THE JOURNAL FOR ADVANCED MICROCOMPUTING

$2.95

Microsystems

Volume 5 / Number 2

February, 1984

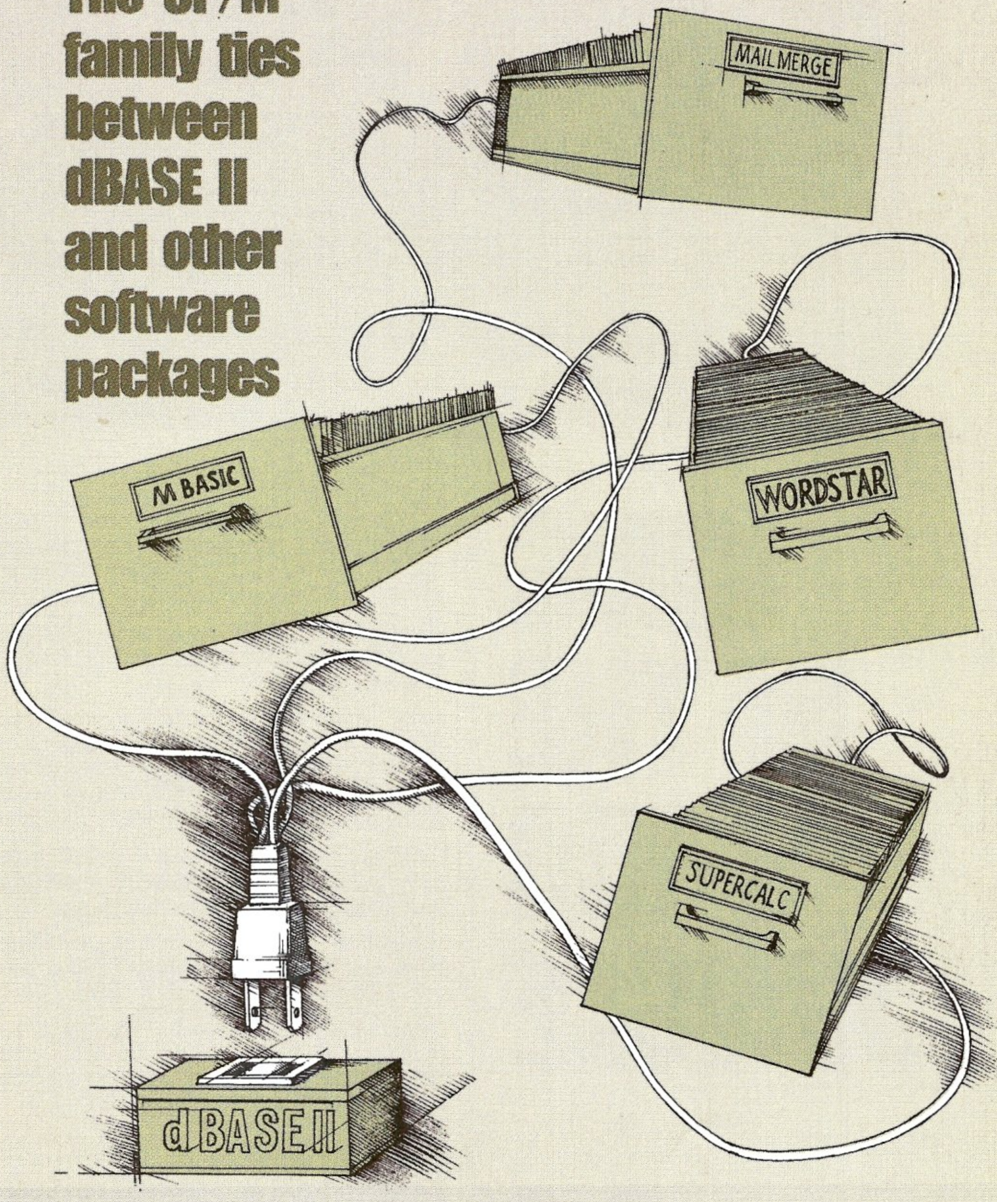
dBASE II
Overview
Interfacing
Datestamps

CP/M Plus
Bringing up Baby
RSX

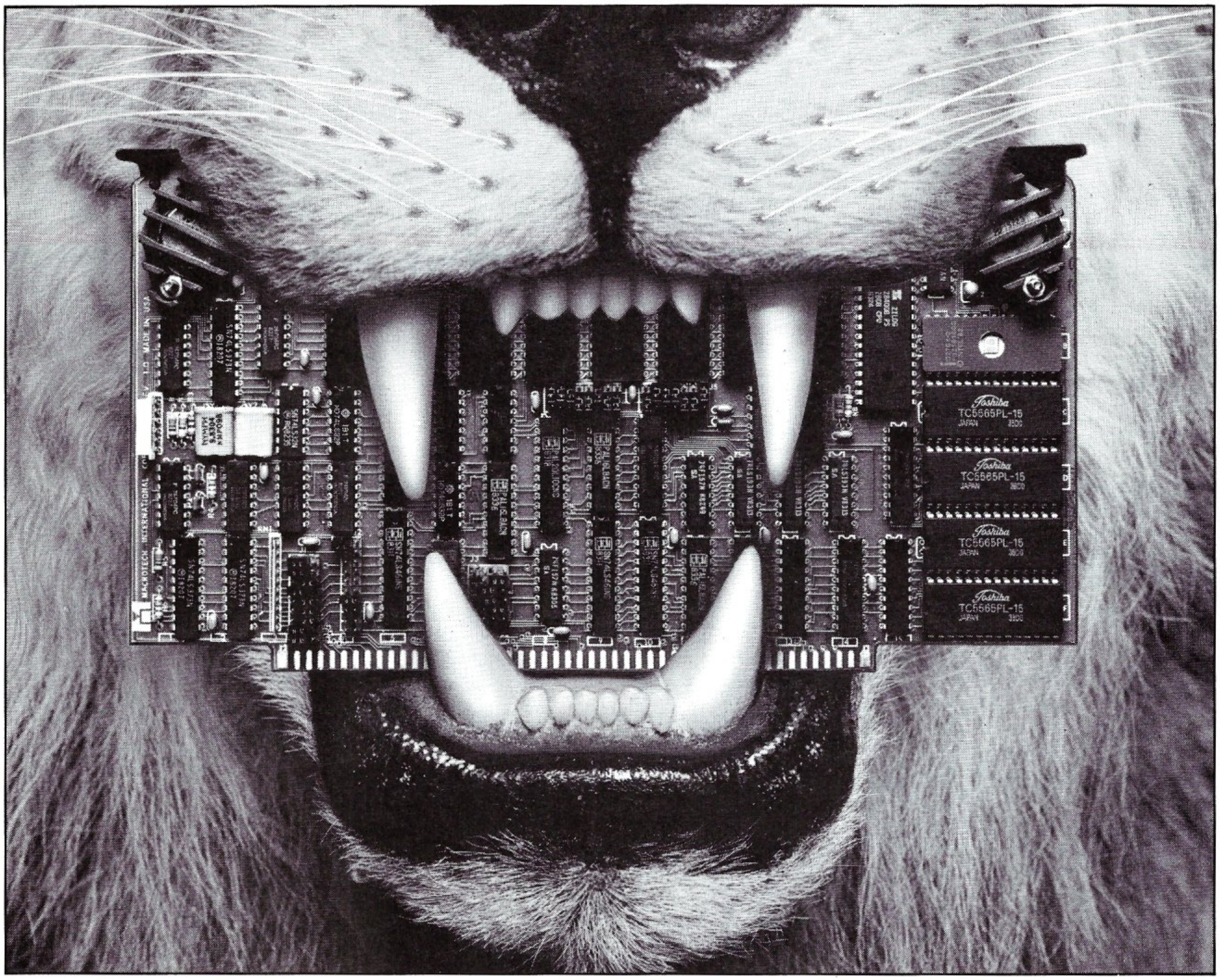
CP/M 2.2
Disk Parameters
Deblocking

Hardware Reviews
Cabinets

Strengthening The CP/M family ties between dBASE II and other software packages

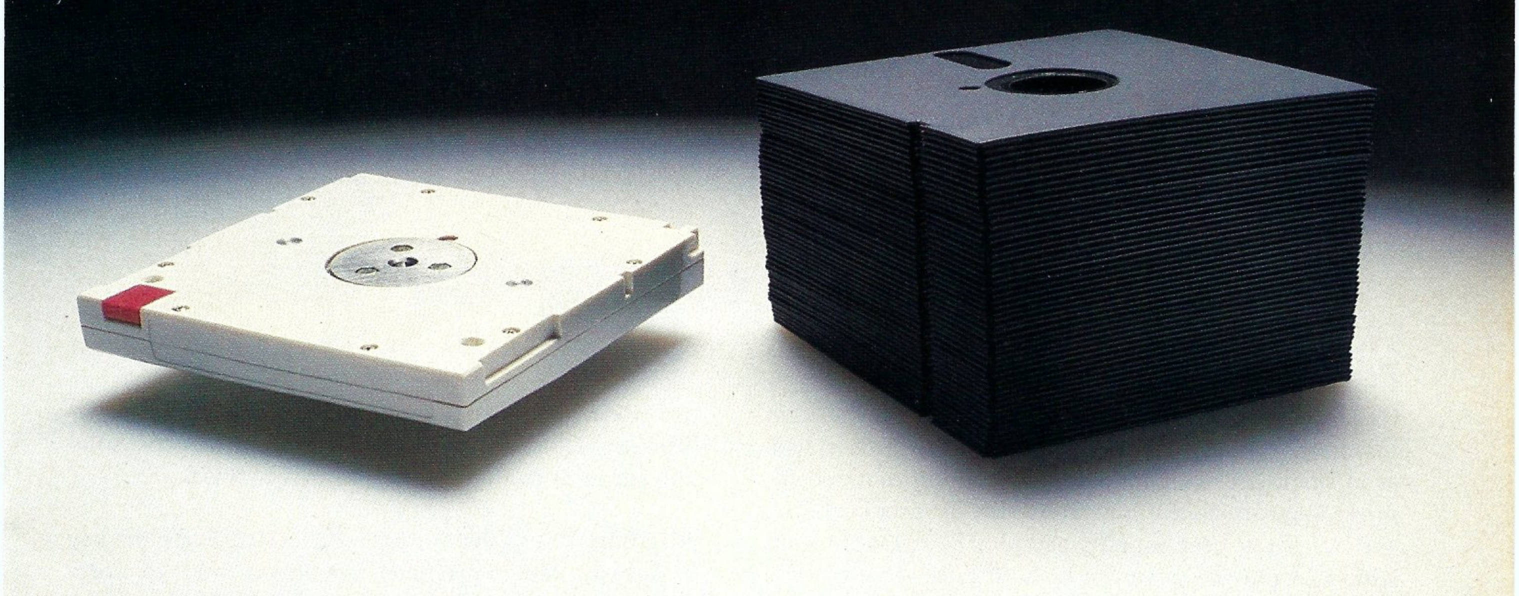


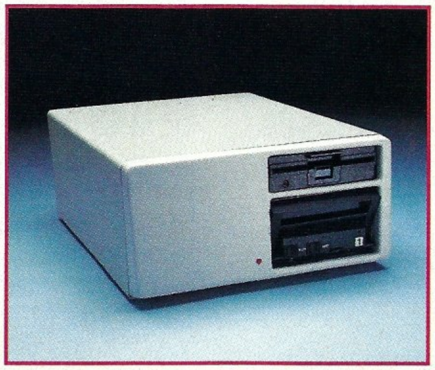
0 14024 14352 02



HOW DOES 1 = 50?

With SUPER STAR. . . The First S-100 Computer To Use a 5 MB Removable Cartridge.





Advanced Digital's new SUPER STAR™ is the first of a family of S-100-based computer systems to be equipped with a 5-1/4″ 5-MB Fixed/5-MB Removable Cartridge Winchester disk drive (10 MB on-line storage capacity) as standard. Now you can have the capacity of up to 50 floppy diskettes and the speed and convenience of 100% backup using the industry-standard Dysan 5-MB cartridge.

SUPER STAR is the ideal system for business, professional, and educational applications. Its CP/M* operating system (standard) gives you access to a vast library of readily available software programs. The six-slot motherboard allows plenty of room for expansion.

And as your business grows, SUPER STAR can grow with you. If you need more memory, just plug it in. SUPER STAR is based on the S-100 IEEE standard. To add more users, simply add Advanced Digital's SUPER SLAVE™ processor boards and TurboDOS* operating system. Now you've got an extremely powerful, multi-user, multi-tasking, multi-processing SUPER STAR computer system for up to four users.

SUPER STAR is truely a Super computer system. Look at these outstanding features:

- New Slim-Line Profile
- 5-1/4″, 10-MB Winchester Disk Drive, 5-MB Fixed, 5-MB Removable Cartridge
- 5-1/4″, Half-High, 48TPI Floppy Disk Drive (Osborne Format Compatable)
- Six Slot Motherboard
- Switch-selectable 110/220V Power
- Up to 4 Users With SUPER SLAVES And TurboDOS
- CP/M Operating System Standard
- One Year Warranty

SUPER STAR is the perfect system for applications requiring large data storage capacity, speed, performance, reliability, and dependable, convenient back-up capability. And at a suggested retail price of $5,000, SUPER STAR is a SUPER VALUE.

See the entire Advanced Digital product line, including the new SUPER STAR, at your local, quality computer dealer or contact:

ADVANCED DIGITAL CORPORATION

5432 Production Dr.,
Huntington Beach, CA 92649
Phone: (714) 891-4004
Telex: 183210 ADVANCED HTBH

**CP/M is a trademark of Digital Research Corp. TurboDOS is a trademark of Software 2000 Inc.*

SUPER STAR, SUPER SIX, SUPER SLAVE are trademarks of Advanced Digital Corp.

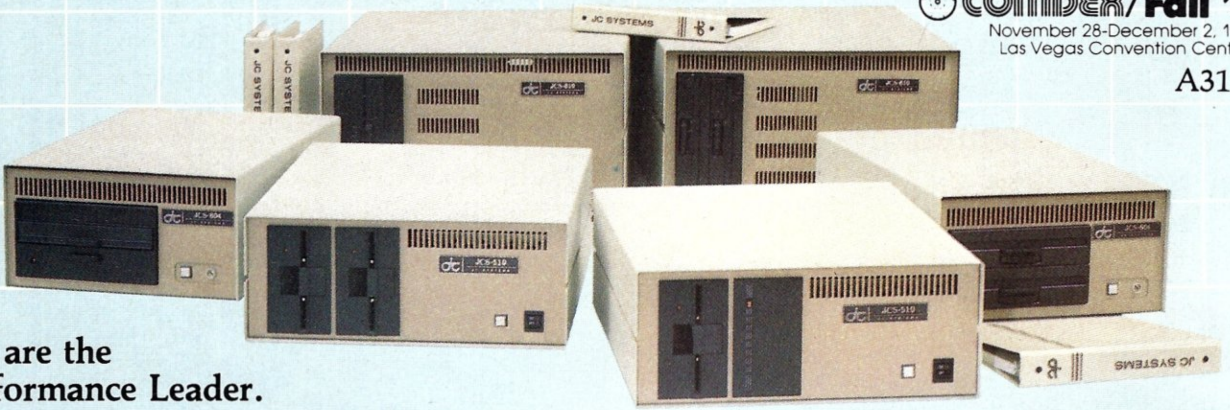
For European Distributor, contact: (714) 891-4004

CIRCLE 148 ON READER SERVICE CARD

THE MOST OUTSTANDING MULTIPROCESSOR, MULTIUSER ORCHESTRA.

PERFORMING ARTS of COMPUTER SCIENCE

See us at
COMDEX/Fall '83
November 28-December 2, 1983
Las Vegas Convention Center
A314

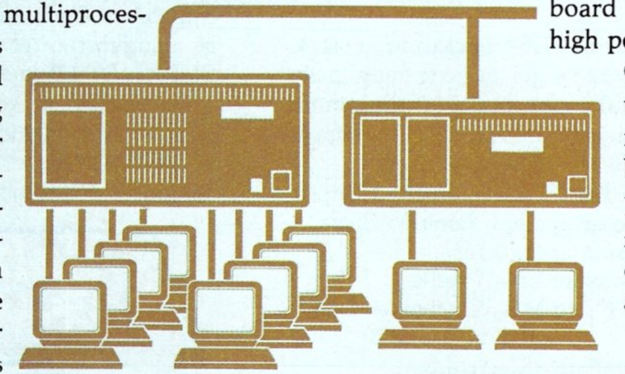


We are the Performance Leader.

Our Duet Product (**MCM*80 and DCM*80**), installed in thousands of multi-processor multi-user systems as well as in single user systems worldwide, certifies our field-proven performance. Our state-of-the-art products, first and best, have been soundly accepted by the various institutes, from the government to the universities, and utilized by the various applications, from office automation to control automation. Our product superiority is recognized for the following reasons.

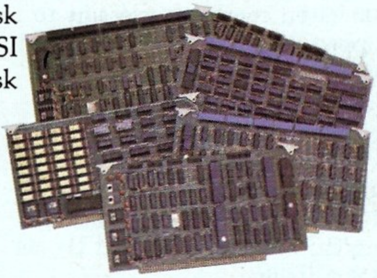
Identical Master/Slave SBC (Single Board Computer) architecture results in a prime advantage, never letting the entire system down. Not only that, the master/slave status is programmable, so it is truly possible to build redundancy in a multiprocessor system.

Unlimited Expandability differentiates our products from other multiprocessor based systems which are bound to the networking limitation. Our TURBODOS implementation on multiple networks, which connect multiple multiprocessor systems, matches the capacity of a mainframe, thus surpassing the mini computers.



Low Cost Local Area Networking has always been a subject but never a solution. Our approach to the LAN of SBCs, with integrated LAN control thru the high speed and reliable IPC (Inter-Processor Channel), provides an ultimate low cost solution for each computing station in a network.

Superior S-100 board Products are the key to superior system products. We now provide two prominent single board computers: **MCM*80**, based on Z80 (4 MHz, 6 MHz, or 8 MHz) processor family, and **MCM*186** based on 80186 (8 MHz) processor. There are two disk controller products, **DCM*80** and **DCM*80 II**. **DCM*80** is the first disk controller to integrate the floppy disk controller and the SASI (ANSI SCSI) hard disk host adapter in one board. **DCM*80 II** is **DCM*80** plus 8K Track Buffer with on-board DMA for high performance.



Our **RAM*80** is the first memory board which has quad-mode capability. It works not only as an 8-bit and 16-bit memory board, but also as a memory disk board.

LINK*80 is an intelligent I/O board designed for high performance foreground processing capability with on board processor (Z80 A) and memory buffer.

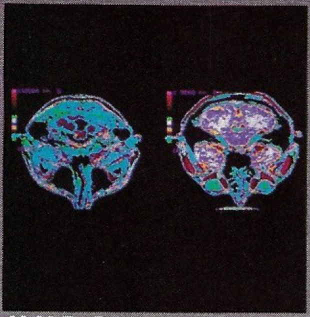
JC SYSTEM
469 Valley Way
Milpitas, CA 95035
408/945-0318 TWX 910-381-7041

JC Systems products are available worldwide thru 60 dealer bases which are growing continually. All of our products are serviced factory-direct or by our dealers. We maintain highly confident technical support, both in the hardware and software products, CP/M and TURBODOS.

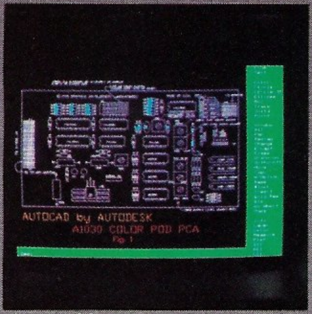
CIRCLE 238 ON READER SERVICE CARD

GDL's A-1000™

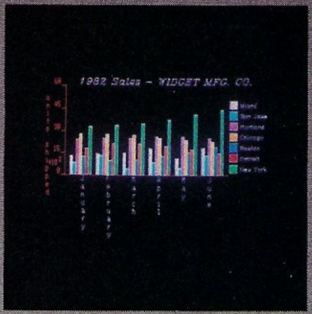
Why Pay More For Less?



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



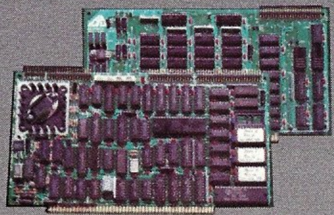
Autocad by Autodesk
R. H. Hymes, GDL



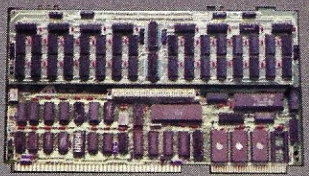
Supercalc Data Graph
Pacific Basin Graphics

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 Medicine, CAD, Education, Science, 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.
GRAFTALK	Business graphics package.
UGRAF	Business graphics software that interfaces to various spreadsheets, **DBASE II** and **Condor**.

OEM's, Dealers 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.

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 Kales 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.

DEALERS

CUSTOM COMP. TECH.
1 Craftsman Ct. Box 4160
Sedona, Ar. 86340
(602) 282-6299

MICROSET
9 Dearham Wood
West Hill, Ont. M1E 1R7
Canada
(416) 439-0474

FRONTENAC DESIGN
RD1 Box 147
Trumansburg, N.Y. 14886
(607) 387-5303

MENTZER COMP. SYSTEMS
1441 Rollins Rd.
Burlingame, Ca. 94010
(415) 340-9363

MOJAVE MICRO
17956-D Sierra Hwy.
Canyon Country, Ca. 91351
(805) 252-0155

GENESYS SYSTEMS
6 N. Goethe Box 277
Ellinwood, Kan. 67526
(316) 564-3636

EDWARDS ENGINEERING
1288 E. Mission Blvd.
Pomona, Ca. 91766
(714) 629-4349

MARTIAN TECHNOLOGIES

MOMS COMPUTING

W. W. COMPONENT SUPPLY
1771 Junction Ave.
San Jose, Ca 95112
(408) 295-7171

TOTAL ACCESS
2054 University Ave.
Berkeley, Ca. 94709
(415) 540-8066

TRACK COMPUTER CENTER
1514 University Ave.
Berkeley, Ca. 94703
(415) 845-6366

REPRESENTATIVES

MARTIAN TECHNOLOGIES
8348 Center Dr. Suite F
La Mesa, Ca. 92041
(619) 464-2924

MOMS COMPUTING
Bldg. 1055 Fort Cronkite
Sausalito, Ca. 94965
(415) 331-2043

GRAPHICS DEVELOPMENT LABORATORIES
2832 Ninth St., Berkeley Ca. 94710
(415) 644-3551
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.

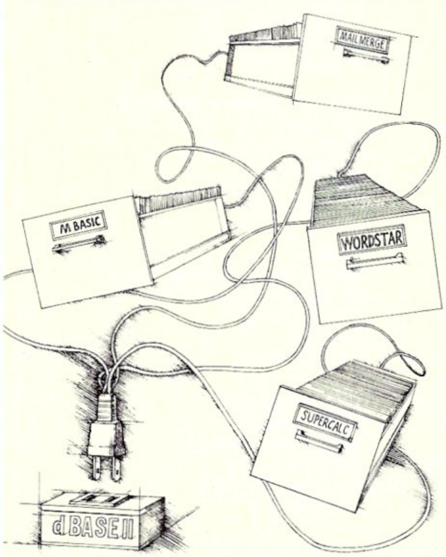
Teletek's HD/CTC Offers A Hard Disk Controller, Plus Cartridge Tape Controller, All On One Board.

TELETEK

4600 Pell Drive Sacramento, CA 95838 (916) 920-4600 Telex #4991834 Answer back-Teletek

CIRCLE 124 ON READER SERVICE CARD

Microsystems Feb



dBASE II

Strengthening the CP/M family ties between dBASE II and other software packages.

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
James Buklarewicz	art assistant
Jim Beloff	advertising director

ADVERTISING SALES OFFICES

New England, Midatlantic
Michael Mahana, *Microsystems*
Ziff-Davis Publishing Company
One Park Avenue
New York, NY 10016
(212) 725-7670

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

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

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

Canada
Frank Lederer, *The Pattis Group*
501 Eglinton Ave., E. #202
Toronto, Ontario
M4P 1N4
(416) 482-6288

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

CONSUMER COMPUTER & ELECTRONICS DIVISION

President	Larry Sporn
Vice President Marketing	Jeff Hammond
Vice President Circulation	Carole Mandel
Vice President General Manager	Eileen G. Markowitz
Creative Director	Peter J. Blank
Promotion Manager	Ronni Sonnenberg

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. Hefferman, *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* Rory Parisi, *Vice President* Selwyn Taubman, *Treasurer* Bertram A. Abrams, *Secretary*

MICROSYSTEMS (ISSN #0199-7955) is published monthly by Ziff-Davis Publishing Company, One Park Avenue, New York, N.Y. 10016.

Second Class postage paid at Los Angeles, CA 90052 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 Dept., One Park Avenue, New York, NY 10016 or call Jim Beloff at (212) 725-3452.

LOOKING FOR A MULTI-USER, HARD DISK COMPUTER SYSTEM YOU CAN AFFORD?

It's the Morrow Decision I.™

Starting at $5,495 retail. That's the basic Morrow 10MB system with 256K of RAM. Including a 14-slot S-100 (IEEE-696) motherboard, a 400KB floppy, one parallel and three serial ports.

A bundle of software. Beginning with our multi-user Micronix™ operating system, with both UNIX™ and CP/M® 2.2 compatibility. The Decision I also includes WordStar® word processing, the Correct-It™ spelling checker, Microsoft® BASIC 80, the LogiCalc™ electronic spreadsheet and Personal PEARL,™ a relational data base manager.

Economical expandability. For individuals with growth on their minds, the Decision I offers very affordable possibilities. Another $2,000 buys 6 MB more memory, 256K more RAM and the ability to add three more users.

Give us a call. We'll get you the name of your nearest Morrow dealer. We'd like to introduce you to Morrow's complete line of S-100, multi-user microcomputers, disks and peripherals. Systems you can afford.

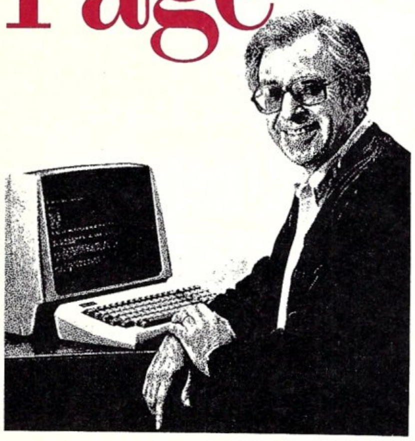
MORROW

600 McCormick Street
San Leandro, CA 94577
(800) 521-3493
(415) 430-1970 In California

Decision I, Correct-It, and Micronix are trademarks of Morrow, Inc.
CP/M is a registered trademark of Digital Research, Inc.
Microsoft is a registered trademark of Microsoft Corporation.
WordStar is a registered trademark of Micro Pro, Inc.
Personal PEARL is a trademark of Relational Systems, Inc.
UNIX is a trademark of Bell Laboratories, Inc.
LogiCalc is a trademark of Software Products, Int'l.

CIRCLE 183 ON READER SERVICE CARD

Editor's Page



A response to the criticism of the IEEE: A guest editorial

by Robert G. Stewart

In the October 1983 issue of *Microsystems,* page 10, in his "News and Views" column, Sol Libes—the editor of *Microsystems*—gives readers instructions as to how to purchase a copy of the IEEE/696-1983 bus standard, but then takes the IEEE to task. With his permission, I quote his statement:

"It should be noted that the IEEE has refused to give me permission to reprint that standard both in *Microsystems* and in my book on interfacing to the S-100 bus, as we did with the proposed version. And furthermore, even though I am a coauthor of the standard, I too must pay to receive a printed copy of the standard. The IEEE has taken the position that they own the copyright and are free to distribute it as they see fit. And the authors of the document, who have worked on it without compensation, have no say in how the document is to be distributed. My personal opinion is that the IEEE is operating as a business and is protecting their vested interests at the sacrifice of serving the needs of the computer engineering community. I have therefore dropped my membership in the IEEE and will no longer participate in their money-making efforts."

After reading this, I discussed the above situation with Sol and with Burt Stanleigh of the IEEE Standards Office in New York. Let me tell it as I see it.

The key issue for the computer engineering community is the development of badly needed standards in the computer field. The conflict of forces within the commercial markets has led, through the decades, to an unending sequence of unnecessary hardships to professional computer engineers and scientists, and the end users. This has been coupled with an equally unending sequence of improvements to the hardware (and sometimes the software) which has benefitted them. It is a mixed and muddy picture, and the question is: How do we keep the good and eliminate the bad? It seems to me the IEEE, as well as other interested organizations and societies, should be strong and effective leaders helping to apply the talents of their members and staff to develop the standards needed *in a timeframe relevant* to the technology.

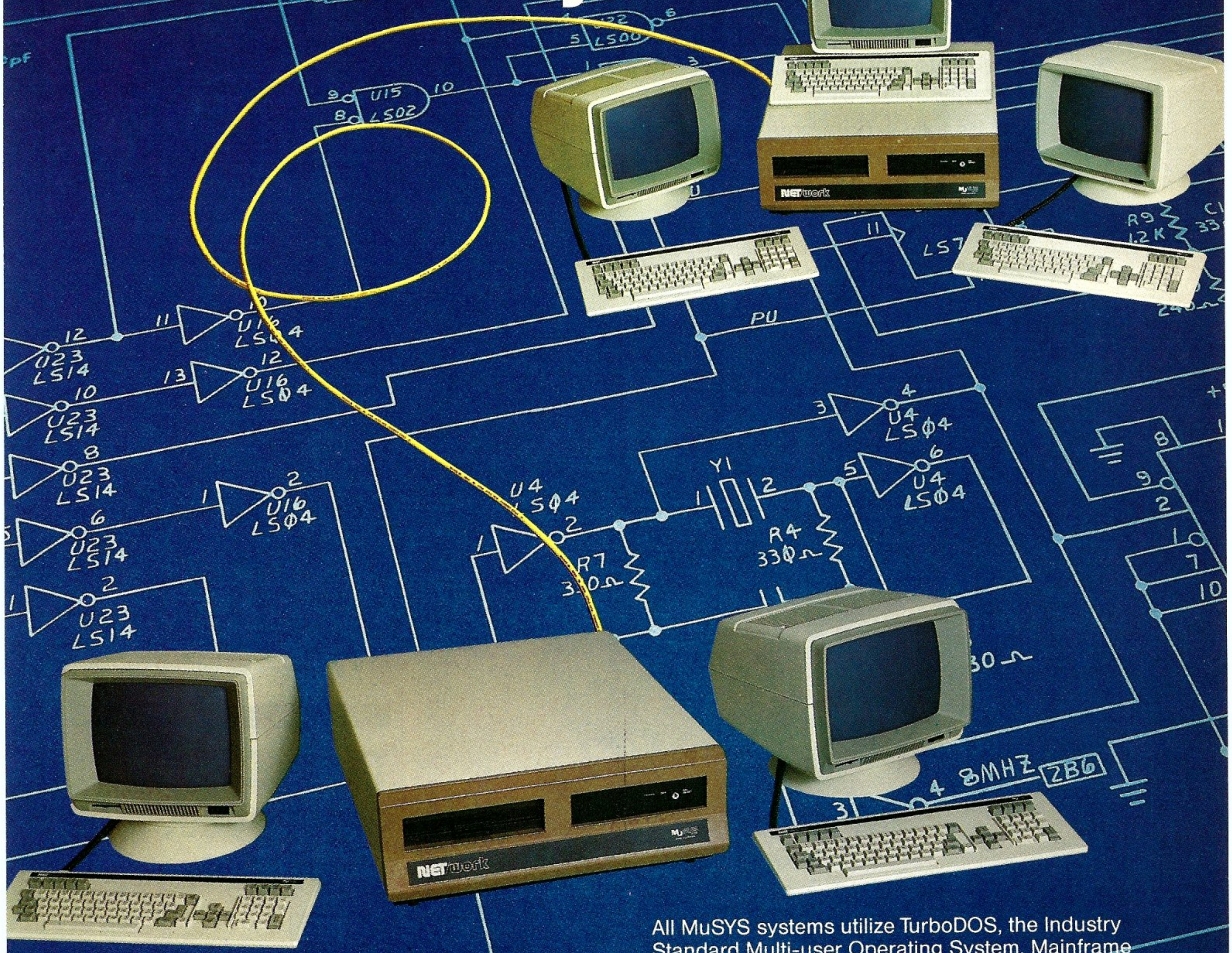
Sol said the IEEE Standard Office was very arrogant and refused to send him a copy of IEEE 696. Sol acted as secretary for the working group for about two years, and had spent much of his own time and money to travel cross country to attend meetings. Burt Stanleigh said it is inconceivable that anyone in the Standards Office wouldn't give Sol a free copy (I asked Burt to send him several free copies). Burt indicated that the chairman of the working group had so far failed to send him a list of the addresses of the working group members so that he could send them copies.

The IEEE Standards Office *does* give other organizations the right to reprint standards provided they appear in a form which is different from that published by the IEEE. For example, if elaborations, analyses, and comments are interspersed, then permission to reprint is granted. The document reprinted by Sol Libes in his book and in his magazine was a copy from *Computer* of the proposed draft as it existed in the working group in 1979. Some changes were made to that document by the working group (and were summarized in the November 1982 and February 1983 issues of *IEEE Micro*) before the final standard was promulgated. Two major reasons for publishing drafts are to expose the document to a wide circle of readers and to solicit comments while the working group can still change the draft. That reason was very much evident in the case of the 696 draft. Furthermore, publication led to a ground swell of implementation which resolved a serious "prima donna" problem with that bus.

The question of copyright ownership is quite simply answered. *Every* meeting of the people involved in developing IEEE 696 was clearly as an activity fostered and sponsored by the IEEE Computer Society and approved as a standard development project by the IEEE Standards Board. The drafts were created by very dedicated individuals *based on committee-sanctioned decisions.* Moreover, the longterm maintenance will be carried out by an IEEE Computer Society Committee, whether the original participants are still involved or not. Thus there is a very reasonable basis for the contention that the copyright for the standard should belong to the IEEE rather than the individuals involved.

Sol observes that the IEEE seems to be operating as a business. Now I too have seen the dollar signs in the eyeballs of both IEEE staff and volunteers. I wonder about that. When I was a chapter chairman here in the Santa Clara Valley I had trouble keeping our chapter earnings small! Remember the IEEE is a nonprofit organization. We presented a Saturday full day tutorial with copies of lecture notes and a delicious buffet lunch with wine at the Stanford Faculty Club for a member cost of

It Meets Your Needs...
Whatever They Are



Large Multi-user Capability

A truly professional, reliable solution—the NET/work 8816 has been specifically designed for installation in demanding multi-user applications. The low starting price allows you to start with two users and grow later to many thousands without the expense of replacing existing equipment or software as you grow. We offer three types of Local Area Networking, including Ethernet, with full shared multi-system resources. AND IT WORKS. Gone are the bottlenecks that make shared-processor multi-user systems too sluggish for real time applications. What's more, we offer an unbeatable combination in a video terminal with our NET/worker. You get styling, operating comfort, value and reliability.

All MuSYS systems utilize TurboDOS, the Industry Standard Multi-user Operating System. Mainframe capability at microcomputer prices, and compatible with virtually all CP/M Software. NET/work systems give you advanced features such as 8MHZ processors, high speed hard disk drives with storage capacities up to 280MB, and 16 bit future upgrade path.

This system meets your needs...WHATEVER THEY ARE. Call or write, MuSYS Corporation, 1752-B Langley, Irvine, California 92714. (714) 662-7387 (toll free outside California 1-800-852-5362. TWX 910-595-1967. Cable MUSYSIRIN.

Dealer and OEM inquiries welcome.

MuSYS CORP

We design the future.

CIRCLE 83 ON READER SERVICE CARD

NET/work is a trademark of MuSYS Corporation. TurboDOS is a trademark of Software 2000, Inc. CP/M is a trademark of Digital Research, Inc. Ethernet is a trademark of Xerox Corporation.

EDITOR'S PAGE

Continued from page 8

$6.00, which was $1.00 below actual cost. We still made money because so many nonmembers came. If you've ever eaten at the Stanford Faculty Club you may understand part of the inducement.

In comparison, on November 7, the IEEE presented a 4½ hour short course by satellite on robot sensing and intelligence and charged IEEE members $130 without lunch. I consider such exhorbitant fees an affront to the members of the IEEE, since the organization is not providing services to them at costs that make sense. Rather the costs are so high they act as a strong deterrent to the members willing or able to participate in the event. The $7 charge for a copy of IEEE 696 is reasonable at least.

Burt Stanleigh tells me the IEEE Standards Office receives no allocation of funds from member's dues. The only income to the office is from sale of standards, and funds received to directly support some standards development activities. Thus it is easy to understand the reticence of the Standards Office to freely grant permission to reprint final standards. So far about 600 copies of IEEE 696 have been sold. That is an income of about $4,200. It certainly does cover the cost to typeset, create artwork, and print a standard. As a paying subscriber to Sol's fine magazine, I can't help noting that there must be similar costs associated with it, even though

The IEEE is owed a debt of gratitude by its participants in the computing field.

he has advertising to defray some.

But the key point, Sol, is not to resign your membership, but rather to become an insider—to work constructively within the organization to make it responsive and change in those ways you think it should. The IEEE is owed a great debt of gratitude by the entire

EDITOR'S PAGE

Continued from page 10

group of participants in the electrical engineering and computer field—users, governments, manufacturers, professionals, and students. One way I show my gratitude is by renewing my membership each year, even when it's hard. The journals of the IEEE, which fill our libraries and bookshelves, are a never-ending bulwark to our calling. The conference and meetings are great meeting and learning places.

However, the IEEE is not perfect, and improvements are needed. The load imposed on volunteers in some of our standards working groups is excessive. There has been no IEEE staff help to any of our Computer Society working groups in developing standards, even when help was requested. The IEEE publications could be less pompous, more readable, and more relevant to engineers and programmers. Therefore, I urge you, Sol, not to resign but to help solve the problems that do exist. Who knows, you might win.

Sol Libes Responds

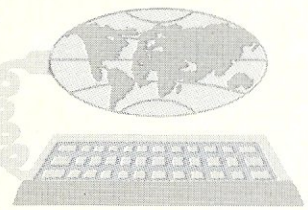
This editorial appeared in the December 1983 issue of *IEEE Micro,* an IEEE Computer Society magazine. It is reprinted with the author's permission.

Dr. Stewart is First Vice President for Technical Activities of the IEEE Computer Society and previously headed up its IEEE Computer Standards Committees. He was also a member of the working group that developed the IEEE 696 bus standard. Further, Bob was instrumental in my getting permission from the IEEE to reprint the S-100 proposed standard in the January 1980 issue of *Microsystems.* Without his help it never would have happened. Thus our readers owe him a debt of gratitude.

I still stand by the view that I expressed in the October 1983 issue. The IEEE operates a very large publishing business and operates it in a very money-making fashion. *Microsystems* is a commercial enterprise as well, operated by the Ziff-Davis Company. However, we grant permission to many other publications and organizations to reprint articles from our magazine. Further, most *Microsystems* authors retain reprint rights to their articles.

The IEEE is a non-profit organization that, among its other responsibilities, is supposed to educate computer engineers. Restricting the dissemination of a computer standard seems to me contrary to this responsibility.

Dr. Robert Stewart, Stewart Research Enterprises, 1658 Belvoir Dr., Los Altos, CA 94022



News & Views

Random rumors and gossip, plus a view of the industry's latest trends

by Sol Libes

Morrow Designs is expected to announce transportable and portable machines compatible with the IBM PC. . . . The first IBM PC clone from Taiwan is expected shortly from Multitech Electronics. They are already selling an Apple clone. . . . North Star is soon expected to introduce a multiuser system that is compatible with the IBM PC and supports 12 users. . . . IBM is rumored working on versions of the PC-XT as integrated data/voice workstations and PABX peripherals. . . . Microsoft is shortly expected to announce its new version of MS-DOS with window capability. Delivery to OEMs is expected before the summer. Digital Research has already announced that it will add windows to CP/M-86.

P-CP/M in beta test

Digital Research of Japan has disclosed that it is beta testing preproduction home computers using the new CP/M-in-ROM operating system. Reportedly, these units use an IC that integrates the system software ROM and processor into one IC and will make possible the selling of CP/M-based systems for under $250. This version of CP/M is being referred to by DRI as "P-CP/M".

Systems running under P-CP/M are expected to compete with home computers that have the new MSX operating system from Microsoft. Several Japanese personal computer makers have already announced their intention to introduce a system based on MSX, and Fujitsu Ltd. has already introduced a personal computer running the Microsoft MSX operating system. Sales of the unit are currently being limited to Japan.

Large LCD screen introduced

CrystalVision, Sunnyvale CA, is the first company to announce a flat-panel LCD display device capable of displaying 25 lines by 80 characters. In fact, the company is promising that initial production quantities will be available this month at $345 in 1,000 quantity lots.

The unit will have an 8″ diagonal measurement (1″ larger than the Osborne Executive) and will store permanent images. Thus it will not have to be refreshed, as do the current LCD displays. It will provide a pixel density of 640 × 250 (IBM PC has 640 × 200) and a nearly 90-degree viewing angle (currently LCD displays have a typical 20-degree viewing angle). The unit is less than an inch thick and is 10″ × 7″ overall. CrystalVision also claims that the contrast ratio is far better than current LCD displays.

This appears to be a significant jump over LCD devices that were shown at the recent Japan Electronics Show by Japanese companies and may put the U.S. back in the running in display technology.

IBM drops Josephson device project

IBM has halted all development work on Josephson technology and closed its pilot production plant. IBM had hoped to use Josephson devices in its next generation of high-performance mainframe computers. These superconducting devices promised operating speeds 50 times faster than current systems.

The nearly 20-year-old project had cost IBM an estimated $100 million and had gotten to the point where IBM had built prototype cache and RAM memory systems using the devices. The decision to cut back was reportedly based on the rapid advances in conventional silicon technology and the problems related to producing reliable Josephson devices.

Osborne news

Osborne is rumored to be negotiating with two computer companies seeking to acquire the company and provide the funds for it to continue in operation and introduce its IBM PC compatible version of the Executive and a new portable computer. Working prototypes of both systems are reported to exist. Osborne is reported to have a substantial quantity of Osborne I and Executive models in inventory, both assembled and in parts.

Robert Jaunich, president of Osborne Computer, has stated that if the company is forced to liquidate, unsecured creditors will at best get 10 to 15 cents for each dollar owed.

The war of the windows

Visicorp demoed their VisiOn window-type frontend software package for the IBM PC in November of 1982 at the Comdex show in Las Vegas, promising delivery in June of '83. The package is designed to make personal computers more "user friendly" and employs a mouse, rather than a keyboard, for system control. These packages allow users to work with more than one application software package at the same time. To some degree, this provides the features of the Apple Lisa and Xerox Star.

However, the task of moving from a demo package to actual, saleable software proved to be a more formidable undertaking than they had originally envi-

NEWS & VIEWS

Continued from page 12

sioned. The result was that initial shipments to dealers did not begin until November '83, a year later. VisiCorp figures that it took them three years to develop the package, with an investment of $12 million. In the interim, Quarterdeck Software introduced their "DesQ" windowing package for the IBM PC, claiming that it will be released shortly. By this summer, if not sooner, Microsoft is expected to release a new version of MS-DOS with window capability. And Digital Research has also disclosed that it, too, is working on windows for its CP/M-86 operating system and that it might be out even before Microsoft's windows. Windowing packages are also expected from some other sources.

It is unlikely that all of these suppliers will be successful in the marketplace, and intense competition among them is expected. The real battle for leadership is not expected to occur until late this year.

User group news

In the October 1983 issue of *Microsystems,* we published a listing of CP/M User Groups. This prompted letters from several groups that did not get on the original list. We published these omissions in the last issue. Here are some more received during this past month.

• **The Valley Computer Club,** *Box 6545, Burbank CA 91510* has a very active CP/M user group and a group devoted to CP/M Plus (version 3.0). For information, call Jim Stine (213) 371-8836.

• Ben Cohen wrote to say that there are over 50 Osborne User Groups. The largest is **FOG (First Osborne Group,** *Box 3474, Daly City CA 94015* with about 6,000 members, an extensive disk library, and monthly newsletter. They have 53 affiliated local groups. Ben recommends writing to FOG to find out about the Osborne user group nearest you.

• Glen L. Moulder wrote to inform us of the **Morrow User Group** of Washington DC (MUFW). For information, write: **MUGW,** *3501 Hamilton St., Suite 4, Hyattsville MD 20782,* or call (301) 277-3760.

• **UCSD Pascal System User's Society (USUS),** *Box 1148, La Jolla CA 92308.* The group has 27 volumes of source code software in its library, publishes a quarterly newsletter, EMail (Telemail/Telenet), and a CompuServe SIG (MUSUS).

• **The Yavapai County CP/M User's Group,** formed this past summer, includes owners of Osborne, Kaypro, IMSAI, and Commodore-64 computers, as well as a VAX running CP/M. Contact *Julie G. Woodman, P.O. Box 68, Kirkland AZ 86332; (602) 442-3820.*

• The Nevada Cobol User's Group has had to raise its dues to $18/year.

• **Sacramento Area Users of TeleVideo Equipment (SAUTÉ),** a small, newly formed group, is for users of TeleVideo systems interested in learning to use the available software and in developing utility and applications programs. The group meets every two weeks; meetings usually have a lecture followed by an open discussion and exchange of information. Dues are $12/year. An exchange of information or membership with any similar users groups would be welcomed. Contact *Ron Odon, SAUTÉ, 8200 Longdon Circle, Citrus Heights CA 95610.*

• A newly organized **EPSON QS-10 Users Group** will publish a newsletter, offer a co-op for discounts to members, and organize public domain programs for distribution. The newsletter will carry free ads for members, plus hardware and software reviews. Dues: $20/year; family, $25; student, $15. *Contact Richard Shoemaker, P.O. Box 1076, Lemont PA 16851.*

• And last there is the **Napa Valley CP/M Users Group** (NVCPMUG). You can call their bulletin board system at (707) 257-6502. or write: **NVCPMUG,** *Box 4096, Napa CA 94558.*

New public domain software

SIG/M (Special Interest Group for Microcomputers, Amateur Computer Group of New Jersey, Inc.) has issued

NEWS & VIEWS

Continued from page 13

six new volumes of public domain software, bringing their total up to 151 volumes. The new volumes are:

Vol.	**Description**
146	CP/M-86 versions of FIND, PRINT, SD, VFILER & WHATSNEW
147	More CP/M translations of popular SIG/M CP/M-80 programs
148	Pascal-Z programs including InitLisp
149	CP/M-86 versions of Small-C, MODEM, XMODEM & LU written in C
150	Bulletin Board system written in C & Z8000 Forth
151	DEC Rainbow Utilities & Modem

For complete information of SIGM software, send $2 ($2.50 foreign) for printed catalog to: **SIG/M,** *Box 97, Iselin NJ 08830.* **P/D Software,** *4691 Dundas Street West, Islington Ontario, Canada M9A 1A7;* (416) 239-2835 is distributing the CPMUG and SIG/M software libraries in 25 different disk formats. These include 8", Apple, Kaypro, Osborne, IBM and others. Prices range from $10 to $20, depending on disks required.

VisiOn front end software package uses a mouse for system control.

Rick Conn has completed version 3 of ZCPR. It will be released in February, 1984, in the SIG/M library.

The New York Amateur Computer Club, *Box 106, Church Street Station, NY NY 10008,* has released four new volumes of software for systems running MS-DOS and PC-DOS. The disks are $6 postpaid (add $3/order for foreign). A printed catalog is $10 ($15 foreign). The new volumes are:

Vol.	**Description**
39	Editor, word processor and CAI programs
40	Statisical package, modem and communications programs
41	Galaxy Trek game
42	"Cheap Assembler", RAM disk, WordStar utility, Num/Caps lock switch, and sort/list programs

The S-100 Bus

Diagnostic hardware for front panel-less S-100 machines

The new S-100 frames, with their 6 MHz terminated motherboards, constant-voltage power supplies, and venturi-styled cooling vents are a great deal more sophisticated than the original old Altairs and IMSAIs were, but most of them lack one of the very ideas for which the S-100 machine was originally designed: the front panel.

Most S-100 manufacturers have done away with the front panel because it isn't really needed for normal operations and can really be a hindrance, especially in commercial use with unskilled operators. Turnkey systems are much better suited to these sorts of environments, where the only front-panel control is usually a big red RESET button.

Even the new S-100 standard (IEEE-696, that is) frowns on the dinosaur front panel. Many of the old S-100 front-panel related pins have been done away with or changed to more closely conform to the front panel-less world.

Unfortunately, the original reason for the front panel is still with us. S-100 computers, like all computers, occasionally stop working, and troubleshooting a dead S-100 frame without a lot of test equipment can be a very difficult task. At least a front panel gives us a window into the machine.

