 12th Ave., Conshohocken, PA 19428

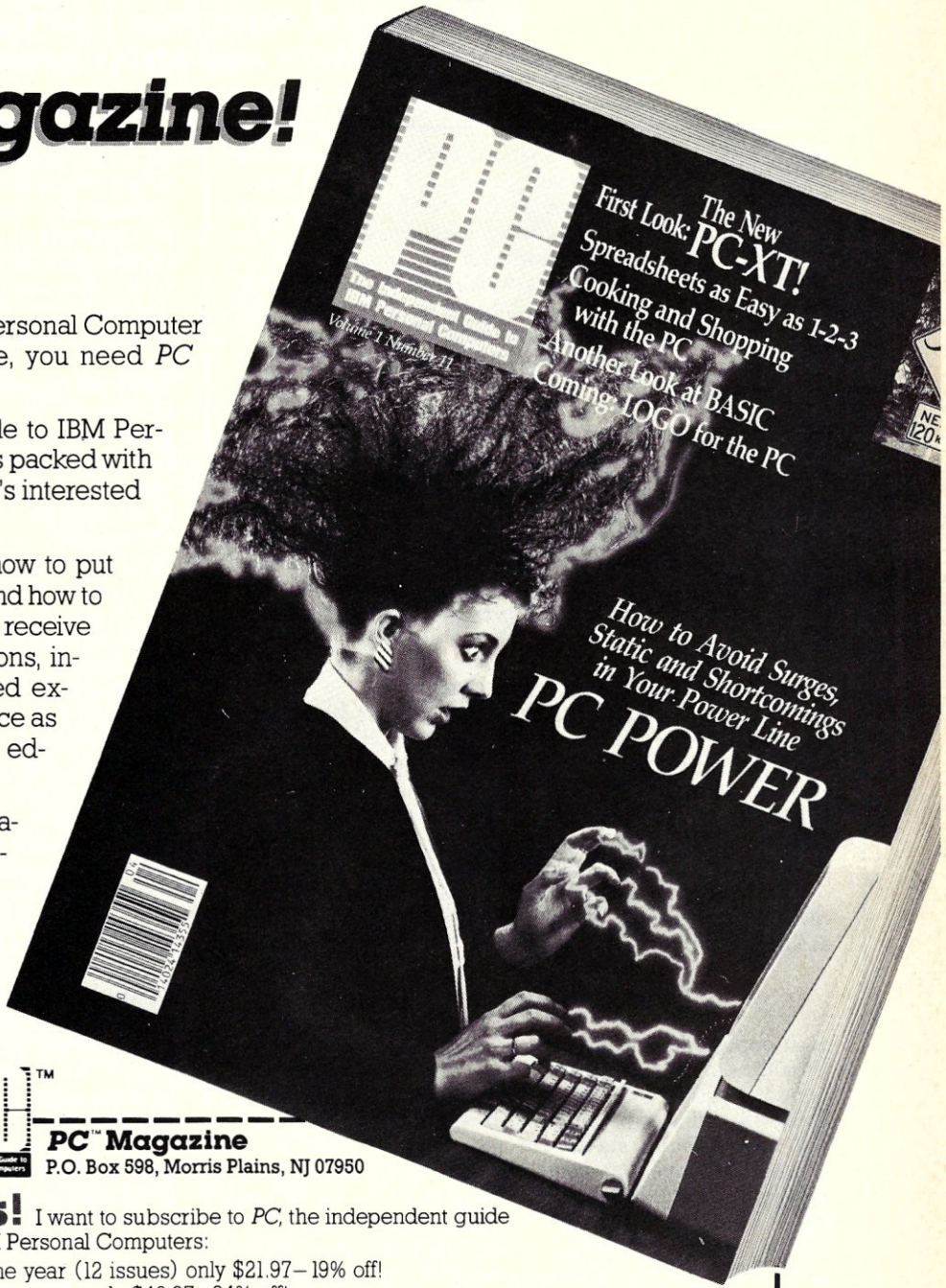
This is usually no problem unless you want to use WordStar. The Control-X character is trapped by TPM and not passed to WordStar.

The following modification to TPM will allow the user to change the command character for ZAPPLE to any character he or she chooses. Control-X will then be passed to WordStar, and the user should have no problems.

```
>                    )
>                    ) Bring up Zapple (DDT can be used)
>                    )
>Z                   ) Address end of memory
EFFF
>F0,EFFF,0          )Fill all of memory with zeros

>
>K.I                )
>G0                 ) Cold boot TPM

59K TPM PIOS V.2.0 - 02/10/80
SERIAL NO: 090701

A>                                  )
                                    ) Start a SYSGEN
SYSGEN (FOR TPM), Version 4.0       )

Get system? (Y/N or Disk ID)A                       )
Source disk on A. Type return to proceed.           )Read the OS into memory
Last address = 2280H                                )(IPA) and do Ctrl-X to
SYSGEN function complete.                           )Zapple.
Put system (Y/N or Disk ID)@E8CB                    Ctrl-X does not print out!
>YFE 18                             )Search the IPA for string FE 18;
1F45                                )Only one address is valid (i.e.,
E8C5                                )the Ctrl-X trap.

>D1F40,1F55                         )Dump memory around the address to see
                                    )that it contains the correct trap.

1F40  CD 03 F0 E6 7F FE 18 C0 CD 1E F0 18 ED E5 21 50
1F50  EA 7E 23 B7 20 0F

>S1F45FE-FE 18-1E                   )Substitute 1E (ctrl-shift-^) for the 18
                                    )Note: you can use any ASCII character
                                    )you want.

>D1F40,1F49
1F40  CD 03 F0 E6 7F FE 1E C0 CD 1E     )Verify the change
>G                          )Return to SYSGEN and answer the question.
A                           )Write new IPM back on drive A.
Destination disk on A. Type return to proceed.
Last address = 2280H
SYSGEN function complete.
Do another? (Y or N)N
You just altered the system on A
***** COLD Start (PM) *****

>
>
>
>K.I                 )Cold boot the new, modified system
>G0

59K TPM PIOS V2.0 - 02/10/80
SSERIAL NO: 090701

A>^X                        )Note that
: ?                         )ctrl-X
A>^X^X^X^X^X                )no longer
: ?                         )traps to ZAPPLE.
A>
```

Get all the facts about the IBM Personal Computer.

Get PC™ Magazine!



Whether you already own an IBM Personal Computer or you're thinking about buying one, you need *PC* Magazine.

PC Magazine is the independent guide to IBM Personal Computers. Each monthly issue is packed with the latest information for everyone who's interested in IBM Personal Computers.

This is *the* magazine for finding out how to put together the best IBM "PC" system...and how to get the most out of it. Every month you'll receive hundreds of colorful pages of evaluations, insights, and straight talk from respected experts—professionals in computer science as well as lawyers, businessmen, writers, educators and many others.

PC covers software, hardware, applications, and every other topic of importance to the thousands of IBM Personal Computer users who read it. To make sure that we give you the facts you need, we include a special "User-to-User" section, a "PC Wish List," and news about IBM Personal Computer clubs, events and publications.

Right now you can save up to 31% on an introductory subscription. And if you're ever dissatisfied with *PC*, just write and tell us—you'll receive a prompt refund for all the unmailed issues remaining in your subscription.

PC™ Magazine
P.O. Box 598, Morris Plains, NJ 07950

Yes! I want to subscribe to *PC*, the independent guide to IBM Personal Computers:

- ☐ One year (12 issues) only $21.97—19% off!
- ☐ Two years only $40.97—24% off!
- ☐ Three years only $55.97—31% off!

Savings based on full one-year subscription rate of $26.97.

CHECK ONE: ☐ Payment enclosed. ☐ Bill me later. 8H286

Mr./Mrs./Ms. _____
(please print full name)

Company _____

Address _____

City _____ State _____ Zip _____

Offer valid only in the U.S., its territories and possessions. Please allow 30 to 60 days for delivery of first issue.

The 50-line Text Formatter Gets Running Fast

A quick substitute for a text processor, this minimal formatter can be used for letters and short papers

by Ian F. Darwin

You've just sent away for an expensive word processing package that is oozing its way through the mail system somewhere, and you've just *got* to get these letters and manuscripts out *now*. You don't want to spend hours writing a text formatter because you've just spent a bundle on what you hope will turn out to be a good text processor. For now, all you have is a partial-screen user-hostile editor—and a C compiler. Where do you go from here?

When faced with this dilemma recently, I came up with the 50-Line Text Formatter for my CP/M system. Also called the Barebones Formatter of Text (or BFT for short), the formatter has one special character, no commands, no options, no parameters, and was written and debugged in about an hour. It does ragged-right alignment of text and provides for line breaks. Surprisingly, these two features provide an acceptable bare minimum for doing letters and short papers!

The programming language

BFT was developed using the C/80 dialect of C, from the Software Toolworks, with the addition of one 'standard I/O' library routine patterned after the UNIX 'stdio.h' library. The routine I added, 'is space', returns 'true' if or when the single character passed to it is a space, a tab, a newline or a formfeed. I also defined 'FILE" for compatibility.

It is perfectly common in C to use the fact that one can test an assignment in the same statement as the assignment itself. The construct

```
if ((x=y)==5) ...
```

both copies the current value of *y* onto the memory location called *x*, *and* tests whether the value so assigned is equal to 5. The main program, for example, need only consist of 'while ((ic=getchar())!= EOF' and a compound statement which is executed as a result. This says: "assign to 'ic' the result of function 'getchar()'; as long as the assignment value is not EOF, then perform the statement part (from 'if ((c==\'\n') . . . ' to 'outchar (c);') and repeat the test". This provides a concise way of processing a file a character at a time, and is almost an 'idiom' in the C language.

Also, the notation '\n' deserves attention. This represents a *single* character, the "newline'. In UNIX and UCSD this is a single character; on CP/M, the newline consists of two characters, a carriage return and a line feed (CR-LF). C implementations on CP/M systems normally map CR-LF to '\n' on input, and vice versa on output. Similarly, '\t' is a single character, the tab. C provides a series of these representations for hard-to-get characters, and they can be used in constructs such as 'switch' statements (unlike Pascal's CASE statement, in which one must code the numerical equivalents or a defined CONSTant). This makes it easier to write systems programs and text formatters in C.

Ian F. Darwin, Computing Services, Univ. of Toronto, Toronto, Ontario, M5S 1A1 Canada

The program

Consisting of about 50 lines of code in the C programming language, BFT has a main program and two short functions (see listing). The main routine looks at each character in the text. If the escape character is seen (defined by BREAKCHAR, and only changeable by recompiling the program), then a new line is started and the column counter is reset to zero. Otherwise, if the character is a newline ('\n') or a tab ('\t'), we change it to a space. Newlines are normally eaten to make the input stream look like a single long line, which BFT has to break up into lines of a reasonable length. Tabs become blanks just because it's simpler than working out the spacing—after all, this *is* a barebones text formatter! If you want to add tab handling, there's a good discussion in the *Software Tools* books by Kernighan and Plauger.

The main routine inserts a newline at the first word boundary after column MINCOL has been reached, or when a user-provided break character is found in the input. There is no provision for breaking long words via hyphenation. My text doesn't use many very long words. With German or Turkish, which tend to use longer and still longer words, BFT would have serious problems.

Whenever the main program needs a line break, it just calls the routine '1break', which prints a newline ('\n'). After printing the newline, '1break' sets the current column indicator to zero, which is the left margin.

Normal characters are printed by calling 'outchar', which passes the character to 'putchar', the C library routine for output. 'Outchar' also increments the column counter.

There is thus no true justification of line—the output is 'ragged right' and, except for the problem of very long words, appears very much as a decent typist might produce it.

Usage

Prepare the input text using any program which will put what you type into a file. I used CP/M's ED for this manuscript, not because it's my favorite editor (believe me, it's not!), but because it was there.

Insert a 'break character' wherever you want a new line to start; one for a new line, two for a new paragraph. For example, if you compiled BFT with

BREAKCHAR defined as '$', then you would enter the following:

```
produce it.
$$Usage
$Prepare the input text...
```

to produce the end of the preceding paragraph and the start of this one. An alternate form would be

```
produce it.
$$
Usage
$$
Prepare the input text...
```

You could even enter it as

```
produce it.$$Usage$$Prepare
the input text...
```

but the former versions produce a manuscript that is easier to update for later revisions. It's a good general rule with any separate text formatter to begin each new sentence on a new line.

BFT is quite easy to use. You just say

```
A>bft infile
```

to get a listing on the terminal, or something like

```
A>bft infile >lst:
```

to get a printout. To get a listing in a file on disk, you just type

```
A>bft infile >outfile
```

These examples assume that you are using a version of C (such as C/80) which allows redirection of the output. In fact, the program could be made simpler by eliminating the code to open the input file, and just redirect the 'standard input' from a file. The above examples would become

```
A>bft <infile
```

and you would shave another half-dozen lines from the code.

Conclusion

This program provides a usable, albeit nearly trivial, working text formatter. It has problems with long words at the end of lines, but fortunately this doesn't happen very often. For a few hours' worth of work, however, it did the trick until my expensive formatter arrived. I did get those letters out on time. And the price was right.

This 50-line Text Formatter can be operational in 5 to 10 minutes on most any C compiler. A few minutes is all it takes to learn, since there's only one special character, and almost any editor can be used to prepare the input.

```
/* bft -- barebones formatter of text */
#include <stdio.h>
#define BREAKCHAR '#'
#define MINCOL 57
int     column;
FILE    infile;
main(argc,argv)
char *argv[];{
char c,d;
if      (argc!=2){
        printf("usage: bft file\n");
        exit(1);
        }
if      ((infile=fopen(argv[1],"r"))==NULL){
        printf("can't open %s\n",argv[1]);
        exit(1);
        }
while ((c=getc(infile))!=EOF){
        if ((c=='\n') || (c=='\t'))   c = ' ';
        if (isspace(c) && (column>MINCOL)){
                lbreak();
                outchar(c);
                }
        else if (c==BREAKCHAR) lbreak();
        else outchar(c);
        }
    outchar('\n');
}

lbreak(){
        outchar('\n');
        column=0;
}

outchar(s)
char s;{
        if ((s==' ') && (column==0)) /* do nothing */;
        else {
                putchar(s);
                column++;
                }
}
```

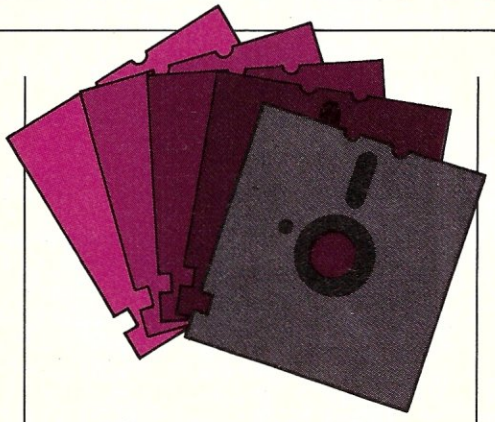
Software Directory

Program name: SOFTPLOT/BGL
Hardware system: CP/M or MS-DOS system with dot matrix printer, plotter, or graphics display
Minimum memory size: 12K Basic workspace, 48K TPA
Language: CP/M or MS-DOS Microsoft Basic interpreter or compiler
Description: SOFTPLOT/BGL (Basic Graphics Language) is a device-independent graphics extension system that supports Microsoft Basic and can make use of dot matrix printers for high-resolution graphics, as well as plotters and displays. Its minimal hardware and software requirements can significantly reduce the cost of producing and using graphics applications, since even the simplest Basic program can easily support a wide variety of hardware configurations. A preprocessor and mergeable subroutines are used to permit programming in high-level HP and Tektronix-like commands such as PLOT X,Y. It supports viewport, window, rotation, text justification, color, dashing, and 3D perspective. Full use is made of available device features.

When released: June 1983
Price: $200
Included with price: Tutorial manual, EMUPLOT printer dump module, pre- and de-processor, library and driver subroutines for various displays, plotters, and printers. A demonstration program is also supplied.
Available from:

Graphic Software, Inc.
1972 Mass. Ave.
Cambridge, MA 02140
(617) 491-2434
CIRCLE #117 ON READER SERVICE CARD

Program name: FILEBASE®
Hardware system: Any CP/M system
Minimum memory size: 64K
Language: Object code
Description: An interactive, menu-driven processor using



comma-delimited field records. Options include record selection, merging, sorting, creating new files, appending to existing files, rearranging fields or adding new ones.

Sorts even on last name if embedded in a full-name field; sorts on zip code even if part of a "city, state, zip" field.

Records can be selected or excluded by testing field contents with comparators (GT LT EQ GE LE BT), or against up to 300 user-entered values. Other methods include a range of record numbers or list of up to 1000 different record numbers, or by viewing each record to include/exclude by a keystroke.

FILEBASE can print labels or envelopes and line listings to screen or printer. Fields can be printed under each other with or without record and field numbers. Keyboarded comments can be added to each record when printing. Files can be combined into one new file or divided into two, using sort and select/exclude methods. Variable-length records can be converted to fixed length, and CB80/CBasic-type fixed-length records can be converted to variable length.

The user does not need to know the number of fields or their sequence in the record because the program determines and displays the data.
When released: April 1983
Price: $75
Included with price: 8"disk or various 5¼" formats and 70-page manual.
Available from:

EWDP Software, Inc.
P. O. Box 40283
Indianapolis, IN 46240
(317) 872-8799
CIRCLE #115 ON READER SERVICE CARD

Program name: Mailer
Hardware system: 8080/Z80, CP/M, CRT with addressable cursor, reverse video, and half-intensity.
Minimum memory size: 64K
Language: Object code (Source PL/I 80)
Description: A complete menu-driven mailing list management program with sort (name or zip) and merge features included. A screen form speeds and simplifies data entry, checking, correction and updating. The program is "friendly" and "forgiving" and requires no programming knowledge. The data fields are reference category, first name, last name, prefix, 3 address lines, city, state, zip (5 or 9 digits) and telephone. Labels are formatted to print 1, 2, or 3 across. Record access for scan or update is virtually instantaneous. You can select records for printing as labels while (if desired) simultaneously creating an address file in Mail-Merge (ASCII variable-length comma-delimited) format for use in word processing. You can also read in your existing "Mail-Merge" files.
When released: March 1983
Price: $100, includes SSSD 8" disc manual; inquire about other disc formats.
Available from dealers, or:

Maurizi Associates
1344 Fitch Way
Sacramento, CA 95825
(915) 486-2993
CIRCLE #116 ON READER SERVICE CARD

Program name: Quic-N-Easi AG®
Hardware system: Z-80, CP/M, Addressable Cursor
Minimum memory size: 64K
Language: C and Assembly
Description: Quic-N-Easi AG is an applications generator for the first-time microcomputer

Software Directory
continued . . .

user. It is simple to use and there is no coding, yet it is very powerful. The system consists of two major modules: the Applications Generator and the Report Generator.

The Applications Generator is used to set up, access, and maintain data files. It is used to write the data input, data editing, and data filing parts of the application. Easy data entry is accomplished via "fill-in-the-blanks" type forms on the CRT. To write an application, a form is typed on the screen. Then the entry areas are identified by placing the cursor where the data are to be entered. A form is displayed to specify the entry field definition. Full editing, justifying, and forced entry can be specified. In addition, range checks and calculations may be specified.

The Report Generator sorts files, performs calculations, manipulates data, and prints reports to the printer, the screen, or to disk (any combination). Reports are created on the screen as they are to appear with up to 300 columns of horizontal scrolling.