Fortunately, most of the functions of the old front panel can be easily added to any IEEE-696 frame. In fact, the two most important functions of the old front panel, the ability to monitor bus lines and to stop and start the processor, can be added to any S-100 frame with just a few simple ICs, LEDs, and a couple of switches.

Figure 1 shows the simple circuit used to monitor a single S-100 bus pin. To monitor all of the address and data lines, you'd need 40 of these (24 address and 16 data) circuits, which could be a bit of a wiring job. In addition, you might also like to monitor some of the status lines, the power fail line, the ERROR* line, and many others, depending on your needs.

Of course, these are all just useless flashing lights unless the machine can be stopped to allow the bus lines to be examined, which is why the circuit in Figure 2 is needed: to allow the machine to be stopped long enough for you to examine the lights of the "front panel."

This circuit also lets the machine be single stepped from one instruction to the next, so that you can now see not only the current address of the machine's PC, but also where it is going. Notice that this rather remarkable feat is easily accomplished on the S-100 bus by just lowering the Auxiliary Ready line (XRDY, pin 3) to tell the master processor to wait before executing the next bus

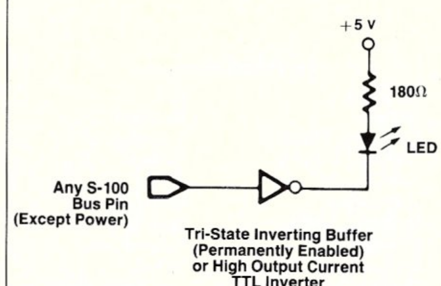


Fig. 1

Figure 1. *LED display circuit for any S-100 bus line.*

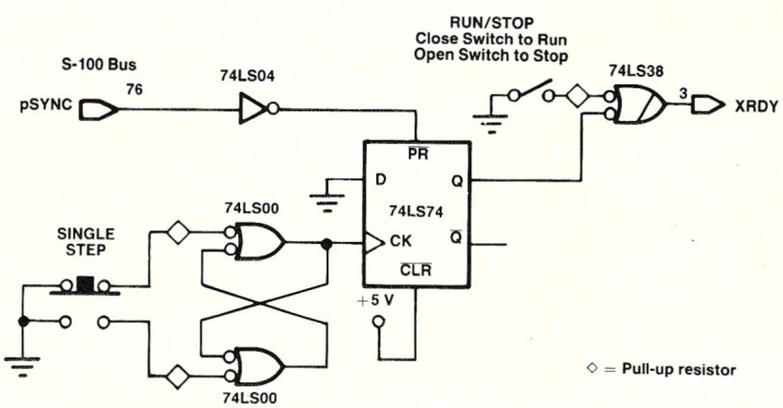


Fig. 2

Figure 2. *Run/stop and single-stepping circuit for S-100 systems.*

by Dave Hardy

THIS IS THE PASCAL COMPILER EVERYBODY'S BEEN WAITING FOR...

EVERYBODY EXCEPT THE COMPETITION!

Extended Pascal for your IBM PC, APPLE CP/M, MS DOS, CP/M 86, CCP/M 86 or CP/M 80 computer features:

- Full screen interactive editor providing a complete menu driven program development environment.
- 11 significant digits in floating point arithmetic.
- Built-in transcendental functions.
- Dynamic strings with full set of string handling features.
- Program chaining with common variables.
- Random access data files.
- Full support of operating system facilities.
- And much more.

ORDER YOUR COPY OF **TURBO PASCAL** TODAY TO TAKE ADVANTAGE OF OUR INTRODUCTORY SPECIAL.

For Visa and MasterCard orders call toll free

1-800-227-2400 X 968

IN CA: 1-800-772-2666 X 968

(lines open 24 hrs. a day, 7 days a week)

Dealer & Distributor Inquiries welcome.

	Turbo Pascal	IBM Pascal	Pascal MT+
PRICE	49.95	300.00	595.00
Compile & Link speed	**1 second!!!**	97 seconds	90 seconds
Execution speed	**2.2 seconds**	9 seconds	3 seconds
Disk Space 16 bit 8 bit	**33K w editor!** **28K w editor!**	300K + editor Not Available	225K + editor 168K + editor
8 and 16 bit	**YES**	NO	YES
built-in editor	**YES**	NO	NO
Generate object code	**YES**	YES	YES
One pass native code compiler	**YES**	NO	NO
Locates RunTime errors directly in source code	**YES**	NO	NO

Benchmark data based on EightQueens in "Algorithms + Data Structures = Programs" by N. Wirth, run on an IBM PC.
Turbo Pascal is a trademark of Borland International. MT+ is a trademark of MT MicroSystems. IBM is a trademark of International Business Machines.

LEARN TO WRITE A SPREADSHEET

Our Introductory offer includes MICROCALC, a spreadsheet written in **Turbo Pascal**. It will be on your disk, and ready to run. And we've included the source code to show you exactly how a spreadsheet is written! (Introductory offer expires March 1, 1984)

Turbo Pascal includes a 250 page bound manual with extensive explanations and many illustrative examples.

Turbo Pascal $49.95 + $5.00 shipping per copy.

Check ______ Money Order ______
VISA ______ Master Card ______

Card #: ______
Exp date: ______ Shipped UPS

BORLAND INTERNATIONAL

Borland International
4807 Scotts Valley Drive
Scotts Valley, California 95066

My system is: 8 bit ______ 16 bit ______
Operating system: CP/M 80 ______
CP/M 86 ____ MS DOS ____ PC DOS ____
Computer: ______ Disk Format: ______
Please be sure model number and format are correct.

NAME: ______
ADDRESS: ______
CITY/STATE/ZIP: ______
TELEPHONE: ______

California residents add 6½% sales tax. Outside North America add $15.00. Checks must be on a U.S. bank, and in U.S. dollars. Sorry, no C.O.D.

QDP-300
The peace of mind computer.

Introducing our third generation computer... the all-new QDP-300. Now, you can rest assured you've found the most advanced microcomputer on the market today. The QDP-300 is a user-friendly system – its on-line "Help" system gives even untrained operators access to its full power. It uses CP/M* and MP/M* operating systems that assure the user of an abundance of compatible software. The QDP-300 even has a dual voltage system that allows worldwide operation (115 or 220 volts) at the flick of a switch. For word processing, financial forecasting, engineering design and manufacturing, inventory, payroll, bookkeeping and more, the QDP-300 will work hard for you for years and years to come. It's also comforting to know that the QDP-300 is fully expandable and readily upgradeable as your computer needs grow. There's even more. It might make you sleep better to know that the QDP-300 is backed by one year, on-site warranty with service provided by General Electric Apparatus and Engineering Services, with more than 50 service locations nationwide. If these features and all of the others we've built into the QDP-300 don't bring you peace of mind, then the low price tag will.

- **More Flexibility –** Easily upgradeable to 16 bit capability which gives the user 8 or 16 bit operation. IEEE 696/S-100 Bus.
- **More Speed –** Unique "cache memory" disk operation makes the QDP-300 one of the fastest operational 8 bit systems on the market.
- **More Power –** Advanced single board design utilizing Z80B** CPU operating at 6 MHz.
- **More Storage –** Dual 8" floppy disk drives provide a total of 2.4 MB of formatted storage (10 MB and 15 MB internal hard disk system optional; 30 MB external hard disk system also available.)

Call or write for complete specifications and literature.

QUASAR DATA PRODUCTS
QDP computer systems
10330 Brecksville Road, Cleveland, Ohio 44141
(216) 526-0838, Telex: 241596

Specifications subject to change

*CP/M and MP/M are trademarks of Digital Research Corp.
**Z80B is a trademark of Zilog Corp.

S-100 BUS

Continued from page 16

cycle. The processor Synchronize signal (pSYNC, pin 76), which indicates the start of every bus cycle, is used to detect when a single bus cycle has occurred, so that the flip-flop can stop the processor via the XRDY line. Each time the flip-flop is clocked by pushing the "Single-Step" button, the machine will be allowed to execute a single bus cycle, then forced into a wait mode.

Figures 1 and 2 allow us to stop the machine whenever we want, and to examine its bus lines. Although they are certainly very useful functions, it would be tedious, to say the least, to single step through an entire computer operation like, for example, loading and running a 30K program. The next piece of hardware needed, then, is something that would *automatically* stop the machine whenever certain conditions were met. Figure 3 is a circuit that does just that, by stopping the processor whenever a certain bus address is reached. Its output is connected to the Preset line of the flip-flop in figure 2, in place of the inverted pSYNC signal. With the address line decoders, the flip-flop will now see only the pSync signal when the address set on the switches in Figure 3 is

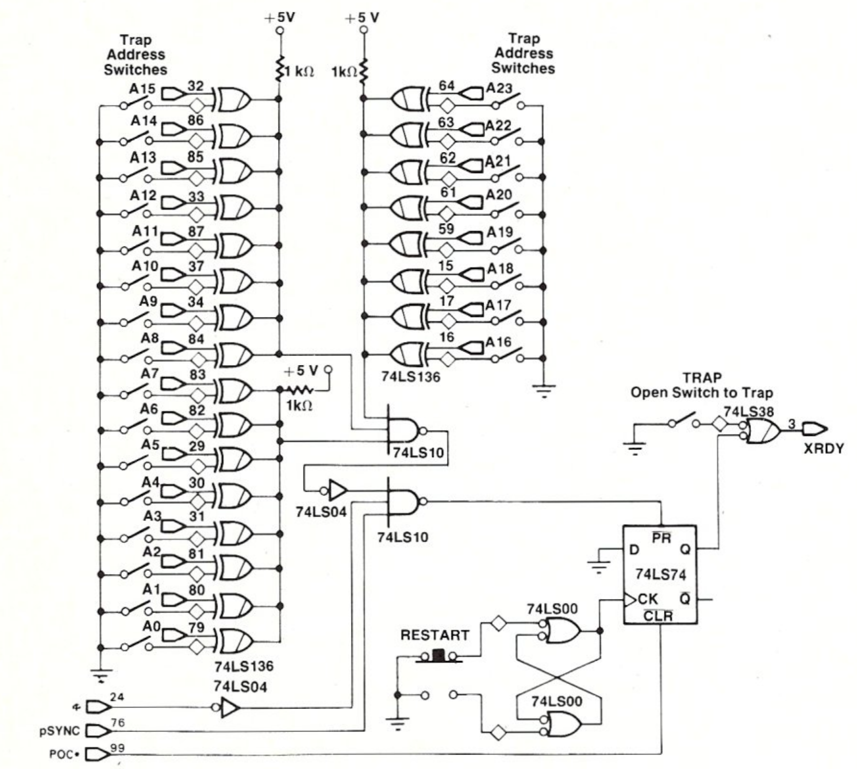


Figure 3. *Address trapping circuit to stop on reaching preselected address.*

S-100 BUS

Continued from page 19

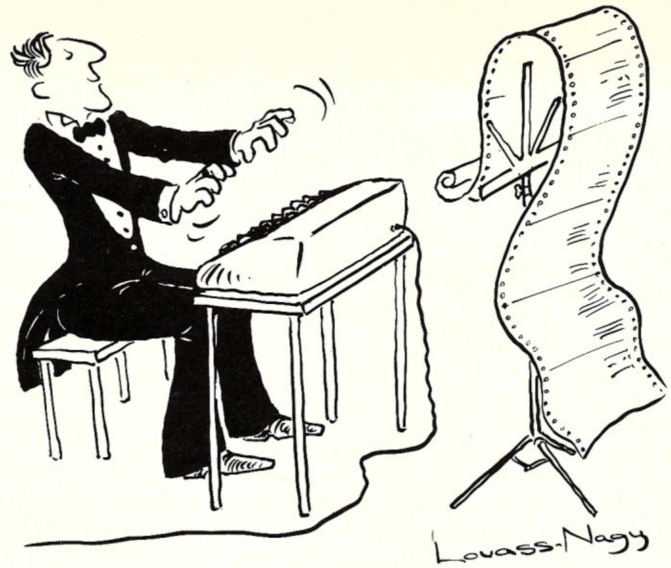
matched. In other words, the machine will be stopped whenever its PC arrives at the preset address. This allows you to let the machine run full speed until it reaches a certain address, then you can single-step the machine after that.

These circuits, from *Interfacing to S-100/IEEE-696 Microcomputers*, by Sol Libes and Mark Garetz (Osborne/McGraw-Hill) can be easily expanded for a specific troubleshooting purpose. Without them, trying to see what is really happening on the S-100 bus would be difficult, at best.

North Star revisited

I've received several favorable replies from readers who would be interested in seeing an S-100 Bus column discussing the differences between the North Star version of the S-100 bus and the IEEE-696 bus. As soon as I finish analyzing all of the information I have received about it, I will try to furnish a concise comparison of the two S-100s, along with a pin comparison.

Readers are encouraged to send in questions about the S-100 bus. Please write to: Dave Hardy, 736 Notre Dame, Grosse Pointe, MI 48230.



•NEW PRODUCTS•

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.

The Well-Tempered Cross-Assembler

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.

5¼" disk formats available at no extra cost include Osborne, Xerox, H-P, IBM PC, Kaypro, North Star, Zenith, Televideo, Otrona, DEC.

Avocet Cross-assembler	Target Microprocessor	CP/M-80 Version	•CP/M-86 IBM PC, MSDOS** Versions•
•XASMZ80	Z-80		
•XASM85	8085		
XASM05	6805		
XASM09	6809		
XASM18	1802		$250.00 each
XASM48	8048/8041		
XASM51	8051	$200.00 each	
XASM65	6502		
XASM68	6800/01		
XASMZ8	Z8		
XASMF8	F8/3870		$300.00 each
XASM400	COP400		
XASM75	NEC 7500	$500.00	
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)

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 $389
Options include:
• Software Driver Package --
• enhanced features, no installation
• required.
• CP/M-80 Version ....$ 75
• IBM PC Version .... $ 95
RS 232 Cable .... $ 30
8748 family socket adaptor . . . $ 98
8751 family socket adaptor . . . $174
• 8755 family socket adaptor . . . $135

• **G7228 Programmer by GTek** -- baud
• to 2400 ... superfast, adaptive program-
• ming algorithms ... programs 2764 in one
• minute.

• Programmer $499

• Ask us about Gang and PAL programmers.

• **HEXTRAN Universal HEX File Con-**
• **verter** -- Converts to and from Intel,
• Motorola, MOS Technology, Mostek,
• RCA, Fairchild, Tektronix, Texas
• Instruments and Binary formats.

• Converter, each version $250

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.

CALL TOLL FREE 1-800-448-8500
(In the U.S. except Alaska and Hawaii)

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 **Trademark of Microsoft

AVOCET SYSTEMS INC.™

DEPT. 0284M
804 SOUTH STATE STREET
DOVER, DELAWARE 19901
302-734-0151 TELEX 467210

The CP/M Bus

While MS-DOS is approaching the power of UNIX, CP/M is still the same OS familiar to 8080 and Z80 programmers.

by Anthony Skjellum

In this installment of CP/M Bus, I will compare the MS-DOS 2.0 and CP/M-86 operating systems. Also, the possible future roles of the CP/M operating system family will be discussed.

One of the original goals of this column was to answer reader questions concerning various aspects of CP/M. Very few questions have been posed and we expect this to remain the case. Therefore, a series of CP/M turtorials will be presented in future columns. Readers are invited to submit suggestions for topics that they would like covered.

Introduction

MS-DOS is the dominant 8086/88 operating system because IBM supports PC-DOS (IBM PC version of MS-DOS) for its PC and XT computer systems. The CP/M-86 operating system is also available for the IBM PC and XT at essentially the same price as MS-DOS 2.0. In the following paragraphs, we will discuss the various aspects of the operating systems and indicate which operating system provides the greater capability in each area. Concurrent CP/M-86 (CCP/M-86) will not really be included in the present discussion.

Starting with the May/June 1981 issue of *Microsystems,* I presented a four-part discussion of how I thought CP/M 2.2 should be enhanced to provide greater capability in 8-bit implementations. I was convinced then that CP/M required enhancement. Many of the suggested enhancements discussed in those columns involved UNIX-like features. Such features have not been added to CP/M but are available in MS-DOS 2.0. Interested readers may wish to review those articles before continuing with this column.

File systems

One of the most important aspects of an operating system is the way it handles files and peripheral devices. The CP/M-86 system handles files in essentially the same way as its ancestor, CP/M-80 2.2. The CP/M-86 file system provides a single directory which contains a fixed number of directory entries. The particulars of a disk format are determined by the Basic Input/Output System (BIOS) as for CP/M 2.2.

The CP/M file system provides no capability to use tree-structured directories. (A tree-structured directory system permits multiple levels of file organization by storing files under different headings or subdirectories.) This limits the usefulness of large capacity drives, since all files must reside in the "main" directory. To alleviate this difficulty under CP/M, hard disk drives are generally partitioned into several logical drives so that some degree of organization can be achieved.

The MS-DOS release 1 file system was no more sophisticated than CP/M's. No tree-structured directories were available. However, the MS-DOS 2.0 operating system does provide for such directories. A thorough complement of MS-DOS functions allows programs to access files from any part of the directory structure, and create/delete directories.

The MS-DOS file structure is also more flexible in handling media errors. Because of the way data are allocated, bad zones may be marked at formatting and be eliminated. With CP/M, transient programs must be created that seek out bad sectors and allocate them to "bad sector" files.

Additionally, MS-DOS dates and time stamps files. This is extremely convenient for the purpose of identifying versions of the same file from different sources. CP/M evidently does not support this feature.

Peripheral devices

Another important feature of MS-DOS is that it treats devices and files identically. For example, the console device is called "con" and may be used anywhere that an input or output file name is required. This is handled by the operating system and is completely transparent to application programs. CP/M is not as flexible.

The original version of CP/M provided limited input/output reassignment capability through the IOBYTE system. IOBYTE was designed to permit different logical and physical assignment of certain devices (select the logical printer from a set of physical printers etc.) These features are still available in CP/M-86, but have not been enhanced. Typically, the logical devices are accessed through the PIP program only, and set via STAT. Users may not replace file names with device names unless a particular application program takes the pains to support this type of operation.

Input/Output redirection

Input/Output redirection and pipes are two UNIX-like features which currently receive considerable attention. These features make programming and file maintenance easier. Input (output) redirection lets any data that would normally come from (to) the console be transferred from (to) a file. Pipes let the

TURBO FORCE™

- ★ MULTI-USER HARD DISC SYSTEM
- ★ TURBO-DOS OPERATING SYSTEM
- ★ UP TO 4 INDEPENDENT USERS
- ★ S-100 BUS ARCHITECTURE – 6 SLOTS
- ★ 8-BIT AND/OR 16 BIT PROCESSOR BOARDS

The "Turbo-Micro" is a revolutionary new business system designed for single or multi-user, multi-processor networking capabilities for fast response time. The "Turbo-Micro" is an attractive desktop computer with modular designs for fast and easy maintenance, yet it is rugged and well packaged for safe shipping and handling. It represents a state-of-the-art advancement with reliable components and disk drives. It is a powerful, yet compact system, 16" W x 7½" H x 22¼" D.

PACKAGED DEALER PRICE $4995.

PRICE INCLUDES:

- Complete desktop system with 110/220V power
- Master processor board — Z80B 6MHz, 64K. Byte memory, 2 serial/2 parallel ports, DMA
- Two slave processor boards, Z80B 6MHz, 64K. Byte memory, 4 serial/2 parallel ports, per board
- 25M. Byte winchester hard disk, 5¼" drive
- Floppy disk drive DS/DD, 1.6M. Byte, 8" drive
- Turbo-Dos* multi-user operating system with print spooling capabilities
- 100% CP/M* compatibility
- Complete manuals with schematics and diagnostics

SYSTEM OPTIONS:

- 16-bit processor, Intel 186 CPU with 128K. Byte or 256K. Byte memory
- Master/Slave CPU upgrade to 128K. Byte memory
- Battery back-up for up to 10 minutes of operation
- System upgradable to the powerful stand-alone cabinet for future expansion
- CP/M 2.2* or CP/M 3.0* operating system
- Complete library of application programs, general business accounting, CPA, medical, dental, pharmacy, construction, manufacturing, POS, PTY management and many others
- Terminals and printers are optional

CIRCLE 8 ON READER SERVICE CARD

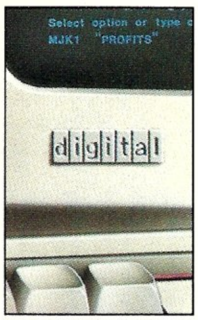
Please call today or visit us, to reserve one for you, and
MAY THE TURBO FORCE BE WITH YOU!

OVERSEAS INQUIRIES ARE INVITED.

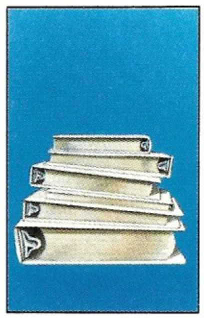
ADVANCED COMPUTER TECHNOLOGY, INC.
5575 MAGNATRON BLVD., SUITE D
SAN DIEGO, CA 92111 (619) 571-2746

Now, the most significant advance in DEC™ software ever!

Announcing DIGITAL REVIEW...the monthly magazine exclusively for your DEC microcomputer.



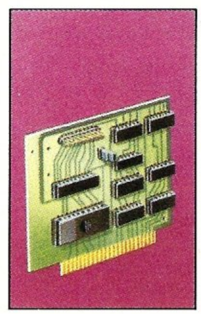
HARDWARE: Each and every month, you'll get all the latest technological developments—minicomputers... microcomputers... peripherals... telecommunications—and what they mean to you.



PROGRAMMING: All the latest in programming techniques with special features devoted exclusively to traditional and new software applications.

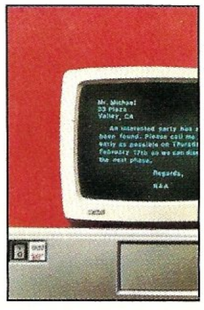
COMPUTER GRAPHICS: Color visual tools for the financial analyst, the laboratory technician, and the physics major. DIGITAL REVIEW reports on the latest innovations and how you can use them in your field.

WORD PROCESSING: You'll learn the newest techniques that will make your office as efficient and as cost-effective as possible.



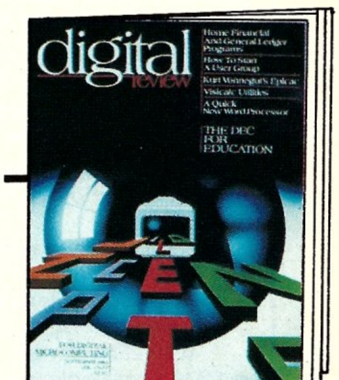
SOFTWARE: Whether it's the newest in operating systems... or advances in computer networks, DIGITAL REVIEW keeps you up-to-date... and spotlights releases that are changing the way businesses conduct their business.

I/O FEEDBACK: A chance for you—the user—to network your application needs and ideas.



Save up to 36%!

Now, through this Special Introductory Offer, you can save up to 36% when you subscribe to DIGITAL REVIEW.



digital review
P.O. Box 2989
Boulder, CO 80322

YES! I want to subscribe to Digital Review for:

☐ One year (12 issues) for \$24.97—I save 16%!

☐ Two years for \$43.97—I save 27%!

☐ Three years for \$57.97—I save 36%!

Savings based on full one-year subscription price of \$29.97
Add \$13 for Canada and all other foreign countries.

CHECK ONE: ☐ Payment enclosed ☐ Bill me
Charge my: ☐ American Express ☐ MasterCard ☐ VISA

Account No. ______________ Exp. Date ______

Name ______________

Address ______________

City ______________ State ________ Zip ________

Please allow 30 to 60 days for delivery of first issue.

Would you like to receive special offers from qualified users of our mailing list? ☐ Yes ☐ No

8H179

CP/M BUS

Continued from page 22

console output of one program become the console input of another. Neither of these features is available under CP/M. Many CP/M-80 utility systems and C compilers emulate these capabilities, but they are not a part of the operating system. However, MS-DOS incorporates these features internally.

Under the UNIX operating system, programs in a pipeline are run simultaneously in an asynchronous fashion. MS-DOS does not support multiple processes, so that pipes are handled as follows: the output of the first program (in the pipeline) is placed in a temporary file. Next, this temporary file serves as the input for the second program in the pipe. When the second program is finished, the temporary file is deleted. If there are more than two programs in the pipeline, this process is repeated sequentially. While this process is not ideal, it is the same concept used successfully under CP/M-80 to provide the same feature.

Program editors

Neither operating system provides a really worthwhile line editor. The ED editor under CP/M is essentially the same editor provided with the early versions of the operating system. However, it is a workable editing tool. The MS-DOS EDLIN program is harder to use. Its command structure leads users to replace lines with commands intended for the text processor.

Overall, whether one uses MS-DOS or CP/M-86, a suitable screen editor will be required. Editors such as CompuView's VEDIT fulfill this requirement more than adequately.

With the basic CP/M-86 users can develop, assemble and debug 8086 assembly language programs.

VEDIT is available for both operating systems.

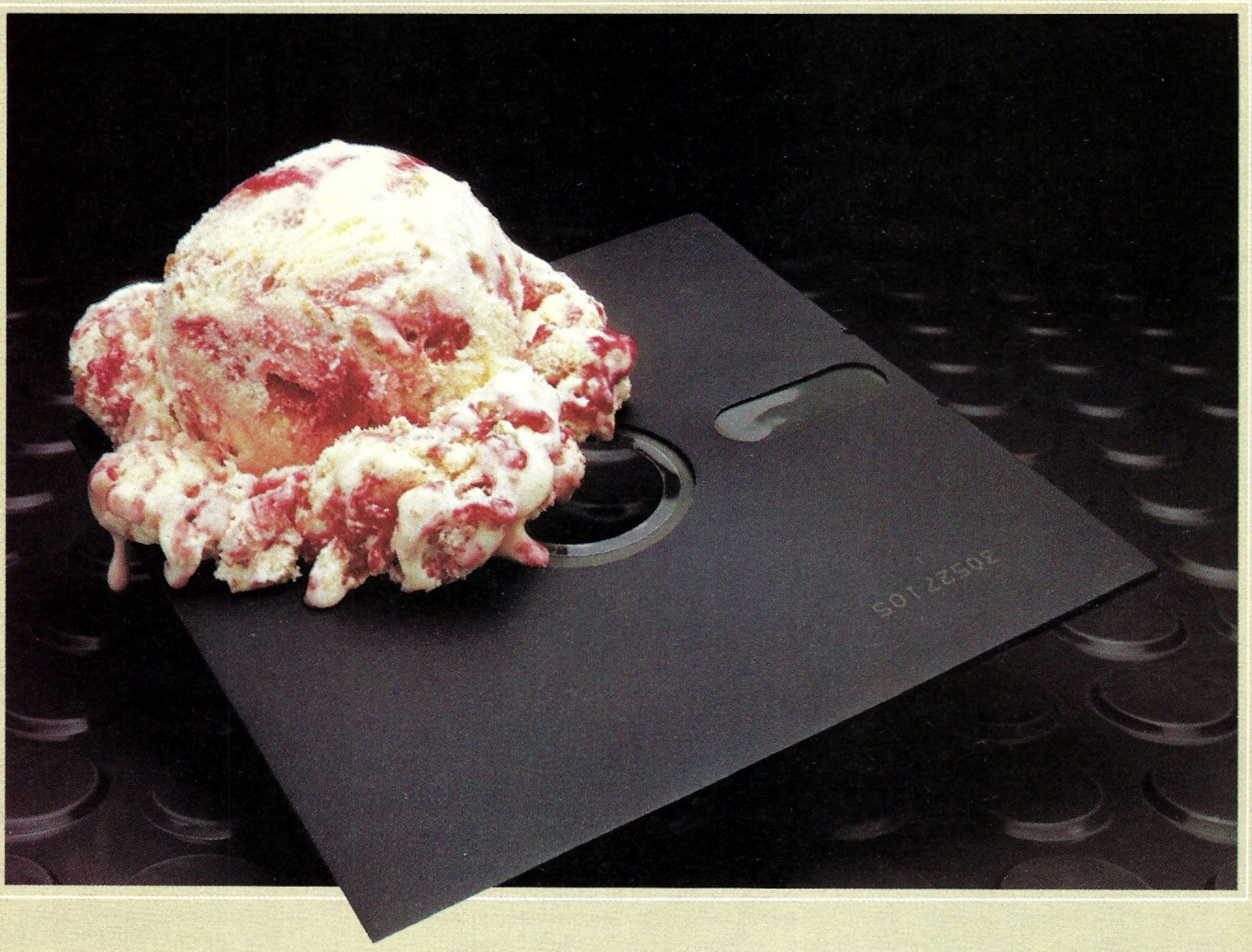
Program development

CP/M-86 is a complete development system. It includes a nonmacro assembler, debugger, and loader. With the basic CP/M-86 operating system, users can develop, assemble and debug 8086 assembly language programs. The basic MS-DOS system is not as complete. It does not include an assembler. Thus, an additional purchase must be made before a source language is available for use with the operating system.

Hard disk support

CP/M-86 provides no special support for hard disk drives. However, MS-DOS 2.0 provides the BACKUP/ RESTORE system. This permits hard disk files to be backed up to one or more floppy disk drives. While useful, there is still room for improvement in the BACKUP/RESTORE system.

Another important feature of MS-DOS is the ease with which FORMAT eliminates bad parts of disk media. Once bad data is found, it is eliminated. However, there is no mechanism for locking out bad sectors without reformatting a whole disk surface. This is inconvenient, since hard disk errors do develop during the life of the system. Reformatting and reloading a hard disk is quite a



NOW THERE ARE TWO WAYS TO MAKE dBASE II A TREAT.

(ONLY ONE IS GUARANTEED.)

Let Autocode write your programs for you.

For those of you with no desire to join the ranks of computer programmers just to get your work done, treat yourself to Autocode. Autocode is the easiest, fastest way to get what you want from dBase II, because it generates complete, high quality programs from simple instructions. Whether you're a dBase II master or just learning, you'll like the way Autocode writes powerful, elegant code with a minimum of effort. And you'll appreciate the fact that Autocode, unlike slower, less capable program generators, supports sophisticated dBase II features like on-screen calculations and multi-tiered criteria for report generation. Even if you're an experienced programmer, the consistent quality of Autocode's programs will win you over.

Use Autocode without risk.

Give it a try. For only $195.00, you get the unabridged Autocode disk and our concise, entertaining, clearly written manual. We're so sure you'll like the simplicity of Autocode's easy to use menu-driven operation, the only risk is that you'll never go back to dBase II alone. So for a taste of the best thing that's ever happened to dBase II, try Autocode. Contact your dealer, or call toll-free 800-262-8800 or (415) 777-3800. Or write Axel Johnson Corporation, 666 Howard St., San Francisco, California 94105.

Autocode 1

☆ A member of the Axel Johnson Group.

CIRCLE 104 ON READER SERVICE CARD

Continued from page 26

job, just to lock out a small amount of bad data.

Another problem along these lines involves FORMAT's inability to find all the bad areas on a disk surface. I have noticed on several IBM XTs that hard disks which format perfectly still may have bad sectors (such systems also pass the IBM diagnostics and Advanced Diagnostics). The only way to keep the bad sectors at bay is to keep files which contain the bad sectors and rename them (eg XXX.BAD). Perhaps FORMAT is not stringent enough when testing tracks and thus does not lock out weak (i.e. marginal) sectors.

Documentation

The standard CP/M-86 documentation is not unlike that provided with CP/M 2.0. It is the standard, concise documentation of previous years. The IBM PC version includes documentation of a tutorial nature. CP/M documentation includes all the information needed, although it is somewhat terse at times.

The MS-DOS documentation is aimed at the beginner. Advanced topics such as the debugging, linking and system calls are reserved for later chapters or appendices. This is just as well, since interested readers can find the information they require more readily. MS-DOS system calls are not always described in full detail; many are very powerful and require further clarification. Most notable are those involving subprocesses. Both operating systems should have examples included with the documentation on system calls.

Environment

The MS-DOS operating system provides environment variables to programs. These variables are up to 32K of data strings. Each of these strings is available to the program via an environment block. Furthermore, the strings may be set via the MS-DOS command language. Such environment variables can provide additional flexibility to the programming environment, if they are used wisely. So far, I have seen only one program which uses an environment variable.

MS-DOS includes the COPY utility as part of the user interface, so copying is easier than under CP/M.

CP/M does not incorporate the concept of environment variables nor that of environment blocks.

User interface

The user interface is an important part of an operating system. As with other aspects of CP/M, the CCP (Console Command Processor) has not really evolved from the CP/M 1 level. Most notable under CP/M-86 is the absence of the useful XSUB feature provided with CP/M 2. Thus, batch file processing has actually suffered a setback in the transition to the 8086 world.

MS-DOS 2.0 has a nice command processor. Batch files do not require a special program to invoke them. Instead, they are treated like other commands, i.e., files with the extension .BAT may be executed like other commands (i.e. .COM or .EXE binary program files). MS-DOS supports parame-

CP/M BUS

Continued from page 29

ter substitution, looping, branching via primitive conditionals. Thus, batch files are much more useful under MS-DOS than they are under CP/M.

MS-DOS includes the file transfer utility (COPY) as an integral part of the user interface. Thus, copying is more easily accomplished than under CP/M where the PIP command must be loaded in from a disk.

MS-DOS 2.0 also includes a PRINT command. PRINT is a background process that prints files on the system printer while other work is under way. Several files may be in the queue at any given time.

Summary

The above discussion includes a survey of various aspects of MS-DOS 2.0 and CP/M-86. In most cases, CP/M-86 falls short of MS-DOS 2.0. It seems that while MS-DOS is approaching the power of UNIX (and will essentially be single-user UNIX in version 3.0), CP/M is still the same operating system familiar to 8080 and Z80 programmers.

I began to sense the real power of MS-DOS once I started using the new MS-DOS 2.0 system calls. Complex file manipulation is trivial under MS-DOS. In my opinion, MS-DOS does everything that CP/M-86 does and a lot more. Thus, my preference in operating systems is MS-DOS 2.0.

Future of CP/M

One can only wonder how long CP/M-86 can survive with the competition of MS-DOS. CP/M-86 does provide the one feature not currently available in MS-DOS: multitasking. However, MS-DOS 3.0 promises to include this feature too.

On the other hand, I think that CP/M will continue to be a prominent operating system for 8-bit computers. Furthermore, I think that more powerful 8080/Z80-type microprocessors will continue to appear, giving CP/M continued importance in the 8-bit world (especially with the myriad of public domain software.) Although this is likely, CP/M could stand enhancement, even beyond CP/M 3.0. In any case, I don't see CP/M-86 as a strong competitor in the 8086 environment.

Conclusion

Because CP/M is likely to have a long and continued importance, CP/M Bus will continue to discuss topics of interest to CP/M users, but mainly in the 8080/Z80 realm, where CP/M is most dominant.

TURN-KEY, TURBO-DOS BUSINESS SYSTEM

This system is tailored to the needs of the individual customer and as such provides that touch of freedom deisred by the business system buyer. **SYSTEM-1** is truly **"TURN-KEY"**, just unpack the equipment, connect the cables, turn it on, and you are up and running. There are numerous **SYSTEM-1**'s meeting the requirements of the small to moderate size business currently in todays market.

SYSTEM-1 can support up to fifteen users, a batch processor, and up to eight printers. Other options, such as **"SEMIDISK"**, are available to the user when the need arises. The attractive all metal enclosure will complement any office decor, as will the **ADM-11** consoles, and the **MPI 150G PRINTMATE** printers.

- 5 USERS
- 2 PRINTERS
- 5 CONSOLES
- 1 Z80A CPU
- 6 Z80B CPU'S
- 2 8" DRIVES
- 15 MEGABYTE HARD DISK
- 832k BYTES OF MEMORY
- ATTRACTIVE ENCLOSURE
- TURBO-DOS 1.22
- TURBO-PLUS 1.03
- CCD CONFIG 1.02
- ALL CABLES
- MANUALS
- PREPAID SHIPPING

SAVE OVER $3000.00

The system software is **"TURBO-PLUS"**, an extended version of **"TURBO-DOS"**. This package is compatible with **"CP/M"**, and **"MP/M"**, providing the user with a wealth of commercial software. **TURBO-PLUS** provides the user with several major advantages over other **"CP/M"** compatible systems, such as a **TWX** command for interconsole messages, a **MAIL** command to leave a message, special **"LOGON"** and **"LOGOFF"** commands for proper access and daily **BULLETINS**. If desired, the system will also maintain daily log entries including system access notations.

Users can be assigned their own work areas, thus one user can not affect another. All user printing is **"SPOOLED"** and will not tie up the users console. In addition, time consuming printing and other processes may be **"QUEUED"** or **"BATCHED"** to be run later.

Included is a free membership to **"TUG"** (Turbo Users Group) which provides a forum for **TURBO-DOS** users.

The complete **SYSTEM-1** will be shipped prepaid to your location ready to operate, with the exception of your application software.

ORDER #CCD-09001-00 .... **$15,499.95**

STD HEIGHT 5.25" FLOPPY DISK DRIVES
CCD08207-02 MPI B51 40 trk, 1 side $199.95
CCD08201-02 MPI B52 40 trk, 2 side $264.95
CCD08206-02 MPI B91 80 trk, 1 side $269.95
CCD08202-02 MPI B92 80 trk, 2 side $369.95

HALF HEIGHT 5.25" FLOPPY DISK DRIVES
CCD08204-05 SA-465, 80 trk, 2 side $379.95
CCD08203-02 MPI 902, 80 trk, 2 side $369.95

STD HEIGHT 8" FLOPPY DISK DRIVES
CCD08205-02 Qume 842, 77 trk, 2 side $579.95

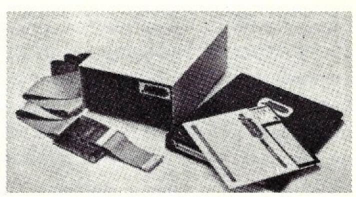
HARD DISK DRIVES, 5.25"
CCD08501-02 ST-419, 19 meg $999.95

HARD DISK SYBSYSTEMS, 5.25"
CCD09201-08 11 Megabyte, CP/M $1699.95
CCD09202-08 22 Megabyte, CP/M $2199.95
CCD09203-08 44 Megabyte, CP/M $3099.95

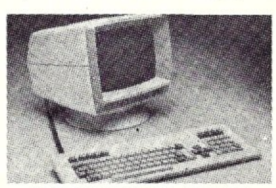
PRINTERS, RS-232 SERIAL
CCD08701-02 MPI Printmate 150G w/softkey, & 4k buffer $1199.95
CCD08703-04 Prowriter 2, 1550 .... $ 739.95
CCD08705-04 Prowriter 8510 .... $ 569.95

Turbo-Dos is a trademark of Software 2000 Inc.
Turbo-Plus is a trademark of MicroServe Inc.
CP/M and MP/M are trademarks of Digital Research Inc.

CCD08204-05 SA-465



HARD DISK SUBSYSTEMS



CCD08601-02 LSI, ADM-11

CRT TERMINALS, RS-232
CCD08601-02 LSI, ADM-11 .... $559.95
CCD08602-09 QUME, QVT 102 .... $559.95

TELETEK IEEE-696 S-100 CARDS
CCD-08101-01 Systemaster CPU $589.95
CCD-08102-01 SBC-1 4mhz, 64k, Slave $579.95
CCD-08103-01 RAM-64k, Expand to 256k ... $359.95
CCD-08104-01 RAM-256k $659.95
CCD-08105-01 PSIO, Quad RD232 .... $219.95
CCD-08106-01 HDTC, Disk/Tape Control $524.95

NOVATION MODEMS
CCD-08801-06 Novation 212 auto cat $589.95
CCD-08802-06 Novation D-cat modem $164.95
CCD-08803-06 Novation J-cat modem $119.95

8" CP/M SOFTWARE
CCD-07001-01 CP/M 2.2, Teletex $150.00
CCD-07002-06 M.S. BASCOM-80 .... $249.95
CCD-07003-06 M.S. Cobol-80 .... $469.95
CCD-07004-06 M.S. Fortran-80 .... $314.95
CCD-07005-06 M.S. MBASIC-80 .... $219.95
CCD-07006-06 M.S. Multiplan $179.95
CCD-07007-06 Perfect Calc/Filer .... $ 349.95
CCD-07008-06 Perfect Writer/Speller .... $ 299.95
CCD-07009-06 Supercalc $124.95
CCD-07010-06 Supercalc II $195.95

(713) 488-9754

Circle Reader Service number 90 for System 1 Information
Circle Reader Service number 121 for Computer Component Information

COMPUTER COMPONENT DISTRIBUTION
P.O. Box 590467
Houston, Texas 77259

VISA & M.C. welcome, Sorry No C.O.D.
Checks take 4 weeks.
Prices subject to change.
ADD 5% shipping, plus 4% tax (Texas).
Terms to rated Firms only

The UNIX File

"To use UNIX well, you must understand not only how to use the programs, but also how they fit into the environment." —Kernighan & Pike

by Ian F. Darwin

This month, "The UNIX File" reviews a UNIX book aimed primarily at practising programmers, answers a reader's question on XENIX, and touches a few other topics.

The UNIX programming environment

The latest UNIX book is unabashedly intended for use by programmers. *The UNIX Programming Environment,* by Brian W. Kernighan and Rob Pike, is the latter's first book, although Kernighan is well known as co-author of many UNIX books and papers. But Pike is no newcomer to UNIX: an old fifth edition UNIX Manual in my possession includes some local manual pages for programs Pike wrote at U of T—in 1975! More recently, Pike was head of design for the BLIT terminal, available in subset form as the Teletype 5620 terminal.

The authors state their intentions at the outset: "Our goal in this book is to communicate the UNIX programming philosophy. Because the philosophy is based on the relationships between programs, we must devote most of the space to discussions about the individual tools, but throughout run the themes of combining programs and of using programs to build programs. To use the UNIX system and its components well, you must understand not only how to use the programs, but also how they fit into the environment" (p. viii).

The book begins with a chapter called "Unix For Beginners," patterned after a paper by the same name by Kernighan. This describes what UNIX is, how it looks, and how to use it for simple tasks. The next chapters discuss the file system, the shell or command interpreter, and filters (programs that make a single change to their input to produce output). "Although our main target is programmers, the first four or five chapters do not require programming experience to be understood, so they should be helpful to other users" (p. viii). Indeed, chapter 5 discusses a topic that defines the boundary between programmers and nonprogrammers. Shell programming, or advanced use of the command language, is a topic that can be approached by nonprogrammers who have learned the more basic uses of the commands. At the same time, as the examples in this chapter make clear, the advanced UNIX programmer will ignore shell programming at his/her own risk. Many programs can and should be written as shell files, using existing tools, rather than coded in languages such as C, both to reduce reinvention of wheels and for ease of maintenance. By presenting this topic *before* C programming, the authors remind us to think of writing a shell file—using existing tools—before reinventing the wheel.

Chapters 6 and 7 are descriptions of what can and should be done in C. The first describes use of the "standard I/O" library; the second, the use of "System Calls" to process files, manage the interrelationships among processes, handle interrupts and errors, and the like. They assume that you know or are learning C, and don't try to teach you the language. It belongs in a book by itself, and Kernighan is the co-author of an excellent one—*The C Programming Language.*

The next chapter deals with use of the "program development tools" such as **yacc, make,** and **lex.** The authors provide many examples of code: this chapter features the step-by-step development of a small-scale interpretive programming language implemented as a few hundred lines of grammar, lexical analysis, and action statements. The last chapter shows you how to document your programs, something you should do before or while you write them. UNIX provides several quite powerful tools for this purpose, and these are examined here.

Appendices describe the standard UNIX text editor **ed,** and present the complete documentation and source code for the programming project developed in chapter 8.

The UNIX Programming Environment lays a good groundwork for developing programs that not only work on UNIX, but work *well* with the rest of the UNIX environment. The overall quality of the book is excellent, and I recommend that anyone developing programs for UNIX should read this book at his/her earliest convenience.

My only criticism of the book, and it is a minor one, is that the authors downplay the **lint** utility, saying that they have found it to produce many superfluous error messages. True, but using the options "-ha" and paying attention to the output is a very good way to find problems in C code. We have found this very useful, both in cleaning up code received from other institutions and in debugging our own code!

The book closes with some abuse of the present thrust of UNIX development efforts, and a review of the philosophy which has both made the system successful and can make your programming effective. "The UNIX system has . . . become one of the computer

WESTERN ELECTRIC • PEACHTREE SOFTWARE INC. • CRAY RESEARCH • UPJOHN • RCA • DYSAN CORP. • WESTINGHOUSE • AMERICAN BELL •
SOFTWARE BANC • JOHNSON & JOHNSON • HOFFMAN LaROCHE • AT&T • TIME LIFE • BALLY MANUFACTURING • APPLE COMPUTER •

AZTEC C ...the most portable and comprehensive C software development system available

Manx Software Systems with over 6,000 licensed users is one of the leading suppliers of "C" compilers and cross compilers for the world's most popular micro computers. The MANX AZTEC C compiler is available as a cross compiler or native compiler for the following environments:

AZTEC C86	AZTEC C II	AZTEC C65
PC DOS	CP/M-80	APPLE DOS
MS DOS	TRSDOS	COMMODORE 64 (12/83)
CP/M-86		ProDOS (2/84)

The AZTEC C product is a complete development system. In addition to a full v7 "C" compiler, the basic product includes assemblers, linkage editors, development utilities, and full run time libraries. New products to be released in late 1983 and early 1984 include graphics development tools, data base managers, program editors, screen management systems, and other development tools that work in conjunction with the AZTEC C system.

CROSS DEVELOPMENT SYSTEMS

The AZTEC C cross development systems include all of the utilities and library support routines available with the native versions including a cross assembler. The binary image created in the host environment is downloaded and tested in the target environment. MANX has been using its own cross compilers on a daily basis since 1980.

HOSTS

PDP-11	68000	8086/8088	8080/8085/Z80
UNIX	UNIX ports	UNIX ports	CP/M-80
		PC DOS	
		MS DOS	
		CP/M-86	

TARGETS

8086/8088	8080/8085/Z80	6502/65xx
PC DOS	CP/M-80	APPLE DOS
MS DOS	TRSDOS (12/83)	APPLE ProDOS (2/84)
CP/M-86	LDOS (12/83)	COMMODORE 64
	DOSPLUS (12/83)	

Other host and target environments will be released in early 1984.

NATIVE COMPILERS

The AZTEC C native 8080 compiler was first released in 1981. Since that time it has been acquired by more than 300 colleges and universities, thousands of corporations, small business, and government agencies. The compiler has been ported to the 6502 and 8086. Plans for future ports include the 68000, 16032, and IBM 370. All native versions are source compatible. Source developed in one environment can therefore be transferred to another environment and compiled, linked and executed.

For prices and information call:
800-221-0440 (outside NJ)
201-780-4004 (inside NJ)
4995812 (Telex)

Or write to: ***MANX SOFTWARE SYSTEMS***
P.O. BOX 55 • SHREWSBURY, NJ 07701

MANX®
software systems

FTWARE BANC • JOHNSON & JOHNSON • HOFFMAN LaROCHE • AT&T • TIME LIFE • BALLY MANUFACTURING • APPLE COMPUTER • AMDAHL CORP. • B.O.A.C. SOUTHWESTERN BELL • MOTOROLA • CITIBANK • BAUSCH & LOMB • MIT •
-TATE INC. • LOCKHEED MISSILES & SPACE CO. • ZENITH • TIME INC. • MATTEL • WANG LABS • ROCKWELL INTERNATIONAL • EAGLE COMPUTER INC. • CORVUS • PITNEY BOWES • GENERAL ELECTRIC • VISICORP • MIT • STANFORD •

U.S. STEEL • XEROX • NORTH STAR COMPUTERS • FORD MOTOR CO. • MICROSOFT • SPERRY UNIVAC • KAYPRO CORP. • BELL LABS • MODULAR COMPUTER • VIDEX • MICROPRO • NATIONAL SEMICONDUCTOR • DRI • SOFT
SOUTHWESTERN BELL • MOTOROLA • CITIBANK • BAUSCH & LOMB • COMPAQ COMPUTER CORP. • BUNKER RAMO • CONTROL DATA CORP. • DEC • TYMAK • IBM • ROCHE BIO-MEDICAL • 3M • BELL & HOWELL • BENDIX •

FLORIDA COMPUTER GRAPHICS • WESTERN UNION • MERRILL LYNCH • LOTUS DEVELOPMENT CORP. • MONOLITHIC SYSTEMS CORP. • HUGHES AIRCRAFT • ASHTON-
WARE PUBLISHING • FRANKLIN • TELEVIDEO SYSTEMS INC. • PERKIN ELMER • NCR • SIERRA-ON-LINE • NORTON CO. • HEWLETT-PACKARD • U.S. ROBOTICS • TRW • G.T.E. • SO

CIRCLE 245 ON READER SERVICE CARD

UNIX File

Continued from page 32

market's standard operating systems, and with market dominance has come responsibility and the need for 'features' provided by competing systems. As a result, the kernel has grown in size by a factor of 10 in the past decade, although it has certainly not improved by the same amount. This growth has been accompanied by a surfeit of ill-conceived programs that don't build on the existing environment" (p. 316). Is this just Pike pining for the "good old days" of 1975? I think not. The authors do not claim that 1975's UNIX was perfect, any more than they make that claim about today's. But they have both been working at improving UNIX for the past decade—not by blindly tossing in new options and new commands, but by building a consistent set of tools that work with one another. Many of the programs in the book began as examples and grew into tools that are used daily; others are well-honed tools presented here because they are missing from standard UNIX.