When released: October 1982
Price: $295
Included with price: Applications generator, report generator, utilities.
Available from:
Standard Microsystems, Inc.
136 Granite Hill Court
Langhorne, PA 19047
(215) 968-1995
CIRCLE #118 ON READER SERVICE CARD

Program name: KBasic
Hardware system: Any CP/M or Turbodos system
Description: KBasic is Microsoft's MBasic 4.53 integrated with KISS (Keyed Indexed Sequential Search) and 14 file handling verbs for applications that require heavy file transactions with extensive file management capabilities.

It gives Basic a wide range of Indexed Sequential and Direct Access record management. User applications under KBasic allow for multiple key access records, full DBMS ca-

pability and file management reliability. Ease of use and automatic control of record sizes from 64 bytes through records of 512 bytes in length are provided by the Direct Access File manager.

Both the ISAM and DAFM file managers maintain deleted record buffer pools and space allocation checkpoints and thus eliminate all need for user file maintenance operations. Up to four ISAM and sixteen DAFM files can be opened for access at one time. Should it be desired to use a Basic Compiler, or MBasic level 5.x an interface to KISS from MBasic source code using "call" statements is available.

When released: October 1982
Price: $475; $22.50 plus $2.50 S/H for User Guide
Included with price: Disk and manual
Available from:
EIDOS Systems Corp.
Box 3216
Nashville, TN 37219
(615) 297-7125
CIRCLE #119 ON READER SERVICE CARD

Program name: MemoPlan
Hardware system: 8- or 16bit CP/M, PC-DOS, Xenix, UNIX
Minimum memory size: 64K RAM and two floppy disk drives with at least 150K each
Description: MemoPlan, with Concurrent Word Processing, makes up to five documents available at a time. The split-screen feature makes it possible to work on two documents simultaneously and to transfer material from one document to the other.

Designed for use by the executive or secretary who may have limited experience with word processing, MemoPlan is a simple-to-operate system for the composition of memos, letters, and reports directly on the computer. Memo Plan matches all printers and can switch from a dot-matrix printer to a letter quality printer automatically, with true proportional spacing ability.

MemoPlan automatically recovers documents after a power outage and retrieves accidentally deleted material. It has only 28 commands, features the ability to flip from one file to the next with a keystroke, and can be adapted to standard function keys or on-screen menus. It is a perfect subset of DocuPlan, Chang Labs' full-featured document processor.
Price: $195
Included with price: 41-page manual
Available from:
Chang Labs
5300 Stevens Creek Blvd., Suite 200
San Jose, CA 95129
(408) 246-8020
CIRCLE #120 ON READER SERVICE CARD

Program name: FilePlan
Hardware system: 8- or 16-bit CP/M
Minimum memory size: GAK RAM and two floppy disk drives with at least 150K each

Description: FilePlan is an electronic filing system designed for ease of data entry by the end-user. With FilePlan's exclusive worksheet format, multiple records can be viewed simultaneously, and existing records provide examples for data entry. The features include special help commands, an on-screen menu, data validity (attribute verification) and user prompts.

During query, the user is prompted through the process and is able to view data at each stage. The result of a query is displayed as a worksheet and may be changed, saved, or printed. Special prompts are supplied to assist inexperienced users.

Record length can range from 128 to 2048 characters and a record may contain up to 32 variable-length fields; maximum field length is 99 characters. Binary-tree indexing keeps records in proper sequence, automatically updating the sequence as records are modified.

Price: $295

Available from:

Chang Labs
5300 Stevens Creek Blvd., Suite #200
San Jose, CA 95129
(408) 246-8020

CIRCLE #121 ON READER SERVICE CARD

Program name: Font-19

Hardware system: Super-19 advanced terminal firmware ROM. Font-19 ROMs require minor modifications to the Z-19 terminal board; no trace cuts required.

Description: Font-19, a series of advanced graphic firmware ROMs, has been designed to complement the Super-19 upgraded terminal firmware for the Zenith Z-19 terminal and Z-89/90 microcomputer. This allows the user to customize the Z-19 terminal graphics capabilities. Currently six advanced character graphics fonts are available. More will be available in the future. Current fonts include the following:

- *Pixel Graphics ROM:* The pixel graphics ROM upgrades the Z-19 terminal to medium resolution graphics capability (160 by 100 pixels). Sample programs are available.
- *VT-100 Graphics ROM* provides an emulation of the DEC VT-100 graphics character set.
- *Tandy Graphics ROM* emulates the Radio Shack Model III graphics and includes an excellent line-drawing and bar-chart facility in addition. The full ASCII character set can be displayed with the graphics characters using this ROM.
- *APL ROM* provides the APL character set as an alternate to normal ASCII characters.
- *Math-Greek-Scientific ROM* provides a full set of mathematics, Greek, and scientific symbols, equivalent to the symbols found on 128-character capability letter-quality printers such as the NEC Spinwriter.
- *Script ROM* provides a handwriting-like character set alternative to the normal ASCII

character set.

All Font-19 character graphic ROMs allow use of normal Z-19 character set graphics, and shift to the extended graphics capability mode when in the alternate character set.

Available from:
Extended Technology Systems
1121 Briarwood
Bensalem, PA 19020
CIRCLE #122 ON READER SERVICE CARD

Program name: Super-19 firmware upgrade kit
Description: The Super-19 upgrade kit for the Zenith Z-19 terminal and Z-89/90 microcomputer extends the terminal editing features of the terminal (and microcomputer) to include the editing features of the DEC VT-100 terminal. These features give the user the ability to run programs optimized to the DEC VT-100 editing capabilities rather than in VT-52 mode, currently supported by standard Z-19 terminals. Since the keys on a Z-19 do not correspond to those of the VT-100, a special set of key codes has been implemented to provide convenient EDT operation. A template is provided that can be copied and placed around the Z-19 keys.

Most of the VT-100 features have been implemented, except for large characters, smooth scroll, 132-character lines and keyboard LEDs. Escape sequences relating to these functions are processed, but no operation is performed. Other features of the Super-19 include:

- Definable Scroll Areas: from 2 to 25 lines can be set scrollable while the remaining lines are fixed
- Extended Transmit Capability: the top of bottom fixed region, the scroll area, the current line or the current character may be transmitted
- Complete Tab Facility: tab stops can be individually set and cleared or set at fixed intervals
- Clock and Calendar: reports hours, minutes, and seconds in ASCII format, similiarly for date
- 8-bit Data Mode: all 8 data bits are transmitted and displayed
- 8-bit escape sequence mode
- Hardware handshaking
- Slow transmit mode
- Light pen position reporting
- Expanded cursor control functions

In addition, all function keys produce different escape sequences when shifted. Full duplex, half duplex, and baud rates up to 38,400 baud are supported. All Heath functions are retained, and all new features are supported in both Heath/VT-52 mode and ANSI/VT-100 mode.

Price: $49.95
Available from:
Extended Technology Systems
1121 Briarwood
Bensalem, PA 19020
CIRCLE #123 ON READER SERVICE CARD

Software Directory
continued . . .

Program name: SCREEN MASTER
Hardware system: CP/M, MP/M, CP/M-86, MP/M-86, IBM-PC, Sirius under MS-DOS, and PDP/11 under UNIX
Languages: C, Bascom, PCBasic, Pascal MT+, PL/I, Cobol
Description: SCREEN MASTER is a very extensive screen management system. It is dictionary-driven and automatically handles all aspects of screen layouts, visual screen characteristics, cursor movement, help text, data editing, and integrity checking during data entry. Screens are terminal and program independent.

SCREEN MASTER consists of two powerful languages: Screen Description Language (SDL), which enables the application developer to define screens for an end user (these screens can be used for both data display and data entry); and (2) Screen Manipulation Language (SML), which consists of screen manipulation commands that can be invoked from a programming language.

SCREEN MASTER cuts screen-generation time and application development costs through use of a high-level language instead of traditional cursor-position commands used in host programming languages. Advantages include its sophisticated error-checking routines and the independence of the actual application program itself. SCREEN MASTER can be used with the MDBS III data base management system or in the "stand-alone" mode.
When released: November 1982
Price: Depends on OS, hardware, and host language: $2,500 to $18,750
Included with price: SDL, SML, comprehensive documentation
Where to purchase it:
ISE-USA
85 W. Algonquin Rd., Ste. 400
Arlington Heights, IL 60005
(312) 981-9200
(800) 323-3629
CIRCLE #124 ON READER SERVICE CARD

New Products

80 MB super micro

Dual Systems is now shipping its System 83/80, the fastest member of its popular System 83 series. The Model 83/80 combines the MC68000, UNIX, 10 MHz, IEEE-696/S-100 bus with 80 MB of hard disk storage and its SMD controller. The SMD disk storage and its SMD controller. The SMD disk controller offers the highest data throughput attainable in a controller with a single track buffer. It transfers all sectors on a track within essentially a single disk rotation, regardless of the place over which the head first settles. This controller is proprietary to Dual Systems.

The 83/80 can be expanded to accomodate 16 users and 168 MB of storage, and its 20-slot backplane can accomodate up to 3.25 MB of parity RAM. The 83/80 technology can also be supplied as a set of circuit boards to qualified systems integrators.

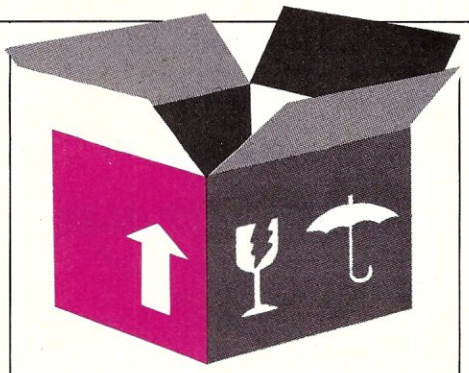
Price: $2,990

Dual Systems Corp., *2530 San Pablo Ave., Berkeley, CA 94702; (415) 549-3854.*

CIRCLE #125 ON READER SERVICE CARD

RACOM Datafinder

The RACOM Datafinder provides fast solutions to file searching, text processing, and pattern recognition problems. The Datafinder is an S-100 board running asynchronously at over 16 MHz, with a programming interface absorbed in only a few minutes. Control is via a single port address for in-

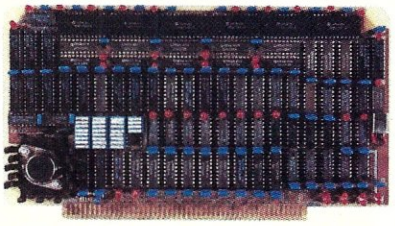


ward status and outward command. Data is accessed via a 24-bit addressed store. The board compares, bit by bit, Test Data with programmable Reference Data, using only bits that are marked Valid for comparison. The results of these tests are recorded in a corresponding Solution bit.

A record may be compared on a field-by-field basis, and each Field may be tested for "Test > Reference", "Reference > Test", or "Reference = Test".

There is a facility for providing bit and byte offsets to the Test Data prior to the comparison process, to facilitate searching for text within a block of text. All of the 12K on-board static RAM store is user programmable.

This board can be used to implement a Content Addressable File Store by bringing information in from a Winchester



disk and then analysing this data according to criteria preprogrammed in the Datafinder.

Data can be simultaneously written to a number of Datafinders and asynchronously analysed according to different criteria at the same time. Status information from the Datafinder gives the total number of hits (successful comparisons) together with the address of the first hit.

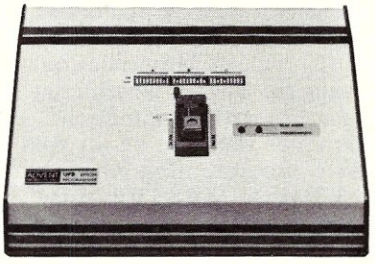
Price: £850

RACOM, Decision Systems Division, *5 Neston Way, Handforth, Wilmslow, Cheshire, SK9 3BX, England; phone: (0625) 52-8979.*

CIRCLE #126 ON READER SERVICE CARD

Universal EPROM programmer

The UP-8 EPROM programmer supports both single- and three-voltage NMOS and CMOS EPROMs, from 512 x 8 (2704) through 32K x 8 (27,256), without the need for personality modules or socket adapters. EPROMs supported by the UP-8 include: 2704: 08, 16, 32, 64, 128, 256; 2732A; TMS 2516: 32, 64; TMS 716; MCM 68732: 64, 66; and all generic equivalents (more than 125 devices).



The UP-8 operates with any 8080, 85, Z80 CP/M-based computer containing a minimum 56K of RAM. Interface to the computer is through a single RS-232 communications port. Baud rates of 50 to 19,200 are switch-selectable.

The UP-8 is supplied with two software packages: BLOCKER and PRGMR. BLOCKER furnishes all utility functions necessary for partitioning long object files into the appropriate lengths necessary for programming EPROMs. Any one of the three checksum types may be computer automatically. Partitioned files are then written to disk as a group of named files for future use in EPROM programming.

The PRGMR software package provides 21 utility commands, including ERASE CHECK, COPY, EDIT, PROGRAM, and COMPARE, all

New Products
continued . . .

activated via single-keystroke commands. An AUTOMATIC mode executes ERASE CHECK, PROGRAM, and COMPARE functions consecutively without operator intervention. The EDIT command permits the amendment of existing files and/or the creation of new files.

Price: $695; includes software on 5¼" or 8" disk, user's manual, and documentation detailing how to write a non-CP/M driver.

Advent PRODUCTS, Inc., *965 N. Main St., Orange, CA 92667; (714) 997-0800.*
CIRCLE #127 ON READER SERVICE CARD

Telephone interface with speech

The CompuFone is a versatile telephone interface and speech storage/output device for IEEE/696/-100 bus computers. FCC-approved for direct connection via a standard modular phone jack, the CompuFone originates and answers calls. An on-board Touch-Tone decoder allows the entering of commands from a remote telephone. The voice digitizer subsystem allows speech recording from the telephone or from either the MIKE IN or LINE IN connections on the board. Previously recorded speech can be sent out the phone line or reproduced locally via an audio amp hooked into the LINE OUT connector.

Possible applications include: intelligent phone answering, voice mail systems, database query with voice output, order entry and inventory control, automatic telephone soliciting and data gathering, ham-radio repeaters and phone patches, paging/announcement systems, sophisticated alarm systems, dictation and transcription, computer-aided instruction, remote control of the computer from any telephone, and (with two boards in a system) call rerouting.

Price: $995; manual only, $30; software on disk, $20.

Computalker Consultants,

New Products continued . . .

1730 21st St., Santa Monica, CA 90404; (213) 828-6546.
CIRCLE #128 ON READER SERVICE CARD

EPROM programmer/ real-time clock

TimEPROMmer (TP-100) is unique in that the calendar/ clock and the programmer are both on one board and independently addressable. The EPROM programmer supports 2708, '16, '32, '32A, and '64 type single-voltage EPROMs. Port mapping is implemented so that no memory space is used. Access to the board is not mandatory due to software control of voltages' this allows remote exchanging of EPROMs via an extender harness (not included).

Extensive CP/M software is executable from the menu and allows read, verify, programming, disk transfers, etc. Inexpensive personality headers are used for maximum device compatibility with future EPROMs having up to 28 pins.

All versions include extensive source software and documentation.

Prices: Bare board, $75; complete kit, $195; A&T fully guaranteed units, $295.

Optronics Technology, *P.O. Box 81, Pittsford, NY 14534; (716) 377-0369.*

CIRCLE #133 ON READER SERVICE CARD

68000 computer

By incorporating half-height floppy disk drives, Sage Computer Technology has further reduced the dimensions of the already compact-size Sage II micro to just 3.9" high by 12.5" wide by 16.7" deep. The low-profile Sage II comes standard with 128K and can be expanded to 0.5 MB of on-board RAM. The minimum configuration includes one built-in, half-height, 640K floppy drive with room for a total of two. All Sage 16-bit, 68000-based

New Products
continued . . .

systems operate at 8 MHz.

Price: Sage II, with one floppy drive, 128K RAM, and a p-System operating system is less than $3,600. CP/M-68K is available optionally.128K RAM expansion blocks, consisting of only two devices, are $250.

Sage Computer Technology, *35 N. Edison Way, Suite 4, Reno, NV 89502; (702) 322-6868.*
CIRCLE #129 ON READER SERVICE CARD

Full CAD program for microcomputers

AutoCAD is a two-dimensional computer-aided drafting and design package that runs under CP/M-80, CP/M-86, or MS-DOS/PC-DOS. It is a general-purpose package suitable for a wide variety of applications such as architectural and landscape drawings; mechanical, electrical, chemical, structural, and civil engineering; and printed-circuit design.

AutoCAD acts like a word processor for drawings. It lets the user make drawings from simple components such as lines (of any width), circles, arcs, and solid-filled areas. Drawings may be created through keyboard commands, with a light-pen and on-screen menu, or from existing paper drawings via a digitizing tablet.

Systems currently supported include CP/M-80 machines with Scion MicroAngelo graphics subsystem and optional light-pen, Victor 9000 with 256K RAM and optional Sun-Flex Touch Pen, and IBM PC with 128K RAM and monochrome and color graphics card. All systems support Summagraphics and Houston Instruments digitizers, and complete range of Hewlett-Packard and Houston Instruments plotters.

COMPUPRO USERS

- BIOS-80, BIOS-86, XIOS-86
- Track buffering, Interrupt driven I/O
- Execute CP/M-80 programs with CP/M-86 or MP/M-86
- Morrow M5, M10 hard disk support
- Memory disk implemented
- Source code included
- Prices from $15-$100

Lanier Computer Systems
3603 23rd Ave.
Shawmut, AL 36876
(205) 768-2616

CIRCLE 30 ON READER SERVICE CARD

MAGNOLIA MICROSYSTEMS

Makes the ZENITH 89/90 a REAL Business Computer

The solid **Dependability** of the Z89/90 with the **Flexibility** to configure it the way **you** need.
- Memory Boards: 16-128 KBytes
- Hard Disk Interfaces and Subsystems: 5-50 MBytes
- Floppy Disk Controller: Mix 8" and 5" drives
- MMS-Net™ Networking
- High-Speed/High-Capacity Auxiliary Processor
- The most powerful, flexible CP/M™ available

Ask your Zenith dealer about Magnolia products

MAGNOLIA MICROSYSTEMS, INC.
2264 15th Ave. West • Seattle, WA 98119
(206) 285-7266 (800) 426-2841

CP/M is a trademark of Digital Research

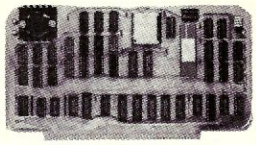
CIRCLE 244 ON READER SERVICE CARD

SAL/80® and **SAL/86™**
do for assembly language what RATFOR does for FORTRAN but emits
OPTIMALLY DENSE code.
SAL/8X includes console I/O primitives which trivialize the task of writing complex interactive user interfaces. Improves programmer productivity by a factor of two and program maintainability by an order of magnitude.
Extensively documented, available for all CP/M compatible disk formats.
SAL/80 version 2.1, $59.00, requires 64K and MAC or RMAC.
CALIFORNIA RESIDENTS ADD 6% SALES TAX.

PROTOOLS®
"Software Tools for the Professional"
24225 Summerhill Avenue
Los Altos, CA 94022
(415) 948-8007

CIRCLE 256 ON READER SERVICE CARD

S-100 Boards from S. C. Digital



NEW 80186 CPU BOARD

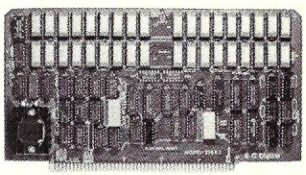
features: **Model: 80186 CPU**
● Intel 80186 Based ● Executes 8086 codes plus 10 additional ● built in DMA channels, timers, interrupt controller ● Interface to Numeric Data Processor, 8087 ● 8 or 16 bit data transfer, with 4 or 8 MHZ clock ● Provision to run 2 different CPU's on the bus, such as our M:Z80 CPU.