Many programs became tools through daily use. Others are well-honed tools missing from standard UNIX.

"Creeping featurism encrusts commands with options that obscure the original intention of the programs. Because source code is often not distributed with the system, models of good style are harder to come by." Expect this book to be successful, because it provides those models of good style.

TRS-XENIX and proliferation

Keith Nush of Illinois writes: "I have a TRS Model 16B using the XENIX operating system. Radio Shack advertises that the XENIX system was 'derived' from the UNIX operating system. Are UNIX programs capable of running on XENIX, and vice versa?"

Well, XENIX in fact *is* UNIX. To make XENIX, Microsoft bought the source for seventh edition (V7) UNIX from Bell and changed the compiler to generate code for the 8086 and (later) other micros. Microsoft also added features they considered useful in a small business—computer environment (see *BYTE* magazine, June 1981, p. 248). Radio Shack bought a version of this software for the TRS-80 model 16B, which actually has two microprocessors in it (on which a paper was presented at the Toronto USENIX conference, summer 1983).

Moving UNIX to microcomputers was not such a radical idea as it seemed. Bell has been writing papers on the topic of UNIX portability since at least 1977. The UNIX version running on the largest total number of microcomputers is XENIX, but the UNIX variant running on the largest number of different manufacturers' equipment is UniPlus+, by UniSoft. A number of other companies (including Human Computing Resources, Toronto) port UNIX to other machines. UC Berkeley took the 32/V

Inter-Office Correspondence Made Simple

SEATTLE OFFICE

```
Memo editor: Enter mail                              Mode: Add
Date: 11/30/83
                                         Special delivery: No
From mailbox: 709 - Jack Spears           Return receipt: Yes
To mailbox:   101 - Will Pierce               Copies to: None
Subject: Electronic Mail Manager          Attached files: No
--------------------------------------------------------------
Will,

     Since the installation of the "EMM" system, our
inter-office correspondance has become effective and
more productive.  The system is a real "bargan" at
even 10 times the price!  See you next week.
```

LYME STREET OFFICE

```
Memo editor: Enter mail                              Mode: Add
Date: 11/30/83
                                        Special delivery: Yes
From mailbox: 101 - Will Pierce           Return receipt: Yes
To mailbox:   102 - John Fail          Copies to: 5 mailboxes
Subject: Board Meeting tomorrow           Attached files: Yes
--------------------------------------------------------------
Hi John,

        The board meeting will be held tomorrow morning at 10
 A.M.  This Electronic Mail Manager sure makes communications
 between our offices easy and fast. The new budget forcast is
 attached in xyCALC format.  Let me know if you approve.
```

EMM™ PLUS COMMX™

LA OFFICE

```
Memo editor: Enter mail                              Mode: Add
Date: 11/30/83
                                        Special delivery: Yes
From mailbox: 102 - John Fail             Return receipt: Yes
To mailbox:   101 - Will Pierce               Copies to: None
Subject: Board Meeting tomorrow           Attached files: Yes
--------------------------------------------------------------
Will,

     The budget forcast looks good except that I have
revised the year end projections to include  new market
trend figures.  I have returned the new setup in xyCALC.
See you at tomorrows meeting!
```

SOUTH STREET OFFICE

```
Memo editor: Enter mail                              Mode: Add
Date: 11/30/83
                                         Special delivery: No
From mailbox: 908 - Lara Wilks            Return receipt: Yes
To mailbox:   101 - Will Pierce                Copies to: All
Subject: New Product Release               Attached files: No
--------------------------------------------------------------
Will,

    We have completed the integration testing of the
fortran version of COMMX on the DEC 10 with links to
all the micros.  The error free file transfers work
like a dream!  The COMMX protocol does it again!
```

EMM™ Electronic Mail Manager using dBase II™ and COMMX™ COMMunications eXchange software team-up to provide your office personnel up to 900 mailboxes on your dial-up Inter-Office Network Send $20 for your demo disk today! Available for all CP/M 80, CP/M 86 and MS DOS systems.

HAWKEYE GRAFIX

23914 Mobile Street, Canoga Park, CA 91306 • (213) 348-7909 or 634-0733
Modem (300 B): 634-0441 Data

CIRCLE 18 ON READER SERVICE CARD

"The Game Board" Now with:

SUPERGRAPH

"The Game Board", combined with your S100 computer, offers an economical solution to the high cost of computer graphics. Now with superGraph, "The Game Board" provides high quality terminal emulation and GSX compatible graphics output in one affordable package.

Features:

- User selectable resolution up to 512x576
- On board CPU performs graphics functions
- Terminal data rate in excess of 9600 baud
- User defined character set
- Partial emulation of VT100/H19
- Full emulation of VT52
- Full GSX command set
- Circles, Arcs, Pie slices, Lines, Points
- Polygon Fill (hatch, solid and pattern fill)
- Windows and Viewports
- Graphics Text (with rotation and scaling)
- Sprites and Models

For more information and ordering contact:

paraGraphics
58 Needham St.
Norfolk, MA 02056
(617) 620-4513
(617) 528-7093

Assembled — $595.
Complete Kit — $525.
Partial Kit — $295.
P39 Monitor — $195.

VISA MasterCard

paraGraphics

CIRCLE 21 ON READER SERVICE CARD

RP/M T.M.

By the author of Hayden's "CP/M Revealed."

New resident console processor RCP and new resident disk operating system RDOS replace CCP and BDOS without TPA size change.

User 0 files common to all users; user number visible in system prompt; file size and user assignment displayed by DIR; cross-drive command file search; paged TYPE display with selectable page size. SUBMIT runs on any drive with multiple command files conditionally invoked by CALL. An automatic disk flaw processing mechanism isolates unuseable sectors. For high capacity disk systems RDOS can provide instantaneous directory access and delete redundant nondismountable disk logins. RPMGEN and GETRPM automatically self-install RP/M on any computer currently running CP/M® 2.2. Source program assembly listings of RCP and RDOS appear in the RP/M user's manual.

Manual alone $55; manual with RPMGEN.COM and GETRPM.COM with utilities on 8" SSSD $75. Shipping $5 ($10 nonUS). MC, VISA.

microMethods

P.O. Box G, Warrenton, OR 97146
118 SW First St. (503) 861-1765

CIRCLE 41 ON READER SERVICE CARD

UNIX File

Continued from page 35

version of UNIX (as ported by Bell to the DEC VAX-11) and made it into the 4BSD system. What does this proliferation mean to Mr. Nush and others with small systems? First, there is a very large number of programs available for use with UNIX. Second, most of these different UNIX systems have a common ancestor and thus many features in common.

The net result is that, in general, *well-written* programs can be moved from one version to another. Binary or executable programs, however, can at best only be moved from one UNIX to another running on the same CPU type—but even this is by no means universal, and should not be relied on.

The bottom line? Buy software that comes in source, or buy software that is

Buy software in source code, or software intended for your particular configuration.

intended for your particular configuration.

There is a small industry growing around the differences in these systems. Software manufacturers must either arrange local access to all the different machines on which they want their programs to run, or make arrangements with a company specializing in making software run on a variety of different machines. The former is possible if you live in a metropolis, but the latter may be your only option if you're growing software on a farm away from the rush of the city. Expect these companies to grow as long as the diversity of UNIX systems continues.

My one complaint about the Radio Shack offering is that when you buy the Model 16B with TRS-XENIX™, all you get is the UNIX kernel and a few utilities. Because XENIX is descended from UNIX, however, people might buy the system thinking that they're getting full UNIX. Not so. The C compiler and more than 200 additional modules—all the UNIX software tools—are hidden away in an extra-cost "development package" (26-6401) which costs $750 in the U.S., $950 in Canada (from the *1984 TRS-80 catalog RSC-10*). In selling TRS-XENIX without this package, Radio Shack is selling a cart without a horse. *Caveat emptor.*

Reference cards

If you want a concise reference to UNIX, tradition suggests getting a pocket reference summary. The Waite book on UNIX (reviewed in this column in November) includes a reference card. Here's another source: Specialized Systems Consultants (Box 7, Northgate Station, Seattle, WA 98125, phone 206-FOR-UNIX) sells four different pocket references. One is for UNIX; another for **vi**; a third for the UNIX C library; and the fourth for the C language itself. Prices range from $2 to $6, with discounts for quantities greater than one of each and with a package deal on the first three. *I haven't seen these summaries yet,* but if you want a pocket reference, check this one out.

Rumour mill

The rumour mill heralds for early 1984 the announcement of a single-user UNIX-like system from Microsoft (pos-

UNIX File

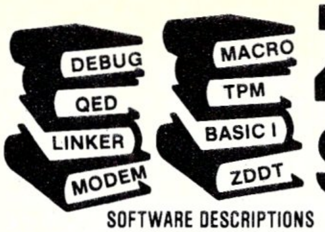
Continued from page 37

sibly sold as a version of their MS-DOS), a single-user PC with UNIX from Western Electric, and the announcement of another release of System V UNIX (called "V.2" or "5.2"). We'll see how many of these are borne out by facts.

The UNIX Book by Banahan and Rutter arrived just as I was shipping this column off to the editor. Watch for my comments on it in a future column, as well as a few tips on "shell programming" and some information on typesetting and networking.

The UNIX File looks at many aspects of the UNIX operating system. If you have comments or questions about UNIX or this column, feel free to write me at the University of Toronto Computing Services (UTCS), 10 King's College Rd., Toronto, Ontario, Canada M5S 1A1. If you have UUCP or UNIX mail access to the USENET network, contact me at "decvax!utcsstat!ian".

The opinions presented here are my own, and not necessarily those of the Univ. of Toronto or UTCS.



Z80 Software

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.

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)

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

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

ORDERING INFORMATION:

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

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

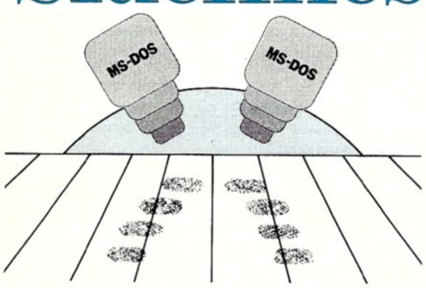
For information and Tech Queries call
(609) 599-2146

Computer Design Labs

CIRCLE 84 ON READER SERVICE CARD

342 Columbus Avenue/Trenton, NJ 08629

From the Sidelines



The battle for the personal computer market

by Hank Kee

This is a new monthly column that will be devoted to topics of interest relating to MS-DOS as well as CP/M. MS-DOS is the generic version of PC-DOS, which is the operating system on the IBM PC. The IBM PC has been accorded a very successful reception since its introduction. In recognition of this popularity, this column will review topics that overlap MS-DOS as well as CP/M. MS-DOS was originally written by Seattle Computer Systems with CP/M very much as a model. Although IBM originally sought to implement CP/M on its PC from Digital Research, it settled on MS-DOS for a variety of reasons. The rights and support for MS-DOS were later acquired by Microsoft. Many newly introduced systems run MS-DOS and CP/M-86. This month, I will comment on IBM clones and compatibles on the marketplace, the IBM PC*jr*, and the newly introduced Palette computer image recorder from Polaroid.

The battle for the personal computer market has resulted in a plethora of new product announcements supporting MS-DOS. Some of these may offer cosupport for CP/M-86. Clearly IBM has been the dominant force in the shaping of personal computing architecture. This has been beneficial, as de facto standards have stabilized an overabundant number of systems that could not freely exchange data disk formats with each other. The negative aspect of all this is that everyone thinks only of Big Blue, and as a result this has forced non-IBM manufacturers to offer me-too look-alike systems. This tends to inhibit innovation and stifle competition.

A major factor in the rapid acceptance of personal computing is the increasing availability of meaningful off-the-shelf software. Software development has truly been cottage industry driven. The overhead expense is low. Most prospective software authors look at getting a small piece of the market. If the market is big enough, then a small piece is still quite significant. They have perceived the IBM PC to be the dominant personal computer, and have therefore concentrated their development exclusively on the IBM PC. They wait in a queue to purchase the IBM PC, which in turn fuels the backlog demand for the PC. Since the demand for the PC is so high, they all figure there must be some huge market for them to tap. The name IBM has created a lot of magic, with the result that the development of commodity software on the IBM PC has spearheaded the increasing demand for all manufacturers to be either clones or compatibles.

Buying an IBM clone or compatible

The introduction of the IBM PC has given new meanings to the word clone. I always thought that a clone was a copy of an original, but nowadays there are virtual copies, near copies, and even better-than-original copies. At least, that is what the various manufacturers of IBM clones and compatibles are telling us.

There are a number of factors that the consumer should be aware of if purchase of a clone or compatible is being contemplated. When the IBM PC was introduced, in August of 1981, I used two single-sided double-density disk drives. These were upgraded to double-sided, double-density 40 track (48 tracks per inch recording density) $5^1/_4''$ disk drives. But there are also 80 track (96 tracks per inch recording density) disk drives available which increase capacity by a factor of 2. However, these higher density drives can read 40-track disks but are unable to write information in the same compatible density. An example of this is the DEC Rainbow 100 and its variants. If you don't need to migrate data between systems, then this difference is not significant.

The Victor 9000 is even fancier. It has a variable number of records per track using a recording density of 96 tracks per inch. The typical disk drive has the same number of records per track whether the track is on the outer edge or the inner edge of the disk. The people at Victor properly deduced that the tracks toward the outer edge can accommodate more records at the same density level as records on the inner tracks. They very cleverly designed their system to take advantage of this feature. If the migration of data is one-way, from the IBM PC to the Victor 9000, this is fine. This assumes the world revolves around the Victor 9000. But recent marketing surveys have shown that IBM is the center of the PC universe.

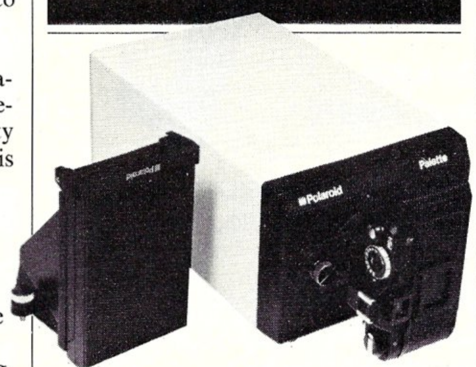
Another variable to consider is the differences in screen attributes. Most of the software written for the IBM PC tends to be screen dependent. That is only part of the problem. The color adapter board offered by IBM lacks high resolution and continuous viewing while scrolling at 80-column width. Many manufacturers opted to give the consumer better resolution, but thereby caused problems in running software that was IBM screen cursor control dependent. The name of this market is not whether or not your hardware is better, but whether you can run the off-the-shelf software that is popularly available. An example of this is the Zenith Z100. It has better color resolution to date (alas, in an S-100/IEEE-696 bus) and has had most of the popular commercial software rewritten for it. The caveat is that popularly available software exists under the same compatible operating system.

The DEC Rainbow 100 has now a very high percentage of overlap of software when compared to the IBM PC, but the Rainbow is supplied with two operating systems; MS-DOS 2.0x and CP/M-80/86. The obvious question is: How does one migrate data between operating systems? It's a very interesting question. In fairness to DEC, the Rainbow can run CP/M-80, CP/M-86 and MS-DOS programs. The plethora of commercial and public domain programs make the Rainbow a very attractive offering. Most of these programs can easily be adapted or are available to support the VT100 protocol on the Rainbow. However, DEC hasn't told the public how to easily move this treasure trove of software from 8″ disk media. There is no 8″ disk drive capability on the Rainbow. What really matters is that the software which you require is readily available.

IBM's Basic is dependent on the ROM. That ROM cannot be legally copied. I'll concede the term "virtual copy" to the manufacturer if the clone product offered to the user includes a functionally equivalent Basic. Keeping all other variable differences to zero enables some of these clones to claim virtual look-alike status. These clones have been fairly successful to date. In this group are the Compaq and the Columbia Data Systems PCs. These two manufacturers have even gone to the extreme of duplicating the PC keyboard with all its "peculiarities." Despite the efforts by Compaq to "clone" the IBM PC, DR Logo does not run on the Compaq. The Chameleon offered by Seequa does run DR Logo. It is a very interesting unit. It offers a clone unit with color adapter support as standard. It also contains a Z80 co-processor to take advantage of CP/M-80 programs. Right now there is no mechanism for transferring programs into the Chameleon CP/M-80 format. Using the PC-DOS disk supplied by IBM, the screen has a "waterfall-like" cascade effect when scrolling. It even blinks on scrolling of lines at 80-column width. This gets to be annoying on text files. A separate operating system disk supplied by Seequa does eliminate this annoyance. But then one couldn't quite call it a virtual clone. It is a near clone. Although all men are created equal, not all PCs are, even when it is ordained by the non-IBM manufacturers.

Another factor to be aware of is that many of the software authors did not properly follow standard conventions in using the BIOS for outputting to the screen. Since they knew that the

The Palette by-passes the video display to record graphics directly onto color film.



The Polaroid Palette

IBM display adapters were memory mapped, they wrote directly to the memory-mapped devices. This enabled them to be "fast" but it also made the software manufacturer-specific. Conversion within the same or like operating systems becomes more than a trivial task. It took almost a half of a year before Lotus 1-2-3 became available on an IBM-compatible PC with different screen attributes. Don't assume the ready availability or transfer of software. If you plan on buying a system, make sure the software required is available and not just on the drawing boards to be made available at a later date. Promises are easy to give.

While we are on the subject of software, look for the availability of CP/M-86 on these clone and compatible systems. Today's hype and public media is so strongly focused on MS-DOS and its repertoire of software that one sometimes overlooks CP/M-86. There's a lot of good software in the CP/M-86 operating environment, especially concurrent CP/M with windowing functions for most systems. Some may even include MS-DOS emulation. Software is the catalyst for selection of system; the hardware selection process should only be academic.

The expansion capability of the IBM PC look-alike is another consideration. Many of the units offer clone features, but they tend to lack ability for accepting add-on equipment. This enables them to reduce their cost, which in turn allows them to offer their systems at "lower" prices than Big Brother's. The Seequa requires an expansion box for add-on boards. This add-on expense, if required, could very easily bring you close to IBM's list prices. The Hyperion with basic features assumes the philosophy of the Model T as offered by Henry Ford. It has all the functions they think you will ever need. You then therefore obviously don't need any other capabilities.

The reasons for purchasing an IBM clone are three-fold. The first is portability: since IBM does not at present offer a portable unit per se, many manufacturers have tried to create a market within this niche. The most successful to date has been the Compaq and the KayPro. In the case of the KayPro, the system is only compatible. But these units are not really portable. They are being called transportable because of their relative size and weight.

The second reason for the moderate success of IBM clones is pricing. The differences in net price can be real, but measure carefully your long-term expectations. It may very well not be a bargain if expansion is just not physically possible. The third reason for the popularity of IBM clones is availability. The IBM PC success story has resulted in deliveries with long lead times. Where waiting is not practical, the purchase of these clone units makes good sense.

There are other systems on the market that can lay claim to IBM compatibility or being a near clone. I have only touched upon a few. In time all systems sold will be near clones of the IBM PC. If not, they will not survive in the marketplace. The dominance of IBM is due neither to innovation nor to superiority of product; rather it is the public perception that IBM will still be around when the dust settles. It is estimated that over 60% of new personal computing programming development is for the IBM PC market.

Many personal computer vendors have already fallen by the wayside, of which the most notable has been Osborne. IBM's success is their ability to provide support. Others might call it marketing, but their mystique is real. If you are contemplating the purchase of a clone or compatible, first review your needs and then determine what is deliverable within your means.

In time the IBM PC*jr* will be measured against its Big Brother as a clone. The uniqueness of this comparison is that they are both from IBM. At present over 60% of the PCs and over 80% of the XTs are purchased by corporations. The IBM PC will be the corporate PC de facto standard, while the PC*jr* will be the dominant system in the home. The use of the PC*jr* as an extension of work from the office to the home is the strategy IBM has adopted. In all likelihood, IBM will be correct in their assessment of this market. The PC*jr* will probably have expansion capabilities cloning many of the expansion options of the PC, and as a result will have a direct impact on the clone market.

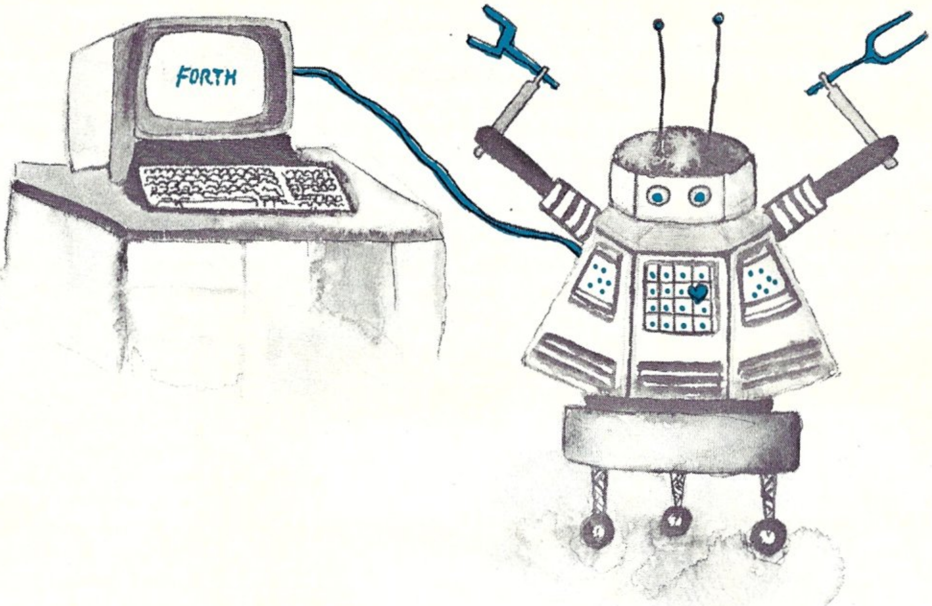
Not all clones are created equal—watch for compatibility problems between systems.

PC*jr* - the product

Much has been written about IBM's entry into the home consumer personal computing market. Most if not all of the commentaries to date have dealt with technical specifications of the PC*jr*. Very little, however, has been said regarding the strategic marketing of this new system. While corporations are struggling with the role of the PC within their organizational culture, we are now assaulted from two fronts—with the IBM PC as a universal workstation in the office, and an extension of this business environment into the home with the PC*jr*. In one of IBM's internal letters, they state very clearly:

"The philosophy is that the PC*jr will allow customers to have a workstation at home that is compatible with their workstations in the office. Customers can communicate with their IBM Personal Computers in the office or can carry*

FORTH gives you **TOTAL CONTROL.**



GRAPHICS • GAMES • COMMUNICATIONS
ROBOTICS • DATA ACQUISITION • PROCESS CONTROL

FORTH: FOR Z-80®, 8086, 68000, and IBM® PC

- ***FORTH*** programs are instantly portable across the four most popular microprocessors.
- ***FORTH*** is interactive and conversational, but 20 times faster than BASIC.
- ***FORTH*** programs are highly structured, modular, easy to maintain.
- ***FORTH*** affords direct control over all interrupts, memory locations, and i/o ports.
- ***FORTH*** allows full access to DOS files and functions.
- ***FORTH*** application programs can be compiled into turnkey COM files and distributed with no license fee.
- ***FORTH*** Cross Compilers are available for ROM'ed or disk based applications on most microprocessors.

Z-80 and 8080 *FORTH* require 48 Kbytes RAM. 8086 and 68000 *FORTH* require 64 Kbytes RAM. Disk formats available include: 8" standard CP/M SSSD, Northstar 5¼" QD, Kaypro 5¼", Apple 5¼", Micro-Mate 5¼", MS-DOS 5¼", Osborne 5¼" DD, and Victor 9000. Most other formats can be special ordered.

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. The entire memory address space of the 68000 or 8086/88 is supported directly for programs and data.

PC FORTH + ... \$250.00
8086 FORTH + for CP/M-86 . . . . . . \$250.00
68000 FORTH + for CP/M-68K . . . \$400.00

Nautilus Cross Compiler allows you to expand or modify the FORTH nucleus, recompile on a host computer for a different target computer, generate headerless and ROM-able code. Supports forward referencing. Produces executable image in RAM or disk file. No license fee for applications. Prerequisite: Application Development System for host computer, \$300.00.

FORTH Native Code Compiler, requires Z-80 FORTH, CP/M 2.2, \$100.00.

Extension Packages

Software floating point (Z-80, 8086, PC only), \$100.00; AMD 9511 support (Z-80, 8086, 68000 only), \$100.00; Intel 8087 support (8086, PC only), \$100.00; Advanced color graphics (PC only), \$100.00; Symbolic interactive debugger (PC only), \$100.00; PC/TERM Communications/file transfer for Smartmodem, \$60.00; Cross reference utility, \$25.00; PC/GEN (custom character sets, PC only), \$50.00; Curry FORTH Programming Aids, \$150.00; Hierarchical file manager, \$50.00; B-Tree index manager, \$125.00; B-Tree index and file manager, \$200.00; QTF + Screen editor for IBM PC, \$100.00;

AUGUSTA, Ada subset compiler from Computer Linguistics for Z-80 CP/M 2.2 systems, \$90.00.

"Starting FORTH" tutorial by Brodie, softcover, \$16.00.

INTEL 8087-3 Numeric Coprocessor, \$250.00.

83 – Standard version of all application development systems available soon. All registered users will be entitled to software update at nominal cost.

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.

VISA® MasterCard

Laboratory Microsystems Incorporated
4147 Beethoven Street, Los Angeles, CA 90066
*Phone credit card orders to **(213) 306-7412***

See us at SOFTCON La Superdome New Orleans Tuesday-Thursday February 21-23, 1984

CIRCLE 26 ON READER SERVICE CARD

SIDELINES

Continued from page 42

disks home. In addition, many of the widely accepted PC applications will operate on PC*jr within its memory and disk constraints."*

There is a high degree of compatibility between the PC*jr* and the PC. There are actually two models of the PC*jr:* the Model 4 and the Model 67; both are compatible members of the IBM PC family (Intel 8088 microprocessor). The Model 4 has 64K, two cartridge slots and a 33-watt power supply. This basic system displays 40 characters per line. IBM has priced it at $669.

The Model 67, priced at $1,269, is the enhanced version of PC*jr,* containing 128K, two cartridge slots, and one 360K disk drive; it has a display expansion of 80 columns per line. The power supply is the same as the basic unit. Each system includes a 62-key cordless infrared keyboard providing all of the functions of the standard 83 keys as on the regular PC.

Since the video display is dynamically mapped, modifications will be required before we can expect to see CP/M-86 for the PC*jr.* The standard operating system for the PC*jr* is PC-DOS 2.1. DOS 2.1 also supports the PC and the XT. Contrary to earlier reports in the press, this version will support one or more $5^1/_4$" disk drives, single- or dual-sided, and one or two fixed disks. DOS 2.1 replaces DOS 2.0 and requires the same amount of storage; however, DOS 2.1 is the only operating system supported by IBM for the PC*jr.* DOS 2.1 has the same function and memory requirements as DOS 2.0. The DOS 2.0 manual has been divided into three manuals. Two manuals (User's Guide and Reference) are included with DOS 2.1. The third manual (Technical Reference) is a separately purchased item. The information contained in the Technical Reference was contained in the DOS 2.0 manual. This is called unbundling. The DOS 2.1 operating system cost $65 asnd the Technical Reference will cost $30.

The PC*jr* connects to a regular home TV with the 0020 RF modulator option and can be connected to the IBM Color Display via the 0021 option. The 0007 option, standard on the Model 67, is required for the higher density display modes. The PC*jr* has the display attributes shown in Table 1.

Software is the catalyst for selecting a system—know your software requirements.

The PC*jr* is not as low cost as it may seem. There are many options this system requires to make it functional, which are normally standard on other competitive systems. Some of these standard options include the parallel printer port attachment at $99; the operating system at $65; the TV connector for $30; and the adapter cable for cassette priced at $30. This seems more like buying a car and adding the engine later at optional expense.

With the announcement of the PC*jr,* the word "portable" may take on a different meaning. The PC*jr* weighs

USE MAGIC/L™ TO WRITE JUST ONE PROGRAM

Bet you can't stop with one.

For just $295 MAGIC/L™ is yours. And suddenly you find you've bought more than a language. MAGIC/L is an entire interactive environment; an assembler, a compiler, an editor, an I/O package, and a system call facility are all wrapped up and delivered. And now MAGIC/L is available for CP/M-based computers.

MAGIC/L is easy to learn. It has syntax similar to C and Pascal, and because it's extensible as well as interactive, it dramatically increases productivity.

Program development features include a built-in text editor, command line recall, CCP, STAT, and PIP command emulation, and the ability to store keyboard dialog on disks.

Key language features include: CHAR, INTEGER, LONG, REAL, and String data types; record structures similar to the STRUCT facility in C; and a complete I/O package that can provide random access, variable length I/O to any CP/M file.

And MAGIC/L offers great portability. Source code which runs on CP/M can be compiled unmodified and run on any other processor.

Typical applications include hardware interfacing, process control, games creation, interactive graphics and image processing. MAGIC/L has made programming easier for DEC, 68000, and Data General users. Now it's working for CP/M users too.

MAGIC/L provides everything you need to write a complete program. But the only way to be convinced is to try it yourself. Send us your $295 check or money order—we also accept MasterCard and VISA—we'll send MAGIC/L for CP/M to you at once. A full money back guarantee is part of the package. Once you've sampled that first program, you'll have to try another . . . and another . . . and another. MAGIC/L . . . it's more than a language.

MAGIC/L . . . It's more than a language

LOKI ENGINEERING, INC.
55 Wheeler St., Cambridge, MA 02138 (617) 576-0666
CIRCLE 61 ON READER SERVICE CARD

MAGIC/L is a trademark of Loki Engineering, Inc.
DEC is a trademark of Digital Equipment Corporation.
CP/M is a trademark of Digital Research, Inc.

SIDELINES

Continued from page 44

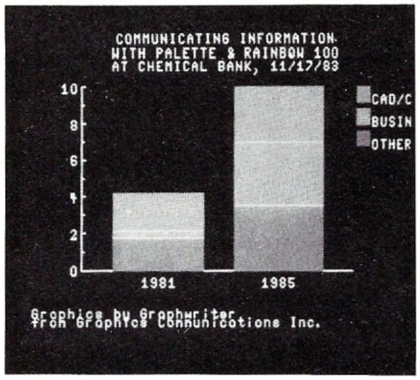
under 9 lbs with the single disk drive. A carrying case is available from IBM. Carry along a Watch-Man-like unit with an RF modultor adapter or the Sony 8½″ color monitor, and you have one heck of a portable personal computing system. Of course the Watch-Man unit would probably result in eyestrain. But in any case, the transportables of today will decline in popularity.

The system has been advertised as having a maximum memory limitation of 128K. But with options, the actual memory available is much lower, as outlined in Table 2.

The following is an initial list of related business/financial application systems available for the PC*jr.*:

- **Multiplan** (spreadsheet)
- **Easywriter** (line editor)
- **VisiCalc** (spreadsheet)
- **Macro Assembler** (software tool)
- **Basic Compiler** (software tool)
- **PFS File/Report** (database query)
- **Dow Jones Reporter** (comm)
- **Time Manager** (project management)
- **Animation Creation** (graphics)

There will be other application systems added to the list. As PC*jr*s are delivered, there will be verification of these PC tools not normally sold through the IBM Product Centers. Lotus 1-2-3 is currently one of the most popular electronic spreadsheet systems in corporate use. However, a minimum of 192K memory is required. Similarly, there are other packages that exceed the



Example of a Graphwriter display on the Palette.

limits of memory in the PC*jr.* The memory constraints of the PC*jr* will motivate many of the present software authors to either make extensive use of program overlays or to provide smaller scale versions of their application systems. Memory may be cheap, as we are told by IBM, but if you can't address it, the value of low-cost memory is academic. The price for use of program overlays is speed of operation. Authors may elect to maintain two different versions. While Lotus 1-2-3 authors are reviewing the marketplace of the PC*jr*, Multiplan may very well have an advantage on both the PC and PC*jr.*

The Polaroid Palette

Have you ever tried to duplicate either graphics or text from your system onto another media? One simple way is to use a camera to photograph directly off the video monitor. The overall quality may not be perfect, but it would do in a pinch. Now Polaroid has introduced the Palette computer image recorder. This system permits a PC user to record graphics as well as text information directly from the PC, bypassing the video display, onto instant 35mm color or monochrome slides. At the time of the announcement it sounded very nice, but it made no impact on me since availability was 4th quarter 1983. In the PC world, that is like a millenium.

It was not until the recent CP/M-83 East show last fall in Boston that I had the opportunity to take a quick gander at the Polaroid Palette. My reaction was very predictable: I wanted immediate delivery!

I have to make numerous presentations during the course of the year. Normally my material is on $8\frac{1}{2} \times 11$ foils. But the foils are constantly being changed, depending on the audience, and quite often the changes are last minute. Also, I am sometimes asked to take part in an overall presentation where slides are required. Doing the artwork on the foils and conversion to slides becomes very tedious. Since the timing is often short, the presentation materials are very amateurish in appearance. It struck me that the Palette was the ideal solution.

Recently I have had the Palette for demonstration and review. The Palette was connected to a DEC Rainbow 100 via the RS-232 serial output port. It very easily mapped information from the Rainbow directly onto Polaroid's 920×700-line resolution monochrome video screen with a tricolor filter wheel. That, by the way, is higher in resolution than any existing personal computing color display on the market today.

The Palette is supplied with a feedthrough for the video signal to allow a monitor to be connected downstream of it. A 75-ohm termination is provided for those systems that do not terminate the video output into a 75-ohm lead.

The Palette system eliminates the need for most of the complex circuitry by having the host computer supply the monochrome video signals and simple ASCII commands to control exposure. The Pallette system gives the user the ability to create greatly enhanced images. Typically, subsets of selected colors from a 72-color Palette are made available to the user for creating individual images. The predefined 72-color Palettes, or "Colorkeys," are in the form of data files encoded with the necessary exposure information for several film types. Predefined color sets simpli-

Table 1. Numbers of colors supported

Resolution	Standard density	High density
160 x 200	16	16
320 x 200	4	16
640 x 200	2	4

Table 2. Actual memory available

		User memory remaining	
	Memory used	Model 4	Model 67
Total system memory		64K	128K
Video buffer	16K	48K	112K
Cassette Basic (ROM)	4K	44K	108K
DOS	24K	24K	88K
Cartridge Basic (w/o DOS)	6K	42K	106K
Cartridge Basic (with DOS)	30K	18K	82K

NEW LOWER PRICES

"THE ORIGINAL BIG BOARD"

OEM - INDUSTRIAL - BUSINESS - SCIENTIFIC

SINGLE BOARD COMPUTER KIT!

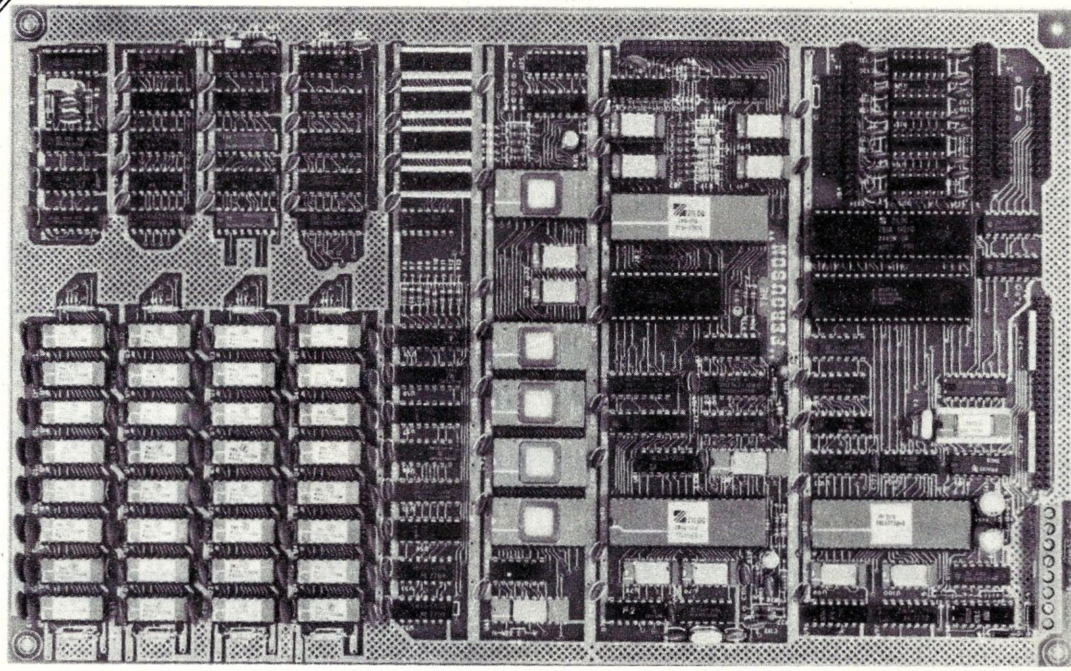
Z-80 CPU! 64K RAM!

(DO NOT CONFUSE WITH ANY OF OUR FLATTERING IMITATORS!)

NEW!

ALL SALES MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. FREE COPY ON REQUEST.

PARTIALLY ASSEMBLED KITS
For All Sockets Installed
And Soldered Add $50.
(Not For Blank PCB)



WANT MORE INFO?
Full Documentation and
Schematics — $5.

THE BIG BOARD PROJECT: With thousands sold worldwide and over two years of field experience, the Big Board may just be one of the most reliable single board computers available today. This is the same design that was licensed by Xerox Corp. as the basis for their 820 computer.

The Big Board gives you the right mix of most needed computing features all on one board. The Big Board was designed from scratch to run the latest version of CP/M*. Just imagine all the off-the-shelf software that can be run on the Big Board without any modifications needed.

$259.00 ** (64K KIT BASIC I/O)

SIZE: 8½ x 13¾ IN.
SAME AS AN 8 IN. DRIVE.
REQUIRES: +5V @ 3 AMPS
+ - 12V @ .5 AMPS.

FULLY SOCKETED!

FEATURES: (Remember, all this on one board!)

64K RAM
Uses Industry standard 4116 RAM's. All 64K is available to the user, our VIDEO and EPROM sections do not make holes in system RAM. Also, very special care was taken in the RAM array PC layout to eliminate potential noise and glitches.

Z-80 CPU
Running at 2.5 MHZ. Handles all 4116 RAM refresh and supports Mode 2 INTERUPTS. Fully buffered and runs 8080 software.

SERIAL I/O (OPTIONAL)
Full 2 channels using the Z80 SIO and the SMC 8116 Baud Rate Generator. FULL RS232! For synchronous or asynchronous communication. In synchronous mode, the clocks can be transmitted or received by a modem. Both channels can be set up for either data-communication or data-terminals. Supports mode 2 Int. Price for all parts and connectors: $39.95

BASIC I/O
Consists of separate parallel port (Z80 PIO) for use with an ASCII encoded keyboard for input. Output would be on the 80 x 24 Video Display.

BLANK PC BOARD — $99.95
The blank Big Board PC Board comes complete with full documentation (including schematics), the character ROM, the PFM 3.3 MONITOR ROM, and a diskette with the source of our BIOS, BOOT, and PFM 3.3 MONITOR.

24 x 80 CHARACTER VIDEO
With a crisp, flicker-free display that looks extremely sharp even on small monitors. Hardware scroll and full cursor control. Composite video or split video and sync. Character set is supplied on a 2716 style ROM, making customized fonts easy. Sync pulses can be any desired length or polarity. Video may be inverted or true. 5 x 7 Matrix - Upper & Lower Case.

FLOPPY DISC CONTROLLER
Uses WD1771 controller chip with a TTL Data Separator for enhanced reliability. IBM 3740 compatible. Supports up to four 8 inch disc drives. Directly compatible with standard Shugart drives such as the SA800 or SA801. Drives can be configured for remote AC off-on. Runs CP/M* 2.2.

TWO PORT PARALLEL I/O (OPTIONAL)
Uses Z-80 PIO. Full 16 bits, fully buffered, bi-directional. Uses selectable hand shake polarity. Set of all parts and connectors for parallel I/O: $19.95

REAL TIME CLOCK (OPTIONAL)
Uses Z-80 CTC. Can be configured as a Counter on Real Time Clock. Set of all parts: $9.95

CP/M* 2.2 FOR BIG BOARD
The popular CP/M* D.O.S. to run on Big Board is available for $139.00.

DOUBLE DENSITY ADAPTER BOARD — $149.95 (A&T)
Requires no cuts or MODS to an existing Big Board. Gives up to 670K storage on a single sided 8 in. diskette. With software to patch your CP/M* 2.2.

PFM 3.3 2K SYSTEM MONITOR
The real power of the Big Board lies in its PFM 3.3 on board monitor. PFM commands include: Dump Memory, Boot CP/M*, Copy, Examine, Fill Memory, Test Memory, Go To, Read and Write I/O Ports, Disc Read (Drive, Track, Sector), and Search PFM occupies one of the four 2716 EPROM locations provided. Z-80 is a Trademark of Zilog.

Digital Research Computers
(OF TEXAS)
P.O. BOX 461565 • GARLAND, TEXAS 75046 • (214) 271-3538

TERMS: Shipments will be made approximately 3 to 6 weeks after we receive your order. VISA, MC, cash accepted. We will accept COD's (for the Big Board only) with a $75 deposit. Balance UPS COD. Add $4.00 shipping.
USA AND CANADA ONLY

*TRADEMARK OF DIGITAL RESEARCH. NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE ORIGINATORS OF CPM SOFTWARE
**1 TO 4 PIECE DOMESTIC USA PRICE.
CIRCLE 15 ON READER SERVICE CARD

SIDELINES

Continued from page 46

fy color selection for individual images and interactive procedures to allow user definition and storage of "colorsets," comprised of any color combination from the 72-color "Colorkey."

A Minolta XG-A body with motor wind and collar adapter as well as Polaroid's Auto Processor are included as part of the system. With the additional option of a normal lens, the 35mm camera becomes an everyday SLR. The 35mm instant slide system by Polaroid is part of the Palette. But other normal 35mm film may be used. Polaroid also supplies a $3^1/_4 \times 4^1/_4$ instant print camera body. The overall system costs $1,500—quite reasonable when one looks at the parts of the system.

The present personal computers that Polaroid supports are the Apple IIe/II Plus, the DEC Rainbow 100, and the IBM PC. The Palette is not interchangeable with different makes of PCs. Since the software drivers are different with each personal computer, I have suggested interchangeable personality modules to Polaroid in the Palette for different PCs as compared to the present product line of a dedicated Palette for each PC model. The graphics package that I used was Graphwriter running under CP/M-80/86. Other graphic softare packages such as Chartmaster and Sign-Master will also make use of the Palette. In bypassing the video display, the actual output resolution is equal to the user's video memory map. There is no degradation in im-

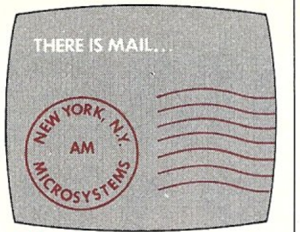
PC*jr* is not as low cost as it seems—"options" that make it functional are standard on other systems.

age quality other than the limits of Polaroid's video monitor system. Therefore, the finer or higher the pixel image map, the greater the detail on the output. On the Rainbow, the results were outstanding. The accompanying photograph shows a very thin yellow line within the bar chart. With normal video output, many systems cannot display this. (see photograph of Graphwriter display on page 46).

For my particular presentation requirements, there are a number of enlargement slide-to-transparency print devices available in the photographic market. Foils can be easily made with any of these units. The per slide unit list cost for the color film from Polaroid is about 80¢. This goes down to about 60¢ per slide when one elects to use 36-exposure rolls. For my work, the 12-exposure roll is more practical. I normaly cannot afford to wait until all 36 exposures have been made before the cartridge film is "instantly" processed by the Auto Processor.

This is overall an excellent product, but perhaps a little steep for the purse strings of a typical hobbyist. But for that matter, what true PC hobbyist has ever worried about cost when he could be the first on the block to own such a unit? The line for these units forms after me. You can always tell the spouse that it is really photography-related and it's for taking pictures of the kids.

Hank Kee, 42-24 Colden St., Flushing, NY 11352



Letters to the Editor

This month... trouble on a static RAM board, a new version of MP/M 8-16, and a special treat for CP/M hackers

Since writing "Digital Research Computers' 16K and 32K Static RAM Boards" (Dec. '83) a problem with the 32K board has occurred. I had purchased an Industrial Micro Systems S8000 DMA floppy disk controller, and the 32K board appears to be incompatible with it. The system would boot, but some utilities would not work correctly. PIP was the utility that brought the problem to my attention, since PIP would not output to the LST: OR CON; device correctly. After much testing and other "finagling," it was determined that the 32K board was causing the problem. Considerable testing was done on the 32K board, none of which revealed anything. It passed all memory tests correctly—it just would not load from the disk. To this day, I do not know why the board is incompatible with the IMS S8000 disk controller.

This is not to say that the board is incompatible with all DMA disk controllers, just that the board doesn't work quite right in my system. A friend of mine has one in a North Star Horizon, and it works perfectly. I still recommend the board if you're sure that it will work in your system. I replaced it in mine with an IMS 370 32K memory board, and am using the DRC board as a storage buffer over on bank 2 where the disk controller isn't addressing. I feel regret that the board doesn't work; perhaps it's a defect in this particular board. I subbed all the memory devices, thinking of a speed problem, but this did not appear to be the problem. I then drug out the logic probe, and proceeded to attempt to troubleshoot the board. Growing tired of this, I subbed all the support chips as well. Nothing worked. The processor appears to address the board correctly, but the disk controller does not. The 16K board appears to work correctly; I haven't had any problems with it, not with the old 2102 8K boards that are in the system. I give up!

Mark D. Pickerill
80 Desmond Rd.
Salinas, CA 93907

Dear Mr. Libes,

Thank you for "Logging-on CP/M" by Ralph J. Jannelli (August, 1983). I have been looking for a logon procedure such as his for sometime. However, I discovered one problem that I presume is a misprint. When I first tested the procedure by entering three incorrect logons, the software went into its loop as it was supposed to do. However, it never came out of it! The problem is in the listing on page 87, in the subsection headed "loopl:". (See below as it appeared and how it should be.)

Figure 1.

```
Loopl: dcx   h
       mov   a,l
       ora   h
       jnz   loopl
       dcr   b
       jnz   loopl
       jmp   byel
```

Figure 2.

```
Loopl: dcx   h
       mov   a,l
       ora   h
       jnz   loopl
       dcr   b
       jz    loopl ←— correction
       jmp   byel
```

As you can see, the flag was being reset, causing the program to continue in the loop. Keep up the good work!

James F. Widner
555 Richards Rd.
Columbus, Ohio
July 30, 1983

Dear Mr. Hunter,

Thank you for taking the time to review MP/M 8-16 (January 1984). We have just released a new version of the MP/M 8-16 operating system which incorporates a number of significant enhancements, including a 325-page user manual. Our new version, 2.1F, beefs up system security by allowing the system manager to require users to "log in" before being allowed access to the system. Login information is maintained in a file called "PASSWD", which can only be modified by the system manager, who can assign users encrypted passwords and restrict them to specific user areas. MP/M-86, as shipped from DRI, has no facility for user login, and can password protect only files, not user areas.

Other features of the new version of MP/M 8-16 include a "HISTORY" file, which keeps an optional record of all system activity. This complements our TIMELOG facility, which generates reports that tabulate the time, date, and duration of each session of every user account and terminal, and totals this data in a statement suitable for use in professional time accounting.

The Gifford implementation of MP/M 8-16 is available from any Gifford Sales Center or from most authorized Full Service CompuPro System Centers.

Jerry Houston
VP Marketing
Gifford Computer Systems
2446 Verna Court
San Leandro, CA 94577


INFINITE SOLUTIONS
Like the molecules in a snowflake, the elements of a computer database can be structured and related in an infinite variety of combinations. Being able to present these combinations quickly and efficiently, with maximum flexibility and minimum programming knowledge, is the mark of excellence which sets the sophisticated database management system apart from the ordinary.
Thousands of satisfied users, from amateur and professional programmers to government agencies, major corporations and industries, have found that DataFlex has no equal in applications development software.
Write or phone for our latest literature and a list of existing applications.
Compatible with CP/M, CP/M-86, MP/M-86, MSDOS, PCDOS, TurboDOS, Novell Sharenet, PC-Net, Molecular N-Star, Televideo MmmOST, Action DPC/OS, Omninet, IBM PC w/Corvus and OSM Muse.
DATAFLEX™
APPLICATIONS DEVELOPMENT SOFTWARE
DATA ACCESS CORPORATION
CIRCLE 215 ON READER SERVICE CARD
8525 S.W. 129th Terrace, Miami, Florida 33156 (305) 238-0012 TELEX 469021 Data Access CI
MSDOS is a trademark of Microsoft. CP/M and MP/M are trademarks of Digital Research. DataFlex is a trademark of Data Access Corp.

LETTERS

Continued from page 50

Dear Mr. Terry,

There are some points in the review of our CPU 86/87, entitled "The CompuPro System 8/16 Model 86/87 Computer (November 1983) that have caused some reader consternation, and I would like to bring them to your attention.

Dr. Bender's excellent article is actually a hardware review of the CPU 86 and two other boards, the RAM 21 (briefly mentioned) and the DISK 1 controller. CompuPro's System 816/D Model 86/87 comes with a CompuPro Desk Enclosure 2. It also includes dual Qume Trak 842 double-sided, dual-density disk drives in another CompuPro enclosure (see photo); the stepper motors on the Qume drives *are* rated for continuous operation. The CompuPro desk enclosure also includes a CompuPro-originated Faraday-shielded active-terminated motherboard as standard equipment; also provided are an Interfacer 3 and 4 and a System Support 1 card. These offer a total of 12 serial ports, one full duplex parallel port, and one Centronics printer port.

In addition, there is a full vectored-interrupt support, a triple 16-bit interval timer, and realtime (day-date-time) clock. The standard System 816/D (86/87) also comes with 512K of 16-bit memory (two RAM 22 256K boards) and 1.5 MB of our MDRIVE/H solid-state disk emulator. As you can see from the foregoing, the hardware reviewed and our System 816/D (86/87) bear little resemblance to each other.

System 816/D (86/87) is a *trademark* of CompuPro used to identify that hardware configuration described above, not that which was reviewed. The title of the review is misleading and has caused confusion in the mind of the public concerning what our trademark "System 816/D (86/87)" stands for.

Jeffrey Swartz
VP, Corp. Communications
CompuPro
3506 Breakwater Court
Hayward, CA 94545

Dear Mr. Libes,

*Re: N*DOS ⟷ CP/M*

I wish to express my deep appreciation for the attention your magazine devotes to the North Star user, and especially for the two articles by David Yates and Allen Ashley in the Sept. 83 issue just to hand.

Here's hoping that some kind soul will transmit them to INSUA (no ad. in September) for those of the lazy finger persuasion like myself.

While it is not fitting material for an article, there is a "quick and dirty" way of transferring from N*-DOS to CP/M. Follow these short steps:

1. In N*DOS, load any monitor that does not come within a bull's roar of 100H.
2. LF FILENAME 100
3. Remove disk from drive 1.
4. Insert and boot CP/M disk.
5. Save #(For No. of Blocks) FILENAME.EXT (.B for my version of North Star Basic under CP/M, .COM for COM files, etc.).

For Basic programs, this would appear to require at least that your basic loads programs for execution at 100H. Mine does, and it works—but it only dawned on me after hand coding a program called CBASE from the INSUA newsletter which does the same thing but deposits the program as an ASCII listing which, as David Yates points out, is not tokenized Basic. In fact it is anything but.

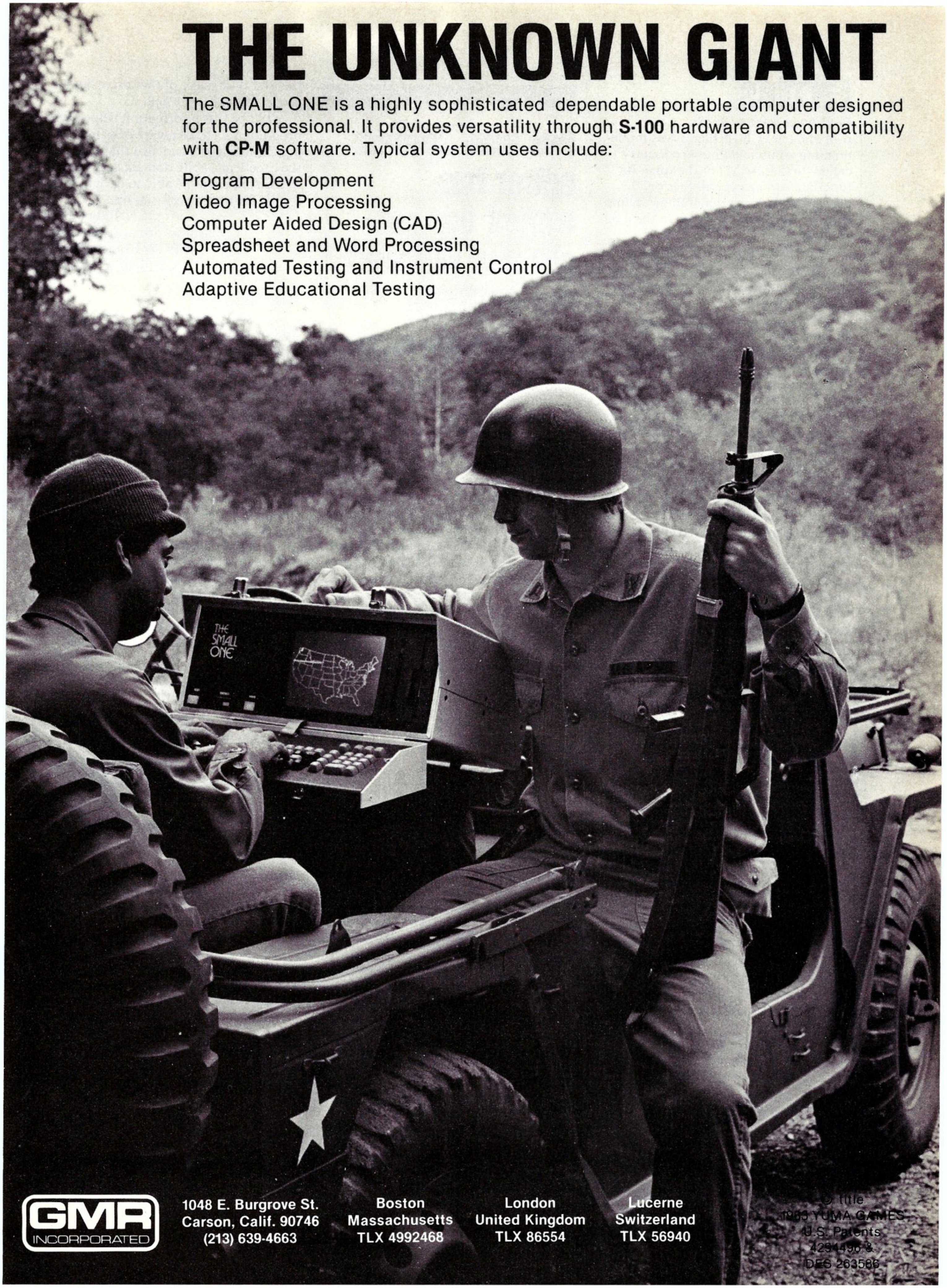
If this helps any other lazy person, I shall be glad.

John Dent
P.O. Box 36
Moonee Ponds, 3039
Victoria, Australia

Dear Mr. Libes,

RE: CP/M 2.2 BIOS Function: SELDSK.

Here is a special treat for CP/M hackers. This tidbit just turned up as the


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
THE SMALL ONE
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
© title
1983 YUMA GAMES
U.S. Patents
4294496 &
DES 263586

LETTERS

Continued from page 52

root of a tricky little bug. I was implementing a public domain program called "FILE.ASM" at the time. Its function is to display all files on all drives and user areas matching the ambiguous file reference given in the command line.

On my system it was stopping after completing drive A: if I had just done a cold boot and hadn't referenced any other drives yet!

Debugging revealed that there is a slight difference in the version 2.2 CP/M BIOS function SELDSK from all earlier versions. There is an additional sentence in the 2.2 manual under SELDSK that says, *The least significant bit of register E is zero is this is the first occurence of the drive select since the last cold or warm start."*

There is no reference at all to register E having to be preset to any special value in earlier versions of the CP/M manual. My BIOS (Morrow) takes advantage of this information and does not do a selection of a drive if the bit is nonzero. It simply returns a value from a local variable it assumes was properly set during the first disk select. I suppose that CP/M keeps this bit correct when calling SELDSK, but any programs that do direct BIOS calls also have the responsibility of managing this bit.

Our new screen editor has tripled our output and reduced our epithets.

One problem with this logic is that an application program has no idea if a drive has already been referenced or not and therefore must always set the bit to 0 for its first reference, even though this may be redundant to the BIOS. Oh well!

Anyway, since this public domain program had random nonzero data in register E, my BIOS was returning a bad value from SELDSK. In fact, the bad value happened to be 0, which the program interpreted to mean nonexistent drive and of course ended its operation!

My simple fix was to clear register E before the call to Seldsk: thus, every disk select is treated as though it were the first disk select of that drive, and a proper value is returned.

I hope this information helps some programmers to be aware of a fine-tuning CP/M detail, and also that users of the excellent public domain software, FILE.ASM, will be able to update their source for better performance.

Billy Smith
Select Information Systems
919 Sir Francis Drake
Kentfield, CA 94947

Dear Mr. Libes:

I rarely respond to "Letters to the Editor", but Dean Dwyer's letter in your November 1983 issue hit a nerve.

I have sent Mr. Dwyer a copy of a screen editor my company has just produced, which is currently under alpha test. The editor is configured for televideo 925/950 terminals and Z80 micros running CP/M 2.2. I will offer this editor, free of charge, to the first 10 people who call me after March 1, 1984.

The editor was produced by my company because no sensible program source code editor was available. My company has been using this editor—and improving it—since July 19, '83. The editor has more than tripled our output, and has reduced four letter words to hell, damn, and nuts!

In short, my programmers were fed up with "You asked for it, you got it"

LETTERS

Continued from page 54

editors, and produced a "What you see is what you got" editor.

My company has no plans to market this editor, as it has already returned our effort tenfold. Those who receive it may use it with our blessings!

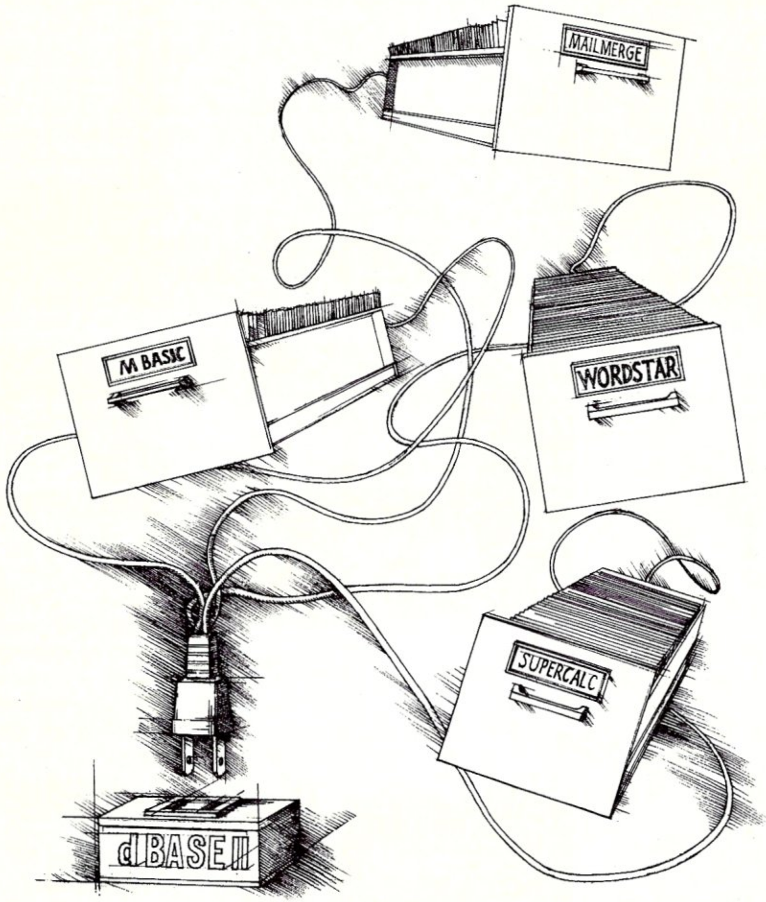
Richard J. Henry, Sr.
Logic Systems of Iowa
1716 4th Ave. SE
Cedar Rapids, IA 52403

Dear Mr. Libes:

Dan Dugan's comments on the DIMS system and the Osborne 1 are well taken, but there is a different solution. As noted in the review last May, the DIMS system is modular. By judiciously choosing which modules to put on the working disk—hardly any database will need *all* of the modules, certainly not in any one working session. You can make enough space even on a 90K Osborne 1 single-density disk for a respectable database. For many operations DGET and DLETTERS are not needed. By eliminating those two files and the "sample" format files you can get the overhead down to 50K, leaving room for a 40K data file. Eliminating DCFORM and the DHELP.DOC saves another 14K.

I'll admit there are some trade-offs in convenience when using a single-density Osborne 1, but the price-to-performance ratio was fantastic when the Osborne came out and was still impossible to even approach when I bought mine in July of 1982. (Okay, I'll admit: I had mine upgraded to DD and added the 80/104 column option. But I could survive without either. This letter was written looking at the little 52-column screen. My only real gripe is the failure to put the "INSERT ON" flag where I could see it without scrolling!)

Benjamin H. Cohen
Box 1674
Chicago, IL 60690



by Bruce Hunter

It's a Database ...an Editor ...a Language ...It's dBASE II

What is dBASE II, a database management tool or a programming language? Rather than fitting neatly into either category, dBASE II can be described more accurately as a cleverly interwoven system consisting of interpreter language created specifically to work with a disk-based database, a database management system, and a set of utilities to create, manipulate and output a database and related programs. The entire package is carefully groomed and orchestrated to work together as a whole; it is also designed for use by both programmers and nonprogrammers.

Only a few short years ago the microcomputer industry was faced with a dilemma. Hardware was available at a price that was affordable by both small businesses and home computerists alike, but there was precious little in the way of software for nontechnical people. Meeting the needs of the nonprogrammer meant instant success for packages like WordStar, VisiCalc and dBASE II, and they soon became industry standards. Of the areas covered by these three software packages (word processing, spread sheets and database managers), only dBASE II continues to have little in the way of real competition.

The major appeal of dBASE II however, lies in its ability to manage a database. Dealing with large amounts of data is now part of daily life, and being able to quickly put your hands on the data you need is a valuable feature. The corollary of this fact, however, is not always apparent to computer novices: if a database is not properly managed, there is no fast and easy way to find and retrieve a specific piece of data.

There are, in effect, only three ways to find and retrieve data from a list of data items. One way is do a sequential read, but of course this has a distinct disadvantage: it takes too long! Another way is to sort the data and then search through it, but the disadvantage of this method is that once you add more data, you have to sort everything all over again. The most efficient way is to maintain index files which are pointers to the data. As new data are entered, indices are extracted and effectively "sorted." Individual data items can then be retrieved by B+trees, hash tables, binary trees and other such techniques. The problem with all these methods is that they each require the ability to program; the last, and most efficient method involves numerous sophisticated programming techniques.

dBASE II utilizes complex programming techniques of which the user never needs to be aware. In particular, a B+tree hierarchical data structure for manipulating its indices. The B+tree is the backbone of file systems such as VSAM and ISAM, well-known on mainframes for many years and more recently introduced to the micro in Digital Research's Access Manager (AM-80 and AM-86). This very sophisticated method of index organization make it possible to search through tens of thousands of records and to find a specific piece of information with only a few probes, thus reducing search time to a matter of microseconds. In dBASE II, the user does not see a hierarchical data index manager, but rather a relational database, one organized with all fields related to neighboring fields.

You can not only manage a database with dBASE II, you can create one as well. The user simply requests dBASE II to "create," and the host computer starts a dialogue with the user, asking for information about what the database fields are going to look like. There are only three data types in dBASE II: character, numeric and logical. It takes only a short time to learn how to enter the required information about the description of the data fields.

The data can then be manipulated in any number of ways, e.g.: they can be sorted and searched, two requirements for any data retrieval system. Searching is accomplished on a parameterized basis. By carefully "wording" a request for data, specific information can be gleaned without burying the user in a mountain of worthless data. There is no reason for dBASE II users to become "buried in data but starved for information." In fact, your information requests can be quite specific. For example, if one of the dBASE II databases contained the results of the year's sales of a company, you could request the total sales of an individual salesman over a specific period of time within a limited area. Not only could you retrieve this information quickly, it could be done without programming and with only two or three command lines.

What computer newcomers don't realize is that if a database is not managed, there is no fast and easy way to retrieve data.

Everything in dBASE II is geared to the care and feeding of the database. Any business or organization has two primary purposes: the service or product with which it deals, and the daily tasks involved in handling the data generated throughout the course of operation. dBASE II is ideally suited to the needs of handling the information processing of a business, particularly because databases can be created and manipulated by users with no programming knowledge.

Once the user has created a data structure, he may freely input data. This can be done with the aid of extensive editing capabilities. Similar to WordStar in nature but not in scope, dBASE II has full cursor control plus the ability to delete and/or overstrike characters within any field. The editor also has the capability to scroll forwards or backwards through the database.

Files can be copied in whole or in part, selectively or indiscriminately. Most commands have the capability of filtering the data. The choice of scope is almost always given, thus limiting a command to one record, expanding its scope to the next or last *n* records, or working on the entire file. Through the use of the usual logical operators (equal, not equal, less than, greater than and so forth) coupled with the Boolean *and*, *or* and *not*, very selective filtering is possible in passing data from one file to another or even out to report forms. For example, eligible bachelors could request a list of all the single ladies over 20 but under 50, living in the cities of Arcadia and Rosemead, but not south of Valley Boulevard, to be found in the next 100 records. This would take just one command. Fields of one database can be replaced with those of another. Single commands will select the number of records meeting a specific description on any number of numeric fields.

Additionally, dBASE II has the ability to write reports. Sometimes it seems that businesses are fueled by reports, and certainly companies working on government contracts generate as much paperwork as finished product! Using dBASE II, you can write reports faster than with any other piece of soft-

IT'S A DATABASE

Continued from page 57

ware I know. Like file creation, report writing is handled nonprogrammatically by a dialogue. It asks the user how many fields, what their width is, what is in the fields by name, and so on. It will also ask for the title of the report and whether you want totals and subtotals. Once the dialogue is finished, dBASE II will automatically generate your report with a choice of screen or printer output. A side benefit is that the format of the report is permanently stored as a disk file: the next time a report has to be written, the report form generation process need not be repeated. Although reports created by dBASE II are somewhat limited in scope, the ease with which they can be written makes up for any limitations.

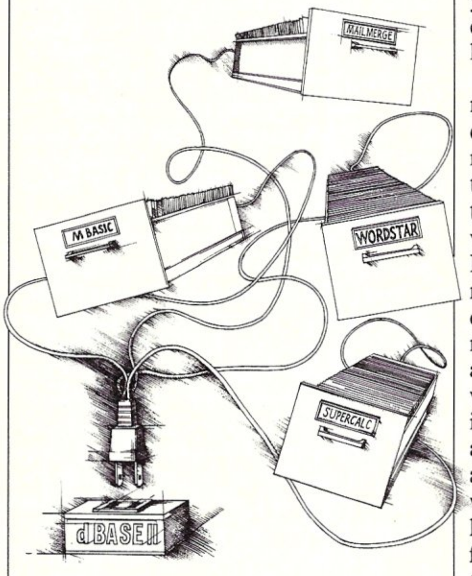
dBASE II as a programming language

Although much of the appeal of dBASE II is due to its ability to function interactively (referred to as the "intermediate mode"), dBASE II is also a fully developed programming language with its own interpreter. It has a do-while, a looping construct, a fully developed if-else that can be nested to any reasonable depth, and a fine "case" structure for multiple branching. It is not a structured language because interpreters are not capable of structured constructs, but it gains a certain degree of structure by the use of "processes"—a "process" is a dBASE II program that is invoked by another dBASE II program. The purpose of this is to allow the programmer to segment his programming tasks and cut them down to size, the philosophy being similar to the "divide and conquer" method of structured languages.

On the other hand, dBASE II does have certain limitations as a programming language. It has no unqualified GOTO (because no line numbers or labels are used); there are neither user-defined functions nor arrays. Also, it is an interpreter, so you can't expect the speed of a compiler. Nevertheless, even with these limitations, dBASE II is a competitive programming language. The reason is simple: no other language available on micros has the built-in capabilities of a database manager.

It's true that micro versions of PL/I (DRI's PL/I-80 and PL/I-86) have indexed files built-in, but the ISAM and VSAM files present in the full set of PL/I are not available in the micro versions. You still have to write your own housekeeping routine for the index files such as the binary tree and so forth—hardly a casual undertaking. Access Manager, used in conjunction with PL/I (or any of Digital Research's programming languages) creates and searches index files, and if your program application involves a database beyond the capabilities of dBASE II (anything in excess of 32,000 records and 16 fields), certainly Access Manager is the way to go.

But dBASE II has advantages of its own to offer the professional program-



Once a user has created a data structure, he may add data freely.

mer. First of all, it's a lot less work to bring up and use! If you're in a hurry, this is a consideration. Also, Access Manager is a database management "tool," so it is not free-standing the way dBASE II is. dBASE II gives you rigidly formatted screens with protective fields, a professional quality screen. Data fields are accessible to the cursor with the advantage of full editing capabilities in certain modes in dBASE II, while protected fields are displayed on the screen and the cursor cannot enter them. This feature can be found only in very sophisticated packages such as Display Manager (DRI's DM-80 and DM-86). A great deal of programming is saved because you don't have to set up the same file information from program to program. In other words, you spend a minimal amount of time on disk output, relieving you of the burden of handling your disk writes. Also, because dBASE II is a relatively easy language to learn, when you train users to use your program, you can take a little longer and also train them to maintain the code. Lastly, dBASE II is a common utility because it is often part of the "free" software supplied with many computers at the time of purchase. Therefore, when you write a dBASE II program for a customer, the odds are that he already has the dBASE II package.

Whether one is new to programming or a "veteran," file handling slows down the programming process. In many languages files seem to be an afterthought. For example, file manipulation in MBasic involves ponderous conversions on the part of the programmer. However, because of dBASE II's data management capabilities, many arduous file handling tasks are cut down to a minimum. It assumes all "valuables" are file names, unless you tell it otherwise. No file housekeeping is done within a program. You don't specify fields at any time in any of the programs—that is all handled by the creation of the file, which is accomplished outside of the language. As a result, a change in the file's structure does not change any of the programs that use it. Files are dBASE II's entire orientation, and this is exemplified by the way it stores memory variables. The command:

```
store 1.98 to amount
```

tells it like it is, notifying dBASE II that there is a variable called "amount" that will have the number 1.98 as its contents. Thus there is none of the confusion of using the equal sign "=" for both assignment and comparison: in dBASE II the equal sign is for comparison only. Files are created so effortlessly and accessed so painlessly that it is almost a surprise when the disk activity makes itself apparent by the noises of head movement.

Looking for information is dBASE II's forte. The secret is the B+tree organization. Any or all file fields can be indexed, and the index files are stored and quietly maintained. Data can thus be maintained within a file in the "as-entered" order, but any field can be searched with lightning speed if the entire file has been sorted on that field. In a B+tree, each indexed filed is sorted, in effect, by the index files. And more than one file can be dealt with at the same time. For example, if one were to create a payroll program, a file of the employee names and hours worked could be opened as the primary file and read sequentially, guaranteeing that no employee would either be missed or

IT'S A DATABASE

Continued from page 58

paid twice. The related files, such as rate files and year-to-date files, could be accessed at random by the use of indices, while a period payroll file could be created sequentially as well.

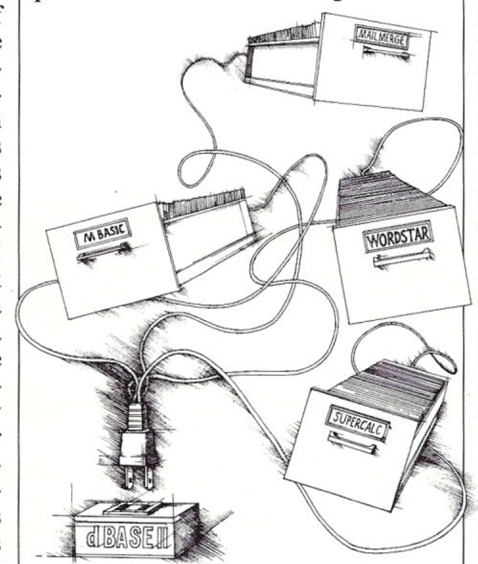
Mail lists, charts of accounts, order entry, inventory, payables, receivables and even general ledger can be programmed in dBASE II. Because dBASE II is free-standing, this can be done without the need of any other software package (except the operating system, of course). Many professional programmers and small systems analysts, myself included, find that dBASE II can be used quite effectively for the rapid development of entire small system packages. For example, it is not uncommon for a consultant to generate 70,000 lines of code in just a few months, which is complex enough to fully automate office procedures that were previously handled inefficiently by many people. I know of one case where a month's work by a number of clerks was reduced to a few hours of data entry and run in a matter of minutes. The capability of the programs generated in dBASE II is limited only by the programmer's ability and imagination and the finite number of records that can be handled. In addition, besides being a quick and inexpensive alternative for the professional, it is easily understood by the customer. As I've already mentioned, with a minimum amount of training, the customer can learn to maintain his own system.

Some general comments on the influence of dBASE II in the industry

Today dBASE II is found everywhere. When an individual buys a microcomputer, he expects that machine to come equipped with software, and informed buyers look for *defacto* standard programs and utilities. An example of a well-balanced software package would include CBasic or MBasic, WordStar, VisiCalc or SuperCalc, and dBASE II. Large system suppliers, such as Gifford Computer Systems, supply dBASE II already installed and integrated within the system. It is also widely used within many industry environments such as JPL.

dBASE II education is proliferating rapidly outside of university environments, particularly in the business world. At computer training centers like ACT (Accelerated Computer Training) in Los Angeles, dBASE II classes are offered weekly as well as many other computer classes. They are primarily designed to introduce people in the business world to computer subjects in a comprehensive but rapid manner, and the most popular class by far is dBASE II. Significantly, programming languages like Basic and Pascal are much less in demand.

Books on dBASE II are beginning to appear. Last year *Everymans's Database Primer* by Robert Byers was initially published by Ashton-Tate, and now Reston, a division of Prentice-Hall, is publishing the book. Reston is coming out with another dBASE II book very soon. Sybex will be offering a dBASE II book entitled *"Understanding dBASE II"* by Alan Simpson, in a few months as well. The presence of dBASE II computer books on the market gives users



dBASE II is a programming language with the built-in capabilities of a database manager.

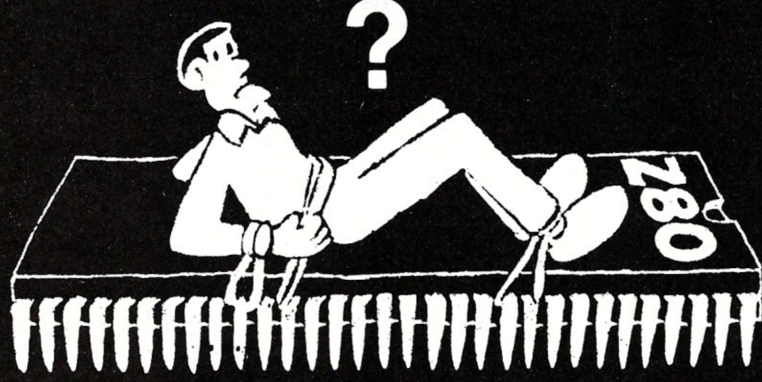
the opportunity for self-education, a vital ingredient to the continued success of any piece of software.

Anything as well accepted in the marketpalce as dBASE II ensures that a contingent of complementary packages will be offered in abundance, all designed to work with and enhance it. Quick-coding packages, graphing packages, scientific packages and numerous others are available in abundance. dBASE II is capable of standing on its own feet, so some of these extra packages are not really necessary: the money for them would often be better spent buying something like a printer! However, some of them are quite handy, such as the scientific extensions.

No other dBASE II imitators come close to the power of the original, as far as I know. The potential use of dBASE II as a database system development language is tremendous. For more complex programming applications, Access Manager, used in conjunction with one of Digital Research's languages offers an enviable amount of programming power which will "outperform" dBASE II in many ways; a comparison of the two is not really viable, because they are not in the same league. Superb in its own right, a database management tool like AM-80 requires the knowledge of a trained programmer to handle it. It is not free standing, requires the use of a programming language to go with it, and it can't store a single character of information on its own. Depending on the application, dBASE II is a better choice. For example, as a consultant I have frequently advised individuals and businessmen to have their first database systems developed in dBASE II. The system can be rapidly developed, thus reducing consultant fees. The customer has the advantage of gaining experience with a computer while running it, and if it should become too slow or his needs start to exceed the record capability of dBASE II, a more sophisticated database system can be developed with a high level language and Access Manager. The money he'd already have made with the dBASE II system usually more than pays for the development of a more sophisticated system; because the algorithms have already been worked out, tried and tested, the cost of creating a professional database management system is substantially reduced.

Since dBASE II is an interpreter, there are two resulting obstacles that inhibit its use as a major development language by professional programmers. First of all, interpreter source code is always available. This is a mixed blessing because the code is open to piracy, the bane of the industry. However, Ashton-Tate offers an encryption capability to overcome this. Secondly, the B+tree as the core of dBASE II is the fastest method of data organization and retrieval you could hope for, but interpreted code leaves a bit to be desired in the way of speed! Hardware modifications can overcome this, however. One of my clients ran a dBASE II program to replace the files of 2,000 records on a CompuPro 8085/8088 with Qume DT8

TIED TO 8-BIT PROCESSING



ADD CO-POWER-88 AND BECOME IBM-PC COMPATIBLE

CO-POWER-88 is an 8088 coprocessor for Z80, CP/M computers. Its fast, 5.33 MHz, and has 256k RAM.

MSDOS comes with CO-POWER-88, and CP/M-86 is available as an option. Simple commands move system control from CP/M to MSDOS or CP/M-86 and back.

Most MSDOS, IBM-PC software runs on a CO-POWER-88 system. Run programs from an IBM-PC disk or transfer them to your system's MSDOS disk.

CO-POWER-88 doesn't replace your Z80, so you can still run CP/M... with an added feature, RAM disk. CO-POWER-88's RAM can be used as a high-speed simulated disk drive during CP/M processing.

Best of all, CO-POWER-88's affordable:

256k CO-POWER-88 w/MSDOS and RAM drive software $500.00

Available for: Actrix, ATR8000, Bigboard, Kaypro, Xerox and Zorba computers. And more to come.

Trademarks: ATR8000 and CO-POWER-88, SWP, Inc. IBM-PC, IBM Corp. Z80, Zilog. MSDOS, Microsoft Inc. CP/M and CP/M-86, Digital Research, Inc.

SWP Inc
MICROCOMPUTER PRODUCTS
2500 E. Randol Mill Road — 125
Arlington, Texas 76011
(817) 861-0421 (metro) 469-1181

CIRCLE 75 ON READER SERVICE CARD

Prices and specifications subject to change without notice. Specifications may vary for particular CO-POWER-88 systems.

Yes. I'd like to know more about CO-POWER-88.

Send to: SWP Microcomputer Products, Inc.
2500 E. Randol Mill Rd., #125
Arlington, TX 76011

#300B

name

street

city

state zip

DEALER INQUIRIES WELCOME.

IT'S A DATABASE

Continued from page 60

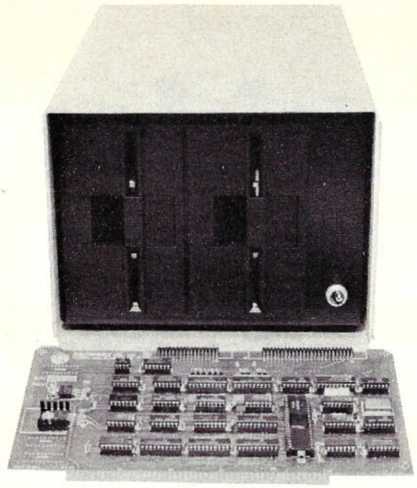
drives, and that took about an hour. The addition of a Gifford hard-disk system to the 8085/8088 front-end board resulted in a program run of five minutes. In this case the cost of both the hard disk and the controller was well justified by the time saved in not having to wrestle with the slower floppy disk drives. In spite of these limitations, the saving grace of dBASE II is that it is a fast, efficient way to create a program that is database-oriented. In many cases coding time can be cut to a quarter of what it would take using conventional programming languages, especially if you have to create your own tools.

dBASE II will have a lasting effect on the industry.

That dBASE II is a definite influence in the industry is aptly demonstrated by a new utility offered by George Eberhard, creator of Computer Innovations CI-C86 16-bit C compiler. This utility is called "C to dBASE II," and it provides read, write and update capabilities to dBASE II format file systems. Because most software packages are being written in C these days, the importance of this utility is readily apparent. This utility would enable the programmer to access dBASE II files from another language, and in the immediate future we will be seeing many other such utilities. Perhaps the ultimate testament of dBASE II's acceptance as an industry standard is the fact that it is used regularly by the Department of Defense. An innovative software utility, dBASE II is not only here to stay, its influence will continue to affect a large portion of the computer industry for years to come.

For more information, contact
Ashton-Tate
10150 W. Jefferson
Culver City, CA 90230
(213) 930-1289
CIRCLE 322 ON READER SERVICE CARD

Bruce Hunter, 1020 South Jenifer, Glendora CA 91740.



DISK MAKER(TM) MAKES *FIFTY* DIFFERENT DISKS

Solving the 5¼" Disk Hassle

Frustrated with the multitude of 5¼" Disk Formats?
Missing sales because you don't have the right format?
Wasting time downloading software?

Solution: **Disk Maker I**

YOUR BENEFITS

- **No more time consuming downloading.** — Finding two computers, connecting them, getting a modem program running in each computer, and then waiting becomes a thing of the past.
- **From yours to theirs** — Your customer needs software on his XYZ but you have it on ABC. Disk Maker does the conversion — at disk speed.
- **One to Many** — Disk Maker can provide disks in any of over FIFTY formats. And the number of formats is increasing monthly.

DISK MAKER II(TM)

Complete standalone Disk Maker II base unit includes one 8" DSDD disk drive, one 48 tpi 5¼" DSDD disk drive, 6 MHZ Z80B, 64K CP/M system with Disk Maker software. An additional 8" drive, a 96 tpi DSDD drive and a 10 M hard disk are available as options. Just plug in your terminal and make disks! Base unit: $2995.00

What is Disk Maker I?
Disk Maker is a product which allows you to format, read and write over FIFTY popular 5¼" disk formats on your existing S-100 computer.

What is included?

- **An S-100 Floppy Disk Controller Board** — supports 4 drives, any combination of 5¼" or 8" — double-sided, double-density, 48 tpi or 96 tpi. And extendable to the new 3" drives in the near future!
- **48 tpi to 5¼" Disk Drive** — double-sided, double-density in a dual drive cabinet and power supply. A second 96 tpi DSDD drive is optional. Drive cables included.
- **Powerful Disk Maker Software.** DMFORM formats diskettes in any of over FIFTY formats. DMSET automatically links Disk Maker with your CP/M system to create up to four additional disk drives on your system. You can then just copy any programs from your system's disk drives to Disk Maker's — using standard CP/M® utilities or our enhanced Multiple Copy (MC) program.

What disk formats can I make?
Any of over FIFTY formats. Osborne, Cromemco, DEC Rainbow and VT180, Epson, Sanyo, TI, Xerox, Eagle, Archives, KayPro, NEC, IBM PC (CP/M 86), SuperBrain, Otrona Attache, Zenith Z-100, Heath (Soft Sector) and TeleVideo to name just a few. And new formats as they are added.

How much does it cost?
Disk Maker I, with S-100 controller board, one 48 tpi DSDD 5¼" disk drive, dual drive cabinet and power supply, cables and Disk Maker software is priced at $1,500.00! (plus shipping.) Please notice: There is no per format charge. All formats currently available are provided at no extra charge. Future software updates are only $25.00.

Options:
96 tpi DSDD 5¼" drive: $385.00
8" DSDD drive, power supply & cabinet: $840.00.

TM - CP/M, CP/M 86 - Digital Research, Disk Maker — New Generation Systems

NEW GENERATION SYSTEMS, inc.

1800 Michael Faraday Drive — Suite 206 — Reston, VA 22090

(703) 471-5598 • **(800) 368-3359**

CIRCLE 123 ON READER SERVICE CARD

Super assemblers plus the world's largest selection of cross assemblers!

Z-80 Macroassembler $49.50

Power for larger programs! This 2500AD macro-assembler includes:

- Zilog Z-80 Macroassembler (with the same powerful features as all our assemblers)
- powerful linker that will link up to 400 files
- Intel 8080 to Zilog Z-80 Source Code Converter (to convert all your Intel source to Zilog Syntax in one simple step)
- COM to Hex Converter (to convert your object files to Hex for PROM creation, etc.)
- 52 page User Manual

8086/88 Assembler with Translator $99.50

Available for MSDOS, PCDOS, or CPM/86! This fully relocatable macroassembler will assemble and link code for MSDOS (PCDOS) AND CPM/86 on either a CPM/86 or MSDOS machine. This package also includes:

- An 8080 to 8086 source code translator (no limit on program size to translate)
- A Z-80 to 8086 translator
- 64 page user manual
- 4 linkers included:
  - –MSDOS produces .EXE file
  - –CPM/86 produces .CMD file
  - –Pure object code generation
  - –Object code and address information only

  Linker features:
  - Links up to 128 files
  - Submit mode invocation
  - Code, Data Stack and extra segments
  - Handles complex overlays
- Written in assembly language for fast assemblies.

Z-8000 Cross Development Package $179.50

Instant Z-8000 Software! This package allows development and conversion of software for the Z8001, 8002, 8003 and 8004 based machines on a Z-80, Z-8000 or 8086 machine. This powerful package includes:

- a Z-80 to Z-8000 Assembly Language Source Code Translator
- an 8080 to Z-8000 Source Code Translator
- Z-8000 Macro Cross Assembler
- Linker and Loader
- COM to Hex File Converter
- a 100 page User Manual
- a Zilog Z-8000 Technical Reference Manual

The Translators provide Z-8000 source code from Intel 8080 or Zilog Z-80 source code. This source code expansion is from 2% to 11%. The Translator outputs a worksheet and a Z-8000 source file. The worksheets show each line of 8080 Z-80 code, with notes to help the programmer to optimize performance, and further lower code expansion. It even comments lines it adds! The Z-8000 source code used by these packages are the unique 2500AD syntax using Zilog mnemonics, designed to make the transition from Z-80 code writing to Z-8000 easy.

All 2500 AD Assemblers and Cross Assemblers support the following features:

Relocatable Code—the packages include a versatile Linker that will link up to 128 files together, or just be used for external reference resolution. The Linker allows Submit Mode or Command Invocation.

Large File Handling Capacity—the Assembler will process files as large as the disk storage device. All buffers including the symbol table buffer overflow to disk.

Powerful Macro Section—handles string comparisons during parameter substitutions. Recursion and nesting limited only by the amount of disk storage available.

Conditional Assembly—allows up to 248 levels of nesting.

Assembly Time Calculator—will perform calculations with up to 16 pending operands, using 16 or 32 Bit arithmetic (32 Bit only for 16 Bit products). The algebraic hierarchy may be changed through the use of parentheses.

CIRCLE 29 ON READER SERVICE CARD

Include files supported—

Listing Control—allows listing of sections on the program with convenient assembly error detection overrides, along with assembly run time commands that may be used to dynamically change the listing mode during assembly.

Hex File Converter, included—for those who have special requirements, and need to generate object code in this format.

Plain English Error Messages—

System requirements for all programs: Z-80 CP/M 2.2 System with 64K RAM and at least a 96 column printer is recommended. Or 8086/88 96K CP/M or MSDOS (PCDOS).

Cross Assembler Special Features

Z-8—256 User defined registers names, directive to engage UPC, Z-80 style syntax.

8748—fully Intel source code compatible.

8051—256 User defined register names, bit addressable naming allowed.

6800 Family—absolute or relocatable modes, all addressing modes supported, Motorola syntax compatible.

6502—Standard syntax or Z-80 type syntax supported, all addressing modes supported.

8086 and Z-8000 XASM includes Source Code Translators

	Z-80 CP/M®	ZILOG SYSTEM 8000 UNIX	IBM P.C. 8086/88 MSDOS	IBM P.C. 8086/88 CP/M 86	OLIVETTI M-20 PCOS
8086/88 ASM			$99.50	$99.50	
8086/88 XASM	$179.50	$750.00			$179.50
Z-8000™ ASM		750.00			299.00
Z-8000 XASM	179.50		179.50	179.50	
Z-80 ASM	49.50				
Z-80 XASM		500.00	79.50	79.50	79.50
Z-8 XASM	79.50	500.00	79.50	79.50	79.50
6502 XASM	79.50	500.00	79.50	79.50	79.50
6800,2,8 XASM	79.50	500.00	79.50	79.50	79.50
6801,03 XASM	79.50	500.00	79.50	79.50	79.50
6805 XASM	79.50	500.00	79.50	79.50	79.50
6809 XASM	99.50	600.00	99.50	99.50	99.50
8748 XASM	79.50	500.00	79.50	79.50	79.50
8051 XASM	79.50	500.00	79.50	79.50	79.50
Subtotal	$ ______	$ ______	$ ______	$ ______	$ ______

Name ____________________

Company ____________________

Address ____________________

City __________ State __________ Zip __________

Phone ____________________ Ext. __________

Make and model of computer system ____________________

☐ C.O.D. *(2500AD pays C.O.D. charges)*

☐ VISA or MasterCard #, Exp. Date (mo./yr.)

____

***TO ORDER.* Simply circle the product or products you want in the price columns above, enter the subtotal at the bottom of that column and add up your total order. Don't forget shipping/handling.**

Total $ __________

shipping/handling ($6.50 per unit) $ __________

Total Order $ __________

Check one:

☐ 8″ Single Density

☐ 5¼″ Osborne

☐ IBM P.C.

☐ Cartridge Tape

☐ Apple (Softcard)

CP/M is a registered trademark of Digital Research, Inc.

Signature ____________________

2500AD SOFTWARE INC.

P.O. Box 441410, Aurora, CO 80014, 303-752-4382

Create Mailmerge-compatible dBASE II Files

Using WordStar to munch those trailing blanks

by Paul W. Heiser

If you've ever used MailMerge to create a personalized letter based upon data in a file derived from a dBASE II file, you've probably run into the "trailing blanks" problem that results in unwanted spaces between words in your letter. This problem arises because of the following two peculiarities:

1. The nature of the internal structure of .TXT files that dBASE II constructs from its own .DBF files.

2. The requirement by MailMerge to have text fields surrounded by quotation marks if the fields contain imbedded commas.

Before getting into the specifics of the problem, let's review a few facts about dBASE II.

To begin with, remember that dBASE II can create records with as many as 32 fields, but each one of these fields has a fixed length that you define when you initially create the file structure. Each field, therefore, has exactly the same length from record to record. Any data field that is not completely filled with characters will automatically include trailing blanks. For example, suppose you have defined a character field called SURNAME and have defined the length of this field to be 10. If you put a 10-character surname into this field, it will not be followed by trailing blanks, since the surname fills up the entire field. On the other hand, if you put a surname like "Smith" into this field, the field will actually contain "Smith" plus five trailing blanks as follows (where the colons indicate the available field width):

```
:Smith     :
```

Let's look at one more field example before moving on. This is an example where commas may be imbedded in the data. Suppose one of the fields in your dBASE II file is COMPANY, which you have defined as a character field of length 15. If you put a company name like "XYZ, Inc." into this field, there is one imbedded comma, and the field contains six trailing blanks as follows:

```
:XYZ, Inc.      :
```

There are two other characteristics of dBASE II files that should also be noted. One is that dBASE II does not insert carriage return/line feed characters at the end of records. The other is that dBASE II does not use commas as delimiters between fields (that is, as field separators). If you try looking at a dBASE II data file with a word processor like WordStar, you will notice that the entire data portion of the file appears on a single line in WordStar due to the absence of carriage return/line feed characters, except for a single carriage return at the end of the STRUCTURE portion of the file.

Since I will later need a sample file to serve as illustration, let's consider a dBASE II file called COMPANY that has the following structure:

```
STRUCTURE FOR FILE:   COMPANY.DBF
NUMBER OF RECORDS:    00000
DATE OF LAST UPDATE:  00/00/00
PRIMARY USE DATABASE
FLD   NAME      TYPE   WIDTH  DEC
001   COMPANY    C      015
002   SURNAME    C      010
003   FNAME      C      010
004   CITY       C      014
005   STATE      C      002
006   ZIP        C      005
**TOTAL**              00057
```

and let's assume that you have added the following data:

```
REC'D   COMPANY          SURNAME
1       XYZ, Inc.        Smith
2       Brown Motors     Brown
3       Fabrics, Inc     Gray

FNAME    CITY      STATE    ZIP
Carl     Topeka    KS       66623
Harry    Boston    MA       02133
Marlow   Miami     FL       33142
```

The other item I will need for illustration is the form letter into which the above data will be merged from a file I will call DATAFILE.TXT derived from the COMPANY file. Let's suppose the form letter is called FORMLET and is constructed as follows, using WordStar:

```
..FORMLET
.op
.df datafile.txt
.rv company,surname,fname,city,
    state,zip

Mr. &fname& &surname&
&company&
&city&, &state& &zip&

Dear Mr. &surname&:
I am pleased to inform you that
&company& has been voted
company of the year.
Congratulations!
                Sincerely,
.pa
```

dBASE II provides several ways for transforming one of its own data files into a structure that can be used by word processors, programming languages and other database management packages. Let's look at two of these ways that are relevant to the problem.

Assume that you are in dBASE II and that you have opened the 'COMPANY' file. You can ask dBASE II to create a copy (which I will call 'DATAFILE') that contains all the data from the 'COMPANY' file but has all text fields surrounded by quotation marks ("). You do this by giving the command:

```
COPY TO datafile DELIMITED WITH"
```

If you exit from dBASE II and look at the DATAFILE.TXT file that dBASE II created (using, for example, the TYPE command in CP/M), you will see the following:

```
"XYZ, Inc.      ","Smith ","Carl
"Brown Motors ","Brown ","Harry
"Fabrics, Inc.","Gray   ","Marlow
","Topeka   ","KS","66623"
","Boston   ","MA","02133"
","Miami    ","FL","33142"
```

Notice that the text in each field is surrounded by quotation marks and that all the trailing blanks have been included in the new file.

dBASE II can transform one of its own datafiles into a structure usable by word processors.

The existence of these trailing blanks is what creates a problem for MailMerge when you print the FORMLET letter. This letter will appear as follows, since all the trailing blanks in the various fields will be printed:

```
Mr. Carl          Smith
XYZ, Inc.
Topeka         , KS 66623

Dear Mr. Smith:

I am pleased to inform you that
XYZ, Inc.     has been voted
company of the year.

Congratulations!
              Sincerely,
```

Now let's look at the second of the two ways in which you might generate a file useable by MailMerge. This way does not result in the carryover of the trailing blanks. Assuming again that you are in dBASE II and the COMPANY file is open, you may give the command:

```
COPY TO datafile DELIMITED WITH,
```

This time, if you exit from dBASE II and look at the DATAFILE.TXT file dBASE II created, you will see the following:

```
XYZ, Inc., Smith, Carl, Topeka,
 KS, 66623
Brown Motors, Brown, Harry,
 Boston, MA, 02133
Fabrics, Inc., Gray, Marlow,
 Miami, FL, 33142
```

On the surface, this appears to be more in line with the kind of structure that MailMerge requires since the trailing blanks have disappeared, but if you look carefully you will see that the records contain varying numbers of commas because of the imbedded commas in some of the field data. Each record contains six fields and should therefore contain five commas, but some of the records have six commas because of the comma that exists in some of the company names.

This is not tolerable for MailMerge, which expects to see commas only as field delimiters unless they are imbedded in text data and surrounded by quotation marks.

Specifically, the form of the file that MailMerge wants to see is:

```
"XYZ, Inc.", Smith, Carl,
 Topeka, KS, 66623
Brown Motors, Brown, Harry,
 Boston, MA, 02133
"Fabrics, Inc.", Gray, Marlow,
 Miami, FL, 33142
```

but it will also gladly settle for:

```
"XYZ, Inc.","Smith","Carl",
 "Topeka","KS","66623"
"Brown Motors","Brown","Harry",
 "Boston","MA","02133"
"Fabrics, Inc.","Gray","Marlow",
 "Miami","FL","33142
```

It would really be convenient if dBASE II provided a command which combined the

```
DELIMITED WITH"
```

and the

```
DELIMITED WITH,
```

features so that the resulting file had no trailing blanks, had comma-delimited fields and had quotation marks surrounding text containing imbedded commas. But such provision is not offered by dBASE II.