Z80B CPU BOARD

features: **Model Z80 CPU**
● 2, 4 or 6 mhz clock. ● 22 bit Address by Memory Mapping in 16K blocks. ● 2 or 4Kbyte EPROM (not supplied) with Phantom generation. ● Jump on Reset. ● Provision to run two different CPU's on the same bus, such model 8186 CPU.

FLOPPY DISK CONTROLLER

features: **Model FDC1**
● Single or Double density, sides, in any combination of up to four 8" or 5.25" drives. ● Digital phase locked loop. ● DMA data transfer with cross 64K boundaries, 24B address, DMA arbitation. ● Monitor/boot EPROM accomodating two different processors. ● CPM Bios programs. ● Serial port to 19.2K baud.



256K DYNAMIC RAM

features: **Model 256KZ**
● 8/16B Data, 24B Address. ● Parity bit per Byte ● Transparent refresh ● Unlimited DMA ● 180nsec. Access time ● Will run 8086, 8088, 68000 to 8mhz, Z80, Z8000 to 6mhz without wait states.

64K STATIC RAM

features: **Model 64KS**
● 8/16B Data 24B Address ● Disable in 2K increments ● 180nsec Access Time (with 64KB) from address on, runs 8086, 68000 to 10mhz, Z80, Z8000 to 8mhz without wait states ● Battery back up capable.

Board Sets: For Limited Time Only!
● Z80B CPU, DMA Floppy Controller, CP/M 2.2, 64KB Static Ram $1,000
● Z80B CPU, DMA Floppy Controller, CP/M 2.2, 250KB Ram $1,250
● For CP/M Plus instead of 2.2 add $50
Please call for latest prices.

All boards conform to IEEE696/S100 specifications, fully socketed, screened legends, masks, Gold contacts. Guaranteed One Full Year.

Model	Prices	with
80186 CPU	$595	16 Bit CPU
Z80 CPU	$325	Memory Mapping, 6 mhz
FDC 1	$395	Monitor EPROM
256KZ	$695	256KB, Parity
256KZ-128	$595	428KB, Parity
64KS	$425	64KB, CMOS
32KUSM	$295	32KB, CMOS
3SPC	$259	3 serial, 1 parallel, cassette
CP/M 2.2	$55	Purchased with FDC1
CP/M Plus	$105	Purchased with FDC1
Z80 Monitor	$55	2K in EPROM, source code, for 3SPC

All Boards come assembled and tested.

Delivery is within 3 to 5 working days. MC, Visa or COD orders accepted. (Add $6 for COD orders) Illinois residents add 5¼% sales tax.

*CP/M is a registered trademark of Digital Research, Inc.

O.E.M. & DEALER PRICE AVAILABLE

S.C. DIGITAL, INC.
1240 N. Highland Ave., Suite #4
P. O. Box 906, Aurora, Illinois 60507
Phone: (312) 897-7749

CIRCLE 259 ON READER SERVICE CARD

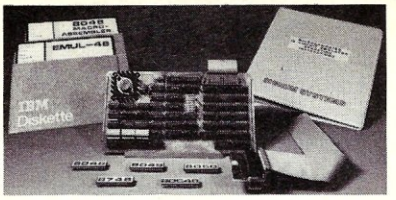
EPROM Programming Monitor (EPM)

EPM is a hardware independent software package that programs EPROMs directly from CP/M* or MP/M II* disk files. EPM can be operated in the standard user friendly menu mode of operation or in the EPROM Editor mode for users needing the capability of modifying files or EPROMs at the byte level. It automatically verifies EPROM erasure prior to programming, provides positive confirmation of successful data transfer and reports any discrepencies directly to the operator. The cost of EPM is $75 and includes all documentation. The EPROM Editor option is $45.

DANTEK Software, Inc.
4550 Schoolhouse Rd
Batavia, Oh 45103
513 / 752-1921

*CP/M and MP/M are trademarks of Digital Research

CIRCLE 18 ON READER SERVICE CARD



8048 IN-CIRCUIT EMULATOR

- IEEE-696 (S-100), CP/M compatible
- Emulates 8035, 8039, 8048, 8049, 8748, 8040, 8050, and 80C48 CMOS.
- Hardware Breakpoints allow Real Time Emulation up to 11 MHz
- Trace includes disassembled code
- Display/Modify Program & Data Memory, Registers, I/O Ports & Flags
- Scope trigger from breakpoint locations

Emulator comes on one S-100 board with a 3 ft. cable and buffer assy. that plugs into the user's uP socket. The board with all supporting software is $995. For 8048 MACRO Assembler add $150.
A portable development system with CRT, two disc drives and emulator is $ 3200 (213) 4515382

SIGNUM SYSTEMS
726 Santa Monica Blvd.
SANTA MONICA, CA. 90401

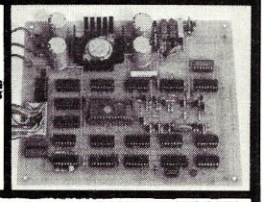
CIRCLE 112 ON READER SERVICE CARD

BURN EPROMS USING CP/M*

EPROM PROGRAMMING SYSTEM RUNS UNDER CP/M

PROGRAMS: 2708, 2758, 2716, 2732, 2732A, 2764 and 27128.

– FEATURES –
- STAND ALONE BOARD
- ELECTRONIC SWITCHING OF EPROM TYPES
- ON BOARD SUPPLIES
- 28 PAGE MANUAL
- DESIGNED WITH EASY-TO-GET PARTS
- WORKS WITH ANY CP/M SYSTEM
- USES 24 VOLT XFMR FOR POWER



– INTERFACES THROUGH ONE 8 BIT INPUT PORT AND ONE 8 BIT OUTPUT PORT. 16 WIRES – NO SPECIAL HANDSHAKE LINES.

– CONTROL SOFTWARE COMMANDS –
- PROGRAM EPROM(S) FROM DISK FILE
- PROGRAM EPROM(S) FROM RAM
- READ DISK FILE INTO RAM
- COMPARE EPROM W/RAM
- VERIFY EPROM IS ERASED
- COPY EPROM
- DISPLAY/MODIFY RAM (MONITOR)

(MONITOR HAS 11 SUB COMMANDS FILL-DUMP-XFER-MODIFY ETC.)

ALSO AVAILABLE FOR OSBORNE 1 (TM)

BARE P.C. BOARD WITH COMPLETE DOCUMENTATION AND SOFTWARE ON AN 8" SINGLE DENSITY DISKETTE **$69.**

TO ORDER SEND CHECK, MONEY ORDER OR CALL

WRITE OR CALL FOR MORE INFO. 513/752-7218

ADD $3.00 SHIPPING Ohio res. add 5.5% tax $3.00 for C.O.D.

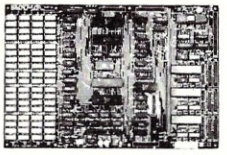
VISA OR M.C. ACCEPTED

AndraTech
P.O. BOX 222
MILFORD, OHIO 45150

* CP/M is a trademark of Digital Research

CIRCLE 199 ON READER SERVICE CARD

The Mega Super Computer



Free Computer with a Purchase of a RAM Disk

★ Z80B Running at 5mhz, ★ Versatile CPM, ★ Math chip 9511 or 9512 AMD, ★ 48 Plus Ports on Mega Expander BUS, ★ CTC, DMA, ★ 2 Parallel Ports—with hand shaking, ★ Serial Ports—with or without hand shaking runs 150 to 19K2 Baud. Runs most terminals, printers, and modems.

★ NOW! CPM 3.0.
★ Hard Disk Interface hooks directly to Priam Drives.
★ Floppy Disk Controller—Handles Single Density IBM compatible disks and Dual Density 1 or 2 sided supports. 8" or 5¼" in various combinations—3 Drives equal over 4 Mega Bytes of Storage. WD2797.
★ 512K Bytes of 64K D-rams with parity configurable as a HIGH SPEED ELECTRONIC DISK or 8 banks of 64K for multi-user or countless other applications.
★ **All this on a state-of-the-art 4 layer card—with accurate documentation—10" x 15".**
★ Prices: Bare Board with Documentation $399
64K Base System A & T $895
512K Base System A & T $1595
CPM 2.2 System A & T $125
512K Base Kit $1299
Manual $25
MPM $325

MEGA CO.
2318 S. Park Street, Madison, WI 53713 (608)255-7400

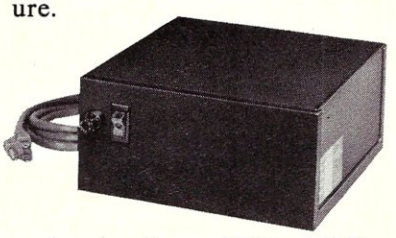
CIRCLE 47 ON READER SERVICE CARD

New Products
continued . . .

Price: $1,000.
Autodesk, Inc., *16 St. Jude Rd., Mill Valley, CA 94941; (415) 381-1819.*
CIRCLE #130 ON READER SERVICE CARD

5V DC power supply with UPS capabilities

The Model MUP-0503 is specifically designed to provide ultra-reliable 5V DC power to a microprocessor logic and memory system. The UPS capability is incorporated within the unit. The MUP-0503 completely replaces a standard power supply. Internal nickel cadmium batteries power the computer memory for over one hour in the event of line failure.



Input voltage: 100 to 132V AC; 47 to 63 Hz; 10 25VA maximum. Output voltage: 5V DC @ 3A. Available with enclosure or an open printed circuit board.
Price: $150
Instrumentation and Control Systems, Inc., *Electro-Pac Division, 520 Interstate Rd., Addison, IL 60101*; (312) 543-6200.
CIRCLE #131 ON READER SERVICE CARD

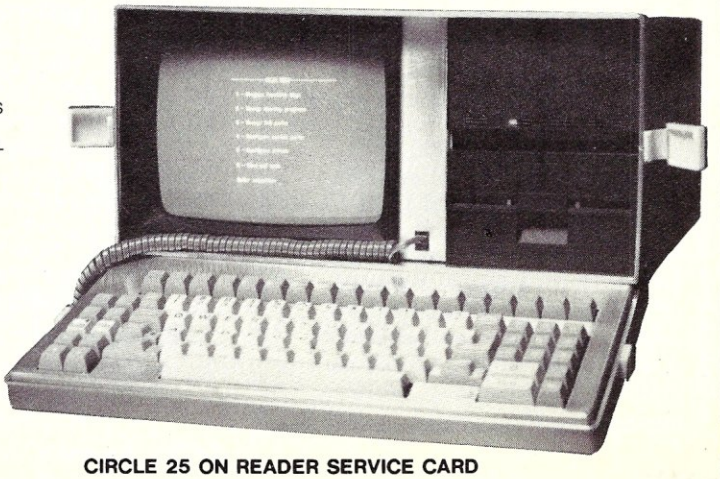
THE WORLD'S FIRST S-100 BUS PORTABLE COMPUTERS
Star-lite HD20 * Star-lite Quad * Star-lite

—THE WORLD'S FIRST PORTABLE COMPUTER WITH BUILT-IN 20 MEGABYTES HARD DISK
* STAR-LITE HD20 — $ 4,995
- 3 OPEN SLOTS
- 20.4 MEGABYTES
- 183K FLOPPY
- ACCOUNTING SOFTWARE GL-AP-AR-PR

—THE WORLD'S FIRST S-100 PORTABLE COMPUTER WITH 1.5 MEGABYTES FLOPPY
* STAR-LITE QUAD — $ 3,495
- 4 OPEN SLOTS
- 1.54 MB FLOPPY
- ACCOUNTING GL-AP-AR-PR

—THE WORLD'S FIRST S-100 PORTABLE COMPUTER
* STAR-LITE — $ 2,695
- 4 OPEN SLOTS
- 400K FLOPPY

FEATURES IN-COMMON FOR ALL 3:
- S-100 BUS (IEEE 696)
- CP/M 2.2
- Z80A CPU 64K RAM
- 9" GREEN SCREEN WITH REVERSE VIDEO, ETC.
- 26 PROGRAMMABLE KEYS
- 1 RS232 SERIAL PORT 1 CENTRONICS PARALLEL
- WORD PROCESSING
- SPREAD SHEET
- 34 LBS.
- 7 5/8" X 16 3/8" X 16 1/2"
- JET-AIRCRAFT-GRADE ALUMINUM ENCLOSURE



COMPUTERSHOP
139 1st ST. CAMBRIDGE, MA 02141
(617) 661-3723, (617) 661-2670

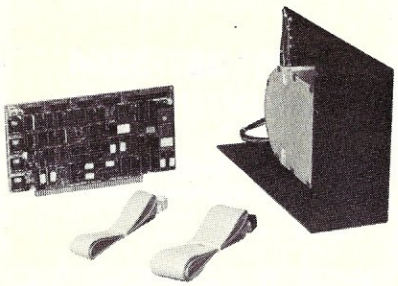
* DEALER INFORMATION AVAILABLE

CIRCLE 25 ON READER SERVICE CARD

New Products
continued . . .

5″ hard disk systems

A new line of 5″ hard disk systems has been introduced by Tarbell Electronics. The systems come in 5, 10, or 16 MB configurations. Each system includes a 5″ disk drive, an 8″ metal frame with 5V and 12V regulators, a hard disk controller and cable, configured CP/M 2.2, and documentation. Features include a data transfer rate of 5 Mbits/second and an average seek time of 120 ms.



Price: The subsystems range from $2,095 for 5 MB, $2,265 for 10 MB, and $2,375 for 16 MB.

The Tarbell Empire Computer Series also now offers the 5″ hard disks, also in the 5, 10, or 16 MB versions. All Empire computers can be purchased as multiuser systems offering additional ports for 8 CRTs and 2 printers, and up to 1MB main memory.

Tarbell Electronics, *950 Dovlen Place, Suite B, Carson, CA 90746; (213) 538-4251.*

CIRCLE #132 ON READER SERVICE CARD

BRIDGE GRAPHICS

PLOTPAK™ is a complete plotting library that runs under FORTRAN-80 and performs a variety of functions:

windowing, linear print arrays, automatic polygon drawing, annotations, plotting symbol/line selection, labeling, coordinate conversions.

PLOTPAK can drive a screen and plotter simultaneously and includes your choice of the following drivers:

SCREENS

- MicroAngelo MA 512
- ADM + Retrographics
- TEK 4010 Compatible Terminals

PLOTTERS

- Houston Instruments DMP-4
- H.P. Plotters 7225B & 7470
- Radio Shack Printer/Plotter

PLOTPAK (.REL file) two drivers **$275**
PLOTPAK (source code) two drivers **$365**

BRIDGE™ Computer Company
DIVISION OF SEA DATA CORPORATION

ONE BRIDGE ST., NEWTON, MA 02158
PHONE (617) 244-8190

We're no longer the best kept secret in the universe. The Martians have landed in La Mesa, Ca. with a complete line of CP/M software.

Let us know your computing problems, maybe we've solved them already.

Timin FORTH RELEASE 3.1 . . . . . **$99.95**
Supplied with a visual editor, CP/M utility package, floating point, and 8080 / Z-80 assembler.

Timin ROMable FORTH **$199.95**
Produces applicaltions in ROM. Contains all vocabulary of Releae 3.1 Stand alone applications as small as 5K.

Timin DUAL-TASKING FORTH **$239.95**
Simultaneous execution of two programs. Plus all the features of Release 3.1

Timin FORTH Manual **$29.95**
ATTENTION all Timin FORTH owners! We have a new revised manual with more examples and documentation. Make your programming life easier.

CP/M USER'S GROUP INDEX GUIDE **$22.95**
Abstracts and documentation from volumes 1-90 of public domain software.

Martian Technologies
8348 Center Dr. Ste. F La Mesa Ca. 92041
619-464-2924

CIRCLE 246 ON READER SERVICE CARD

PARA-SOFT

PARASOL **39.95**
- Structured but readable
- 10 - 20% larger object than ASM
- Full file interface - Record or Sector
- Many utility libraries available
- Binary and BCD (18-digit)
- Full string-handling capability
- 44K CP/M-80 or larger

DISAM **39.95**
- 8080 or Z-80 disassembler
- Output ready for ASM or MAC
- Automatic label generation

Write or call for futher information

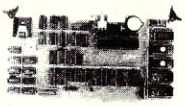
Available on 8″ SSSD - call for other formats

Parametric Software
P.O. Box 6273
Columbia, SC 29260
(803) 782-5458

CIRCLE 23 ON READER SERVICE CARD

TRUE 16 BIT PROCESSING

MICRO SOLUTIONS INC.



Z8001 Microprocessor (upgradable to Z8003)

- 2 Kx16 Eprom with Monitor Program (E²Rom optional)
- 16 Vectored Interrupts
- Full IEEE- 696 (S100) Compliance
- All Z8001 Features available
- Battery backup time of day clock
- Requires RAM capable of word transfers
- CP/M8000 Coming Soon

*Model M8000 CPU Board $425[00]
*Model M8000 EE with E²Rom $500[00]

Memory Management Companion Card available 3rd Quarter 1983
*6 month limited warranty

NEW!!
2 Serial / 2 Parallel I/O Card

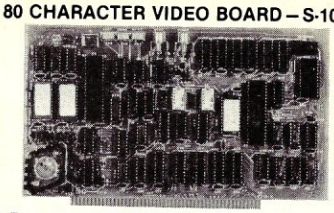
- 2 Async serial ports
- 1 Centronics parallel port
- 1 General purpose parallel port
- 3 16 bit counter/timers
- Uses Zilog 8531/8536

Introductory Price $225[00]

To Order: Call 1-800-821-8858, In NM, 1-505-523-0975 or write:
MICRO SOLUTIONS INC.
Suite 191 E — 1608 El Paseo Rd. — Las Cruces, NM 88001
Include Ck., M.O. or use Master Card or Visa. NM res. add 4.5% tax.

CIRCLE 10 ON READER SERVICE CARD

80 CHARACTER VIDEO BOARD – S-100



All This on ONE BOARD:
- Keyboard port with TYPE-AHEAD buffer
- 8275 CRT controller with light pen port
- Two 2716's – program & character rom's
- Optional 2716 for CHARACTER GRAPHICS
- All screen & keyboard ram
- SIMULTANEOUS I/O or Memory mapped
- Z-80 MPU – 2 or 4 Mhz system clock
- Easy to adapt Software
- Uses only EASY-TO-GET parts
- Use in any S-100 system
- 696 Bus Compliance: D8 M16 I8 T200
- Build for less than $200
- INow includes Crystal and Heat Sink, $9.85 value.

Introducing The VDB-A
Bare board with Documentation **$49.50**
+ $2.50 s&h (Ill.res. add 6% tax)
Add 3% for MC & Visa

Simpliway PRODUCTS CO.
P.O. Box 601, Hoffman Estates, IL 60195
Dealer Inquiries Invited
312/359-7337

CIRCLE 67 ON READER SERVICE CARD

Do you speak CB80?

- MAGIC Library — 21 assy language functions to sort arrays, process dates and more $39.95
- CBUG Newsletter for CB80 Users $12/12 issues

Do you use CBASIC?

- CBASIC programs compiled to fast, compact .COMs

Do you need software?

- Custom CB80 programs

Software Magic
11669 Valerio St. #213
North Hollywood, CA 91605
(213) 765-3957

CBASIC, CB80 are trademarks of Digital Research, Inc.