Until recently, my solution to this dilemma was to do the following:

1. Create an intermediate file using the DELIMITED WITH" form of the command.

2. Write a Basic program to read the intermediate file, strip the trailing

MAILMERGE

Continued from page 67

blanks from all the fields, and write a new file structured in the way that MailMerge expects to see it.

This works, of course, but is not terribly convenient since the Basic program must be tailored to the detailed structure of the file and therefore needs to be rewritten for every new file you create.

I recently discovered that there is a much easier way to accomplish all this, using some of the capabilities of WordStar. The process is as follows:

1. Starting with the original COMPANY file, create a new file using the command:

```
COPY TO datafile DELIMITED WITH''
```

As you have seen, this results in a new file called DATAFILE.TXT, which appears as follows:

```
"XYZ, Inc.          ","Smith ","Carl
 ","Topeka    ","KS","66623"
"Brown Motors  ","Brown ","Harry
 ","Boston    ","MA","02133"
"Fabrics, Inc. ","Gray ","Marlow
 ","Miami     ","FL","33142"
```

Don't specify single blanks in response to the FIND question.

2. Exit from dBASE II and call up WordStar.

3. Select the nondocument mode of WordStar by typing 'N' at the NO-FILE menu.

4. Type DATAFILE.TXT when WordStar asks for the name of the file to be edited.

5. Use WordStar's find-and-replace command (^QA) and respond to the FIND question by typing two spaces followed by pressing RETURN.

6. Respond to the REPLACE WITH question by simply pressing RETURN.

7. Respond to the **OPTIONS** question by typing **GN** (for Global and Change Without Asking) and pressing **RETURN.**

Based on these instructions, WordStar will now proceed to remove all occurrences of two consecutive spaces without affecting any of the single spaces. After WordStar has finished doing this (and you may have to wait a while if the file is fairly large), the file will appear as follows:

```
"XYZ, Inc.","Smith ","Carl",
 "Topeka","KS","66623"
"Brown Motors ","Brown ",
 "Harry ","Boston","MA","02133"
"Fabrics, Inc. ","Gray","Marlow",
 "Miami ","FL","33142
```

This is almost what we want, but there are still a few places in the file where there is a single space followed by a quotation mark. To get rid of these, do the following:

1. Again type ^QA for **FIND AND REPLACE.**

2. In response to the FIND question, type a single space followed by a quotation mark (") then press the RETURN key.

3. In response to the REPLACE WITH question, type only a quotation mark (") and press RETURN.

4. In response to the OPTIONS question, again type GN and press RETURN.

After WordStar has finished the **FIND AND REPLACE** process (during which you can have another cup of coffee if the file is large), the final result will be:

```
"XYZ, Inc.","Smith","Carl",
 "Topeka","KS","66623"
"Brown Motors","Brown","Harry",
 "Boston","MA","02133"
"Fabrics, Inc.","Gray","Marlow",
 "Miami","FL","33142"
```

which can now be used by MailMerge to print your form letter without difficulty. The resulting form letter will appear as follows:

```
Mr. Carl Smith
XYZ, Inc.
Topeka, KS 66623

Dear Mr. Smith:
I am pleased to inform you that
XYZ, Inc. has been voted
company of the year

Congratulations!
                    Sincerely,
```

which is, of course, what you want.

If you are dealing with a large file that contains many long strings of spaces, you can speed up the process by removing the long strings first, and then working your way down to the two-space strings, always being careful not to specify a single blank in response to the FIND question.

For example, suppose there are many fields containing 20 or so consecutive blanks. After entering the **FIND AND REPLACE** mode by typing

Answer the FIND question with (") to remove trailing blanks.

^QA, press the space bar 20 times in response to the FIND question, press RETURN in response to the REPLACE WITH question, and select the GN options. This will result in the removal of all 20-space strings. Then repeat the process for 15-space strings. Continue to work your way down in this manner to two-space strings.

When WordStar has finished doing its thing, you will find that:

```
A. Some of the fields are exactly
   as you want them.
B. Some of the fields will
   contain two sets of quotation
   marks separated by a single
   space (" ").
C. Some of the fields will
   contain two sets of quotation
   marks not separated by a
   space ("").
D. The remaining fields will
   contain a single space between
   the final field character and
   the terminating quotation mark.
```

To eliminate the (" ") combinations, type ^QA, type (" ") in response to the FIND question, press RETURN in response to the REPLACE WITH question and, as usual, select the GN options.

Do the same for the ("") combinations by typing ("") in response to the FIND question and RETURN in response to the REPLACE WITH question.

Finally, remove the single trailing blank (as in case D above) by typing (") in response to the FIND question and (") in response to the REPLACE WITH question. Be sure to do this step last, otherwise you may find yourself with a single quotation mark between two field-delimiting commas and you'll have no way to get rid of it, short of removing it manually with ^G wherever it appears.

This was obviously much more difficult to explain than to execute, but I think you will find it useful.

Paul W. Heiser, 225 Long Meadow Circle, Pittsford, NY 14534

Banish Gremlins from your Interfaces

Make dBASE II talk to WordStar, SuperCalc, MBasic, and CBasic

by Alan Simpson

With so many software packages available for CP/M-based microcomputers, it is sometimes difficult to select just one for a particular task. Ideally, we'd like to interface numerous software packages utilizing the strengths of each. In this article, I'll discuss interfaces among WordStar, MailMerge, dBASE II, SuperCalc, MBasic, and CBasic (CB-80).

You can interface any two software systems directly in most cases, but since this would create far too many combinations to explain in one article, I will use dBASE II as the general "go-between." I choose dBASE II because it is the easiest to work with. Each type of transfer will be discussed independently. I'll use the symbol (RET) to stand for "press the return key," so that you can identify individual commands typed on the screen.

dBASE II to WordStar

dBASE II allows you to print formatted reports from databases using the REPORT command. Occasionally, you might want to embed a dBASE II report in a WordStar document. This is not difficult. All you have to do is get dBASE II up and running. With the dBASE II dot (.) prompt showing, we SET ALTERNATE TO (any data file name), then SET ALTERNATE ON, and generate the REPORT. Then SET ALTERNATE OFF, and QUIT dBASE II. Now you can load your word processor and read the report database into the word processing system. Here is a typical scenario for this sort of transfer:

```
A> DBASE (RET)
(dot prompt appears)
REPORT FORM FORMNAME (RET)
(fill in report characteristics if necessary)
SET ALTERNATE TO TRANSFER (RET)
SET ALTERNATE ON (RET)
REPORT FORM FORMNAME (RET)
SET ALTERNATE OFF (RET)
QUIT (RET)
A>
```

Let's say you want to pull the dBASE II report into a document called "MANUAL.TXT." In order to do this, you need to invoke WordStar and get MANUAL.TXT into the edit mode. You can do so at the system level with the usual command:

```
WS MANUAL.TXT (RET)
```

Now position the cursor to the place in the document that you want the dBASE II report to appear.

```
^KR
```

WordStar asks:

```
NAME OF FILE TO READ?
```

Reply:

```
TRANSFER.TEXT (RET)
```

and that's all there is to it. Don't forget to add drive assignments (e.g. B:TRANSFER.TXT) if you're working with multiple drives.

dBASE II to MailMerge

You may want to send a dBASE II file to WordStar's MailMerge for printing form letters. To do so, USE the mailing list database, and COPY it to another data file in MailMerge-readable form. In this example, we'll copy to a file named MM.TXT on the disk in Drive B. dBASE II allows you to copy its data files to other formats, including MailMerge files, however, if you use the dBASE II command:

```
COPY TO B: MM.TXT DELIM WITH
  (RET)
```

the MailMerge file will only be OK as long as none of the fields has a comma embedded in it. A comma in a field will wreak havoc on form letters. The other option is to:

```
COPY TO B: MM.TXT DELIM WITH
  " (RET)
```

which puts quotation marks around each field, but unfortunately leaves the fields padded with blanks that will not format properly in the letter. The way I solved this problem was to write a simple Basic program which simply 'unpads' the padded fields, and sets the long fields filled with blanks to nulls. Let's look at an example.

Suppose we have a dBASE II file with 10 fields (LNAME, FNAME, TITLE, COMPANY, ADDRESS, CITY, STATE, ZIP, PHONE, SALUTATION). We want to get its data to a usable MailMerge file. In dBASE II, USE MAIL, and COPY TO B:MM.TXT DELIM WITH ". We then exit dBASE II (QUIT), and when the A> prompt reappears, we ask the operating system to TYPE MM.TXT. You will see something like this:

```
"Reagan ","Ronnie ","President
","UnitedStatesofAmerica",
"1600 Pennsylvania Ave.
","Washington ","D.C. ","01234 ",
"(123) 555-1212
","Mr. President"
"Zeppo ","Z. ","Dog
","123 A. St. ","San Diego ","CA
","92122 ","(619) 453-7120 ",
"Dr. Zeppo "
"Frisbee ","Hobart ","Janitor
","M.V. Co. ","P.O. Box 000 ",
"LaJolla ","CA ","12345 ",
"(000) 123-4567
","Hobart "
```

Basically the right data, but a mess. If you run the little program presented here, using either MBasic or IBM Basic

dBASE II allows us to copy its data files to other formats, e.g. MailMerge.

A, then type B:MM.TXT, we'll see the data file now looks like this:

```
"Reagan","Ronnie","President",
"United States of America","1600
Pennsylvania Ave.",
"Washington","D.C. ","01234",
"(123) 555-1212",
"Mr. President"
"Zeppo","Z." "Dog","",
"123 A. St.",
"San Diego","CA","92122",
"(619) 453-7120","Dr. Zeppo"
"Frisbee","Hobart","Janitor",
"Micro Diabolical Incorp. ","P.O.
Box 000","La Jolla","CA","12345",
(000) 123-4567","Hobart"
```

A more palatable format for MailMerge to work with. Here is the Basic program:

```
 10 REM *******************
    Remove trailing blanks from
    MM files.
 20 PRINT "Preparing MailMerge
    File"
 30 OPEN "I" ,#1, "MM.TXT"
 40 OPEN "O" ,#2, "TEMP.DAT"
 50 DIM F$(10)
 60 WHILE NOT EOF (1)
 70 INPUT #1,F$(1),F$(2),F$(3),
    F$(4),F$(5),F$(6),F$(7),
    F$(8),F$(9),F$(10)
 80 FOR IL%=1 TO 10
 90 SPOT%=INSTR(1,F$(IL%),"  ")
100 IF SPOT%)1 THEN F$(IL%)=LEFT
    $(F$(IL%),SPOT%-1
110 IF SPOT%=1 THEN F$(IL%)=""
120 NEXT IL%
130 WRITE #2,F$(1),F$(2),F$(3),
    F$(4),F$(5),F$(6),F$(7),
    F$(8),F$(9),F$(10)
140 WEND
150 CLOSE
160 KILL "MM.TXT"
170 NAME "TEMP.DAT" AS "MM.TXT"
175 KILL "TEMP.DAT"
180 PRINT "DONE"
```

Lines 70 and 130 are actually one long line each, but I had to break them into two lines to fit them on the page. Don't do this in Basic, and don't try to put in the blank lines I've left between the routines in the program.

The system is set up for 10 fields [DIM F$(10), and in the INPUT# and WRITE# commands. If you have more or fewer fields, you'll have to change these lines, as well as the upper-range of the for-next loop in line 80 (FOR IL%=1 TO 10). Also note that in line 90 there are two blanks between the quotation marks ("bb"), NOT one. Don't change this, or you'll get some pretty unpleasant results.

Now MailMerge can access the data file MM.TXT to create form letters. You naturally have to create the form letter in WordStar to read the new MailMerge file. The following WordStar dot commands at the top of the form letter will read the MailMerge file properly:

```
.OP
.DF B:MM.TXT
.RV LNAME,FNAME,ADDRESS,
     CITY,STATE,ZIP,PHONE
&FNAME& &LNAME&
&TITLE&
&COMPANY&
&ADDRESS&
&CITY&, &STATE& $ZIP&
&PHONE&
Dear &SALUTATION&
     (body of letter here)
```

After the form letter has been created and saved, you merely need to print it using WordStar's Merge-Print Option.

If you already have a MailMerge file and are trying to get it into dBASE II, this, too is feasible. Let's assume that the MailMerge file is currently named "MM.DAT." You need to go into dBASE II and CREATE an empty file with the CREATE command. Structure it so that it has the same fields as your MailMerge file. When dBASE II asks "INPUT DATA NOW?" say "no." Then USE the newly created database, and:

```
APPEND FROM MM.DAT DELIMITED
  (RET)
```

You can now SORT your MailMerge file or do whatever you please with it in dBASE II. To get it from a dBASE II file back to a MailMerge file, follow the example above.

BANISH GREMLINS

Continued from page 71

dBASE II to SuperCalc

SuperCalc stores its data files in a format which causes dBASE II and WordStar to gag when they try to read in one of its data files. The solution to this problem is to create an ASCII version of the SuperCalc file that dBASE II and WordStar can swallow. It goes something like this. First load SuperCalc and read in the desired file with the usual /L command. Next, get rid of borders using the /Global Border command. Then create an ASCII file of the data in SuperCalc. To do so, use the /O as usual. Use the D for (D)isplay, and inform SuperCalc of the range (e.g. A1:E20). Then a prompt will ask if you want the data output to screen, printer or disk. Choose D for disk. When SuperCalc asks for the file name, give it any name you wish. We'll use "DBCALC" for our example. SuperCalc will add the extension ".PRN" to the field name. If you want to store the DBCALC file on Drive B, use the B:DBCALC format as usual. Then quit SuperCalc so that the A> prompt reappears.

Now, load up WordStar and name your document. Then, just use a ^KR to READ in your SuperCalc file. When WordStar asks for the name of the file to read, give it the full name (B:DBCALC.PRN) and that's it.

I'm afraid that I do not have the slightest idea of how to get dBASE II files into SuperCalc files. I think SORCIM has a software package available that can help with this. If you'd rather do it yourself, rumor has it that SuperCalc is written in Pascal.

People with limited disk storage capacity may have to juggle files around a bit. If your SuperCalc file and WordStar document are on separate disks, do Step 1 above with SuperCalc in Drive A, and the SuperCalc data file disk in Drive B. When done, remove both disks, put your WordStar document disk in B, your CP/M system disk in Drive A, and type a control-C (to keep the BDOS errors away). Type PIP, and press RETURN. When you get the * prompt, remove the CP/M system disk from Drive A, and put in the SuperCalc data file disk (the one with B:DBCALC on it.) Type B:=DBCALC.PRN (or whatever you've named your file), and over it goes. Your WordStar document disk now has your WordStar-readable SuperCalc file on it.

Then Step 2 is performed with WordStar in Drive A, and the WordStar document disk in Drive B. It sounds like a bit of an ordeal, I know, but hard disks still cost several grand!

At the end of this procedure DBCALC.PRN finally exists as an ASCII file, and can be read directly into a WordStar document. To get this file into dBASE II requires a little more juggling.

First, you must load dBASE II so that the dot prompt is showing. Then, CREATE a database that will pull in the SuperCalc file. The database you create must have a field for each column in the SuperCalc file. In our example we have five columns (I know this because the range I asked to have SuperCalc output was from A1 to E20). You need to be careful about data types and widths here. The dBASE II field widths must be IDENTICAL to the SuperCalc column widths. Placing SuperCalc's cursor under each column individually will show each column's width down near the left-hand corner of the screen.

dBASE II and WordStar gag on SuperCalc files.

Let's say your SuperCalc file had the first field as account number, the second as account title, the third as MTD balalnce, the fourth a QTD balance, the fifth field as YTD balance. You would need to structure our DBASE II file as follows:

```
FIELD NAME,TYPE,WIDTH,DECIMAL
    PLACES
001 ACCNO,N,6,2
002 TITLE,C,20,0
003 MTD,N,12,2
004 QTD,N,12,2
005 YTD,N,12,2
006
INPUT DATA NOW? (RET)
```

Now you need to USE this newly created file and:

```
APPEND FROM B:DBCALC.PRN SDF (RET)
```

And now you can dBASE II a copy of your SuperCalc file to our hearts' content.

dBASE II to MBasic

The Basic language allows you to create data files with either variable-length records (sequential files), or fixed-length records (random access data files). To send dBASE II files to MBasic sequential files, type:

```
COPY TO BASIC.DAT SAF DELIMITED
 WITH ' (RET)
```

To read Basic sequential data files into dBASE II, load up dBASE II, and CREATE a database with a structure that matches the Basic file. Then, just type:

```
APPEND FROM BASIC.DAT DELIMITED
 (RET)
```

dBASE II will then treat the new version of the Basic data file as one of its own. MailMerge can read MBasic sequential data files directly.

MBasic random access files may present more of a problem. MBasic stores its random access files in binary notation, undelimited with fixed field lengths and no carriage return/line feed at the end. This is indeed a problem. The easiest method for getting an MBasic random access file into a dBASE II database might be to first create a sequential file from the random access file in Basic, then just read the data into dBASE II using the commands described above. Since the EOF function does not work properly with MBasic random access files, we've flagged the last record on the file as having a zero (0) as the last individual's last name. An MBasic program capable of performing such a feat looks like this:

```
  10 REM ********* Send random
     access data to a sequential
     file
  20 OPEN "R", #1, "RFILE.DAT",
     60
  25 FIELD #1, 20 AS F1$, 10 AS
     F2$,15 AS F3$, 9 AS F4$, 5
     AS F5$, 10 AS F6$, 30 AS F7$
  30 OPEN "0", #2, "SFILE.DAT"
  40 FOR REC% = 1 TO 90000
  50 GOSUB 1000 :REM Read next
     R/A record
  55 REM***************Assumes
     a zero marks the last record
  60 IF LEFT$(LNAME$,1)="0"
     THEN 100
  70 WRITE #2, LNAME$,FNAME$,
     ADDRESS$,CITY$,STATE$,ZIP$,
     PHONE$
  80 NEXT REC%
  90 REM ********* Done with
     transfer
 100 CLOSE
 110 SYSTEM
1000 REM ****** Read random
     access record number REC%.
1010 GET #1,REC%
1020 LNAME$=F1$
1030 FNAME$=F2$
1040 ADDRESS$=F3$
1050 CITY$=F4$
1060 STATE$=F5%
1070 ZIP$=F6$
1080 PHONE$=7$
1090 RETURN
```

To send dBASE II data into an MBasic random access file, use the COPY command mentioned above, then treat it as a sequential file, and write an MBasic program to translate the MBasic sequential file to an MBasic random file. Here's the MBasic code for that:

```
  10 REM**********Send Sequential
       data to a random file.
  20 OPEN "R", #1, "RFILE.DAT",
       60
  25 FIELD #1, 20 AS F1$, 10 AS
       F2$, 15 AS F3$, 9 AS D4$, 5
       AS F5$,
       10 AS F6$, 30 AS F7$
  30 OPEN "I", #2, "SFILE.DAT"
  40 FOR REC% = 1 TO 9000
  50 IF EOF(2) THEN 100
  60 INPUT #2, LNAME$,FNAME$,
       ADDRESS$,CITY$,STATE$,ZIP$,
       PHONE$
  70 GOSUB 2000 :REM Write next
       R/A record
  80 NEXT REC%
       (90) REM*********Done with
       transfer
 100 CLOSE
 110 SYSTEM
2000 REM*******Write random
       access record
2010 LSET LNAME$=F1$
2020 LSET FNAME$=F2$
2030 LSET ADDRESS$=F3$
2040 LSET CITY$=F4$
2050 LSET STATE$=F5$
2060 LSET ZIP$F6$
2070 LSET PHONE$=F7$
2080 PUT #1,REC%
2090 RETURN
```

MailMerge can read CBasic files directly.

dBASE II to CBasic

CBasic and CB-80 sequential files can be handled in the same way as MBasic sequential data files described above. CBasic and CB-80 handle random access files a little differently, however. CBasic random files are stored in ASCII delimited with commas, with variable-length fields and a carriage return at the end of each record. CBasic random fields will read a dBASE II file that has been "copied to" another file delimited OK, but then the CBasic or CB-80 program needs to deal with the nuisance of the padded fields. An easier way around this is to have dBASE II mimic the CBasic or CB-80 compiler. You need to CREATE a dBASE II database with one field, with the total length of the fields equalling the record length of the CBasic random access file. In the code below we will assume that the dBASE II file CBASIC.DAT has already been created accordingly.

Now, using the MAIL.BF database as an example, we can write the following routines to create true CBASIC/CB-80 random access data files that have an assumed record length of 155 characters.

```
************* Send dBASE file to
     DBASIC/CB-80 random access
     file.
SELE PRIM
USE B:MAIL
SELE SECO
USE B:CBASIC.DAT
SELE PRIM
DO WHILE .NOT. EOF
SELE SECO
APPEND BLANK
STORE TRIM(LNAME) + ','+TRIM
  (FNAME)+','TRIM(ADDRESS)+',';
  +TRIM(CITY)+','+TRIM(STATE)+',
  'TRIM(ZIP)+',';
  +TRIM(PHONE) TO CRECORD
********************* Pad entire
  record with blanks
DO WHILE LEN(CRECORD) < 155
  STORE CRECORD + ' ' TO CRECORD
ENDDO
REPLACE CBFIELD WITH CRECORD
  SELE PRIM
ENDDO (while not EOF)
```

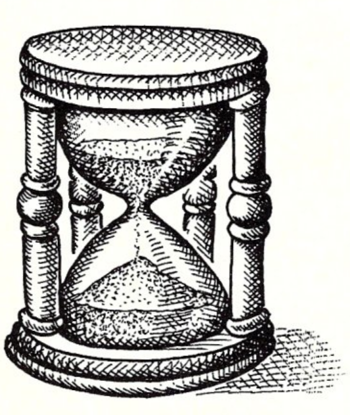
Now the entire CBASIC.DAT file can be COPYed to another file with the SDF option and put in ASCII format to get rid of the dBASE II header. MailMerge can read most CBasic sequential and random access files directly. SuperCalc can read/write MBasic and CBasic data files that have been sent to dBASE II, then sent from dBASE II to Standard Data Format (SDF) files delimited with quotation marks using the procedures described above.

Once you learn the techniques for passing data among packages there is really no reason to 'take sides' with just one CP/M software package when trying to put your computer to work for you. Most CP/M-based software packages are pretty compatible, and we can learn to use the various packages in combination as one massive creative tool. We've merely scratched the surface in this article, the potential is almost limitless.

Alan Simpson, 5328 Bragg St., San Diego CA 92122

Datestamp your dBase II Files

Automatic dating of dBASE II files from a hardware clock/calendar



by Edward Heyman

DBASE II is a powerful microcomputer database manager, and a joy to use! However, there was one petty irritation—the need to key in the date at the beginning of my programs, especially since I have a realtime clock sitting on my bus. The technique that I describe in this article will remove the irritation and will keep you up to date.

The technique is general and will allow you to pass the date to dBASE II from any realtime clock that provides the date. Before going into detail, let me outline the process.

The key is to use a dBASE II memory file called DATE.MEM in conjunction with a C program called DB. Program DB reads the clock, updates DATE.MEM, and then calls dBASE II with the dBASE II command file that you want to run. The first thing that the dBASE II command file does is read DATE.MEM; the command file then sets the internal date value to the correct date, and finally continues with its normal processing.

dBASE II memory files are created with the command "save to [filename]." This command saves all of the variables currently in memory to a file named FILENAME.MEM. The structure of the file can be examined with DDT. Figure 1 is an example of a dBASE II memory file as displayed by DDT as well as a view of the file from within dBASE II. The file is 3K long and has two parts. The first 1K contains the names of the variables and other information. The balance of the file contains the values of the variables.

The structure of the first section of the file can be considered as 64 entries (the maximum number of dBASE II variables allowed) having the following structure:

```
struct variable {
  char   name[10];
   /* the name of the variable */
  char   fill1;
  char   type;
   /* the type of the variable */
  char   length;
   /* the length of the var     */
  char   offset;
   /* location of value of var */
  char   fill2,fill3;
  }
```

The variable types and their codes are:

```
Character          C3 hex
Logical            CC hex
Numerical          CE hex
```

The length is the number of bytes in a

string (dBASE II character variable). The offset is the number of bytes from the beginning of the second part of the file that marks the start of the value of the variable.

The total file can be expressed as follows:

```
struct dbmem   {
  struct  variable  entries[64];
  char   data[1536]
  ;       /* data storage area    */
  char   xtra[512]
  ;       /* not used             */
  }
```

Storage of the value of the variables in the data area is allocated sequentially, and therefore the data area has no predefined structure. The first byte of a character variable value is the length of the string. The characters then follow in order. There is no terminator at the end of a string value.

To create the DATE.MEM file, use the steps shown in Listing 1. The ">" is the CP/M prompt, and the "." is the dBASE II prompt.

Next, let's consider the C program. The program must do the following:

1. Read the DATE.MEM file into memory
2. Locate the variable name MDATE
3. Find the offset to the value of the variable
4. Set the current date from the realtime clock
5. Replace the value of the variable with the new date
6. Write the updated file back to the disk
7. Call dBASE II with the user-selected command file

The program of Listing 2 does the job. A word about the BDS C library file commands:

open(fame,val) opens a file, whose name is FNAME, for reading if val is 0, for writing if val is 1, and for reading and writing if val is 2. The function returns a file descriptor for a good open and -1 for an error.

read(fd,buf,nsec) reads nsec sectors (128 byte) of the file with file descriptor fd into buffer buf; **read**( ) returns the number of sectors read for a good read and -1 for an error.

seek(fd,0,0) resets the file pointer to the beginning of the file. It returns -1 or error.

write(fd,buf,nsec) writes nsec sectors (128 byte) of the file with file descriptor fd from buffer buf. The function returns

FIGURE 1a.

MEMORY DISPLAY FROM WITHIN DBASE

```
 restore from test
. disp memo
A123456789  (C)  ABCDEFGHI
NUMBER      (N)    12345678910
LOGICAL     (L)  .F.
** TOTAL **      03 VARIABLES USED  00015 BYTES USED
```

FIGURE 1b.

DDT EXAMINATION OF THE DBASE MEMORY FILE OF FIGURE 1a.

```
A>ddt test.mem
DDT VERS 2.2
NEXT  PC
0D00 0100
-d100
0100 41 31 32 33 34 35 36 37 38 39 00 C3 09 02 64 00 A123456789....d.
0110 4E 55 4D 42 45 52 00 00 00 00 00 CE 0A 0D 64 00 NUMBER.........d.
0120 4C 4F 47 49 43 41 4C 00 00 00 00 CC 00 14 64 00 LOGICAL.......d.
0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-
-d500
0500 0D 0B 09 41 42 43 44 45 46 47 48 49 07 4A 12 34 ...ABCDEFGHI.J.4
0510 56 78 91 02 00 00 00 00 00 00 00 00 00 00 00 00 Vx..............
0520 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0530 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0540 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0550 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0560 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0570 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0580 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0590 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
05A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
05B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

Listing 1

CREATION OF DATE.MEM FILE

```
A>dbase

ENTER TODAYS DATE  AS MM/DD/YY          OR RETURN FOR NONE  :

***  dBASE II       Ver 2.3B    22 FEB 82
. store '01/01/01' to mdate
01/01/01
. disp memo
MDATE          (C)   01/01/01
** TOTAL **        01 VARIABLES USED  00008 BYTES USED
. save to date
. quit
*** END RUN       dBASE II       ***



A>
```

```
/*                              LISTING 2                                   */

/*              db.c                    Jan  2,1983                         */
/*      program to set the date for DBASE II programs                       */

main(argc,argv)
int argc;
char *argv[];

{
   struct memv {
           char   name[10];          /* the name of the variable           */
           char   fill1;
           char   type;              /* the type of the variable           */
           char   len;               /* for char the length of the string  */
           char   offset;            /* location of value of the variable  */
           char   fill2,fill3;
           } ;
   struct buffer {
           struct memv entries[64];
           char   data[1536];
           char   xtra[512];
           } buff;

   char   newdate[9];                /* the current date as mm/dd/yy       */
   int    fd,sectors;
   int    off;
   int    k;                         /* index of entries[k] array          */
```

DATESTAMP

Continued from page 75

the number of sectors written for a good read and -1 for an error.

The BDS C library command **excel** (prog,arg1,arg2,..,0) loads and executes the program <prog> with the command line arguments [arg1,arg2, ,etc]. The last argument of the command must be zero [0].

Getdate( ) reads the clock and returns the date, formatted as an ASCII string, to the main body of the program.

Two versions of the routine are shown in Listing 2. The first is specific to the Godbout System Support board, and the second to the Scitronics RTC-100 clock board.

If you have a different realtime clock board, write a new getdate( ) and replace the *getdate* given in this article with your new one.

Finally create a dBASE II command file to read DATE.MEM and set the date. Listed below is a simple command file called DATE.CMD that bypasses the dBASE II request for the date and sets and prints the current date.

```
*         DATE.CMD
restore from date     * read the
                        file date.
                        mem
set date to &mdate    * set the
                        dBASE
                        internal
                        date
? date()              * print the
                        date on the
                        console
return
```

To call dBASE II with date setting, use the command

```
A>DB DATE
```

By adding the first two lines of DATE.CMD to any of your dBASE II command files, you can call them with the dating function, by using:

```
A>DB <command file name>
```

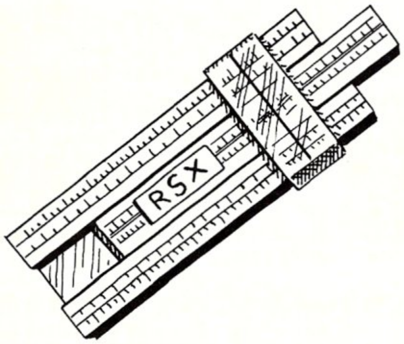
The version of dBASE II used was 2.3b. It is possible that if you have a different version, the memory files will be different. If so, examination with DDT will show you what changes to the structure are necessary.

Edward Heyman, who has a degree in Chemical Engineering from the University of Michigan, is a Product Manager for the DuPont Company. He is the author of several articles on Pascal, which have appeared in Byte.

```
  if( argc != 2 ) exit(puts("usage db <Dbase mem filename>"));

/****   1) read the DATE.MEM file into memory                              ****/
  if((fd = open("DATE.MEM",2)) == -1)
                        exit(puts("could not open DBII mem file"));
  if( read(fd,buff,24) != 24 ) exit(puts("DBII mem file read error"));

/****   2) locate the variable name MDATE                                  ****/
  for(k=0; k<64 && ! (strcmp(buff.entries[k].name,"MDATE") == 0); k++);

/****   3) find the offset to the value of the variable                    ****/
  off = buff.entries[k].offset;

/****   4) set the current date from the real time clock                   ****/
  setdate(newdate);

/****   5) replace the value of the variable with the new date             ****/
  for(k=0; k < 8; k++)  buff.data[off+1+k] = newdate[k];

/****   6) write the updated file back to the disk                         ****/
  if(seek(fd,0,0) == -1) exit(puts("seek error"));
  if(write(fd,buff,24) < 0 ) exit(puts("write error"));

/****   7) call Dbase with the user selected command file                  ****/
  execl("DBASE",argv[1],0);
}/* end program db.c */
/* The following code is specific to the Godbout System Support Clock     */

#define  base           0x50          /* base port address              */
#define  clkcmd         base + 0x0a   /* clock command port             */
#define  clkdata        base + 0x0b   /* clock data port                */
#define  cread          0x10          /* read bit pattern               */
#define  day1           0x07          /* code to enable days one read   */
#define  day10          0x08          /* code to enable days ten read   */
#define  mth1           0x09          /* code to enable month one read  */
#define  mth10          0x0a          /* code to enable month ten read  */
#define  yr1            0x0b          /* code to enable year one read   */
#define  yr10           0x0c          /* code to enable year ten read   */

setdate(newdate)
char *newdate;
{
     int  days1,days10,month1,month10,temp,year1,year10;
     int  dd,mm,yy;

     outp(clkcmd,(cread | day1));       /* set up for days one read      */
     days1 = inp(clkdata);              /* read days one                 */
     outp(clkcmd,(cread | day10));      /* set up for days ten read      */
     days10 = inp(clkdata);             /* read days ten                 */
     outp(clkcmd,(cread | mth1));
     month1 = inp(clkdata);
     outp(clkcmd,(cread | mth10));
     month10 = inp(clkdata);
     outp(clkcmd,(cread | yr1));
     year1 = inp(clkdata);
     outp(clkcmd,(cread | yr10));
     year10 = inp(clkdata);
     outp(clkcmd,0);
     dd = (10 * days10) + days1;
     mm = (10 * month10) + month1;
     yy = (10 * year10) + year1;
     sprintf(newdate,"%2u/%2u/%2u",mm,dd,yy); /* write mm/dd/yy to newdate*/
}/* setdate */
/*
/* The following code is specific to the Scitronics RTC-100 Clock Board */

#define  port0          0x18          /* port0 base address             */
#define  port1          port0 + 0x01
#define  port2          port0 + 0x02
#define  port3          port0 + 0x03

setdate(newdate)
char *newdate;
{
     int  n;
     int  dd,mm,yy;
     char  time[13];

     outp(port1,0xf0);                /* start of stop clock pulse          */
     outp(port0,0x0f);                /* set b0..b3 to output, b4..b7 to input*/
     outp(port3,0xfc);                /* start of clock read pulse          */
     outp(port1,0xf4);
     for(n = 0; n <= 12; n++) {       /* read 13 characters                 */
       outp(port0,n);                 /* set up character to read           */
       time[n] = inp(port0) >> 4;/* read char and shift to ls nibble        */
       }
     outp(port1,0xf8);                /* start sequence to return clock to run*/
     outp(port0,0x0f);
     outp(port3,0xf8);
     outp(port1,0xfc);
     outp(port0,0x0f);

     dd = (time[8] * 10) + time[7];
     mm = (( time[10] & 3) * 10) + time[9];
     yy = (time[12] * 10) + time[11];

     sprintf(newdate,"%2u/%2u/%2u",mm,dd,yy); /* write mm/dd/yy to newdate*/
}/* setdate */
*/
```

Resident System Extensions (RSX)

Enhance CP/M Plus by using your own resident utilities



by Ronald G. Fowler

About a year ago, Digital Research introduced what is probably one of the most ambitious single-task operating systems yet developed for an 8-bit computer: CP/M Plus (also known as CP/M+). This latest enhancement of CP/M adds support for such "big-system" features as time and date stamps for files, file passwords, redirection of input and output, flexible I/O device assignment, and a new concept in micro operating system design: the Resident System Extension, or RSX. In this article, we'll look at RSXs in detail: what they are, how they work, and how to write one. We'll also see how these concepts apply to an actual RSX.

PEEK: an example RSX

Our example RSX is called PEEK, so named because of its ability to give the user a "peek" into the operating system. When activated, PEEK will monitor all file-related system calls made by transients (and the CCP) by presenting a trace of each call on any of three output devices. You can, for example, watch the CCP load a program, or the DIR transient search the directory. It's primary purpose, however, is to act as a debugger for transient programs that access files.

The information displayed by this trace includes the name of the system call (for example, "opnfil" for opening a file, and "erase" for deleting a file) and the value of the DE register pair passed to the system. File control block information is presented, including the drive code, file name, and the FCB information fields EX, S1, S2, RC, and NR. The final item displayed in the trace is the value returned in the A register by the BDOS. All of this information fits on a single 80-character line.

You can send PEEKs output to the console, the list device, or the AUX device. I normally keep a spare terminal attached to my system, assigned as the AUX device, and run PEEK with its output directed to the AUX device. Since most of the programs I write output to the console, this prevents intermixing of output between PEEK and the program under test.

What is an RSX?

The RSX may be viewed as a wedge that is inserted between transient programs and the operating system. The sole function of an RSX is to intercept system calls, modify existing ones, or perhaps even remove some calls from the system altogether.

This wedge is not part of CP/M proper, but is invoked as a transient command after CP/M is initially started up. Before we discuss the specifics of how this is accomplished, however, we must first review program loading in the CP/M+ environment.

The CP/M+ Console Command Processor (CCP), in contrast to that of earlier versions of CP/M, executes as a transient program in low memory; a fresh copy is loaded at system start-up, and again at the termination of each user program. Since one of the functions of the CCP is to load user programs, and since user programs run in the same region of memory as the CCP itself, it follows that the program loader portion of the CCP cannot remain in low memory. And it doesn't.

The program loader is a separate module within CCP.COM (and is itself a special case of an RSX). When the CCP begins execution, it normally relocates the loader to high memory, where it remains until a user program is requested. The loader is then used to bring the user program into memory and transfer control to the new program. At this point, the loader's task is normally finished, and the user program is free to write over the area occupied by the loader; a new copy will be relocated by the CCP when it gets control again, when the user program completes its execution.

The presence of an RSX within the user program, however, modifies this process somewhat (we'll see how the presence of an RSX is detected later). In this case, the loader relocates the RSX code to high memory (just under the loader itself), and modifies the normal system call entry point (at location 5), to point to a special jump field within the RSX. This jump field will transfer control, during each system call, to the RSX system-call intercept code.

The intercept code contains the functionality of the RSX, and may take full advantage of the system, including BDOS calls for file and console I/O. This is accomplished through the use of a field within the RSX called NEXT; this field contains the address of the next function call processor in the system. If there is only one RSX in memory, this field will point to the normal operating system entry point. If, however, there are multiple RSXs in memory, the next field will point to the next older RSX in memory (RSXs always load downward, so the newest RSX is the one lowest in memory). These combined "next" fields thus form a chain that eventually ends at the operating system entry point.

PEEK, our example RSX, uses the next field to pass system calls on to the BDOS (after printing part of the trace display).

It is the responsibility of the call-intercept code to save the calling program's registers (C and DE are the only pertinent registers used in a system call, and the only ones that must be saved). The stack pointer should also be preserved, and a local stack employed, if the intercept code uses any significant amount of stack space.

Normally, RSXs are attached to transient programs and exist in memory only for the duration of the transient. When the transient terminates, and the CCP is reloaded, all RSXs are removed from memory.

The RSX will intercept, modify, and even remove system calls.

The RSX may be maintained in memory, however, by setting a boolean field within the RSX, which will prevent its removal by the CCP. The RSX then becomes permanently resident, and will not be removed from memory until this "permanent" boolean is reset (an action that the RSX itself may take). Note that the presence of any permanent RSXs will cause the program loader to remain resident also; the space occupied by the loader cannot be reclaimed until all permanent RSXs are removed from the system.

RSX communication

A special system call (number 60) has been provided in CP/M+ to allow transient programs to communicate with RSXs in memory. This communication must take place via system-call interception of function 60 by the RSXs; this system call is not acted upon by the CP/M+ BDOS, and any function-60 calls that pass all the way up to the BDOS will simply return the value 255 in the A and L registers.

Our example RSX consists of two sections: the RSX itself (listing 2), and a communication section (listing 1). The two are joined by the GENCOM utility, and exist as a single transient file. The purpose of the communication section is to pass command line arguments to the RSX. It uses function 60 to accomplish this.

Function 60 takes a single argument in register pair DE: the address of an RSX parameter block (RSXPB). The RSXPB is a data structure in memory formatted into fields as follows:

Function number (byte). The function number is similar in concept to the numbered system calls. It is defined by the RSX itself, and may have any value *except* that the values 128-255 are reserved for use by RSXs provided with the CP/M+ system. The implied purpose of this field is to allow the caller (normally a transient running in the TPA) to identify *which* RSX it is trying to communicate with.

Number of parameters (byte). This is the number of word-parameters that the RSX should expect.

Parameters (word). These are values passed to the RSX, the number of which is defined in field #2. These parameters are defined entirely by the RSX, and may contain addresses, characters, numbers, etc.

I should point out that, since the operating system itself takes no action on the parameter block, there is no real need for user-written RSXs to honor the format of the parameter block *unless* your RSX must coexist in memory with RSXs supplied with the CP/M+ system (the GET, PUT and SUBMIT programs are examples of supplied utilities with imbedded RSXs).

In our example RSX, we ignore the function number entirely in the function-60 routine; I felt a more positive identification of the requested RSX was available, since RSXs are named, and the name exists within the RSX itself. Hence, I defined the first word-parameter as a pointer to the name of the RSX. Thus, the caller passes the name of the requested RSX (actually, a pointer to the name), and the RSX verifies the passed name against its own internal name. If the names do not match, the call is passed upward via the next field. Otherwise, the call is processed locally, and in our case, allows the display output device to be changed. In the calling code (listing 1), I set the function number to 127. The purpose of this is to avoid conflict with Digital Research-supplied RSXs (which use codes 128 to 255); hopefully, the use of a high function number will also prevent conflict with any other user-written RSXs that I may find myself using.

Earlier, I mentioned that both our example RSX and the code that communicates with it reside in the same transient program file. It's obvious, therefore, that any invocation of PEEK

USE THE RSX

Continued from page 79

will cause a new copy of the PEEK RSX to be wedged into the system, even if one already exists. To prevent memory from filling up with PEEK RSXs every time I switch the output device, I added some code in the function-60 processor, the logic of which goes like this: PEEK's function 60 always returns A=0. Since CP/M returns A=255 for any function 60 call that reaches BDOS, the highest PEEK that can remain in memory is that which receives A=255 from a function-60 call. If A=0 after the call, then there is a copy of PEEK higher in memory; thus the current peak must set its "remove-me" boolean, causing the CCP to delete the bogus PEEK at the next warm start.

RSX generation

Since RSXs must be capable of execution anywhere in memory, special relocation information must be maintained within the RSX. The format used is called "page-relocation" (2), and allows the RSX to be loaded at any page boundary in memory. The source file must first be assembled with RMAC (supplied as part of the CP/M+ package); the object produced by the assembler is a fully relocatable module, in the industry-standard MicroSoft relocation format. This object module is then passed through the linker (also supplied with the system) with the linker's "P" option invoked. The output file is the "raw" RSX, which must then be processed by the GENCOM utility supplied with the system, to produce the result of the whole process: a finished COM file.

This sounds complex, but is actually quite straightforward. Let's walk through the process, using our example RSX.

Our RSX code is given in listing 2, which should be typed in with your text editor into a file called PEEKBDOS.ASM. Assemble the program with RMAC using the command line

```
RMAC PEEKBDOS $PZSZ
```

Note that the options (denoted by the dollar sign) suppress the creation of the print and symbol files. You may want to omit the options altogether, or send both files out to the printer using the option "$PPSP". In any case, the object file will be contained in the file PEEKBDOS.REL which must then be processed by the linker. We can use the command line

```
LINK PEEKBDOS.RSX=PEEKBDOS[P]
```

for this purpose. The P option instructs the linker to output a page-relocatable output file.

Now we have the RSX portion ready. Next we must generate the communications module.

Type listing 1 into a file and name it PEEK.ASM. Assemble it with RMAC:

```
RMAC PEEK $PZSZ
```

and load it with the linker:

```
LINK PEEK
```

producing the output file PEEK.COM. Now we are ready to combine the two using GENCOM:

```
GENCOM PEEK PEEKBDOS.
```

Note that the first argument to GENCOM is the COM file to which the RSX (specified in the second argument) is to be attached. The output of GENCOM is a special kind of COM file that contains a 256-byte RSX header at its beginning (not to be confused with the RSX prefix, which will be described shortly). The header is identified by the fact that its first byte (and thus the first byte of the COM file) is the value 0C9 hex, which is a "return" code in 8080 assembly language. Since a COM file that begins with a return to the operating system is meaningless, any such files are considered to have an RSX header, and thus, embedded RSXs.

The header contains the number of RSXs present in the file (there may be up to 15), and a 16-byte record for each embedded RSX. These RSX records within the header contain such information as the name of the RSX, its location relative to the beginning of the file, the PRL header information (the RSXs PRL header is stripped out when the RSX is added): bit-map location relative to the start of the RSX, and additional memory requirements of the RSX. All of this is necessary for the program loader to find and relocate the component RSXs.

Since GENCOM moved the entire program up by 256 bytes to make room for the header, the loader must move it back down after extracting the RSXs described by the header. Thus the original COM file that was linked with an RSX by GENCOM is now in memory at its correct execution address (100H).

The RSX prefix

Up until now, I've mentioned various fields within the RSX, without specifying exactly where within the RSX they may be found. In fact, the format of these fields is defined within the *RSX prefix,* a data structure located at the beginning of the RSX. The RSX prefix is 27 bytes long, and adheres to the following format:

Serial (6 bytes, not initialized) Since an RSX must appear to be part of the system after it is relocated, it's origin page number must start with the CP/M serial number. The program loader copies the system serial number to this field.

Start (3 bytes, JMP XXXX, initialized) this is the next field described previously. It contains a jump instruction (set up by the program loader) to the next RSX in line (or the BDOS entry point if there are no more).

Prev (word, not initialized). Contains a pointer to the previous RSX (or location 5, if there are no more). Not normally used by the RSX, this field is

```
                          title   'PEEK.ASM: tpa module for PEEK RSX'
                ;
                ;
                ; copyright (C) 1983 : Ron Fowler, Fort Atkinson, WI
                ;
                ; This is the TPA module for the PEEK rsx; it must be combined with
                ; PEEKBDOS using the GENCOM utility, as described in the CPM+ Pro-
                ; grammer's Guide.  Syntax and usage information is contained in the
                ; companion module, PEEKBDOS.ASM.
                ;
                ; equates
                ;
0005 =          bdos    equ     5               ;system entry point
0009 =          printf  equ     9               ;print console buffer funct #
003C =          rsxf    equ     60              ;"CALL RSX" system call
                ;
005C =          dfcbl   equ     5ch             ;default file control block 1
006C =          dfcb2   equ     6ch             ;default file control block 2
0100 =          tpa     equ     100h            ;transient program area
                ;
000D =          cr      equ     13              ;carriage-return code
000A =          lf      equ     10              ;linefeed code
                ;
                ; by the time we get running, the CPM+ loader has already loaded
                ; and activated the rsp (or it may already be in memory from a
                ; previous load).  So all we have to do is parse any arguments from
                ; the command line, and pass them on to the rsx using function 60.
                ;
0000 215D00             lxi     h,dfcbl+1       ;get pointer to command line arg
0003 7E                 mov     a,m             ;get optional "-" (for unix people)
0004 FE2D               cpi     '-'
0006 C20A00             jnz     nodash          ;jump if not there
0009 23                 inx     h               ;yep, advance past it
000A CD2100     nodash: call    parse           ;parse first argument
000D CD2100             call    parse           ;parse possible second argument
0010 7E                 mov     a,m             ;shouldn't be a third
0011 FE20               cpi     ' '             ;(ie, 3rd position should be blank)
0013 C24700             jnz     synerr          ;if bad, jump
```

```
0016 117500   sendit: lxi     d,rsxpb         ;send to rsp
0019 0E3C             mvi     c,rsxf
001B CD0500           call    bdos
001E C30000           jmp     0               ;all done, warm-boot
              ;
              ; parse an argument
              ;
0021 7E       parse:  mov     a,m             ;fetch argument
0022 23               inx     h               ;advance pointer
0023 FE20             cpi     ' '             ;allow nulls
0025 C8               rz
0026 FE50             cpi     'P'             ;pause-flag?
0028 C23100           jnz     npars           ;jump if not
002B 3E01             mvi     a,1             ;pause-flag
002D 328400           sta     args+1          ;where it goes
0030 C9               ret
0031 CD3800   npars:  call    valid           ;validate
0034 328300           sta     args
CP/M RMAC ASSEM 1.1   #002    PEEK.ASM: TPA MODULE FOR PEEK RSX