CIRCLE 68 ON READER SERVICE CARD

SIMPL-SIMON SOFTWARE

EXPLORING CP/M

Explore the mazes of the CP/M operating system with this menu-driven 8-inch disk which allows you to (1) look into every hidden cranny of a CP/M disk; (2) change any ASCII or hex byte; (3) recover erased files; (4) create "autoload" disks; (5) find and lock out bad sectors; (6) reconstruct files from crashed disks. And more! **$39.95.**

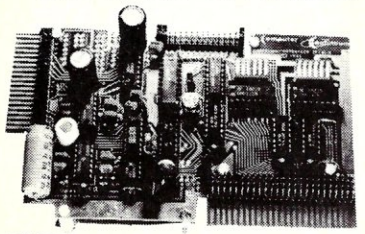
Hard-copy instructions included. For recommended parallel reading: CP/M PRIMER by Murtha and Waite, $14.95. Send check for prompt shipment. Add $2.25 per order for shipping. Return for refund if not satisfied. CP/M is a registered trademark of Digital Research, Inc. Requires two drives, 32K.

TELEPRINT, INC.
P.O. Box 10B, Sylvania, Ga 30467

CIRCLE 270 ON READER SERVICE CARD

Microsystems Mart

A/D 8 channels + D/A 8 channels



Fast 1.6 µs A/D conversion time. 100 ns acquisition window. 1 µs D/A settling time. 8 bits. 0-5V or 0-2.5V full scales. Board allows re-arrangement of the pins and other features for easy interfacing to many types of host computers including Sinclair, Apple, TRS80, others. Switches set address. Address picks channel. Inquire about KITS and software. Assembled and tested **$195**. Aluminum box and ribbon cable **$15.**

Computer Continuum
301-16th Ave
San Francisco, CA. 94118 (415)752 6294

THE UNIX™ GUIDE

An inexpensive way to discover the benefits of UNIX System III

- Orients computer users to UNIX software
- Special coverage of the UniSoft Edition with Berkeley enhancements
- Application oriented, clearly written and concise
- Helps readers determine that they are buying full UNIX System III
- Complements standard UNIX reference documentation

Order the guide by sending a check for $25.00 payable to:

PACIFIC MICRO TECH
5819 Poinsett Ave.
El Cerrito, CA 94530
(415) 233-3596

*UNIX is a trademark of Bell Laboratories

IMPROVED CP/M® TO ATARI® INTERFACE!

The Critical Connection™

ONLY $175

- ★ Automatic install for many systems
- ★ Hardware connects Atari to CP/M system
- ★ Software allows Atari to use disk drives, printer and keyboard of the CP/M system.
- ★ Ability to simulate, with the CP/M system, 4 Atari disk drives simultaneously.
- ★ Software buffers your keyboard & your printer.
- ★ Error detection and recovery.
- ★ Documentation and technical support.

(408) 997-0264

USS ENTERPRISES
6708 Landerwood Ln., San Jose, CA 95120
Call/Write for Free Brochure

RAM'S AND EPROM'S FOR SALE

4164 Pla	150ns	$4.60
4164 Cer	150	4.95
6116P-3	150	3.75
6116LP-3	150	4.50
2716	450	2.95
2716-1	350	CALL
2732	450	3.75
2732A	250	CALL
2764	250	7.00
2532	450	4.25
2564	450	8.50

- Minimum order is $25.00
- C.O.D. only
- OEM inquiries invited

Excel-Tec Industries
Post Office Box 2205
Silver Spring, MD 20902
(301)434-4123

SELL THIS MAGAZINE

in your store for BIG PROFITS!

FOR DETAILS:

CALL COLLECT
(212) 725-6851

DIRECT RETAIL SALES DEPT.
4th FLOOR, ONE PARK AVENUE
NEW YORK, NY 10016

I WILL BEAT ANY COMPETITOR'S PRICE PROVIDED IT IS NOT BELOW MY COST. **TRY** TO BEAT THESE IC PRICES:

DYNAMIC RAM		
64K	200 ns	$5.25
64K	150 ns	5.35
64K	120 ns	5.90
16K	200 ns	1.35
EPROM		
2764	300 ns	$6.25
2732	450 ns	4.05
2716	450 ns	3.20
2532	450 ns	4.60
STATIC RAM		
6264LP-15	150 ns	$60.00
6116P-3	150 ns	4.10
2114	200 ns	1.45
Z80A FAMILY		
CPU, CTC, or PIO		$3.39
DART		7.99
DMA or SIO/0		12.50

Prices 6/1/83

MasterCard/VISA or UPS CASH COD
Factory New, Prime Parts
MICROPROCESSORS UNLIMITED
24,000 South Peoria Ave.
BEGGS, OK. 74421
(918) 267-4961

Prices subject to change. Call for volume prices. Subject to available quantities. Shipping & Insurance extra. Cash discount prices shown.

FREE CATALOG!

Just let us know and we'll mail you a FREE *Creative Computing* Catalog—16 pages filled with books, buyer's guides, magazines, and more!

To get your FREE catalog, write to: *Creative Computing* Catalog, Dept NA1X 39 East Hanover Ave., Morris Plains, NJ 07950.

Automatic Micro

Smartboard™ RC100 - Remote Control Relay Interface
Reliable, flexible control of up to 16 bi-stable relay switches.

- S-100 compatible
- 16 channels relay control & status
- All inputs/outputs optically isolated
- Onboard 8035 CPU and 4 Status/Command registers allow easy control & communication with main CPU
- Relay state unaffected by power outages
- Zero voltage switching for extended relay contact life
- Port or Memory mapped addressing
- Fully Documented
- External 2716 or 2732 EPROM allow custom programming for application flexibility
  (Designed for GE RR8 remote control relays)

RC100 Remote Control Relay Interface	$495.00
RR8 relay (ea)	$17.95
DB-25 connector cable (ea)	$24.95

Dealer inquiries invited!

Automatic Micro, Inc.
P.O. Box 20007
Columbus, Ohio 43220

16 BIT S100 SUPER MICRO

EXPANDABILITY
- 18 SLOT S-100
- 128K-1M RAM
- 1.2M PER 8– DISK
- 640K PER 5¼ DISK
- 15.8M PER HARD DISK

FLEXIBILITY
- IEEE 488
- A-TO-D AND D-TO-A
- OTHER IEEE 696 BOARDS

GRAPHICS
- 512x480 SCION or
- 640x400 w/NEC 7220
- .001– 11x17– PLOTTERS

SOFTWARE
- MICROSOFT MSDOS
- BASIC, FORTRAN, PASCAL
- 8087 SUPPORT
- PLOTTING PACKAGE
- SCREEN DUMPING
- MATRIX ROUTINES
- DATA ACQUISITION

WILLIAM J. CLAFF
(617) - 235-9505

classified ads

CLASSIFIED RATES: Per Word, 15 Word Minimum. REGULAR: $1.00. EXPAND-AD: $1.50. GENERAL INFORMATION: Prepayment discounts available. Payment must accompany order except credit card — Am. Ex., Diners, MC, VISA (include exp. date) — or accredited ad agency insertions. Copy subject to publisher's approval; must be typewritten or printed. First word set in caps. Advertisers using P.O. Boxes MUST supply permanent address and telephone number. Orders not acknowledged. They will appear in next available issue after receipt. Send order & remittance to: Classified Advertising, MICROSYSTEMS Magazine, 1 Park Avenue, New York, NY 10016. Direct inquiries to: Kenneth L. Novotny (212) 725-3927.

COMPUTER EQUIPMENT/SUPPLIES

COMPUTER PAPER—Save $$$: Lowest single carton prices. Free samples. Bob's charts, 405 East Third, Long Beach, CA 90802.

BUSINESS OPPORTUNITIES

COMPUTER CASH—101 exciting, new, sparetime home businesses. Report. $4.50, Associated Research, Box 22485MI, Houston, TX 77227.

INDEXING AND MICROFORM AVAILABLE

This magazine is indexed in THE MAGAZINE INDEX available from Information Access Corporation, 404 Sixth Avenue, Menlo Park, California, and in microform from Xerox University Microfilms, Ann Arbor, Michigan and Bell and Howell, Micro Photo Division, Wooster, Ohio.

ADVERTISERS INDEX

Four Answers To Your S-100 Multi-User Problems.

Intercontinental Micro Systems makes everything you need for S-100 bus multi-user systems, networks or single user systems.

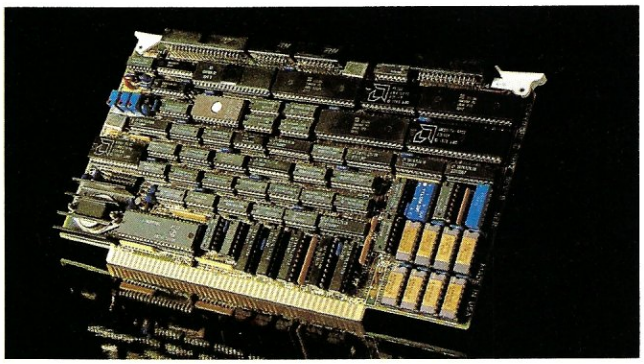
At a price that won't break your budget.

Quite simply, our single board computers, slaves, 256K memories and personality boards let you build a system *now*, not later. The hardware works, the software works, and the prices are what you'd expect from a company that uses the most advanced design, software and production techniques to keep costs down.

What you won't expect is the almost awesome sophistication of Intercontinental Micro System's products.

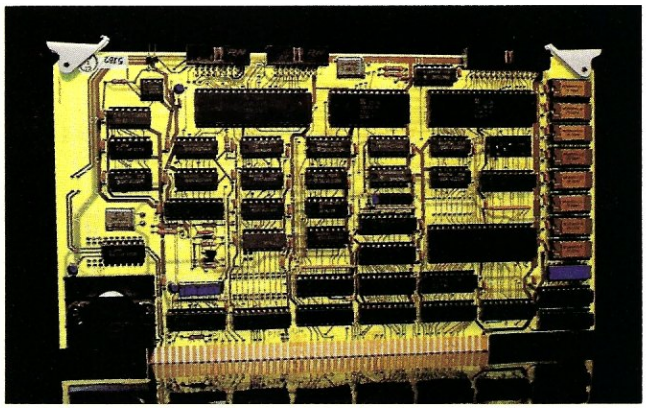
So stop messing around with multiple sourcing, hardware integration problems and software nightmares. Come to Intercontinental Micro and get it all — price, performance and delivery.

Read the specs, then call, write or circle the bingo number below. We'd be glad to send more information and help solve your S-100, multi-user system problems.



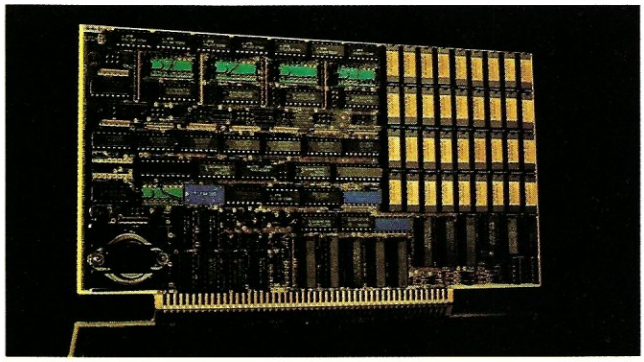
1 CPZ-48000 SINGLE BOARD COMPUTER.

☐ IEEE 696.1/D2 S-100 compliance. ☐ Z80A,™ 4MHz Operation. ☐ Floppy disk controller (FDC). Single or double sided. Single or double density. 8" or 5¼." ☐ Two synchronous or asynchronous serial I/O channels (SIO). ☐ Two parallel I/O channels (PIO). ☐ Four channel DMA controller. ☐ 64K on board RAM. ☐ Memory management unit (MMU). Addresses up to 16 megabytes of system memory. ☐ Eight Vectored priority interrupts. ☐ Provisions for 2K or 4K onboard EPROM. ☐ Software selectable baud rates. ☐ IBM Bisync, HDLC, SDLC and other protocols. ☐ CP/M,™ MP/M,™ and TurboDOS™ operating systems available. ☐ Turbo-Disk® implementation included.



2 CPX-MX SLAVES.

☐ IEEE 696.1/D2 S-100 compliance. ☐ Compatible with CPZ-48000 SBCP, any Z-80A based CPU with extended address capability or 16 bit based CPUs complying with IEEE 696.1/D2 bus specification. ☐ Z-80B™ 6MHz (CPS-6X) or Z80A 4MHz (CPS-4X) operation. ☐ Two synchronous (CPS-MS) or asynchronous (CPS-MA) serial I/O ports. ☐ TurboDOS™ & CP/NET™ compatible. ☐ Master confiscation of slave memory for diagnostic purposes. ☐ Two parallel I/O ports; eight data bits + 2 handshake lines per port. ☐ 64 Kbytes of onboard dynamic RAM. ☐ Master/slave memory-to-memory transfers under DMA control @ 571 Kbyte/sec transfer rate when used with CPZ-48000 SBCP. ☐ Software selectable baud rates. ☐ Usable as an intelligent I/O processor in single user system.

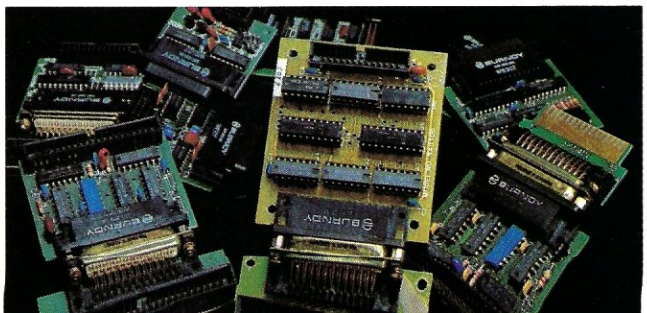


3 256KMB-100 256K MEMORY.

☐ IEEE S-100 bus, spec 696.1/D2 compliance. The 256KMB-100 is compatible with most IEEE S-100 board products now on the market. ☐ Linear addressable to 2 megabytes. ☐ 225 nano-second access time, maximum, 160 nano-seconds, typical. ☐ 295 nano- second read-write time, minimum. ☐ Bank selectable 16K increments. ☐ I/O port address bank selection. ☐ Configures for phantom deselection. ☐ Parity error detection, visual and/or interrupts. ☐ Bank selection compatible with CROMIX,™ CP/M2.2,™ MP/M,™ Alpha Micro, and other major systems.

4 PERSONALITY BOARDS.

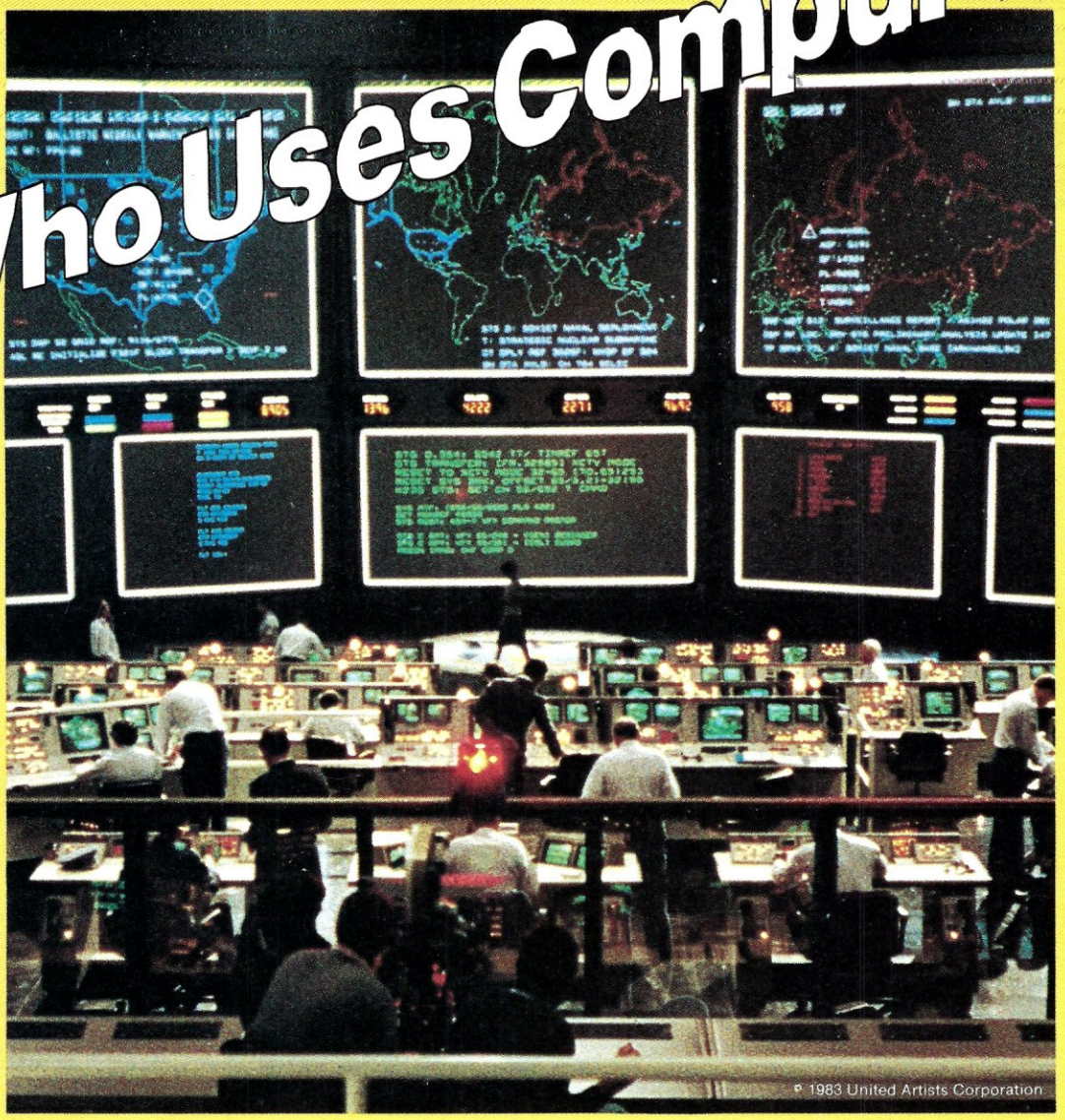
☐ Centronics printer. ☐ 8 inch floppy disk. ☐ 5¼ inch floppy disk. ☐ RS232 serial communications. ☐ Synchronous/asynchronous modem. ☐ Priam smart/smart E hard disk. ☐ Long distance serial communication (2000 ft @ 9600 baud). ☐ Shugart Associates Systems Interface (SASI). ☐ Clock/calendar. ☐ Konan David,Jr.™ hard disk. ☐ Archive tape drive.



INTERCONTINENTAL MICRO

1733 South Douglass Road,Ste.E
Anaheim, California 92806

Who Uses CompuPro?



More than 100 highly placed military strategists, one scientist and one very ingenious young computer hobbyist used **CompuPro** systems — in MGM/UA's new film, "War Games."

Two off-stage **CompuPro** systems ran a variety of on-screen computers, creating clever effects programmed by the film's computer consultant, Steve Grumette.

For a gigantic war-room set, Mr. Grumette programmed 120 monitors to flash hundreds of different battle plans at eye-boggling speed — all using just two **CompuPro** systems. He credits **CompuPro's** high operating speed and optional solid state disk drive with making that scene possible. "We couldn't have done it with any other machine because at that time there wasn't any other M-Drive," he said.

Reliability was another reason that Mr. Grumette chose **CompuPro**. As he put it, "When people are busy making a movie, they don't want to wait for somebody to run out and get a printed circuit board fixed."

On the set and on the job, **CompuPro** delivers performance, quality and reliability.

Call (415) 562-0636 for the **Full Service CompuPro System Center** nearest you.
CIRCLE 12 ON READER SERVICE CARD