0037 C9               ret
              ;
0038 FE51     valid:  cpi     'Q'             ;quit...
003A C8               rz
003B FE41             cpi     'A'             ;out to aux
003D C8               rz
003E FE43             cpi     'C'             ;out to con
0040 C8               rz
0041 FE4C             cpi     'L'             ;out to list
0043 C24700           jnz     synerr          ;what then?
0046 C9               ret
              ;
              ; syntax error
              ;
0047 115700   synerr: lxi     d,synems        ;oops, bomb out
004A 0E09             mvi     c,printf        ;print string function #
004C CD0500           call    bdos
004F 3E51             mvi     a,'Q'           ;insure removal of rsx
0051 328300           sta     args
0054 C31600           jmp     sendit          ;go send, exit
              ;
0057 0D0A436F6Dsynems: db     cr,lf,'Command Line Syntax Error',cr,lf,'$'
              ;
              ; rsx parameter block
              ;
0075 7F       rsxpb:  db      127             ;function # (we ignore)
0076 02               db      2               ;2 parameters
0077 7B00             dw      myname          ;parm1 = rsx name
0079 8300             dw      args            ;parm2 = command line arguments
              ;
              ; name of rsx
              ;
007B 5045454B42myname: db     'PEEKBDOS'      ;name of RSX we're calling
0083 41FF     args:   db      'A',0ffh        ;output-device, pause-flag
              ;
0085                  end
CP/M RMAC ASSEM 1.1   #001    BDOS-PEEK RSX FOR CP/M 3.0

                      title   'BDOS-PEEK RSX for CP/M 3.0'
              ;
              ;
              ;                         ***************************
              ;                         *     PEEKBDOS.ASM        *
              ;                         *     Version 1.0         *
              ;                         *   Copyright (c) 1983    *
              ;                         *          by             *
              ;                         *      Ron Fowler         *
              ;                         ***************************
              ;
              ;
              ; This program is an RSX running under CP/M 3.0 (only).  It's purpose is
              ; to trace file-related BDOS calls, and display information regarding each
              ; call on the AUX device.  It must be combined with the companion module,
              ; "PEEK.ASM" using GENCOM, as described in the CP/M+ programmer's guide.
              ;
              ; PEEK's output device defaults to the system console, and can be changed
              ; by doing "peek a" (aux), "peek l" (list), or "peek c" (console).  Note
              ; that "peek p" sets the pause flag, causing the trace to pause after the
              ; call to bdos.  Note also that a "*" is printed at the pause;  the program
              ; then waits for input.  Any character other than "Q" will simply cause a
              ; return to the caller.  A "Q" resets the pause flag, and the program then
              ; displays without pausing.
              ;
              ; "PEEK Q" removes the RSX from memory.
              ;
              ; I find this very useful for debugging programs.  Typically, I keep a term-
              ; attached to my AUX device (well, actually, it's a separate computer, run-
              ; ning a modem program across a serial link, but that's another story...) --
              ; this RSX outputs to the AUX device to keep the output from the program
              ; from getting mixed up with the output from this RSX.  Hence, I get a full
              ; "snapshot" of file operations made by the program under test.
              ;
              ; Note that this RSX uses one level of caller's stack.  Also, if you modify
              ; this code, be VERY careful with the registers;  I save only what is absol-
              ; utely necessary, in order to conserve memory.
              ;
              ;                         Ron Fowler
              ;                         Fort Atkinson, WI
              ;                         08/22/83
```

USE THE RSX *Continued from page 81*

```
                 ;
                 ; UPDATE INFORMATION:
                 ;
                 ; -- if you update this program, please pass along a copy to Fort Fone File
                 ;    Folder (FFFF) RCP/M, Fort Atkinson, WI: (414) 563-9932 (no ringback).
                 ;    Please update the version number in both modules (PEEK.ASM also).
                 ;
                 ; updates (in reverse order to minimize reading time):
                 ;
                 ; 1.0   originally written by Ron Fowler   08/22/83
                 ;
                 ;-----------------------------------------------------------
                 ;
                 ; CP/M equates
                 ;
0002 =           conotf  equ     2                 ;output character to console
CP/M RMAC ASSEM 1.1      #002    BDOS-PEEK RSX FOR CP/M 3.0

 0004 =           auxotf  equ     4                ;aux output
 0005 =           lstotf  equ     5                ;list output
 0006 =           diriof  equ     6                ;direct console i/o
 0009 =           printf  equ     9                ;print string function
 000F =           openf   equ     15               ;open file
 0010 =           closf   equ     16               ;close file
 0011 =           srchf   equ     17               ;search for file 1st occurance
 0012 =           srchn   equ     18               ;search for file next occurance
 0013 =           erasf   equ     19               ;erase file
 0014 =           readf   equ     20               ;read sequential
 0015 =           writf   equ     21               ;write sequential
 0016 =           creat   equ     22               ;create file
 0017 =           renmf   equ     23               ;rename file
 001A =           sdmaf   equ     26               ;set transfer address
 0021 =           rdran   equ     33               ;read random
 0022 =           wrran   equ     34               ;write random
                  ;
                  ; character equates
                  ;
 000D =           cr      equ     13               ;carriage-return code
 000A =           lf      equ     10               ;linefeed code
                  ;
                  ;
                  ;
                  ; RSX prefix
                  ;
 0000 0000000000          db      0,0,0,0,0,0      ;space for CPM3 to insert serial #
 0006 C31B00              jmp     begin            ;jump to entry point
 0009 C3          next:   db      jmp              ;jump instruction op-code
 000A 0000                dw      0                ;next RSX or loader
 000C 0000                dw      0                ;previous RSX or 0
 000E 00          rmvflg: db      0                ;removes RSX when non-0
 000F 00                  db      0                ;flag for non-banked only
 0010 5045454B42myname:   db      'PEEKBDOS'       ;our name
 0018 00                  db      0                ;loader flag
 0019 0000                db      0,0              ;reserved
                  ;
                  ; This is the BDOS call intercept.  Here we determine if the
                  ; call is one of those we process.  If not, we simply pass
                  ; control on to the next module.
                  ;
 001B 79          begin:  mov     a,c              ;get function code
 001C FE3C                cpi     60               ;for RSX's only?
 001E CADC00              jz      rsxfnc           ;go if so
 0021 218701              lxi     h,fnctbl         ;nope, test for one of ours
 0024 7E          scan:   mov     a,m              ;get first/next table opcode
 0025 23                  inx     h
 0026 B7                  ora     a                ;end-of-table?
 0027 CA0900              jz      next             ;then quit now
 002A B9                  cmp     c                ;ours?
 002B CA3600              jz      trap             ;go trap if so
 002E 3E06                mvi     a,6              ;nope, calculate next table entry
 0030 CD8201              call    addha
 0033 C32400              jmp     scan             ;continue
                  ;
                  ; Here when we've determined that we must display the system call.
CP/M RMAC ASSEM 1.1      #003    BDOS-PEEK RSX FOR CP/M 3.0

                  ; First save caller's stackpointer and load a local stack.
                  ;
 0036 22F301      trap:   shld    tblptr           ;save pointer to function string
 0039 210000              lxi     h,0              ;get user's sp
 003C 39                  dad     sp
 003D 22F701              shld    spsave
 0040 311102              lxi     sp,stack         ;load local stack
                  ;
                  ; begin the trace: do newline, then print the function string
                  ;
 0043 3E0D                mvi     a,cr             ;carriage
 0045 CD6F01              call    type
 0048 3E0A                mvi     a,lf
 004A CD6F01              call    type
 004D 2AF301              lhld    tblptr           ;get string pointer
 0050 CD4B01              call    prathl           ;print it
 0053 CD4501              call    ilprnt           ;colon, open paren
 0056 3A20                db      ': '
 0058 A8                  db      '('+80h
```

```
                   ;
                   ; print fcb address (or dma address)
                   ;
0059 7A                     mov     a,d             ;get parameter hi
005A CD5E01                 call    hexout
005D 7B                     mov     a,e             ;then lo
005E CD5E01                 call    hexout
0061 CD4501                 call    ilprnt          ;closing paren
0064 29A0                   db      ')',' '+80h
0066 79                     mov     a,c             ;get function code
0067 FE1A                   cpi     sdmaf           ;set-dma?
0069 CA9F00                 jz      exit            ;then done
                   ;
                   ; Not setdma function.  Print out the FCB information
                   ;
006C D5                     push    d               ;save fcb pointer
006D 1A                     ldax    d               ;get drive code
006E 13                     inx     d
006F CD5601                 call    hexsp           ;print hex byte, space
                   ;
                   ; Print the file name
                   ;
0072 060B                   mvi     b,11            ;eleven characters
0074 1A            fnprnt:  ldax    d               ;get first/next char
0075 13                     inx     d               ;advance fcb pointer
0076 CD6F01                 call    type            ;print it
0079 05                     dcr     b               ;all 11
007A C27400                 jnz     fnprnt
007D 3E20                   mvi     a,' '           ;separate filename
007F CD6F01                 call    type
                   ;
                   ; print ex,s1,s2,rc
                   ;
0082 0604                   mvi     b,4             ;four of these
0084 21DC01                 lxi     h,idbase        ;ascii id's for each field
0087 CD4B01        idloop:  call    prathl          ;print id
008A 1A                     ldax    d               ;get field
CP/M RMAC ASSEM 1.1         #004    BDOS-PEEK RSX FOR CP/M 3.0

008B 13                     inx     d               ;advance fcb pointer
008C CD5601                 call    hexsp           ;print in hex, space
008F 05                     dcr     b               ;all 4
0090 C28700                 jnz     idloop
                   ;
                   ; print the nr field
                   ;
0093 CD4B01                 call    prathl          ;last id string
0096 211000                 lxi     h,16            ;offset to nr field
0099 19                     dad     d
009A 7E                     mov     a,m             ;fetch it
009B CD5E01                 call    hexout          ;print it
009E D1                     pop     d               ;restore fcb pointer
                   ;
                   ; here after fcb info printed.  Allow the call to pass upward,
                   ; then print the returned value.
                   ;
009F CD0900        exit:    call    next            ;give call to bdos
00A2 22F501                 shld    hlsave          ;save returned registers
00A5 D5                     push    d
00A6 C5                     push    b
00A7 F5                     push    psw             ;save returned value
00A8 21EB01                 lxi     h,rtnmsg        ;print "Ret: "
00AB CD4B01                 call    prathl
00AE F1                     pop     psw             ;restore, resave returned a
00AF F5                     push    psw
00B0 CD5E01                 call    hexout          ;print it in hex
00B3 3AF201                 lda     pause           ;do a pause?
00B6 B7                     ora     a
00B7 CAD100                 jz      nopaus          ;jump if not
00BA CD4501                 call    ilprnt          ;yep, print prompt
00BD 20AA                   db      ' ','*'+80h
00BF 0E06                   mvi     c,dirIof        ;get a character
00C1 1EFD                   mvi     e,0FDH          ;don't come back 'till we have one
00C3 CD0900                 call    next
00C6 E65F                   ani     5fh             ;convert char to upper case
00C8 FE51                   cpi     'Q'             ;quit-pause?
00CA C2D100                 jnz     nopaus          ;no
00CD AF                     xra     a               ;yes
00CE 32F201                 sta     pause
00D1 F1            nopaus:  pop     psw             ;restore 'em all
00D2 C1                     pop     b
00D3 D1                     pop     d
00D4 2AF701                 lhld    spsave          ;restore stackpointer
00D7 F9                     sphl
00D8 2AF501                 lhld    hlsave
00DB C9                     ret                     ;back to caller
                   ;
                   ; Handler for function 60: call RSX. Here we insure that the
                   ; call is for us.  The RSX parameter block passed in DE must
                   ; contain two parameters, the first of which is a pointer to
                   ; our name (near absolute insurance that there won't be any
                   ; confusion).  The second parameter is a pointer to a request
                   ; code (one of: "Q", quit, "C", console, "L", list, "A", aux).
                   ;
00DC D5            rsxfnc:  push    d               ;save user de
```

USE THE RSX

Continued from page 83

needed by the program loader as a backward link for use in adding and removing RSXs.

Remove (byte, initialized). This field is the warm start removal boolean discussed earlier. If set to 0, the RSX will remain in memory after the next warm-start. If set to 0FFH, the next warm start will remove the RSX. The RSX may set this flag at any time to cause its own removal.

Nonbank (byte, initialized). This is another boolean field. If set to 0FFH, the loader will not load the RSX in a banked CP/M+ system. If set to 0, the RSX will be loaded in both banked and nonbanked systems.

The RSX is a powerful tool for extending the operating system.

Names (8 bytes, initialized). This is the name of the RSX; it is used only by GENCOM, and is otherwise available to the user. I use it in the example RSX to identify function-60 calls.

Loader (byte, not initialized). This boolean identifies the program loader, which has the field set to 0FFH. The loader zeroes this field in all RSXs that it loads. This field is used to identify the last RSX in memory, which is always the loader.

Reserved (2 bytes, not initialized). The final field is reserved for future versions of CP/M.

PEEK operating instructions

A few final notes are in order regarding the use of the PEEK, the example RSX.

Peek understands a simple command line syntax of the form

PEEK <dev-name><pause-token>

where < dev-name > is the device to be used by PEEK to display its output, and < pause-token > is the character "P"; if active, pause mode will cause the system to wait after each system call is traced for a keystroke from the console. < dev-name > is any of "A", "C", or "L", for

```
CP/M RMAC ASSEM 1.1     #005    BDOS-PEEK RSX FOR CP/M 3.0

00DD EB                 xchg                    ;rsx pb pointer to hl
00DE 23                 inx     h               ;ignore function code
00DF 7E                 mov     a,m             ;get # parameters
00E0 FE02               cpi     2               ;two?
00E2 C23A01             jnz     pass            ;quit if not
00E5 23                 inx     h               ;skip # of parameters
00E6 5E                 mov     e,m             ;fetch parameter #1
00E7 23                 inx     h
00E8 56                 mov     d,m
00E9 23                 inx     h               ;hl points to 2nd parm
00EA 22F701             shld    spsave          ;save that pointer
00ED 211000             lxi     h,myname        ;check for our name
00F0 0608               mvi     b,8             ;8 characters
00F2 1A         ckname: ldax    d               ;compare first/next character
00F3 13                 inx     d
00F4 BE                 cmp     m               ;compare with our name
00F5 23                 inx     h
00F6 C23A01             jnz     pass            ;if fails, pass to next rsx
00F9 05                 dcr     b               ;all 8
00FA C2F200             jnz     ckname
                ;
                ; rsx request is for us, get rsx pointer, load 2nd parameter
                ;
00FD D1                 pop     d               ;recall passed parameter
                ;
                ; call next rsx's -- we may be an accidently loaded rsx (which
                ; will happen when the tpa module is used only for communication
                ; with us).  PEEK always returns 0, CP/M returns 0FFH.  So...
                ; unless we get an ff back, there are more of us in memory, and we
                ; are only a clone, and must commit suicide (does this sound
                ; like science fiction to you too??)
                ;
00FE CD0900             call    next            ;are we real?
0101 FEFF               cpi     0ffh            ;a=0ffh?
0103 CA0C01             jz      wereal          ;jump if so
0106 210E00             lxi     h,rmvflg        ;sad...
0109 36FF               mvi     m,0ffh
010B C9                 ret
                ;
010C 2AF701     wereal: lhld    spsave          ;get rsx pointer
010F 7E                 mov     a,m             ;fetch next parameter
0110 23                 inx     h
0111 66                 mov     h,m             ;into hl
0112 6F                 mov     l,a             ;hl points to request
0113 23                 inx     h               ;point to pause flag
0114 7E                 mov     a,m             ;fetch it
0115 FEFF               cpi     0ffh            ;null? (don't alter pauseflag)
0117 CA1D01             jz      pntreq          ;jump if so
011A 32F201             sta     pause           ;no, stuff it
011D 2B         pntreq: dcx     h               ;point to request
011E 7E                 mov     a,m             ;fetch request
                ;
                ; a now has the function request character passed from the tpa
                ;
011F FE51               cpi     'Q'             ;quit?
0121 CA3E01             jz      quit
CP/M RMAC ASSEM 1.1     #006    BDOS-PEEK RSX FOR CP/M 3.0

0124 FE41               cpi     'A'             ;output to aux?
0126 0E04               mvi     c,auxotf        ;prep function code
0128 CA3401             jz      setout
012B FE43               cpi     'C'             ;output to console?
012D 0E02               mvi     c,conotf        ;prep code
012F CA3401             jz      setout
0132 0E05               mvi     c,lstotf        ;process of elimination (list)
0134 79         setout: mov     a,c             ;get new output device code
0135 32F101             sta     curdev          ;stuff it
0138 AF                 xra     a               ;return a=0
0139 C9                 ret
                ;
                ; here when call 60 is not for us
                ;
013A D1         pass:   pop     d               ;clear stack
013B C30900             jmp     next            ;to next rsx
                ;
                ; here to quit
                ;
013E 3EFF       quit:   mvi     a,0ffh          ;flag loader to remove us
0140 320E00             sta     rmvflg
0143 AF                 xra     a               ;we still return 0
0144 C9                 ret
                ;
                ;       ***************
                ;       * subroutines *
                ;       ***************
                ;
                ; print in-line message terminated by char w/bit7=1
                ;
0145 E3         ilprnt: xthl                    ;msg pointer to hl
0146 CD4B01             call    prathl          ;print it
0149 E3                 xthl
014A C9                 ret
                ;
                ; print string pointed to by hl, until char w/bit7=1
                ;
014B 7E         prathl: mov     a,m             ;fetch first/next char
```

```
014C 23                 inx     h               ;advance pointer
014D CD6F01             call    type            ;print it
0150 E680               ani     80h             ;terminator?
0152 CA4B01             jz      prathl          ;continue if not
0155 C9                 ret
                ;
                ; output a in hex, followed by space
                ;
0156 CD5E01     hexsp:  call    hexout          ;print it
0159 3E20               mvi     a,' '
015B C36F01             jmp     type
                ;
                ; output a as 2 hex digits
                ;
015E F5         hexout: push    psw             ;save it
015F 0F                 rrc                     ;hi nybble into lo
0160 0F                 rrc
0161 0F                 rrc
CP/M RMAC ASSEM 1.1     #007    BDOS-PEEK RSX FOR CP/M 3.0

0162 0F                 rrc
0163 CD6701             call    nybble          ;print it
0166 F1                 pop     psw             ;now restore, fall into lo
0167 E60F       nybble: ani     0fh             ;zap any garbage in hi nybble
0169 C690               adi     90h             ;this little beauty...
016B 27                 daa                     ;...is from the old (and maybe
016C CE40               aci     40h             ;...still existing) Intel library
016E 27                 daa                     ;fall into type
                ;
                ; type char in a on console via current output device.
                ;
016F F5         type:   push    psw             ;save all
0170 E5                 push    h
0171 D5                 push    d
0172 C5                 push    b
0173 E67F               ani     7fh             ;allow no parity bits to pass
0175 5F                 mov     e,a             ;align character
0176 3AF101             lda     curdev          ;get device function code
0179 4F                 mov     c,a
017A CD0900             call    next            ;output the character
017D C1                 pop     b               ;clean up
017E D1                 pop     d
017F E1                 pop     h
0180 F1                 pop     psw
0181 C9                 ret
                ;
                ; add a to hl
                ;
0182 85         addha:  add     l
0183 6F                 mov     l,a
0184 D0                 rnc
0185 24                 inr     h
0186 C9                 ret
                ;
                ; Table of locally-processed system function calls. Format  is function
                ; code  followed by id string. Note that more functions may be added to
                ; this table, but if they are not fcb-related functions, you'll have to
                ; modify the TRAP routine to process the new data types.
                ;
0187 0F6F706E66fnctbl: db       openf,'opnfi','1'+80H
018E 10636C7366         db      closf,'clsfi','1'+80H
0195 1472642073         db      readf,'rd se','q'+80H
019C 1577722073         db      writf,'wr se','q'+80H
01A3 1A73657464         db      sdmaf,'setdm','a'+80H
01AA 1173726368         db      srchf,'srch ','1'+80H
01B1 1273726368         db      srchn,'srch ','n'+80H
01B8 1772656E61         db      renmf,'renam','e'+80H
01BF 1365726173         db      erasf,'erase',' '+80H
01C6 2172642072         db      rdran,'rd ra','n'+80H
01CD 2277722072         db      wrran,'wr ra','n'+80H
01D4 1663726561         db      creat,'creat','e'+80H
                ;
01DB 00                 db      0
                ;
                ; table of fcb id strings
                ;
CP/M RMAC ASSEM 1.1     #008    BDOS-PEEK RSX FOR CP/M 3.0

01DC 6578BD     idbase: db      'ex','='+80h
01DF 7331BD             db      'sl','='+80h
01E2 7332BD             db      's2','='+80h
01E5 7263BD             db      'rc','='+80h
01E8 6E72BD             db      'nr','='+80h
                ;
01EB 205265743Artnmsg: db       ' Ret:',' '+80h
                ;
                ; data area
                ;
01F1 04         curdev: db      auxotf          ;system-call # for output
01F2 00         pause:  db      0               ;non-0 = pause after bdos calls
01F3 0000       tblptr: dw      0               ;saved pointer into fnctbl
01F5 0000       hlsave: dw      0               ;save for hl
01F7 0000       spsave: dw      0               ;stackpointer save
01F9                    ds      24              ;12-level stack
                stack:
                ;
                ;
0211                    end
```

output to the Auxiliary, Console or List devices, respectively. A special device name, "Q" (Quit) will terminate the RSX and remove it from memory.

Some examples of PEEK command lines are

```
PEEK AP
PEEK L
PEEK C
PEEK Q
```

The first example sends PEEK's output to the AUX device, with pause mode active. The second example switches PEEK's output to the list device while the third causes PEEK to output to the console. The final example terminates PEEK.

Pause mode, once enabled, will print the call detail information followed by an asterisk prompt; PEEK will then wait for a console key. Any key will continue, but remember that the Q key will deactivate the pause function.

The RSX acts as a wedge between transient programs and the OS.

Conclusions

The RSX is a powerful tool for extending the operating system, implementing debuggers, and creating novel and exciting utilities. I hope this tutorial will stimulate the use of RSXs in the development of new and unusual CP/M applications.

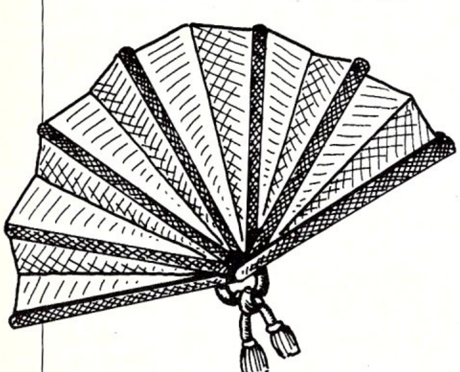
When multiple RSXs are resident, it is possible that an RSX may be deleted while another in lower memory is still active. In this case, the higher-memory RSX remains in memory, but is deactivated. The space occupied by this higher-memory RSX is not reclaimed until all RSXs in lower memory are removed.

For a more detailed discussion of page relocation, see *A Simple Technique for Static Relocation of Absolute Machine Code* by Gary Kildall, Dr. Dobbs Journal, February, 1978.

Ronald G. Fowler, 1113 Elsie St., Fort Atkinson, WI 53538.

Ways to Keep your Cool

Para Dynamics' special ventilation methods increase cooling efficiency on mainframes.



by Dave Hardy and Ken Jackson

Probably the most important part of any S-100 system is the S-100 frame itself. Regardless of its contents, it is the frame, with its power supply and motherboard, that is ultimately responsible for the overall functioning of the computer system.

Since their early beginnings in the mid-1970s, S-100 frames have evolved from simple boxes with power supplies and S-100 sockets, to complex mainframes with filtered-air cooling, terminated-bus motherboards, and sequenced power supplies.

The first successful attempt at a real S-100 frame was probably the old workhorse IMSAI 8080 frame, which is still in widespread use today. Its shortcomings included a simple power supply, a motherboard with an unterminated bus, and inadequate cooling. In addition, it had no provision for the most common computer peripheral, the floppy disk drive.

In the years that followed the introduction of the IMSAI frame (and others, like Altair from MITS), the S-100 frame saw many changes and additions, but most of the "new" frames were still basically the same as their predecessors. Most did away with the front panel, many had terminated-bus motherboards, and some even had constant-voltage power supplies. Unfortunately, few of the new frames had all of these features, and many other problems still remained.

Then, in the latter part of the '70s, S-100 frame design began to shift from the lab/experimenter/hobbyist to the business world, where dependability and turnkey operation were the most important requirements. Designers began to include industrial-quality features in their machines, like Constant Voltage Transformers (CVTs) to withstand powerline surges and brownouts, and filtered air cooling to eliminate heat problems as well as keep the dust and dirt of the commercial workplace out of the S-100 environment.

The Para Dynamics 3000 series S-100 mainframes are an attempt to put all of the desirable characteristics, including those mentioned above, into one box. Para Dynamics' frames have been totally redesigned to fit the needs of the new S-100 bus, and also to fit the needs of the end-user. Their three most popular models are the 3020D, which is a 20-slot S-100 frame, the 3510D, with 10 slots and room for two $5^1/_4$" floppy drives, and the 3820S, which is probably the ultimate S-100 system frame.

Common characteristics

Although made for different applications, the Para Dynamics 3020D, 3510D, and 3820S have several features in common. The most obvious is their high-quality construction. Each frame is made primarily of heavy-gauge aluminum and sheet metal, with all internal parts securely mounted. Terminal blocks are used for most wiring connections, and all wiring is oversized to insure minimum voltage drop with high-current loads. Oversized heatsinks for the power supplies' power transistors are used throughout. In addition, the power supplies used are a great deal more than adequate, typically supplying 30 amps at 8 VDC over an input range of 95 to 130 VAC or 190 to 260 VAC (50/60 Hz).

For safety, all machines have three-wire AC supply cords and internal double-pole circuit breakers. In addition, each frame is equipped with a double-bitted AC-power keyswitch, and a reset switch with built-in "power on" indicator light.

The motherboard in each unit is a high-quality glass-epoxy PC board, completely populated with S-100 sockets. Terminal blocks for connection to ground. +8 VDC and ±16 VDC are mounted directly on the PC board. Active termination is provided for the S-100 lines. The active bus terminators can optionally be terminated to ground, if desired. An option is also provided to use pin 53 as a signal line (IEE-696 says it should be a GROUND line) in case your CPU is an older one that requires the old SSWDSB* signal to enable its data input buffers, as was the case with most of the old front panels.

One of the most interesting features of the Para Dynamics products is their unique air funnel cooling scheme. The S-100 compartment in each frame is closed off from the rest of the machine, and a large air funnel is fitted over one entire side of the motherboard. At the other (narrow) end of the funnel, an exhaust fan is used to draw air into the front of the compartment, across and between the S-100 boards, and through the funnel. The advantage of this method is that the funnel shape can actually control the airflow in different parts of the S-100 compartment, and it is shaped in such a way that every board in the frame receives the same amount of airflow. This feature provides the uniform cooling that has previously been lacking in other S-100 frames. As demonstrated in the test results below, this air funnel scheme is responsible for the dramatic increase in cooling efficiency in the Para Dynamics products. In all of our tests

In all our tests, we were unable to raise the temperature in the S-100 compartment by more than 4°.

(and somewhat to the amazement of the technicians performing them), we were unable to raise the temperature at any point in the S-100 compartment by more than four degrees above the ambient (room) temperature.

Specific descriptions

The 3020D. The 3020D is a desktop (also available as a rackmount) 20-slot S-100 mainframe. It has all of the features mentioned above, plus mounting slots for 16 DB-25 connectors, one DB-37 connector, and four RF (BNC) connectors cut into the back panel. A clamped slot is also provided on the back panel to accommodate ribbon cables. The motherboard is mounted so that S-100 cards are inserted vertically and are facing the right side of the machine (unlike the old IMSAI, which placed the cards facing the front of the machine). The S-100 compartment occupies the entire front half of the frame, and the back half contains the power supply, air funnel, and cooling fan. Nearly one-third of the back half of the machine is occupied by the large Constant Voltage Transformer.

Although the power supply is not directly in the airflow of the cooling system, it does not become excessively warm during normal operation because of the convection cooling provided for the entire back half of the machine, and also because part of the air drawn into the S-100 compartment is taken from the power supply compartment through a set of double walls along the sides of the S-100 compartment.

The rackmounted version has a sheet metal cover that attaches with four screws and provides an air inlet for the cooling system and convection cooling slots for the power supply. The desktop version adds a heavy-gauge aluminum cover, similar to the old IMSAI cover, but without air slots, since there is no airflow required along the sides of the machine.

Like the old IMSAI, this frame expects that any peripheral devices (i.e., disk drives, etc.) will be powered from an external source.

Power supply ratings common to all machines (all DC voltages are fused, all ratings are for continuous load):

+16 VDC at 5 amps
−16 VDC at 2 amps
+8 VDC at 30 amps
−8 VDC at 2 amps

The 3510D. The 3510D, like the 3020D, is available in a rackmount or desktop configuration. It is similar to the 3020D in layout, except that half of the S-100 compartment is partitioned off to hold two 5¼" floppy disk drives (or the equivalent), and the motherboard has only 10 slots instead of the usual 20. To support the floppy disk drives, a second power regulator board has been added that supplies them with additional voltages: +5 VDC at 7 amps, and +12 VDC at 3 amps. Because the floppy disk drives sit in the S-100 compartment, they are well ventilated by the air funnel system.

The back panel contains cutouts for 16 DB-25 connectors, one DB-37 connector, four RF (BNC) connectors, and a clamp for ribbon cables.

The 3820S. The 3820S is the full-blown deluxe stand-alone "Pronto" mainframe. The machine itself is about 25" high, 19" wide, 23" deep, and is mounted on casters (it weighs about 80 pounds). The 3820S contains a full 20-slot S-100 mainframe, with room and power supplies for up to two 8" floppy drives and one 8" hard disk.

The S-100 frame is contained in the bottom half of the 3820S, and is mounted on slides so that it can be pulled forward out of the frame for access to S-100 cards. Inside the 3820S, behind the S-100 frame, are the two large Constant Voltage Transformers and the power supplies for the S-100 frame and the drives.

Wiring is all done between terminal blocks, so wiring in a hard-disk should be fairly simple. Wiring for two 8" flop-

Oversized heat-sinks for power transistors are used throughout.

PARA DYNAMICS

Continued from page 87

py drives is already installed.

Note that two entirely separate power supplies are used for the frame and drives. This arrangement allows a power sequencer to be used that will allow the system start up to power up the frame *first*, then the drives, and power down the drives before the frame. This feature ensures that the computer will be properly operating when the drives are turned on or off, so that no erroneous writes will take place. In other words, you won't have to worry about the computer accidentally blowing away data on the drives when you turn the machine on or off.

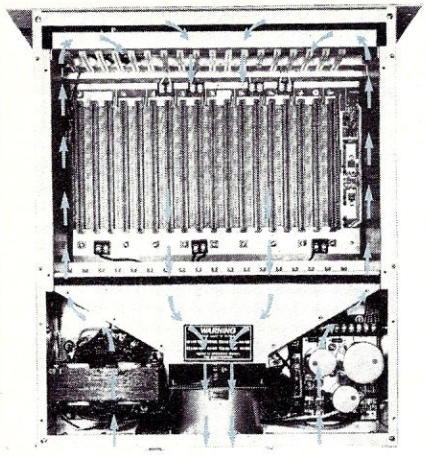
The 3820S also provides a washable air-filter in the front of the machine, something not offered in the 3020D or 3510D frames. In an industrial environment, this filter is invaluable.

Because of the additional heat generated by the 8″ drives and the hard disk, the 3820S has a second fan that provides cooling for the top half of the machine, where the drives are mounted. No filter is provided for this fan because of the many openings in the top part of the frame for connectors, drives, etc.

Also wired into this machine is a separately fused display AC outlet, rated at 15 amps, that can be used to drive peripheral devices such as terminals, printers, etc. Although these outlets are wired past the dual 30 amp system circuit breakers, they are not switched on and off by the keyswitch mounted on the front of the machine.

The back of the 3820S contains cutouts for 20 DB-25 connectors and two DB-37 connectors. Also mounted on the back of the 3820S are the large heatsinks for the drive power supplies.

Unlike the two desktop frames, this machine is "opened up" for access by removing panels. The front panel is removed to provide access to the S-100 frame, which slides directly out about 10 inches. The top panel removes to give access to the drives and power supply fuses. The side panels remove to provide access to the power transformers, the lower wiring, and to installl or remove the drives. With all of the panels removed, the machine is completely accessible for maintenance, wiring, repairs, etc. Accessibility is a feature often overlooked by other manufacturers.



The funnel design ensures an unusually even distribution of air over the board faces for maximum cooling.

The 3820S provides the following voltages:

+24 VDC at 8 amps
+16 VDC at 5 amps
−16 VDC at 2 amps
+8 VDC at 30 amps
−8 VDC at 2 amps
+5 VDC at 7 amps
−5 VDC at 3 amps
+12 VDC at 3 amps
−12 VDC at 3 amps

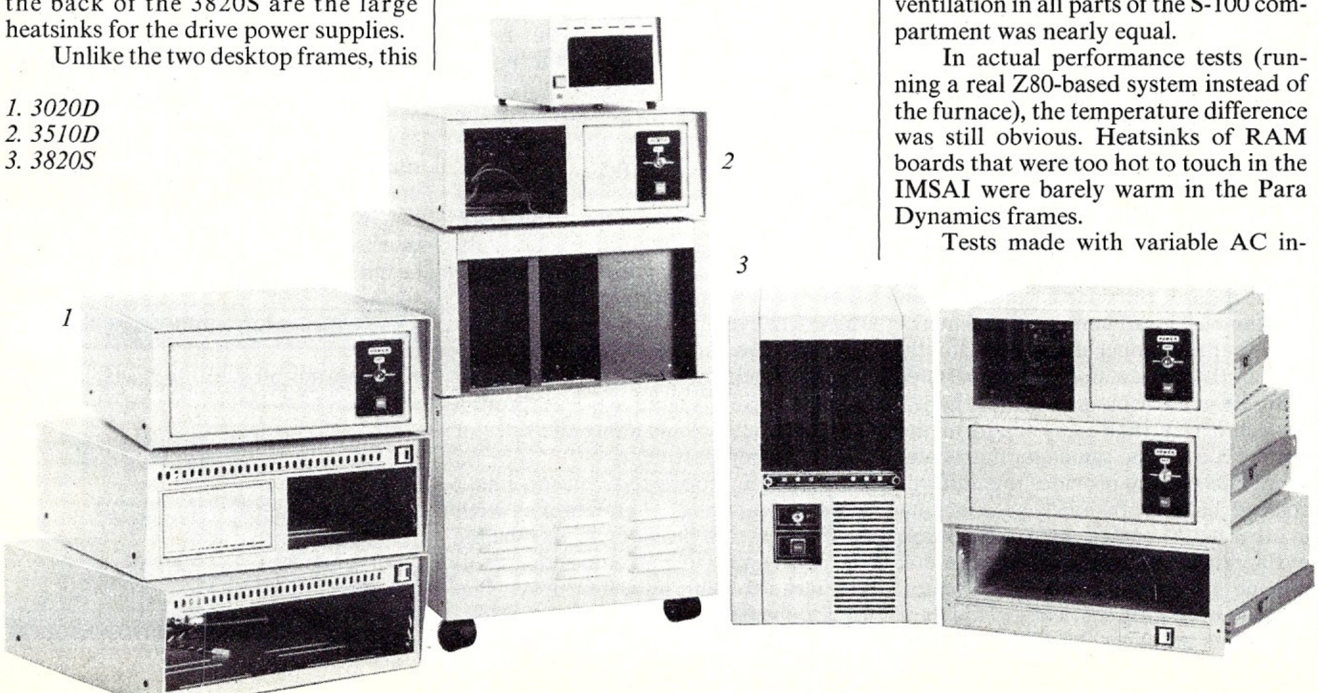
Test results

Our tests showed that the power supplies were able to outperform their listed specifications by about 10 to 20%. In all cases, the machines were able to perform well at 100% of their rated load capacity with no noticeable degradation over a two-hour period of time. We didn't bother to make any absolute maximum tests, mostly because manufacturers take a dim view of people blowing up their evaluation products. It is safe to say that most end-users will not require more power than these machines can deliver, unless the power supplies are used to power large external devices.

Cooling and airflow measurements provided the most interesting results of all of our tests. Relative cooling tests were made using an IMSAI 8080 mainframe and a 3820S, both equipped with identical multiple and extremely inefficient S-100 boards (actually eight boards, each with identical power resistors used to produce a great deal of heat). Although both machines exhausted about 80 cubic feet of air per minute from their S-100 compartments, the temperature difference (internal temperature minus room temperature) in the IMSAI was about 16 degrees, while the temperature difference in the 3820S was about 1½ degrees! This is by far the most impressive feature of the Para Dynamics frames. Test results for the 3020D and 3510D were similar, except that we could fit only 10 boards into the 3510D. Temperature tests made at the center and each end of the S-100 compartment also confirmed that ventilation in all parts of the S-100 compartment was nearly equal.

In actual performance tests (running a real Z80-based system instead of the furnace), the temperature difference was still obvious. Heatsinks of RAM boards that were too hot to touch in the IMSAI were barely warm in the Para Dynamics frames.

Tests made with variable AC in-



1. 3020D
2. 3510D
3. 3820S

puts revealed that acceptable output levels were available from below 80 VAC to 135 VAC. The frames would have probably operated acceptably over a wider range of voltages; however, these were supposed to be nondestructive tests, and any wider voltage ranges could possibly cause damage to the power supplies or the cooling fans.

Documentation

Documentation provided with each machine consists of a single User's Manual. Each manual is brief, but contains the necessary information to set up, test, and service its frame. An introduction, a general description, warranty information, and initial procedures are covered in Section I of each manual. Section II contains a parts list, wire lists, illustrations, and schematics. Section III describes initial startup; Section IV, maintenance and servicing.

Conclusions

Only three small complaints can be made about these machines: First, they are *heavy.* The Pronto weighs 80 pounds (which is why it's on casters). There is really nothing that can be done about this problem, since most of the weight is due to the quality of the machine. Unless you have a pet gorilla, unpacking this device could become a full-time job.

Two separate power supplies for the frame and drives ensure that you won't accidently blow away data on powering up.

Second, the sheet metal used (like all cut sheet metal) has sharp edges that can do a good job of cutting up fingers. This could be remedied by the manufacturer with little trouble, or by the end-user with some sandcloth or a file.

Finally, in order to have such efficient cooling, the machines all use high-volume muffin fans that are a bit too loud for use in a home or close-quartered business environment. If you are used to having the older computing machines at your site, then the Para Dynamics machines will cause no noise problems. But if you are used to having a Commodore or an Apple, you will have to get used to the sound of the fan. However, these machines are no louder than any other S-100 frame with similar capabilities.

The Para Dynamics frames are the best we've ever seen for S-100 products. They are fully professional inside and out, and are conservatively rated by their manufacturer. They are well designed, with thought given by their designers to both performance and safety. If you are assembling an S-100 system, Para Dynamics frames are definitely worth considering.

Para Dynamics Corporation
7895 East Acoma Drive
Scottsdale, AZ 85260
(602) 991-1600

CIRCLE 300 ON READER SERVICE CARD

Dave Hardy, 736 Notre Dame, Grosse Pointe, MI 48230

CP/M Disk Parameter Tables

The DPB and DPH determine how most information is stored on disk. Use them to optimize the configuration of your disk drives.

by Richard Rodman

The popularity of CP/M version 2.2 is partly due to the great wealth of applications software available to run under it. Yet this bank of software would not exist if there were not so many machines using CP/M. Digital Research's CP/M won out over its competitors in the early days—FAMOS, OASIS and others—because it was relatively easy to configure and transport to other hardware. While the other vendors dealt only with OEMs, insisting on customizing their systems themselves and guarding source code closely, Digital Research documented procedures for first-time installation and included abundant information and sample source listings.

CP/M separates the logical portions of the operating system, common to all implementations, from the physical input/output drivers. The standard portion consists of the Console Command Processor (CCP) and the Basic Disk Operating System (BDOS); the user-written, hardware-dependent module is the Basic Input/Output System (BIOS). Because of this modularity, CP/M can be easily configured for a multitude of I/O devices.

CP/M versions 2.0 and up are also user-configurable to disk drives of almost limitless variety. Many parameters involved in the arrangement ("format") of information on the disk must be specified and optimized by the user. These parameters are stored in two tables within the BIOS called the *Disk Parameter Header* (DPH) and the *Disk Parameter Block* (DPB).

These two tables almost completely specify *how* the information is stored on a disk (either floppy or hard). Setting up the DPH and DPB for a disk format involves many sensitive decisions about the disk subsystem, and can make great differences in storage efficiency and transfer speed. This article will describe the meaning of each parameter, and explain its effect upon the resulting system. If you already know some of what is presented here, please bear with me. A great many of the concepts involved may not be readily apparent to some readers, and so it is healthy to explain them. The core can be reached only by eating the apple around it.

The CP/M file/block system

The CP/M operating system uses a named file system that is accessible from user programs. Disk space is allocated randomly and noncontiguously as needed for each file, thus optimizing storage. This means that a 32K file need not be placed in a 32K free section of the disk, but can be broken up and scattered about; yet this scattering is transparent to the user's program. Thus every portion of the disk can be used, and there is no disk space fragmentation problem.

Disk drives are similar to phonograph records in that the magnetic read head, which corresponds to the needle, can be moved at a moment's notice to any portion of the disk. Computer disk drives segment the phonograph arm's free swing into discrete increments called "tracks." On an 8″ floppy disk, there are 48 tracks per inch.

Note that now, as the disk rotates under the head, the head covers a ring or annulus on the disk's surface. This area is also referred to as a "track." This track is divided up for convenience into units called "sectors." Each sector contains a number of bytes.

Thus we have the following levels of granularity involved in any computer disk format: bytes, sectors, and tracks. Because the numbers of these vary from disk to disk, however, CP/M breaks up the total disk storage into logical units called "blocks." The block size can be decided upon by the user, but this size (in kilobytes) must be a power of two (1K, 2K, 4K, etc.).

Each file in the directory uses some number of blocks. The directory entry for each file contains a description for each block allocated to the file. For example, if a file is 5K in size and the block size is 1K, then five blocks will be allocated. If the block size is 2K, then three blocks will be required, thus using 6K of disk space. Notice that we have wasted 1K of disk space by using the larger block size. In general, for a disk system containing N files with a block size of B, $(N \times B)/2$ bytes will be wasted.

We can see that it is to our advantage to use as small a block size as we can. On the other hand, CP/M has to keep track of which blocks on the disk have been used by the files on the disk, so that it can know where to put new information. This is done by keeping a table in memory with one bit per block, the high-order bit of the first byte of the table standing for the first block on the disk. This table is known as the *allocation vector.* At one bit per block, the allocation vector will require a byte for every eight blocks on the disk.

Let us suppose we have an 8 MB hard disk drive. If we use the smallest block size, 1K, then we will have to set aside $(8\ \text{MB}/1\text{K})/8 = 8192/8 = 1024$ bytes of memory for the allocation vector. While that may not seem like a lot, what happens when we have a 96 MB drive divided into twelve 8 MB logical drives? Can CP/M use the same allocation vector for each drive? No, because more than one may be accessed at a time. Now you're talking about 12K of memory just for the allocation vectors.

Setting up the DPH and DPB involves some sensitive decisions.

Clearly, a trade-off in the block size is called for. This is the first decision to be made in the definition of a disk format, as it will influence the disk storage available to the user. It will also influence the speed of the resulting system somewhat, because using a smaller number of larger blocks reduces the amount of work CP/M needs to do.

The next consideration related to block size is the directory. Unlike some operating systems, which allow the directory to be scattered about on the disk much like the files, CP/M requires that the directory be permanently allocated space right at the beginning of the user disk area. Each file on the disk has an entry in the directory. Each of these entries is 32 bytes in length.

The first half of a file's directory entry contains the user area, name and record count information of the file. The second half, 16 bytes, is used to hold the block numbers which this file "owns." Important point: If there are more than 255 blocks on a disk, it will take a two-byte number to express the number of a block. (One-byte numbers can hold a number from 0 to 255; two-byte numbers from 0 to 65,535.) Hence, if there are 255 or fewer blocks on a disk, each directory entry can point to 16 blocks. If there are more than 255 blocks, a directory entry has to use two-byte numbers, and can point to only eight blocks. If a file needs more than that many blocks, it will need more than one directory entry.

Now, to make the entire disk space usable, we have to be able to point to each block on the disk at least once, so we should have enough directory entries to contain one block number for each block. If we have 255 or fewer blocks, we can divide the number of blocks by 16; otherwise, we have to divide by 8. This gives the absolute minimum number of directory entries we need. Remembering the waste principle described earlier, we should double this number.

Because CP/M allocates disk space in blocks, it would be pointless to have a directory size that is anything less than an exact number of blocks. Since there are 32 bytes per directory entry, we can multiply our tentative directory maximum by 32, then divide by our block size. Round the resulting number up to an integer. This is the number of directory blocks. Is it more than 16? You will have to increase your block size. A directory that big will make for a slow system anyway.

Once you know the number of directory blocks, you can then multiply by the block size and divide by 32 to get your new directory size.

Remember that the allocation vector is stored in memory. How does it get there? When a disk is first accessed, CP/M marks the directory blocks as allocated, then it goes through the whole directory, reading each block number and setting the appropriate bit. All the drives are reset on each "warm boot," so this extensive task is redone over and over. Although this makes the system a little slower, it greatly increases our chances that the vector is correct. CP/M's great reliability is due to its simplicity and thoroughness.

There are two cases which can come up that call the accuracy of this scheme into question. The first is: suppose two files "own" the same data block. Then, suppose we erase one of them. When we erase the file, CP/M goes through its directory entry and "de-allocates" (resets the bits in the allocation vector) the blocks assigned to that file. Now, one of the blocks that is still owned by the second file has been marked as free! How do we prevent this?

This situation can only occur if CP/M's file-handling procedure has been bypassed. Since a block owned by a file is not free, the system won't re-allocate it. If this situation arises, the validity of the data is in question as well. The next warm boot will, of course, correct the allocation vector.

The second question involves changing disks. Suppose that someone removes a floppy disk after the allocation vector has been constructed, and puts in another. The allocation vector is now incorrect, and when CP/M allocates a free block to a file being written on the disk, it will quite probably write over data belonging to another file. How can we prevent this?

This was a great problem with CP/M 1.4, and the solution Digital Research came up with was quite clever. Since the two disks have, by the condi-

DISK PARAMETER

Continued from page 91

tions of our question, different blocks assigned to the different files, the directory information for the two disks must be different. If we detect this difference, we know that the allocation information is incorrect, and we can prevent the data from being written to the disk.

Now, at last, the source of the famous "BDOS ERR ON B: R/O" error is known: A human changed the disk without telling the computer!

CP/M tells the directories apart by maintaining a checksum on each 128-byte logical sector of the directory or each disk in what is called a "check vector." The person setting up the system may make this vector smaller than the number of directory logical sectors if he desires, or even set it to zero to bypass the disk checking (on nonremovable hard disks, for example). Sectors on the disk may be of any size, as we described earlier. CP/M handles what are called "logical sectors" as 128-byte records. It is up to the BIOS to separate and arrange the physical sectors into 128-byte sectors. The CP/M system will usually read these logical sectors in sequential order. The configurer may find, through experimentation, that it is advantageous to arrange the logical sectors in such an order that the physical sectors will be just coming around the disk when the system is ready to read them. This is known as "sector skew." The sector skew can be put in a table, or calculated in an algorithm.

Lastly, we have to determine a track offset. Traditionally, CP/M has been loaded from the outermost (and lowest-numbered) tracks of the disk, which thus cannot be used by files. The user portion is inside these tracks, which are called "system tracks." Enough disk space is available in these tracks to contain the CP/M system; the directory will then begin with the first sector of the next track.

We have now described all of the parameters necessary for CP/M to understand a disk format. Now, we will show how the DPH and the DPB express these parameters to CP/M.

Incidentally, all of the preceding and following apply also to MP/M. There is one minor incompatibility between MP/M 2.x and CP/M, which will be discussed. CP/M 3.x will probably follow the MP/M 2.x convention.

Disk parameter header (DPH)

The DPH is a 16-byte table that must be defined for each logical disk drive on the system. These entries may *not* be shared by more than one disk drive. When the Select Disk (SELDSK) routine of the BIOS is ready to return, it must return the address of the DPH. The DPH specifies the addresses of buffers and the DPB associated with the selected logical disk drive. It consists of a *translate table address,* three *work words,* the *directory buffer address,* the drive's *DPB address,* the *check vector address,* and the *allocation vector address.*

Tables specify how information is stored.

The *translate table address* is provided as a convenience to the user. It is picked up and sent back to the user at the time the Sector Translate (SECTRN) routine of the BIOS is called. If the user desires, he may construct a table wherein the physical sector numbers are stored in order of their corresponding logical sector numbers. If he does so, SECTRN will simply add the logical sector to this address and read the value pointed to as the translated sector number, the physical sector to be read.

The system itself does no checking on the value stored as the translate table address. Further, it does no checking on the value returned from SECTRN, other than to immediately pass it to the Set Sector (SETSEC) routine. Some user diagnostics, such as DU, display the intermediate value. While it is not necessary to do so, it is preferable to preserve the meaning of the SECTRN routine by using it to perform logical-to-physical sector number mapping. The value of the translate table address may be used as a parameter to some calculation algorithm, if desired.

The *first work word* is used by the BDOS to point to the last-used directory entry. This value is filled in at the time the initial allocation vector is constructed. Having this pointer greatly increases the speed of the system when the directory is less than half full. In case the check vector ever indicates that the directory doesn't match, the disk is set to read-only and this word is set to the maximum directory entry, so that no files will be missed. The user should never do anything to this value. If it is damaged, the disk should be reset.

The *second and third work words* are used by the BDOS in its sector calculation routine. The CP/M 2.2 BDOS calculates a two-byte logical sector number from the block number (hence the 8 MB limit). This number must be divided by the number of logical sectors per track to obtain the logical track position. CP/M 2.2 uses successive subtraction to perform this operation, saving time for accesses that are sequential or nearly sequential (which most are) by deriving its result from the result of the previous calculation. These intermediate results are saved in the second and third work words.

The *directory buffer address* tells the BDOS where the directory buffer is. The user is required to have only one directory buffer, which is one logical sector (128 bytes) in size. The directory buffer is used for directory searches while accessing or looking for files, and for reading the directory during initial allocation vector construction. There is no reason for having more than one directory buffer.

The *DPB address* points to the DPB, not surprisingly. Note that while the work words of the DPH are changed by the BDOS, the DPB parameters are never changed. Thus, several logical drives may point to the same DPB.

The *check vector address* is the address of the drive's check vector. You will recall that this vector is a set of one-byte checksums, one per directory sector. The actual size of this buffer is specified in the DPB; the address is specified in the DPH, because each drive must have its own check vector. (It has its own directory, doesn't it?)

The *allocation vector address* is the address of the drive's allocation vector. As we described earlier, this vector consists of one bit for each block on the drive. The user may *not* create a vector that is smaller than necessary. If the user does not set aside enough space for one bit per block, CP/M will go ahead and overwrite whatever follows (crash!). Again, each drive must have its own allocation vector.

Disk parameter block (DPB)

The disk parameter block is the table that specifies the actual parameters of the disk drive. The DPB is a set of fixed values, 15 bytes in length. It consist of: *sectors per track* (word), *block shift factor* (byte), *block mask* (byte), *extent mask* (byte), *data storage maximum* (word), *directory maximum* (word), *directory allocation* (two bytes), *check vector size* (word), and *track offset* (word).

The value of *sectors per track* refers to the number of logical (128-byte) sectors stored on a logical track. Actually, the system keeps a two-byte (in MP/M

DISK PARAMETER

Continued from page 92

2.x, 3-byte) logical sector counter. The logical sector number is obtained by multiplying the desired block number by the number of logical sectors in a block, which depends on the block size. Then, any offset within the block is added. This number is divided by the sectors per track value, the quotient being the logical track number, and the remainder the logical sector on that track. The logical track then has the track offset added to it, and SETTRK is called. The logical sector number is passed to SECTRN along with the translate table address, and . . . you know the rest.

Although there are tracks both in the logical world and the physical world, remember that there is never any need of an exact correspondence. Track numbers are divided as a convenience for the user. A "memory disk," no matter what you call it, is simply a substitution of read/write memory for a disk drive. Even though it has no tracks, a value of sectors per track could be decided upon to simplify any bank selection necessary.

As a second example, it may be desirable on a multihead hard disk drive to consider a track as an annulus for each head, thus making track selection partially head selection and partially seeking. Alternatively, all the annuli together (a "cylinder") may be considered as a single track, and the head selection thus becomes part of sector translation or selection. As long as it is done in the same order, it makes no difference. CP/M does not do anything with track numbers or sector numbers other than send them to the BIOS.

The *block shift factor* is determined by the block size. The block size must be an integer power of 2 in kilobytes, as we said earlier. One kilobyte comprises 8, or 2 raised to the third power, logical sectors. The first block number is zero. If a block number, then is shifted up in a binary fashion by the integer power plus 3, the result will be the logical sector number at which this block begins. This integer power plus 3 is the block shift factor, and CP/M uses it in exactly this way.

The block shift factor is also used in another way. If we want to access the 1,223rd logical sector in a file, for example, shifting the number down by the block shift factor gives the block of this file in which CP/M will find this sector.

Where within that block will we find the particular sector, though? The answer to this question is found by the logical AND of the logical sector number (1223) with the *block mask.* The block mask is a number which is one less

A DPH is defined for each drive.

than the number of sectors in one block. Since the number of sectors in block is always a power of two, this number is used as a mask value for determining the sector number within a block.

The *extent mask* results from a combination of many factors, some of them historical. Under CP/M 1.x, the only block size allowed was 1K, and the only disk storage devices accessible through the system had storage of less than 256 blocks. Thus, 16 block numbers could be pointed to by each directory entry. Each directory entry, for files larger than 16K and thus needing more than one, was called an extent.

Further, because 16 blocks were thus allocated per extent, and these 16 blocks each addressed 8 sectors, each extent then addressed 128 logical sectors. A logical sector count was maintained in the file control block (FCB) at FCB+15 (called "cr"). A current extent number was maintained at FCB+12 (called "ex").

Under CP/M 2.x, with larger block sizes and larger disk drives, it became possible for each extent to address more than 128 logical sectors. Owing to the fact that CP/M 1.x stored a value of 128 in the "cr" byte on full extents, only seven bits of that byte could be used for the record count. The overflow had to go somewhere. Digital Research hit upon the ingenious idea of redefining the "ex" byte as a record count overflow. Because the number of logical sectors addressed by each extent is always some power of 2 times 128, the "ex" byte, when ANDed with the appropriate mask, would yield the rest of the record count within the extent; the part masked out would increment on each extent. You guessed it: this mask is called the extent mask.

Those of you who survived the previous paragraph probably realize that calculating the extent mask is not simple. Digital Research got around this obstacle by presenting values as shown in Table 1.

While it might be possible to construct a formula for EXM based upon block size and total number of blocks per drive, it is a lot easier to just use the table. At least now you know why!

The value for *data storage maximum* is the block number of the last block in the user space on the drive. It is simply the number of blocks on the disk minus one (they are numbered starting at zero). Remember that this number of blocks must actually fit on the disk. It can be calculated as: (total number of tracks on drive) times (total number of sectors on track) times (size of each sector) divided by (block size), truncated to an integer, minus one. The value of DSM includes the directory blocks, so the user space shown by STAT will differ by the size of the directory.

The *directory maximum* is the number of the last directory entry available, counting from zero. It is thus the number of directory entries desired minus one.

The *directory allocation* is provided to speed up the initial allocation vector construction. One bit per block, remember—obviously, the blocks for the directory must always be allocated. Therefore, before the process of reading the directory is begun, (DSM+1)/8 bytes are cleared in the allocation vector, and these two bytes are stored in there. The directory allocation is *not* a word value. It is stored high-byte first, not in 8080 fashion. Block zero, where the directory starts, gets the high-order bit of the first byte, the other blocks following. If your directory requires two blocks, you would store 11000. . . binary or C0 00 hex in these two bytes. If your directory needs 16 blocks, it would have 11111111 11111111 binary or FF FF hex. If your directory needs more than 16 blocks, now you know why you have to reduce it!

The *check vector size,* as described

Table 1. Values for calculating the extent mask

Block size	BSH	BLM	EXM for <256 blocks	EXM for > 255 blocks
1K	3	7	0	0
2K	4	15	1	0
4K	5	31	3	1
8K	6	63	7	3
16K	7	127	15	7

earlier, is the size of the directory checksum value array. It is usually set to the number of logical sectors in the directory, a lesser value if you don't mind taking chances. You may set it to zero if the disk is nonremovable. Under MP/M 2.x, if the disk is nonremovable, it is also necessary to set the high-order bit of this byte, and store 8000 hex as the value for check vector size. (This prevents most of the "Disk Reset Denied" messages). Unfortunately, if you try to use 8000 hex as the check vector size under CP/M, it thinks you have 32,768 bytes of memory dedicated for use as a directory checksum vector, and uses all

A DPB defines how data are stored on the disk.

it needs. Hopefully, future versions of CP/M will correct this incompatibility. In my personal opinion, having a check vector size greater than 255 is unlikely to be of any benefit.

Last, but not least, is the *track offset.* As we mentioned earlier, this value is added to the logical track to give the user a track number. The track offset is simply the number of system tracks. Because tracks are numbered starting at zero, the track offset also is the track that the directory starts on.

Conclusion

Hopefully, this discussion will clear up a lot of popular misconceptions about the DPH and the DPB, and increase the public bank of understanding about CP/M. The CP/M operating system, no matter what processor it runs on, will be with us for some time. It has achieved a good balance of small code size yet good disk space efficiency, reasonably fast yet reliable operation, plus simple, easy-to-use, yet powerful interface capability. Moreover, the CP/M software base is second to none. As CP/M shows up on more and more new machines, let us hope that system implementers set up their DPHs and DPBs properly.

Richard Rodman, 2011 Dexter Dr., Falls Church, VA 22043

MICROSYSTEMS REVIEWS

Display Manager from DRI

A set of callable routines for constructing data entry screens

by Alan Simpson

Most professional programmers wince a bit when they hear the term "I/O code." Not because writing code for data entry screens and reports is difficult, but rather because it is a boring and tedious task. Typically, the programmer designs displays on graph paper, then laboriously writes line after line of code to format displays on the screen and printer. The process becomes even more unpleasant if the program is to be used with many different terminals. The programmer then needs to take into consideration the control codes for various CRTs. This is a very time-consuming process, particularly if one plans on making one's software compatible with 50 different terminals. Most professional programmers get around some of the tedium by writing general-purpose I/O routines, and storing displays and terminal control codes on data files.

Digital Research has come up with an even better method. You, the programmer, buy Display Manager, then you "draw" your input and output displays directly on the screen exactly as you wish them to appear at runtime. Display Manager then takes care of writing the I/O functions, storing displays on a data file, and providing control codes for a variety of terminals. Sounded good to me, so I thought I'd give it a try.

Display Manager (DM-80) is one of Digital Research Incorporated's productivity tools, and works with any of their 8-bit programming languages (CB-80, Pascal/MT+, or PL/1), and 16-bit languages, including Pascal MT+, CB-86, PL/1, and C. Version 1.0 of DM-80, the one I used for this review, supports 55 different terminals and allows the programmer to include extra terminals. Display Manager also includes a program written in CB-80 that allows the end user to install the program to his particular terminal. DM-80 requires that you use CP/M, CP/NET, or MP/M and have at least 40K available in the Transient Program Area of your main memory. Display Manager will also run under PC-DOS with any of the DRI 16-bit programming languages.

Using Display Manager

When I first received DM-80, I read the manual cover to cover. Like most software manuals, the DM-80 manual tends to be more descriptive than tutorial. The first thought that came to mind after reading the manual was "What?" The manual is about 100 pages in length, with the usual addenda that tell you what the manual forgot to mention, as well as changes that have been made since the printing of the manual (yes, even though this is version 1.0). At first I was dubious as to whether or not this product was truly going to help increase my productivity. Since I've already developed a number of my own canned functions to handle I/O screens, it seemed unlikely that learning this new and seemingly complex tool would be worth the effort. When I actually sat down and used DM-80, I found it much easier to use than expected, and well worth learning.

Using DM-80 is essentially a three-step process: 1) Install DM-80 to your particular terminal; 2) Create and edit displays using the DM-80 editor; and 3) Write the application programs that access the displays. I'll discuss my experiences with each step in the process.

1. Installation. Installing DM-80 to your particular CRT is a simple process, unless you happen to be using 5¼" disks. DM-80 is delivered on two 8" disks, and one needs to do quite a bit of wading through the manual to determine exactly which files must be resident on disk during the various phases of designing screens. I managed to get DM-80 up and running on a single-sided double-density (180K) disk through a little trial and error. Once you have the correct files on disk, the rest is easy. DM-80 is menu driven and the program itself is somewhat tutorial.

If you are using one of the DM-80 supported terminals, installing the program is as simple as selecting that terminal from a menu of choices. Of the 55 terminals that DM-80 supports, many are different models from the same manufacturer. For purposes of brevity, I'll just list the manufacturers here:

A.B.M.	A.D.D.S.	Apple
Beehive	Control Data	Cromemco
Digital Equip. (DEC)	Direct	Hazeltine
Heath	Hewlett-Packard	I.S.C.
Lear-Siegler	Microterm	Osborne
Radio Shack (Tandy)	Soroc	Teleray
Televideo	Toshiba	Vector Graphic
Visual Technology	Xerox	Zenith

If you are not using one of the supported terminals, you will have to provide the control codes for a custom terminal. This is not difficult, provided that the custom terminal has enough documentation to supply the appropriate codes. The DM-80 manual has a simple questionnaire to fill out about custom terminal characteristics. Then, the install program asks the same questions that the questionnaire did, and you fill in the blanks. The installation program has a very convenient test capability that allows you to check to make sure you've installed a custom terminal properly. It does so by trying each function (clear screen, position cursor, reverse video, etc.) on the screen, and asking if the function worked correctly. If you discover a mistake during the test phase, you can edit the terminal codes using a reinstall option. Once you have DM-80 installed for you system, you can begin creating displays.

2. Creating displays. I never thought I'd see the day I'd actually enjoy creating I/O screens. DM-80 changed that by allowing me to draw and edit displays on the screen in an interactive, visual manner.

When you call up Display Manager's editor, it asks if you want to edit an existing display or create a new one. If you create a new one, it must have a unique number, as this number is used by the application program for finding the display. When you are ready to create your display, the editor presents a blank screen with the cursor in the upper left-hand corner, and you can just start drawing your display on the screen as you wish it to appear to the user at runtime.

The manual tends to make this process seem more difficult than it is. There are well over 40 distinct control-key commands (some three characters long!) that the editor uses. Personally, my brain's RAM space for storing control-key sequences is just about full, but DRI was quite considerate in making memorization a bit easier. For instance, many of the control-key sequences are identical to those used in other software packages (^V toggles insert mode, ^OC centers, while ^A, ^S, ^E, ^D, ^X, ^S, ^F move the cursor about on the screen, etc.). DRI also provides abbreviated reference cards, kindly laminated in clear plastic, for quick reference. When you first start designing displays, however, be prepared to do a good deal of wading through the manual. Control-key definitions are interspersed throughout the text, and the reference cards are too brief for first-time use.

When you are developing a display, you simply type out the prompts, headings, and borders where you want them to appear on the screen. You can also enter a control key command to specify that either an input or output field be

I never thought I'd see the day when I'd actually enjoy creating I/O screens.

displayed. You can easily move text and fields about the screen as you zero in on just the format you wish. You can also include template characters in input fields, such as "(___) ___-____" for phone numbers. Then, you can determine visual characteristics for the various fields in a simple and pleasant manner. To do so, simply position the cursor at the beginning of a field and enter a control key command to call up the status window. The table shown on page 98 then appears on the screen.

This window presents the DM-80 default characteristics for a single input field on the screen display. You can use the default characteristics for this field, or change them by simply moving the cursor about the status window. Of course, you can change default characteristics also.

The top line of the status window presents the field number (automatically assigned as the display is being designed), the row, column, and length of the field, whether or not it is surrounded by blank spaces (posts), and the type of field (INPUT or OUTPUT). The letters **rr** and **cc** are the row and column numbers of the cursor's present position on the screen, and **nn** is the number of fields in the display.

The Validate prompt allows the programmer to provide error checking with the simple press of a key. The options are X (accepts any printable character), A (accepts only alpha characters), C (all characters, including control characters), D (decimal numbers only), F (allows function keys only), I (interger only), and U (same as X, but input is changed to uppercase). Beep determines whether or not an illegal entry by the user causes the terminal's bell to ring (Y/N). Format for the fields can be L (left-justify), R (right-justify), N (numeric output), 0-9 (number of digits to the right of the decimal), C (send control keys to the screen), and M (money fields with leading dollar [or other currency] sign and two digits to

Display Manager

Continued from page 97

the right of the decimal).

The AutoRet option determines whether data entry terminates when the field's capacity is full (Y/N). The End-Input options allow the programmer to specify various methods for terminating data entry. If cursor is selected (Y), then the terminal's up-down arrow keys terminate data entry for the field. If BadC is Y, any illegal character for the field terminates entry for that field. F-key terminates entry if a function key is entered.

The remaining options Invs, Half, Invr, Flsh, Undl, Usr1, Usr2, Usr3 allow the programmer to specify visual attributes for a field. By filling in a Y above an option, the programmer can cause the field to be invisible, half intensity, inverse video, underlined, or flashing. The programmer can also define up to three user-defined visual attributes, and include these in various fields.

The whole procedure is simple and fast. You just draw the display as you want it to appear to the end user at runtime, then set the cursor to the beginning of each input and output field and use the status window to determine the basic characteristics and visual attributes of the individual field. Any programmer who has ever written I/O code to include as many options as Display Manager provides will probably see that this is a far quicker and easier method. I was certainly convinced.

As if this were not enough, Digital Research took it a step further and made Display Manager self-documenting. Once the display is created, the programmer can store a print-image ASCII file of the display on a disk file. This image file can be pulled directly into most word-processing systems to ease the development of a user's manual. Also, the disk file contains detailed documentation for each field in the display, which helps with the technical documentation as well as with debugging and modification. The final step in the process is to link your displays with your application program.

3. Write the application program. Once the displays have been designed, you need to write the actual programs that will use the displays. Display Manager adds the following functions to the programmer's present language:

INITDM:	Initializes the application program to use a specific terminal's control codes and capabilities.
OPENDIS:	Opens a DM-80 display file.
RETDM:	Returns visual attributes supported by a given CRT so the programmer knows which of DM-80's options are readily available for a particular CRT.
CLRSCR:	Clear-the-screen command.
CURS:	Set cursor to visible or invisible.
DISPD:	Places a display from the display file onto the screen.
CLSDIS:	Closes a display file.

For managing the actual fields in the display from the application program, DM-80 provides the following functions:

POSF:	Position the cursor to a given field.
NXTF:	Positions the cursor to the next field.
SETF:	Modifies the visual attributes of a field during runtime.
RETF:	Returns the field position, length, and type.
PUTF:	Outputs data to the current field.
GETF:	Accepts and validates data entered to a field by the end user.
UPDF:	Updates and validates data entered for a field.
ENDF:	Determines how user ends data entry.
RESF:	Resumes operation at last field.

```
Field No. Row  Col  Len  Posts  Type-INPUT
  000     000  000  000   YY    *rr, cc* nn

Validate  :X:  X,A,C,D,F,I,U  Beep    :N:  N,Y
Format    :L:  L,R,N,0-9,C,M  AutoRet:N:   N,Y

End Input---Cursor  :N:  BadC  :N:  FKey  :N:   N,Y

:N:  :N:  :N:  :N:  :N:  :N:  :N:  :N:  N.Y
Invs Half Invr Flsh Und1 Usr1 Usr2 Usr3
```

The syntax for using most of DM-80's functions (using CB-80 as an example) is:

```
integer variable=FUNCTION
  (integer expression)
```

Since DM-80 uses functions rather than commands, it is the programmer's responsibility to determine whether or not a function is successful, and to return an error message describing the problem to the user, should an error occur. This adds a bit of bulk to an application program, but then again, it does provide the programmer some flexibility in handling errors.

There are some minor annoying inconsistencies among the functions that the programmer must deal with. For example, some functions return a zero when the function is successful (Boolean false?), and negative value when the function is unsuccessful (Boolean true?). Some functions, like the CURS function, allow various numeric arguments (e.g., 0-3), but the value must be expressed as a string. Other functions do not use strings for numeric arguments. Until you get used to the exact syntax of the various functions, plan on doing a bit of debugging. You may find some of the syntax awkward and counterintuitive at first.

The final step is to write the program in the language of your choice and use the various DM-80 routines to access displays and manage field data. In your source program, you need to include DM-80's prewritten functions. For example, in CB-80, you need to include the command:

```
%INCLUDE DM80EXTR.BAS
```

Digital Research provides external functions for all the supported languages. Then when you link the compiled code, you need to include the DM-80 relocatable library as an overlay. In CB-80 the command to do so is:

```
LK80 TESTPROG,DM80CB80.IRL
```

Digital Research provides runtime libraries for each of the supported languages. The manual provides sample programs written in CB-80, Pascal/MT+, and PL/1 as useful examples of programming techniques.

Incidentally, once the source code is written and is capable of putting displays on the screen, the end user can use control keys or arrow keys to move the cursor about on the screen. DM-80 defaults to both the ANSI standard keys for moving the cursor about (^H, ^J, ^K, ^L) as well as the more popular ^S, ^X, ^E, ^D keys. The programmer need not write any code to provide these capabilities.

Similar products

To my knowledge, there are no products similar to Display Manager for compilable languages. Ashton-Tate's dBASE II, however, includes a program called ZIP that provides a capability similar to, but not as flexible as, DM-80. Both ZIP and dBASE II have one advantage over DM-80 and the DRI compilable languages: they're easier to use. With ZIP, the programmer draws the display on the screen, and follows prompts with the actual field or variable name that the prompt will be expecting. The programmer can also place commands on the screen that will later be embedded in the source code. ZIP then generates source code for the screen displays.

The programmer pays a heavy price for this ease of use however, and here is where Display Manager shows its true advantages. First, DM-80 can be used with high-performance native-code compilers, whereas ZIP can only be used with dBASE II, a slow-running interpretive language. For the independent software developer, DM-80 allows the user to write programs that will run on just about any 8- or 16-bit based system, and Digital Research does not charge royalties to the developer. ZIP and dBASE II narrow the market to customers who already own dBASE II, unless the developer is willing to pay some rather astronomical "royalty" fees ($70-$100 per copy!) to Ashton-Tate for a runtime package that allows non-dBASE II owners to use the package. Unfortunately, the dBASE II runtime package slows the applications programs down even further. Also, dBASE

DM-80 is a pleasure. I'd recommend it to anyone who is fluent in the DRI compiliable languages.

II is a very high-level database management system which, while providing powerful commands, robs the more sophisticated programmer of some lower-level flexibility, such as arrays, mathematical functions, and the ability to have more than two data files active at any time. Basically, if you're already an experienced programmer and you prefer a compilable language, DM-80 is your best bet. If not, perhaps ZIP and dBASE II are preferable.

Recommendation

I found DM-80 to be a very powerful and productive programming tool. It's also a pleasure to work with, though somewhat awkward at first. I'd recommend it highly to any professional programmer who is already fluent in any of the DRI compilable languages. I'd especially recommend DM-80 to anyone thinking about writing marketable software, as it will greatly reduce the labor inherent in making your programs compatible with a variety of terminals.

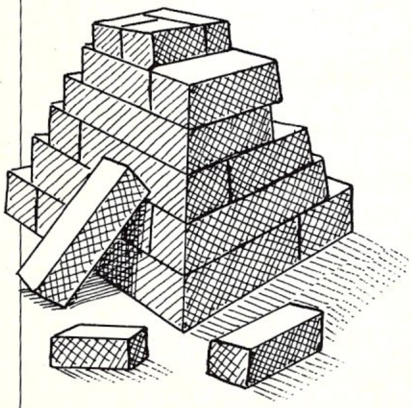
Display Manager is available from Digital Research, and costs $400 for the 8-bit version, $500 for the 16-bit version. You can call Digital Research at (800) 227-1617 ext. 400 outside of California, or (800) 772-3545 ext. 400 from within California for a dealer or distributor nearest you.

Alan Simpson, 5328 Bragg St., San Diego, CA 92122

CIRCLE 301 ON READER SERVICE CARD

Deblocking Routines Under CP/M 2.2

Using large disk sectors can speed program execution



by Richard I. Silverman

This article will explain the blocking/deblocking algorithm (BDA) as published in Digital Research's *Alteration Guide* (appendix G). It is intended for any CP/M user curious about the mechanics of the BDA, as well as for home hackers who want to implement the BDA, but who hesitate in the belief that they can't fix what they don't understand. Briefly, installation of the BDA allows CP/M to work with large disk sector sizes that are multiples of 128 bytes. Larger sector sizes can speed program execution and increase the amount of on-line disk storage space.

When Digital Research (DR) wrote CP/M, the most affordable and accepted disk format was the single-density 8″ floppy disk with 128 bytes per sector (B/S). Technically this is referred to as IBM system 3740 format. DR supported this standard by making every CP/M disk access a request for one sector of data, or exactly 128 bytes. The problem is, a disk format of 128 B/S is not efficient.

First, a floppy drive spends most of its time moving the head to the proper track and sector (this is called seeking), not transferring data. A 512 B/S format means one fourth as many seeks are needed to transfer an equal amount of data. For example, Microsoft's Basic interpreter loads in about 9.8 seconds on a standard 8″ CP/M disk and in 4.8 seconds on a single-density 512 B/S disk. This is a considerable improvement, and, in some instances, it gives the feeling of Winchester-like performance.

Second, selection of a larger sector size (such as 512 B/S) squeezes 23% more storage onto a single-density diskette, and 18% more compared to a double-density 128 B/S disk. This is an effective way to delay purchase of another disk drive. I increased the capacity of my drives from their standard 250K capacity to 580K by installing the algorithms to operate with a double-density, 512 B/S disk format (sufficient for my purposes of word processing and assembly language development). Bob Weidemann's article in the February 1983 issue of *Microsystems* explains the pros and cons of using larger disk sector sizes with CP/M.

The principles of the BDA

Throughout this article, *host* will refer to the actual track and sector on a disk. Using 512 B/S as the host sector size, blocking can be defined as the writing, by CP/M, of a 128-byte block of

dBEST.™

dBASE II® is, quite simply, the best-selling database management system (DBMS) made for any computer, ever.

And because it is so powerful yet so easy to use, it has become the standard for managing data with a microcomputer.

Over 150,000 users. So far.

Doctors and lawyers, accountants and salespeople, stockbrokers and students, big businesses and small are all managing their data better with dBASE II. All over the country, around the world, users have found that dBASE II is the best solution to all their information needs.

Knowledge is power.

With dBASE II, you can quickly and easily create a full business information system that does exactly what you need done, a system that will handle today's problems yet grow with you.

All of your data is at your fingertips with dBASE II. Using simple English-like commands, you add, delete, edit, display, print and manipulate your information.

Once you've decided on what you want done, you save the instructions so that even your least experienced personnel can perform the most complex business functions with two words: *Do Invoices, Do Payroll, Do* anything that needs to be done.

dBASE II is the most advanced information management tool available for your micro. And it's only $700 (suggested retail price).

A legend in its own time.

Already, books have been written about it. Hundreds of independent businesses are based on it. And other microcomputer programs measure themselves against dBASE II.

To see why, drop by your nearest computer or software store and ask for a demonstration. Then take a package home and use it for 30 days. If it's not everything we said it was, return it and get your money back.

But we think you'll keep it.

Can over 150,000 users be wrong?

For the name of your nearest dBASE II dealer, contact Ashton-Tate, 10150 West Jefferson Boulevard, Culver City, CA 90230. (213) 204-5570. (800) 437-4329, ext. 217. (In Colorado, (303) 799-4900).

ASHTON·TATE

CIRCLE 28 ON READER SERVICE CARD

DEBLOCKING

Continued from page 100

data into the host sector, and deblocking as the reading of a 128-byte block from the host sector for passing back to CP/M.

The algorithms, which are implemented as assembly code in the BIOS, intercept the drive, track, and sector request from CP/M and decide if the desired block of 128 bytes is in memory (a host sector of disk data is always kept in memory by the BDA).

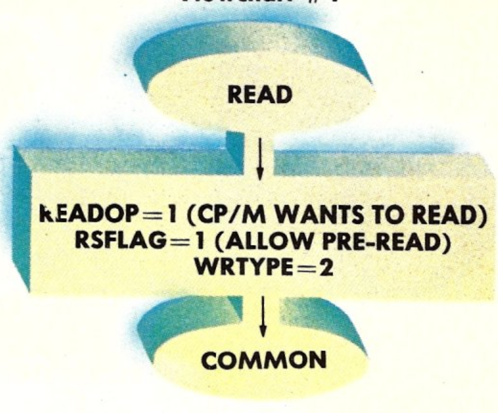
Because applications programs running under CP/M request sequential 128-byte sectors of data, the BDA needs to do only one 512-byte disk access per four requests by CP/M. The other three are fast 128-byte memory-to-memory transfers from the BDA post-memory area to CP/M.

CP/M's first disk read or write requires a preread of the disk. That is, the BDA must fetch a 512-byte host sector into memory before the 128-byte CP/M transfer can be performed. If the CP/M operation was a WRITE, then the BDA writes 128 bytes into the proper area of the 512-byte host memory area. If the CP/M operation was a READ, then the proper 128 bytes from the host memory area are passed to CP/M. After the BDA finishes the 128-byte transfer, it continues to hold the host sector in memory and returns to CP/M. Succeeding CP/M READs/WRITEs affect what the BDA does with the host sector currently in memory.

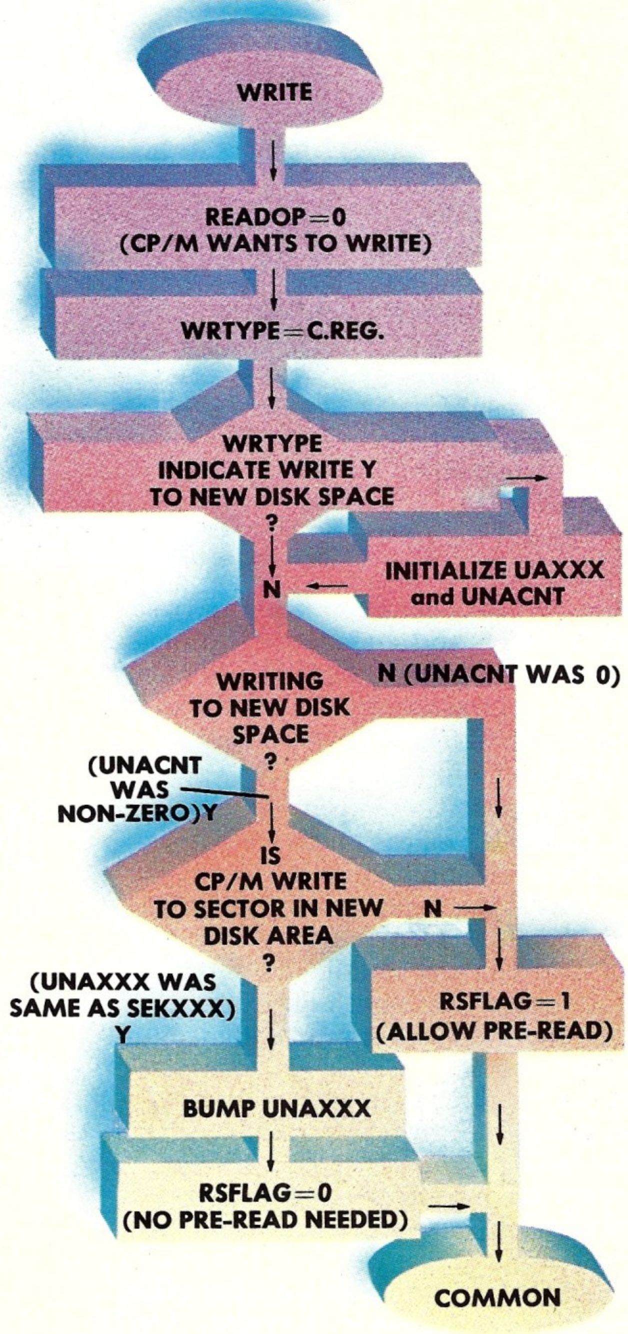
If the next CP/M operation is a directory write, the host memory area is written to disk. This leaves the disk fully updated and in a condition to be physically removed. If it is a normal data read or write, the BDA checks to see if the proper host sector is in memory. In the worst case, the BDA may need to write the host memory area to disk followed by a read before CP/M's request can be performed. CP/M tends to minimize this situation because it works with sequential blocks of 128 bytes.

Finally, CP/M indicates when it is writing to new disk space. This is called a write to an unallocated block and allows the BDA to eliminate prereads. This type of write occurs when, for instance, the command SAVE 20 MEMORY.SPC is typed at the console. In this case, CP/M creates a new file called MEMORY.SPC to hold 20 * 256 bytes of data. The BDA speeds the CP/M operation by eliminating reading of the host sector into memory. The BDA fills the host memory area with data passed to it from CP/M then writes it to disk every fourth time. There is no need to read the disk area first, since a new disk area is being written to.

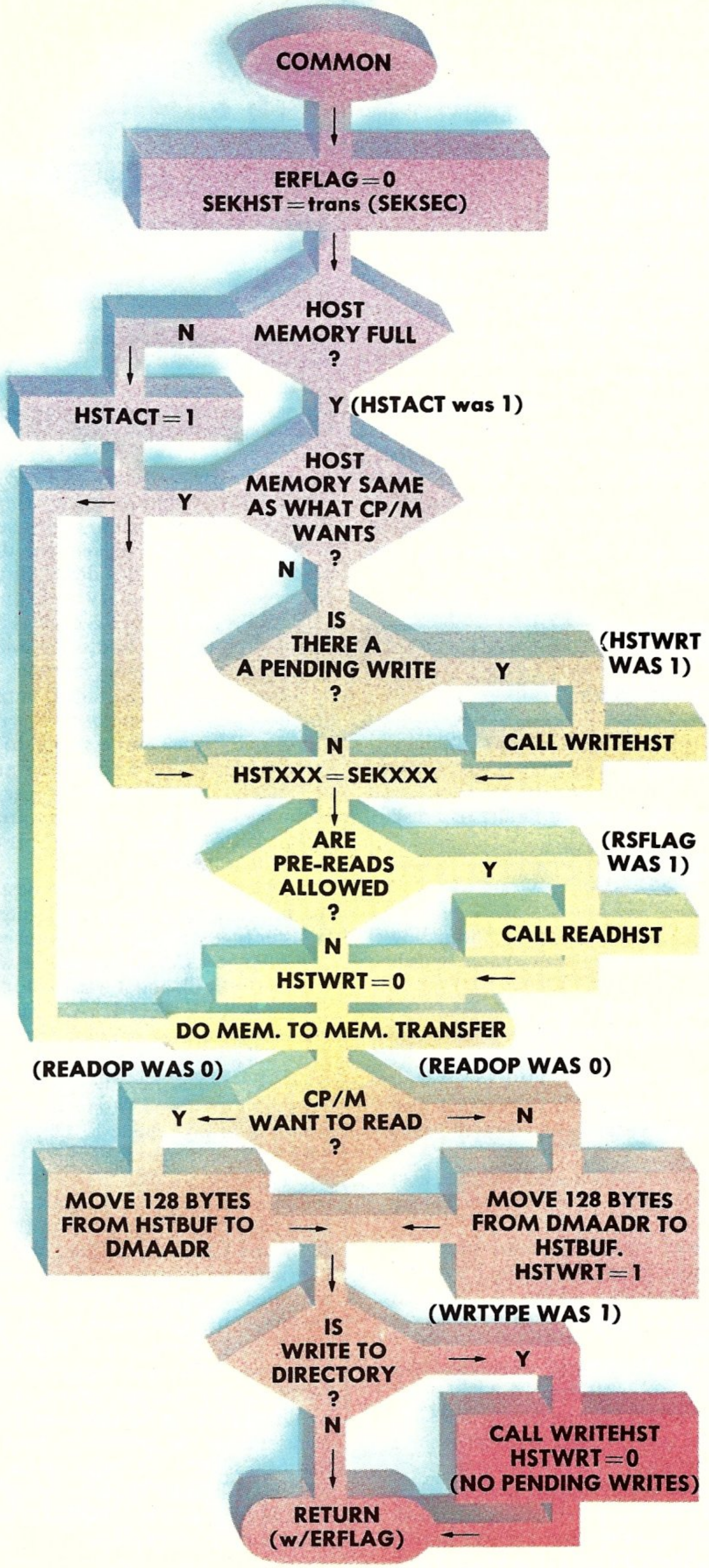
Flowchart #1



Flowchart #2



Flowchart #3



BIOS entry level alterations

The heart of the blocking/deblocking algorithm is contained in new READ and WRITE entry level routines provided by Digital Research. They are listed in appendix G of the *Alteration Guide* (V2.X) and included on the CP/M distribution disk in the file DEBLOCK.ASM. The versions of READ and WRITE originally in the BIOS must be modified somewhat and renamed READHST and WRITEHST respectively. Under this scheme, DR's READ, WRITE, and a new section of code called COMMON implement the BDA and call the hardware-specific routines READHST and WRITEHST when a disk access is needed.

The rest of DEBLOCK.ASM summarizes the changes needed by the remaining entry level routines. For the most part these changes either initialize BDA flags or store CP/M parameters in locations compatible with the BDA. For instance, the entry level routine SETTRK must be modified to store the track number into the variable SEKTRK. The new DR READ, WRITE and COMMON routines use SEKTRK when they are called by CP/M. (Note: A macro assembler is needed to assemble DEBLOCK.ASM. If Microsoft's M80 is used, line 15 must be altered to read: "if @y eq 1".)

Details of the algorithm

READ (Flowchart 1). READ sets the read/write flag (READOP) and the preread flag (RSFLAG) to one, indicating that CP/M wants to do a disk read and that prereads are allowed, respectively. Before entering COMMON, the type of write operation (WRTYPE) is set to a 2, which is significant only because any write by COMMON will not be considered a directory write.

WRITE (Flowchart 2). WRITE first sets the read/write flag (READOP) to a zero to indicate CP/M wants to do a disk write, then preserves the C register (passed by CP/M) in WRTYPE. If the write/type was 2, indicating that this is the first write to new disk space, WRITE preserves the drive, track, and sector numbers in the unallocated disk/track/sector variables (UNAXXX). In addition, the number of writes which CP/M will perform to the new disk area is placed in the variable UNACNT; it will always equal the CP/M block size divided by 128.

If UNACNT is nonzero, which indicates that writes are occurring to new disk space, then WRITE checks to see whether CP/M is trying to write to the next consecutive sector on the disk. It does this by comparing the disk/track/sector variables (SEKXXX) to which CP/M wants to write, to the un-

DEBLOCKING

Continued from page 103

allocated disk/track/sector variables (UNAXXX) initialized on the first write to new space and incremented after each write. The comparison provides a mechanism to detect when CP/M has prematurely abandoned the write to new disk space and is now writing to an allocated (or old) area. This will happen if only part of a new CP/M disk block is needed for a new file.

If the comparison indicates a write to new disk space, the remaining logic in WRITE bumps the UNAXXX variables to point at the next sector in preparation for the next WRITE by CP/M. Finally, the preread flag (RSFLAG) is set to zero, since prereading is not needed when writing to new disk space.

If UNACNT was zero or the comparison failed (indicating a write to the directory or old disk space), the preread flag (RSFLAG) becomes a one to cause prereading of the disk before writing.

COMMON (Flowchart 3). COMMON is a section of code executed by both READ and WRITE. On entry the disk error flag (ERFLAG) is reset and the sector number which CP/M wants (SEKSEC) is translated into a host sector number and put in SEKHST. Translation is needed becuase CP/M always thinks in terms of 128-byte sectors. Using the single-density 512 B/S example, CP/M thinks there are 32 sectors of 128 bytes on a track. If CP/M sets SEKSEC to 5, meaning it wants to read or write to the fifth 128-byte sector, COMMON places a 2 in SEKHST—the host sector that contains CP/M's request.

Next, COMMON checks to see if a host sector is residing in memory (HSTACT=1). If none is there, a condition that would be true on the first disk access or after a directory write, COMMON must fetch the host sector off the disk, then do the memory-to-memory transfer described two paragraphs down. If a host sector is present, COMMON checks to see if it's the

Using large disk sectors speeds program execution and increases disk storage.

proper sector containing CP/M's request. This is done by comparing the drive/track/sector variables set by CP/M (SEKXXX) against the drive/track/sector variables describing the host sector currently in memory (HSTXXX). If CP/M's request is in the host buffer, the CP/M disk request can be a memory-to-memory transfer.

If the comparison failed, implying the host sector in memory is wrong, COMMON must write it out to disk (the drive, track, and sector are in HSTXXX), then read the proper host sector specified by the three variables SEKXXX (note that a READ is done only if RSFLAG=1).

Finally, with the correct host sector in memory, a memory-to-memory transfer is performed. As a last check, if the CP/M operation was a write (READOP=0), and it was to the directory (WRTYPE=1), COMMON writes the host memory buffer to disk so there are no pending writes. This leaves the disk in a finished state so it can be physically removed from the drive. A return is always made to CP/M with ERFLAG in the A register.

READHST/WRITEHST. These routines are the original read and write from the BIOS. They perform the disk R/W and handle such functions as retries and error checking. (A disk error is indicated by setting ERFLAG to one.) On entry to these routines the drive, track, and sector are in HSTXXX. When the BDA was installed in the BIOS, one of the modifications specified by DEBLOCK.ASM eliminated sector translation in the BIOS entry level routine SECTRAN. To make up for this, READHST/WRITEHST should perform the translation of HSTSEC before the disk operation occurs.

Table 1. Glossary of BDA variables

DMAADR	a word sent by CP/M specifying where in memory a 128-byte block of data lies which is to be read or written to a host sector on disk.
ERFLAG	a byte returned to CP/M indicating success or failure of a disk operation.
HSTACT	flag set to nonzero if a host sector of data is currently being held in memory by BDA.
HSTBUF	a buffer in memory of the same size as a host sector.
HSTWRT	flag set to nonzero if a write operation has been performed by CP/M but the host sector in memory hasn't been written to disk yet.
HSTXXX	three variables called HSTDSK, HSTTRK, and HSTSEC which specify the host sector currently being held in memory by BDA.
READOP	set to 1 if CP/M called READ and 0 if it called WRITE.
RSFLAG	set to 1 to allow a preread and 0 if prereads are not needed. The second situation only occurs when writes to unallocated (new) disk space are performed.
SEKHST	a variable computed on entry to COMMON specifying the host sector number where the 128-byte block of data desired by CP/M is contained.
SEKXXX	three variables called SEKDSK, SEKTRK, and SEKSEC which specify the desired 128-byte of data desired by CP/M.
UNACNT	if WRTYPE=2 on entry to WRITE this variable is initialized with the CP/M block size divided by 128 (a CP/M block size of 2048 would cause 16 to be stuffed).
UNAXXX	three variables called UNADSK, UNATRK, and UNASEC which are initialized at the same time as UNACNT. They specify the next consecutive sector to which to write during WRITEs to unallocated (new) disk space, and are used to detect when a WRITE to new disk space ended early.
WRTYPE	a variable stuffed with the type of WRITE being performed by CP/M. These are: 0 if the WRITE is to a previously written disk area, 1 if the WRITE is to the directory, and 2 if its the first WRITE to an unallocated (new) area on disk. During a READ, WRTYPE is always set to 2 which is significant only because it's not a WRITE to the directory.

Richard I. Silverman, 205 S. Prospect, Apt. #8, Burlington, VT 05401

7 Good reasons why you should subscribe to *Creative Computing.*

It's the Number One magazine of computer applications and software!

There's one place you can always be sure of learning more about microcomputer software and applications: *Creative Computing.*

Every month *Creative Computing* provides you with a continuing education on everything related to microcomputers and computer equipment. Useful articles, "how to" tutorials, exciting new applications, games and "no holds barred" reviews of the latest software and equipment make up a major part of *Creative Computing*'s editorial content.

We give you probing features on programming breakthroughs and important news. Plus in-depth articles on elementary, intermediate and advanced software and applications topics—to help you develop your knowledge and skills, save hundreds (perhaps thousands) of dollars in unneeded software, discover uses for your personal computer that you might never have considered. Articles that increase your overall "computer consciousness." Here's how:

1 *Creative Computing* gives you things to actually do with a computer.

Just owning a computer isn't enough. You've got to know what to do with it. That's why applications are our primary focus. Text editing, animation, graphics, business simulations, data base and file systems, music synthesis, control of household devices, communications, games—some of the applications and software you'll learn about in *Creative Computing.*

2 *Creative Computing* discusses business applications in simple, nontechnical language.

If you're a business person who needs to know about the latest developments in word processing and office applications, turn to *Creative Computing.* We clarify such business applications as investment analysis, futures evaluations, data base management, mailing list programs, text editing, word processing and simulations. And all the software available for business people.

3 *Creative Computing* helps you decide which computer equipment is best for you.

Our tough, no-nonsense equipment profiles arm you with the facts before you walk into a computer store. You'll know the right questions to ask and how to cut through the jargon and sales hype. We give you authoritative guidance in deciding what you need, what you don't need—and what's right for you and your pocketbook.

4 *Creative Computing* covers computer education in depth.

We started out as a computer education publication, and we're still committed to the educational community. We regularly carry articles on designing educational software, evaluating educational software, teaching concepts and terminology in computer education, text editing applications for literature and computer simulations in the classroom—plus a great deal more.

5 *Creative Computing* brings you hours of mind-expanding game entertainment.

We've got a soft spot for the computer game addict—and computer game software. We know you want to understand more about the new computer games flooding the market: which ones are easiest to learn? Require the most skill? Offer the most surprises? Give you the best graphics? Provide the most challenge? Contain a new twist? *Creative Computing* brings you the answers.

6 *Creative Computing* features the state of the art.

Columns on the most popular personal computers, a "software legal forum," letters to the editor. Reviews of books, games, organizations, dealers and events. Fascinating interviews with leading innovators, equipment designers, program developers and game inventors—men and women who'll give you a real glimpse of the future!

7 Our price is right.

By subscribing to *Creative Computing* now, you can save as much as 33% off the full subscription price. To learn elsewhere what you'll learn from *Creative Computing*, you might spend hundreds of dollars in course fees and books. Then you'd have to winnow out what you could use from all that you'd learned. But *Creative Computing* does that for you, so you'll have time to enjoy your own computing interests. And that saving of time makes this offer very inexpensive indeed.

Join over 150,000 *Creative Computing* readers by subscribing today! Just use the coupon at right.

SAVE UP TO 33%!

Creative Computing · P.O. Box 5214 · Boulder, Colorado 80322

YES! Send me *Creative Computing* for:

☐ One year (12 issues) for \$19.97—I save 20%!
☐ Two years (24 issues) for \$36.97—I save 26%!
☐ Three years (36 issues) for \$49.97—I save 33%!

Savings based on full one-year subscription price of \$24.97.

Check one: ☐ Payment enclosed. ☐ Bill me later. 8H765

Mr./Mrs./Ms.____________________
(please print full name)

Address____________________ Apt._____

City____________________

State____________ Zip____________

Offer valid in U.S. and possessions only. Please allow 30 to 60 days for delivery of first issue.

MICROSYSTEMS REVIEWS

AUTODEX, a Menu-driven Front End

AUTODEX does everything a novice could wish for

by Bob Kowitt

In the March 1983 issue of *Lifelines*, I did an article on front end programs for CP/M. These utilities are designed to assist the end user, and especially the novice end user, when using CP/M operating system functions. AUTODEX, another such utility, runs under both CP/M and IBM PC-DOS.

AUTODEX will do all the basic CP/M functions: copying, erasing, and listing files, displaying their contents on either terminal or printer, running the programs listed, and has a couple of other features that are unusual. These features are menu driven, usually requiring only a single keystroke.

The manual has both a table of contents and an index and, with one exception, provides clear instructions. I will cover the exception later on.

Installation

The installation is also menu driven. If your terminal is listed on the first menu, that's all there is to it. If, however, it is not, it is suggested that you answer as if your terminal were the Osborne I and then run a second utility called TD.COM. This installation utility leads you through a series of questions about your terminal. To do this, you will need your terminal manual, or if you are a new user, ask your dealer for help. The codes are straightforward if you have had experience, but may prove confusing for the novice. This is not the fault of AUTODEX but of its control codes. Should you have insurmountable installation problems, Automatic Software USA is very cooperative and will give you any phone assistance you might need. They're nice people to deal with.

Usage

The updated version of AUTODEX loads automatically when the operating system is booted. AUTODEX clears the screen, displays its logo, clears the screen again and displays two separate areas. The top area contains a list of disk commands that may be executed by placing the cursor on any of them and pressing the return key; the lower area then displays a sorted list of disk files. This area has its own set of commands executed by a single keystroke.

The following is a description of the disk commands in the upper area.

SORT displays the disk files sorted by filename, type or filesize. It is the file display list that is sorted, not the actual disk directory.

CURRENT displays the current

default drive and permits changing the default. When it is changed, the selected disk's files are displayed below.

BACKUP identifies the backup disk drive, should you decide to backup or copy a disk file or group of files. With a two-disk system, this defaults to the drive that has not been selected as the current drive (above).

DATE. When utilized, the date in this position can be very easily copied to the third column of the disk file area next to any of the filenames and permits dating of each file on the disk.

FLAT prints the file list on your printer.

DISKID permits you to name your disk and open a disk file under this name that will store a date and the assigned descriptions for each file. The filename will be two dashes (. .) followed by the six-character name you have selected (see below).

HELP presents a list of commands available and a short explanation of each.

MULT is one of the more useful facilities provided with AUTODEX. Once you have gone through the file list marking files with an 'M,' MULT will do one of three operations; (1) It will backup the marked files to the designated backup disk, (2) it will list the marked files to the printer, or (3) it will erase all of the marked files from the disk.

USER. Under CP/M 2.x, when the current user area number is displayed, a new user area may be selected.

The ESC key toggles between the upper screen and the lower and at any point, CONTROL-P will print the entire screen on the printer.

When the cursor is located on the lower screen, the cursor control keys that have been installed into the AX system move the cursor to any point of the screen. When the cursor has been moved to a filename and then right to the date column, pressing the "=" key will copy the system date, as shown at the top area, into this column next to the selected filename. The data also may be entered manually. If you move the cursor to the fourth column, you may enter a description of the file that is considerably clearer than the eight-character filename permitted by the operating system. I definitely prefer having filename such as "total disk copying program" to DSKCPY.COM, or "Accounts receivable monthly report" to ARMORPT.COM.

Single character commands in the lower file area permit:

- (B)acking up a file.
- (C)opying the file to another name.
- (E)rasing a single file.
- (H)elp, which presents a list of commands available, each with a short explanation.
- (L)isting a single file.
- (M)arking for the MULT command.
- (N)ext screen (when the disk has more than one screenful.
- (P)revious screen (same as above).
- (R)enaming a file.
- (T), which makes the cursor location the top row of the display.
- (U), which copies to another user area.
- (V)iewing the file at the terminal screen if it is an ASCII file or does a Hex/ASCII dump if not.
- (X), which executes a program.

The one item that I must take to task in the documentation is the (X) command. In its simple form, it will run a .COM file from the disk and create a CP/M submit file to allow return to AUTODEX. I tried and tried to make some sense out of the manual's description of the posssible extended uses of the (X) command and finally came up with a working method.

AUTODEX does all the basic CP/M functions, then adds some unusual features.

Essentially, (X) will run a CP/M .COM file in the usual manner. If, however, you should try to e(X)ecute an ASCII or text file, AUTODEX will assume that you want to act upon that file with some existing .COM file. The possible .COM files are installed into AUTODEX with a utility called PROGLIST.COM. The procedure is as follows:

To implement this utility, first e(X)ecute PROGLIST, then answer the subsequent inquiries with names of various .COM files that are able to manipulate ASCII files in some manner. After the inquiries are answered, determine whether the entire ASCII file name is needed by the desired .COM program, and then allow the list to be combined with AUTODEX. When you have done this, you may exit PROGLIST.

After exiting PROGLIST, AUTODEX will reload. Move the cursor to the ASCII file and type 'X.' The right side of the screen will display the names of the .COM programs that are installed. Move the cursor to one to be run and press RETURN. The selected command file will be run with the ASCII file as its parameter.

For example, the ASCII file is called MY.LTR. I might, at one time or another, want to use one of the following: PMATE (a text editor), WS (a word processor), SQ (a file squeezer), ED (CP/M's editor). I will execute PROGLIST first:

```
Do you want instructions
                (y/n) ? N
Program Name:  PMATE
Update Date    (y/n) ? N
Full Filename  (y/n) ? Y

Program Name:  WS
Update Date    (y/n) ? N
Full Filename  (y/n) ? Y

Program Name:  SQ
Update Date    (y/n) ? N
Full Filename  (y/n) ? Y

Program Name:  ED
Update Date    (y/n) ? N
Full Filename  (y/n) ? Y

Program Name:  <RETURN>
               (to end entries)
 Install this program list
      into AUTODEX (y/n) Y
```

Now when I e(X)ecute MY.LTR, the right of the screen will display

```
PMATE   WS   SQ   ED   Other
```

Placing the cursor on one of these and pressing RETURN will run my choice and act upon MY.LTR.

Suggested improvements

1. The system of disk IDs is almost compatible with Ward Christensen's UCAT.COM, which I use to maintain a catalog of my disks. Ward has selected the single "–" as a filename initial to permit his program to find the disk identifier. This "–" sorts to the beginning of his FMAP and identifies the name of the disk to UCAT, the cataloging program. Because of AUTODEX's use of the double dash, I must rename my disks. I would have preferred a single dash for the start of the ID name.

The most inconvenient feature of this form of disk ID, however, is AUTODEX's use of .DID as the filetype. My disks are named with both filename and filetype. That is, I have named disks as –ARCHIVE.10A or –LETTERS.205. If I want to AUTODEX on all of my disks, I must name the disks, using only three characters for the ID plus a number. This

AUTODEX

Continued from page 108

might force me to change the foregoing disk names to --ARC10A.DID and --LET205.DID. I think the authors should depend upon the "–" at the beginning of the ID and not the filetype to identify the DISKID file. Better yet, the disk identifier character could be modified with the install program.

2. AUTODEX displays the size of each file and the amount of space left on the disk. When I am copying a series of files with the MULT command, I would like to know how much room I have left on my destination disk. This could be calculated just before doing the MULT backup by subtracting the backup file total from the amount of room left on the backup disk, taking into account the minimum file size required by the backup disk's format. If I have exceeded the available room, I can remove some of the files from the MULT list before proceeding.

Installation of AUTODEX is menu driven.

3. If there is a system ID file on the destination disk, the date should be installed with this file after the copy is complete.

4. The manual does not indicate whether file copying is done with or without verification. We should have this option, perhaps during installation of the system.

Conclusion

AUTODEX is clearly one of the easiest and most convenient front-end utilities available for CP/M. While not as powerful as some others, it is simple to get running without a programmer. It does everything a novice end user could want, and some more.

AUTODEX was formerly distributed by Durant Software under the name SIMPLIFILE, version 1.1. AUTODEX is currently available from:

Automatic Software USA
1035 Santa Barbara St.
Santa Barbara, CA 93101
(805) 963-5861

CIRCLE #302 ON READER SERVICE CARD

Bob Kowitt, 1727 N. Jerusalem Rd., E. Meadow, NY 11554

Table 1. Qualitative factors

Factor	Rating
Documentation	
Organized for learning	5
Organized for reference	5
Readability	5
Includes all needed information	4
Ease of use	
Initial startup	6
Application	6
Error recovery	
From input error	7
From full disk	0

Ratings in this table are on a scale of 1-7 with:
1 = clearly unacceptable for normal use
4 = good enough to serve for most purposes
7 = excellent, powerful, or very easy, depending on the category.

Screen 1

```
AUTODEX Screen List            Date: 091083
     * AUTODEX *     Sort      Multi   Current: A   Backup: B   Date: 091083
Space left: 930K     Exit      FLst    User: 0                  DiskId: ADSYS
Row: 2 of 38                   Help
CMD Name    Typ   Size           * * C O M M A N D   S U M M A R Y * *
--ADSYS     DID    4K
CPM13       SYS   12K        **DISK Commands**       ** FILE CMDs**
DU          COM   10K        Sort = Sort filenames    ? = Command
FIND        COM    2K        Mult = Bkup, List, Erase     summary
FORMAT      COM    4K     Current = Current drive     B = Backup file
LDRGEN      COM    2K      Backup = Backup drive      C = Copy file
MEMLIFE     COM    2K        Date = Today's date      E = Erase file
MEMTEST     COM    2K        Exit = Exit AUTODEX      L = List file
MODEM       COM   12K        FLst = List filenames    M = Multiple file
MODEM       PHN    2K      DiskId = Identify disk     N = Next screen
MOVE        COM    2K        Help = Command           P = Previous screen
NEW         SUB    2K               summary           R = Rename file
PRINT       COM    2K        **MISC Commands**        T = Make this
RECLAIM     COM    4K    <CTRL>-P= List screen            Top row
SH          COM   10K       <ESC> = Escape from       U = User number
SMARTKEY    COM    4K               action, or where      copy
STAT        COM    6K               you are           V = View file
SH          OVR   12K                                 X = eXecute program
```

Screen 2

```
AUTODEX Screen List                Date: 091083
     * AUTODEX *         Sort      Multi   Current: A   Backup: B   Date: 091083
Space Left: 930K         Exit      FLst    User: 0                  DiskId: ADSYS
Row: 2 of 38                       Help
CMD Name       Typ   Size  ChgDte  Description
    --ADSYS    DID    4K   091083
=>  CPM13      SYS   12K   091083  Operating System BIOS & BDOS
    DU         COM   10K   011283  Ward Christensen's Disc Utility
    FIND       COM    2K   040483  Ward Christensen's String Find Utility
    FORMAT     COM    4K   091083  Advanced Digital Format program
    LDRGEN     COM    2K   091083  Advanced SYSGEN program
    MEMLIFE    COM    2K   090982  Memory test
    MEMTEST    COM    2K   010183  Memory test
    MODEM      COM   12K
    MODEM      PHN    2K
    MOVE       COM    2K   091083  Single disc copier
    NEW        SUB    2K   091083  Formats, Installs Loader & Power
    PRINT      COM    2K   060682  Paginating printer
    RECLAIM    COM    4K   091083  Reclaims bad discs
    SH         COM   10K   041083  Microshell
    SMARTKEY   COM    4K   050583  Reassign keyboard keys
    STAT       COM    6K
    SH         OVR    6K
```

The Versatile Turbo Pascal

A multi-faceted program development system for CP/M-80, CP/M-86, and PC-DOS

by David Carroll

As we begin our story, we find Clark Mint, intrepid programmer, hunched over his microcomputer keyboard, working to beat the deadline set by his boss, Perry Height. Clark realizes that he won't complete his assignment in time, unless he resorts to extreme measures. After looking carefully around the office to be sure he is alone, Clark opens his briefcase and removes a small package, which he then unwraps to disclose—a floppy disk containing the answer to his problem. Placing the disk into his computer, he commands the system to run the powerful program contained in the microscopic magnetic particles on the disk.

Faster than Basic, more powerful than Fortran, able to compile long programs in a single pass—LOOK! On the screen—Its a word processor!—Its a compiler!—Its Turbo Pascal!

Far-out? Not really. Anyone who has done a large amount of program development on microcomputers is familiar with the frustrations of going back and forth between editor and compiler, the time-consuming disk saves and reads after each code change, and the lengthy searching for typos and other errors. Many have purchased hard disks to cut down on these delays.

Well, so what? You're thinking "just another pseudo-Pascal compiler, who cares?"—aren't you? WRONG! Finally, somebody has done it right. A powerful Pascal Z80 or 8086/88 single pass native code compiler (not a P-code compiler) together with a full screen editor (not a line editor) and error checking to make a super programming development package.

I received the Turbo Pascal prerelease package from Borland with mixed feelings. After several experiences with JRT Pascal and Digital Research's Pascal MT+, I was not looking forward to fighting my way through (yet) another "version" of Pascal. On the other hand, I had reviewed (and used extensively) two other Borland products: WordIndex II and MenuMaster—and found both were well designed and bug free. French-born Borland president Phillipe Kahn had told me some incredible things about their new Turbo Pascal, and I decided to find out if they were true.

First, some background. Borland products are developed by a team of high power professionals. The original word processing tools were developed to help automate document preparation

SAGE™ TECHNICAL BRIEFING

4 It's Fast In Any Language.

When it comes to software development, the difference between a Sage IV computer and other micros is like day and night.

With the Sage Computer it'll take you fewer days and nights to finish your program.

Speed is the reason. Speed resulting from the fast MC68000 microprocessor, fast architecture, fast operating systems, and blinding transfer rates.

The fact is, even we are surprised by the amount of software that's developed on Sage Computers.

So if you're a programmer, maybe you should spend some time learning how much time you could save using a Sage IV.

The Hottest Languages And Operating Systems.

Programs in nearly all of the important languages can be written on the Sage computer.

Included in the price is the **p-SYSTEM** operating system which supports **Pascal, BASIC,** and **FORTRAN.**

Other operating systems are optional. For **UNIX** fans there's **IDRIS,** which runs up to twice as fast (even without a Sage computer). **IDRIS** conforms to /usr/Group Standards Committee standards and programs written under it are highly portable to other micros. **CP/M** advocates please note that Digital Research has developed **CP/M-68K** for Sage hardware, providing a truly complete software development environment. Versions of **Pascal, BASIC, C-BASIC, C,** and **FORTRAN 77,** as well as a very fast **APL,** may be used under this operating system.

The Sage IV is also blindingly fast when running **hyperFORTH** with its extended programmer and user interfaces.

Besides **IDRIS,** other **Multi-User** operating systems that run on the Sage Computer are **PDOS, MBOS,** and **MIRAGE.**

A lot of excitement has been brewing in the Pascal World over Niklaus Wirth's new **MODULA 2**—and it's available for Sage computers. So is **ADA,** for the record.

If you found your language here, you just found your computer. The 16-bit, 2-million operation/sec Sage IV micro with up to one MByte RAM and 18-MByte internal hard disk.

For more information and the name of your nearest SAGE dealer, call us today. And be sure to ask about our new soft ware catalog describing over 200 application programs for Sage computers.

Sage Computer Corporate Office, 4905 Energy Way, Reno, Nevada 89502. Phone (702) 322-6868 TWX: 910-395-6073/SAGE RNO

Eastern United States
Sage Computer
15 New England Executive Park
Suite 120, Burlington, MA 01803 (617) 229-6868

SAGE™ COMPUTER

The computer you don't wait for!

CIRCLE 96 ON READER SERVICE CARD

TURBO PASCAL

Continued from page 110

and paper flow. All were written in Pascal and later hand optimized for speed. When the IBM PC was released, Borland decided to transfer their software to 16-bit, but was unable to find a useable Pascal native code compiler. So, they wrote their own: Turbo Pascal.

Turbo Pascal generates native Z80 code for CP/M machines and 8086/8088 code for MS-DOS and CP/M-86 machines. All commands except for a few operating system specific extensions to the language are fully transportable. Support for the 8087 math coprocessor chip is planned in future 16-bit versions. The package includes the compiler, a WordStar-compatible "editor," a loader and linker, and compile and runtime error checking. The system is driven by a simple user menu system. In addition, a 254 page typeset, softbound reference manual is included. All for $49.95? Yep!

Turbo Pascal is a reasonable superset of the original Pascal as defined by Jensen and Wirth in the *Pascal User Manual and Report.* There are a few differences from International Standards Organization Standard Pascal and several useful extensions to the language. I found that the several standard Pascal programs I tried compiled and ran without difficulty. I found a small error in the installing of the highlighting command. Start and stop highlighting functions appear to be reversed in the install program.

Using Turbo Pascal

Using the system is very simple. First, of course, make a work copy of the distribution disk. Then, type "TINST" to run the install program that will customize Turbo Pascal for your terminal and computer system. The program asks you if you wish to install the screen (terminal) or edit commands. (Turbo Pascal allows you to change any editing commands you wish from the distribution WordStar-like standard.) Type "S" for screen and a list of supported terminals will appear. If yours is on the list, type the appropriate number and your system is installed. If not, answer a few questions about your hardware, and Presto!, a new terminal is added to the list.

Using the work copy of the distribution disk, type "TURBO." This loads the complete system into memory. If you have installed the system for your terminal by running the install program as above, the screen clears and the main menu is displayed. All available commands are shown with the proper command letter highlighted. Select a workfile by typing "W" at the ">" prompt and then "TEST" (for our example). The .PAS filename extension is automatically added to the filename.

To use the editor, the system must be installed for your terminal. Type "E" to enter edit mode. A screen very similar to that of WordStar appears and the user may enter or edit a file using full screen editing (not line editing) like WordStar. An added screen command allows automatic indentation—the return key moves the cursor down a line and positions it under the first character of the previous line—a real help for "structured" programming. Unlike WordStar, the cursor may be freely moved anywhere on the screen without regard for blanks, spaces, or characters at a particular location. Tabs are automatically set to the position of the first letter of each word in the preceding line. Another added feature is the line restore command. If a line is deleted or changed, it may be restored to its original condition with a simple command as long as the cursor remains on that line. The normal "control-K, control-D" sequence ends the edit and returns to the main menu. Note however that the file is NOT saved till the user enters the main menu "S" command.

I compiled all 1,261 lines to a .COM file on a 6MHz Z80 TurboDos system. Wow!

Compiler options may be selected with the "O" command. The compiler may be directed to place the object code program in memory, in a .COM disk file (with the run library), or in a .CHN disk file without the library (for chained overlays). The memory area for the object code may also be specified, to reserve space for machine code or absolute variables.

The compiler is activated by entering the menu command "C" (or by "R"—running a modified program). You may compile in memory or to disk. Longer programs may be compiled to disk files if memory is limited. If a compile-time error is detected, the source code line location is shown (Figure 3). The user than types ESC and returns to the editor with the cursor pointing out the error (Figure 4). The error is then corrected using editor commands (Figure 5). Returning to the main menu, another "C" command starts the compiler again and this time we reach a normal termination (Figure 6).

If the program is compiled in memory (for debugging) and run, Turbo Pascal will locate the point in the source code where a runtime error occurred. If incorrect data are entered as input to our example program, an I/O error will normally occur (unless disabled). The expected input is of type integer and a character is entered generating the I/O error (Figure 7). The compiler searches the source code in memory and locates the error position in the source code. The user types ESC and returns to the editor with the cursor indicating the runtime error position (Figure 8).

The language

The Turbo Pascal implemention is close to the ISO Standard Pascal. The minor differences which exist are:

1. Different (expanded) procedures for Dynamic variables;
2. Get and Put are not implemented—instead Read and Write have been extended;
3. Goto statements may not leave current block;
4. Standard Page procedure is not implemented;
5. Packing is automatic, thus Pack and Unpack procedures are not implemented;
6. Procedures and functions cannot be passed as parameters (strings and pointers may be passed).

Turbo Pascal Integers take values between -32768 and +32767. Byte values are between 0 and 255, while the range of real values is 1E-38 through 1E+38 with a mantissa of up to 11 significant digits. Boolean, Char and user-defined types are also supported.

Turbo Pascal extensions mainly provide machine and operating system interfaces, and allow include files, program chaining, machine code linking, bit/byte manipulation, and random access data files.

The compiler may be instructed to optimize the machine code for arrays to be code size, or speed efficient, and in the CP/M-80 version, absolute nonrecursive code may also be specified. Recursive programs are slower and use more memory. The 16-bit version can address up to 1 megabyte of heap space for dynamic variables automatically (compared to the 64K limit of other 16-

bit Pascals).

Code generated by the compiler to a file is in standard .COM file format (for CP/M) and will run without the Turbo Pascal system. The assembler or loader is not required. The entire 6K runtime library (including a complete set of transcendental functions) is included in the .COM file. A compiler option allows generation of .CHN (chain) files without the library; these overlay the calling program in memory (but not the library or variable areas), allowing large applications to be written with several .CHN modules driven by a single .COM file.

As for speed, a 731 source line 20K demonstration PLO compiler program from Nicklaus Wirth's book *Algorithms + Data Structures = Programs* was compiled to a .COM file on a 4 MHz Multitech M-504 5¼″ Z80 computer system in 38.3 seconds. 10,230 bytes of code were generated plus data storage resulting in an 18K .COM file (2K block size). The same source file compiled on the same machine with Digital Research's Pascal MT+ version 5.5 took 5 minutes 13 seconds to compile 10,372 bytes of code and an additional 47 seconds to link and generate a 42K .COM file. JRT Pascal 4.0 called an undeclared type error at line 58 and quit.

On a smaller scale, the 35 line "Eight Queens" program from Wirth's book (see above) took 1 minute 40.9 seconds for MT+ to compile and link, 39.1 seconds for JRT Pascal 4.0 to generate a P-code .INT file, and 4.7 seconds for Turbo Pascal to make a .COM file. The resulting MT+ .COM file was 20K and the Turbo Pascal .COM file was 8K in size. The JRT 4.0 .INT file was 2K, but the 24K EXEC4.COM. program and the 16K PAS40.LIB are needed to run it.

The editor

The Turbo Pascal editor is a functional subset of the popular WordStar word processing program. It is a full screen oriented editor with the same command structure as WordStar. Major differences include:

1. The entire file to be edited must fit in memory;

2. A screen "indent" command allows structured format by automatically setting the margin to the start of the preceeding line;

3. Deleted line restore capability is included;

4. Single words may be marked as blocks with a single command;

5. Tabs are automatically set to the start of each word on the preceeding line;

6. The cursor may be moved any-

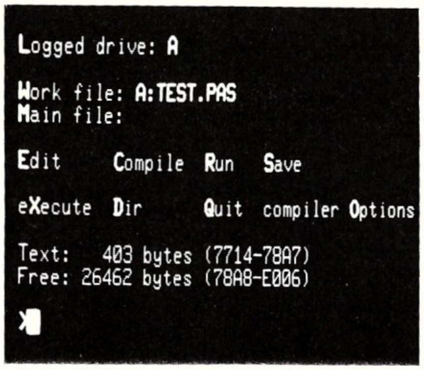


FIGURE 1

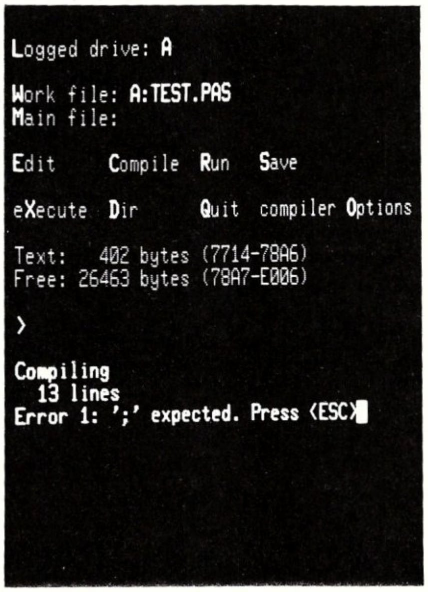


FIGURE 2

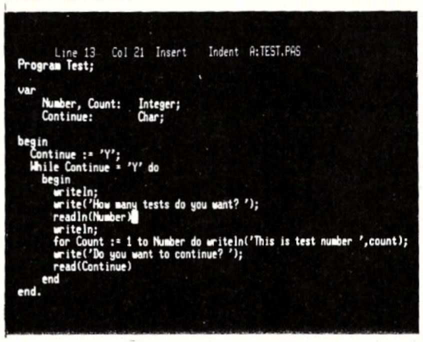


FIGURE 3

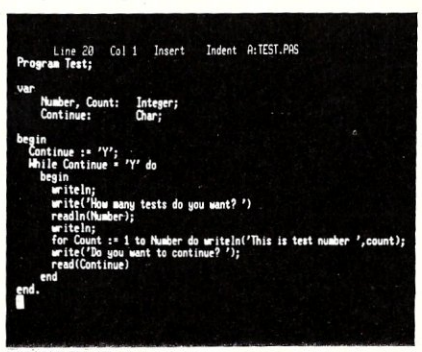


FIGURE 4

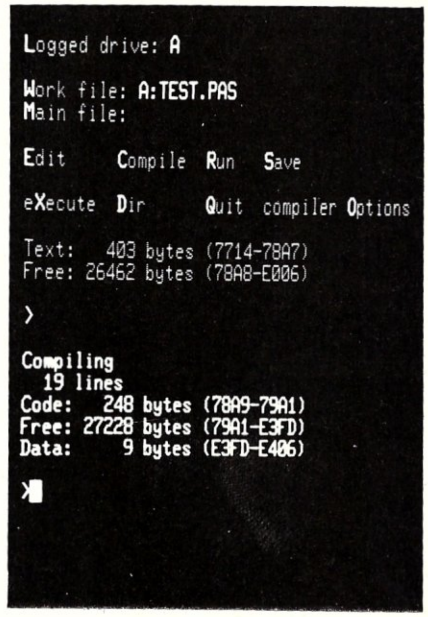


FIGURE 5

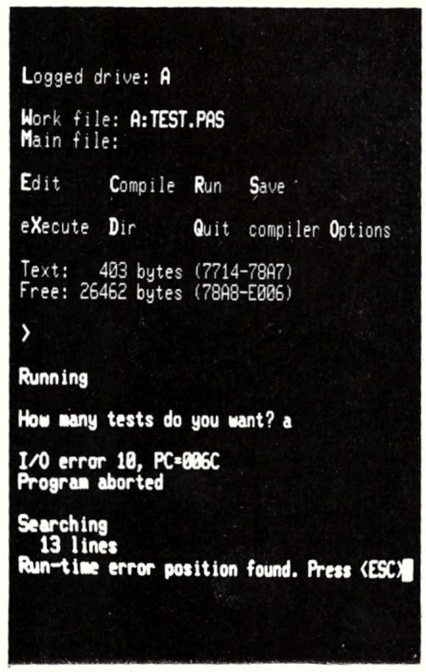


FIGURE 6

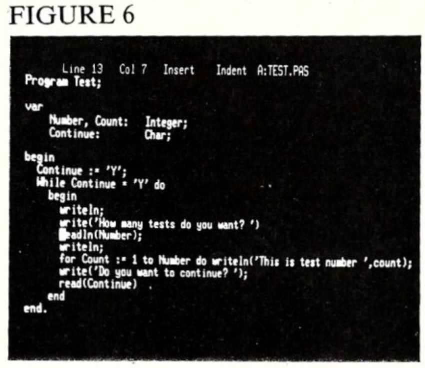
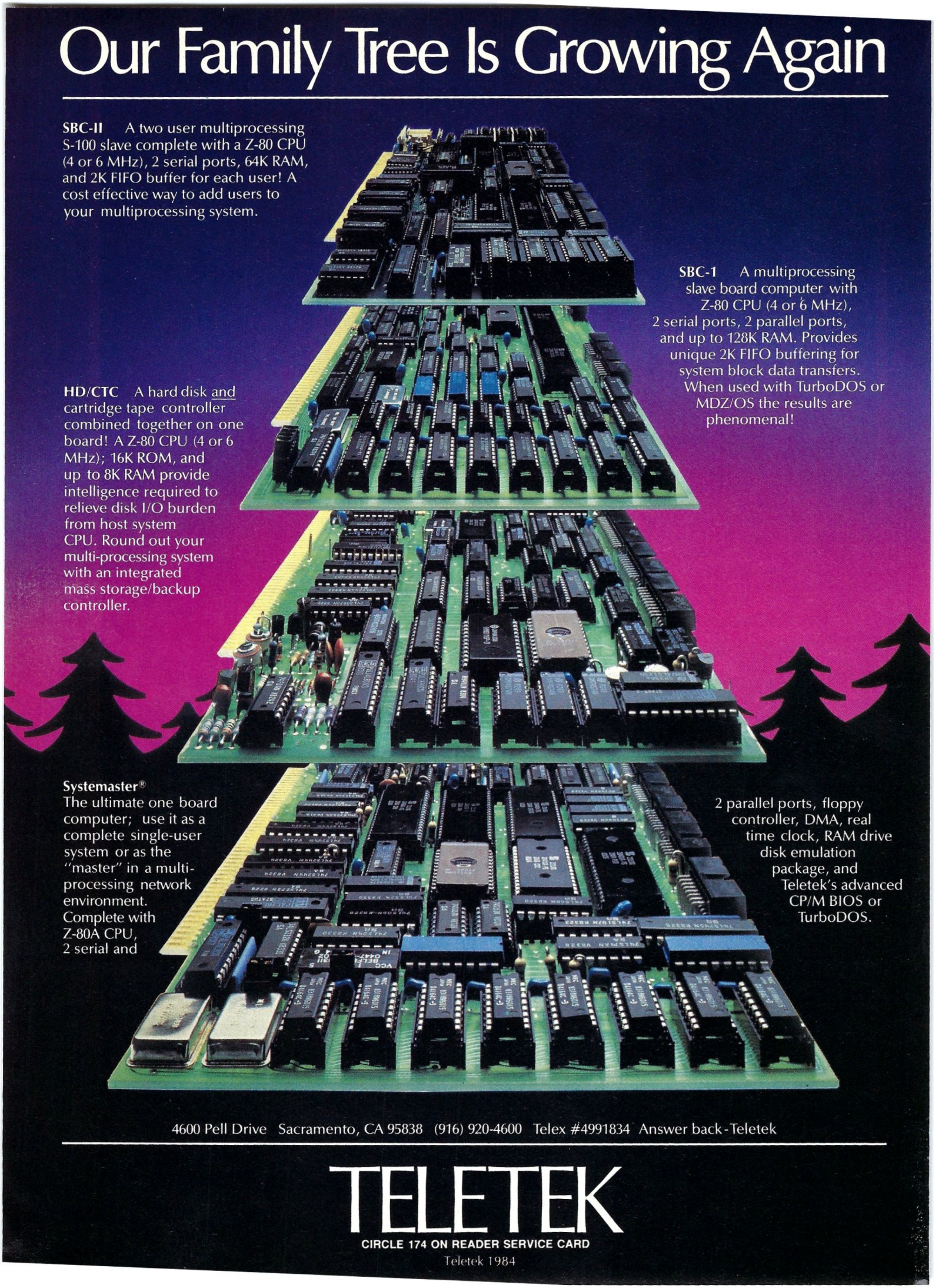


FIGURE 7



Our Family Tree Is Growing Again
SBC-II A two user multiprocessing S-100 slave complete with a Z-80 CPU (4 or 6 MHz), 2 serial ports, 64K RAM, and 2K FIFO buffer for each user! A cost effective way to add users to your multiprocessing system.
SBC-1 A multiprocessing slave board computer with Z-80 CPU (4 or 6 MHz), 2 serial ports, 2 parallel ports, and up to 128K RAM. Provides unique 2K FIFO buffering for system block data transfers. When used with TurboDOS or MDZ/OS the results are phenomenal!
HD/CTC A hard disk and cartridge tape controller combined together on one board! A Z-80 CPU (4 or 6 MHz); 16K ROM, and up to 8K RAM provide intelligence required to relieve disk I/O burden from host system CPU. Round out your multi-processing system with an integrated mass storage/backup controller.
Systemaster®
The ultimate one board computer; use it as a complete single-user system or as the "master" in a multi-processing network environment. Complete with Z-80A CPU, 2 serial and 2 parallel ports, floppy controller, DMA, real time clock, RAM drive disk emulation package, and Teletek's advanced CP/M BIOS or TurboDOS.
4600 Pell Drive Sacramento, CA 95838 (916) 920-4600 Telex #4991834 Answer back-Teletek
TELETEK
CIRCLE 174 ON READER SERVICE CARD
Teletek 1984

TURBO PASCAL

Continued from page 113

where on the screen, without regard to 'blanks' or 'spaces';

7. All editor commands may be redefined in the install program.

As can be seen, this editor was designed by a structured programmer, and is useful for just about any programming task. The editor limitation that files must fit into memory shouldn't cause any problems for Turbo Pascal programmers because of the include file compiler capabilities and the ability to chain programs.

The manual

The Turbo Pascal manual is a well written reference guide to this Pascal language implementation. It is a reference manual—not a tutorial, so a student should use a Pascal text with Turbo Pascal while learning. There are occasional misspellings and gramatical errors in the manual, but not enough to be irritating or prevent understanding. The 254 page guide includes detailed descriptions of all expressions, statements, blocks, data types, procedures and functions, compiler directives, differences, and error messages.

Final release version

I received the Turbo Pascal standard distribution disk from Borland shortly before this article went to press. There were two significant factors that should be mentioned.

First, I noted the CP/M-80 install program now offers nearly 20 terminal choices, but some of the supplied configurations seem to have minor errors which must be corrected for proper screen editor operation. I discovered one such error in the Liberty Freedom 50 terminal installation. The supplied configuration uses control-Z for screen clear. This can cause the main menu to begin flashing when the user returns from the editor. What is actually happening is that the terminal returns a control-Q in response to the control-Z command. The Turbo Pascal main menu won't accept most characters (including control-Q) as input, and so it issues another screen clear, causing the flashing effect. This minor problem is easily cleared up by reinstalling the Turbo system to send an ESCAPE * instead to clear the screen.

A Turbo Pascal disk also includes a demonstration program—1,261 lines of Pascal source code for a complete, operational electronic spreadsheet called MicroCalc. MicroCalc is a 154 cell (7 by 22) spreadsheet similar to VisiCalc and SuperCalc in many ways. It allows text, numeric, and formula entry, 11 digit precision, output to the printer and disk, disk program loads and saves, and it uses the same cursor controls as the Turbo editor. Best of all, you can modify it! As an example of a real Pascal application it is without parallel.

The CP/M-80 version of MicroCalc is made up of masterfile and 6 "include" files to allow compilation. It totals 44K of source code which compiles to a 22K .COM file. I compiled all 1,261 lines (including the 6 "include" files) to a .COM file on a 6MHz Z80 TurboDos system with hard disk in 28 seconds. Wow! I found that MicroCalc ran without problems.

Conclusions

It is impossible to cover all of the features of a sophisticated product like Turbo Pascal in a brief review. However, I can say that Turbo Pascal is a complete program development package for both beginners and serious professionals. Its availability for CP/M-80, CP/M-86, and PC-DOS/MS-DOS and its chaining and include capabilities that make it attractive for developing applications for almost any market. The $49.95 introductory package only allows single CPU use of the compiler and generated code, but an unlimited development and compiled code distribution license is available for additional $100. This package would be a good deal at $499.95 (look at Digital Research's Pascal MT+)—at $49.95 it's a steal. Either the editor or the spreadsheet alone is worth the price. (By the way—the price will not stay this low forever—it is an introductory offer.)

One small negative: the CP/M-80 version is *only* for Z80 processors. It produces Z80 code and is written in Z80, so if you have an 8080 or 8085, get a new computer or forget it.

Borland has contracted with an outside supplier for copying, packaging, and shipping services—hopefully this will prevent the delays in shipping which plagued JRT Pascal 2.0 and 3.0 if Turbo Pascal becomes as popular as JRT. Only time will tell.

For more information contact **Borland, International,** *4807 Scotts Valley Dr., Scotts Valley, CA 95066.*

CIRCLE 303 ON READER SERVICE CARD

David Carroll, P.O. Box 699, Pine Grove, CA 95665

Software

Directory

A selection of new database and communication packages

Program name: Autocode 1
Hardware system: any Z80 or 8080/8085 computer with CP/M, CP/M-86, PC-DOS or MS-DOS
Minimum memory: 130K
Description: Autocode is a program generator that writes customized applications programs which run on dBASE II. Autocode generates reports, analyses, calculations, screens, menus or submenus. Using simple English language prompts and responses, Autocode creates files, defines report criteria, sets up formulas for total and subtotal fields, forms parameters for entering data, etc. Autocode is entirely menu driven and includes no control characters. The 35-page manual contains examples of each definition.
Price: $195
Available from:
Axel Johnson
666 Howard Street
San Francisco, CA 94105
(415) 777-3800
CIRCLE 316 ON READER SERVICE CARD

Program name: REBOS
Hardware system: any CP/M or MS-DOS system, including the IBM PC-XT
Minimum memory: 64K
Language: dBASE II
Description: REBOS automates existing filing and reporting systems, customizes and types frequently used letters, orders legal documents and prepares mass mailings. The REBOS archive database stores information on closed transactions for future use and client follow-up. REBOS prepares cash flows, provides projections for any desired time period, and can project individual associate incomes. REBOS is password protected on three levels, allowing sales associates access to information while protecting management reports and preventing unwanted editing of files. REBOS can create an ASCII file from any of its databases, enabling the user to write his own programs and reports or to interface with his word processor.
Price: $2,500
Available from:
Clover Data Systems, Inc.
P.O. Box 270
Kulpsville, PA 19443
(215) 362-9000
CIRCLE 317 ON READER SERVICE CARD

Program name: Electronic Mail Manager
Hardware system: Any microcomputer running COMMX-PAC (required)
Language: dBASE II
Description: The Electronic Mail Manager provides a private electronic mail network used in conjunction with COMMX COMMunications eXchange software. Data is transferred between all locations, completely unattended, at predetermined times when phone rates are best. Each user is assigned a mailbox number requiring a password for system operation. Users may only access data addressed by or to their mailboxes. A branch manager is assigned a system password allowing privileged system operations. Up to 1,000 mailboxes can be assigned. A master station controls assignment and alteration of mailbox data. The system manager prepares all files for transmission and invokes the communications monitor which allows access via password. The master station then calls all branch locations for mail pickup. Mail is then sorted and delivered to its destination branch. A user may attach other files to a memo such as WordStar, SuperCalc, MBasic or .COM files. Archives are automatically created for all outgoing and incoming memos. EMM has a search function which can be setup to find items by date, subject, source, or destination using and/or expressions. This feature works with the outgoing mail bag, incoming mail bag and archives.
Price: $400 for initial Master/Slave pair; $150 for each additional Slave
Available from:
Hawkeye Grafix
23914 Mobile
Canoga Park, CA 91307
(213) 348-7909
CIRCLE 318 ON READER SERVICE CARD

Program name: Quickcode, dUTIL, Quickscreen, dGRAPH
Hardware system: any CP/M, MS-DOS or CP/M-86 based computer
Minimum memory: Quickcode: 180K; Quickscreen-dGRAPH: 48K
Language: dBASE II
Description: Quickcode automatically generates programs (in the form of "command" files) to run under dBASE II without programming by the user. Quickcode also expands the power of dBASE II with four new types of data: dollars, dates, telephone, and social security numbers. Quickscreen sets up the screen: you just define the data fields and Quickcode writes all the programs you need. dGRAPH automatically loads dBASE II data, computes scales, draws grid lines, and labels charts. It also has automatic shading and graph overlay. dUTIL puts often repeated instructions into a standard text file using any editor or word processor. IF/END IF and DO/ENDDO sequences are automatically indented and aligned.

WHY GAMBLE?

Don't take chances! Isolators provide complete protection.

Pat. # 4,259,705

They prevent:

- CPU/printer/disk interaction
- Lightning or spike damage
- AC power line disturbances
- RFI-EMI interference

Commercial Grade Isolators

ISO-1	3 Isolated Sockets	$ 81.95
ISO-2	2 Isolated Socket Banks, 6 Sockets	$ 81.95

Industrial Grade Isolators

ISO-3	3 Double Isolated Sockets	$122.95
ISO-11	2 Double Isolated Banks, 6 Sockets	$122.95

Laboratory Grade Isolators

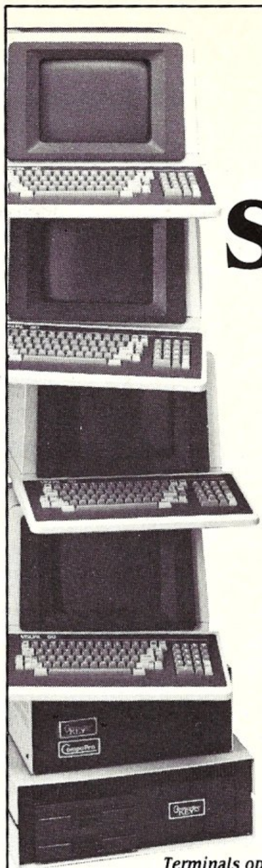
ISO-17	4 Quad Isolated Sockets	$213.95
ISO-18	2 Quad Isolated Banks, 6 Sockets	$180.95

Circuit Breaker, any model (Add-CB)	Add	$ 11.00
Remote Switch, any model (Add-RS)	Add	$ 20.00

ESP Electronic Specialists, Inc.
171 S. Main St., Box 389, Natick, Mass. 01760 (617) 655-1532

Toll Free Order Desk 1-800-225-4876
MasterCard, VISA, American Express

CIRCLE 102 ON READER SERVICE CARD



POWER PRICE & SUPPORT

Key Micro Systems offers a complete selection of expandable single or multi-user systems, available with floppy and hard disk drives. With them you can enjoy the best of S-100 computers and run any mix of 8 and 16 bit programs simultaneously. Our use of the highest quality peripherals with the power of CompuPro boards delivers the reliability you need — at the right price.

If your company needs complete software selection, on-site service and full customer support, you need Key Micro Systems.

Key Micro Systems, Inc. is a Full Service CompuPro® Systems Center

KEY MICRO SYSTEMS INC.

Terminals optional.

1606 Nooseneck Hill Rd., PO Box 715, Coventry, RI 02816 · **401/828-7270**
822 Boylston St., Suite 201, Chestnut Hill, MA 02167 · **617/738-7305**

CIRCLE 240 ON READER SERVICE CARD

LARGE INVENTORY AVAILABLE FOR IMMEDIATE DELIVERY

SMITH SYSTEMS INC.
215 Joliet St., Suite #1
Dyer, IN 46311 • PH. 219/865-6010
MIDWEST DISTRIBUTOR FOR
HDL Research S-100 Main Frames

MCS-112 DESK 12 SLOT W/CVT PWR	$737.
MCS-122 DESK 22 SLOT W/CVT PWR	$881.
RM-12 RACK 12 SLOT W/CVT PWR	$879.
RM-122 RACK 12 SLOT W/CVT PWR	$1036.
DDE-8 DISK ENCLOSURE W/CVT PWR	$910.
DDE-8R DISK ENCLOSURE W/CVT PWR	$1015.
TF-12 12 SLOT W/SP FOR 2-5" DRVS	$960.

Ready to Run Systems Available Now!!
Send For Our System's Flyer
(Dealer Inquiries Invited)

OUR BAR CODE READING/PRINTING SYSTEM

Video Movie Center
Customer ID Card
SMITH COMPUTER SYSTEMS
215 Joliet st Suite 1
Dyer IN 46311
219-965-6010
ID Number 10001
10001

FEATURING
- HP WAND
- OUR 8080 BAR CODE SOFTWARE
- MACHINE LANGUAGE PROGRAM RUNS UNDER DBASE II VERS 2.4
- USES POPULAR 3-OF-9 BAR CODE
- PRINT FROM YOUR DBASE FILES
- READ INTO YOUR DBASE FILES
- SAMPLE PROGRAM INCLUDED
- EPSON FX, MX, OR GEMINI

THE MODEL 3901, OUR INITIAL OFFERING, CONSISTS OF SOFTWARE THAT INSTALLS UNDER DBASE II (VERS. 2.4 UNDER CP/M-80), AND BAREBONES HARDWARE REQUIRING 5V DC (REGULATED) AND A SINGLE BIT TTL PARALLEL INPUT PORT. PRICE: $250. SEND FOR DATA SHEET. AN RS-232 SERIAL VERSION WILL BE DESCRIBED IN OUR NEXT AD.

COMING SOON IBM PC BARCODE LINK

COMPLETE COMPUTER SYSTEMS

MORROW MD-3 64K W/MDT-20 TERMINAL & PERFECT'S SOFTWARE	$1795.
LNW 80 II WITH 2 DRIVES SOFTWARE & AMDEK AMBER MONITOR	$2350.

SYSTEM COMPONENTS

IOMEGA 10-MEG REMOVABLE CART WITH SCSI CONT INTERFACE MASTER			$1795.
DTC S-100 CONTROLLER (WITH CP/M-80 (SOFTWARE DRIVERS)			$395.
8" QUME DT-8 DS/DD	$475	8" MITSUBISHI THIN DS/DD	$464.
CCS 2810 Z-80 4MHZ CPU	$245	TARBELL DD ASM W/CPM	$395.
Memory Merchant 64K Static	$419.	PMMI 300 BAUD S-100 MODEM	$339.
SSM I/O-8 8 PORT SERIAL	$449.	GEMINI 10x PRINTER PARALLEL	$320
Adds View Point ADM-3A+	$545.	AMDEK V300A AMBER	$165.

Terms: We accept Visa/MC, prepay, check or money order. Please allow personal checks two weeks to clear before shipment. $5.00 handling fee on all orders under $50.00. 15% restocking fee. All orders shipped via U.P.S. unless otherwise specified. All U.P.S. C.O.D. orders require cash or cashiers check. Prices subject to change without notice.

CIRCLE 55 ON READER SERVICE CARD

NOW! CP/M-80 SOFTWARE USERS...

KNOW YOUR SOFTWARE PROGRAM BEFORE YOU BUY... CHECK OUT - THE ENCYCLOPEDIA OF CP/M-80 SOFTWARE©...

A comprehensive, indepth, single source of information about the "vast amount of software" for your CP/M-80 8-bit microcomputer. Always current, updated, containing all of the CP/M-80 software available plus the latest price reductions. You can save more than the cost of an annual subscription on your very next software purchase by knowing what is available, what it will do and what it will cost. Answers most questions on every program.

CP/M is a trademark of Digital Research.
© Copyrighted by ENCYCLOPEDIA PUBLISHING CORP.

Rush your order now for the Spring Edition. One year subscription (four quarterly issues) $129.00* *Foreign orders $169.00 payable in U.S. Funds. Visa, Mastercard, American Express, Carte Blanche, Diners Club. 24 hour toll free telephone orders. 1-800-522-1500 Ext. 865

Or mail subscription orders prepaid to:
ENCYCLOPEDIA PUBLISHING CORPORATION
950 Hampshire Road, #108
Westlake Village, CA 91361
(805) 497-8002

CIRCLE 122 ON READER SERVICE CARD

SOFTWARE

Continued from page 116

dUTIL also highlights all dBASE II reserved words by setting them in upper case.

Prices: Quickcode: $295
dGRAPH: $295
dUTIL: $ 99

Available from:
Fox & Geller
604 Market St.
Elmwood Park, NJ 07407
(201) 794-8883

CIRCLE 319 ON READER SERVICE CARD

Program name: CPL: Computer Phone Link
Hardware system: A modem; CP/M, MP/M II, CP/M-86, MP/M-86, MS-DOS
Minimum memory: 48K (8-bit); 64K (16-bit)
Language: C
Description: Computer Phone Link is a powerful and friendly asynchronous communications program that allows you to Upload/Download files from other micros or mainframes. CPL is delivered complete and ready to run. No user software patching is required.
When released: 1982
Price: $165
Available from:
Cawthon Scientific Group
24224 Michigan Ave.
Dearborn, MI 48124
(313) 565-4000

CIRCLE 320 ON READER SERVICE CARD

Program name: W$L: Wall Street Link
Hardware system: A modem; CP/M, MP/M II, CP/M-86, MP/M-86, MS-DOS
Minimum memory: 48K (8-bit); 64K (16-bit)
Language: C
Description: Wall Street Link is communications software specifically designed for the Dow Jones information network. W$L will automatically dial your local tymnet or telenet number, send your password and allow you to use Dow Jones information with helpful prompts rather than cryptic codes. W$L is delivered complete and ready to run. No user software patching is required.
When released: 1982
Price: $165
Available from:
Cawthon Scientific Group
24224 Michigan Ave.
Dearborn, MI 48124
(313) 565-4000

CIRCLE 320 ON READER SERVICE CARD

The North Star HORIZON® 8/16. Now it's turbocharged for better all-around performance.

Introducing a new standard for multi-user CP/M® applications.

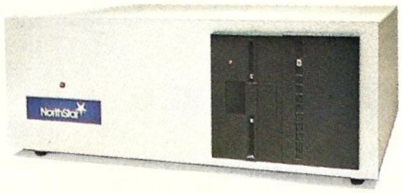
North Star's TurboDOS® HORIZON 8/16 can handle as many as eight users while supporting both 8-bit and 16-bit applications simultaneously. And do it all at a lower cost-per-user than most conventional time-sharing systems.

It's faster, too.

In fact, North Star TurboDOS is several times faster than standard multi-user operating systems. What's more, it's compatible with CP/M-80,™ CP/M-86,® and MP/M.™ It offers networking features like file-sharing, record locking and electronic mail. And when it comes to durability, consider this: more than 30,000 first-generation HORIZONs are still in use. And each one can be easily upgraded to the new 8/16 architecture.

Test drive our turbocharged HORIZON today.

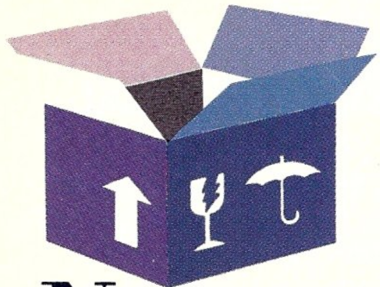
You'll find one at more than 1,000 computer stores and systems houses nationwide. For the location near you, call 800-722-STAR. Or write North Star Computers, Inc., 14440 Catalina Street, San Leandro, CA 94577.



NorthStar
Simply powerful solutions.

Special OEM incentives. Call our OEM Coordinator at (415) 357-8500.

CIRCLE 19 ON READER SERVICE CARD



New Products

A collection of devices that protect your computer against variations in voltage

Z8 Basic 8-Channel A/D converter

The Micromint, Inc., analog-to-digital converter board is an eight-channel device designed for use with the Micromint Z8 Basic computer controller board. It provides the ability to monitor up to eight analog signals in one of two 10-volt ranges. The A/D converter provides eight channels of conversion, with each channel having eight bits of resolution. Because the A/D converter board is memory-mapped, both Basic/Debug and Z8 assembler language programs can easily access it. No special routines are needed to configure the A/D board. It constantly samples the eight input signals and presents them to the computer on request. The eight channels are refreshed every 640 ms, providing over 1,000 conversions per channel per second.

The A/D converter board may be addressed to any of eight locations in 16 base addresses, so multiple boards may be installed on a single Z8 computer/controller system.

Price: $140 in single quantity.

The Micromint, Inc., *561 Willow Ave., Cedarhurst, NY 11516;* **orders only:** *(800) 645-3479;* **technical contact:** *(203) 871-6170.*

CIRCLE 305 ON READER SERVICE CARD

Transient overvoltage supressor for home electronics

A new transient overvoltage protection device designed specifically for use with personal computers has been introduced by Transtector Systems.

Designated the model SL™, the divice plugs directly into any standard 110 volt outlet and accepts all standard three-pronged plugs.

The SL automatically cleans incoming electrical powerlines of overvoltage transients and spikes, thus eliminating the major cause of damage to sensitive solid-state components. Transtector's Model SL responds to a transient in five billionths of a second or less. It features an instant reset to provide continuous, uninterrupted protection and is UL Listed. The SL measures about 3″ × 3″ × 3″ and carries a 90-day replacement warranty.

Price: $99.

Transtector Systems, *E. 5250 Seltice Way, Box 1299, Post Falls, ID 83854; (800) 635-2537; (208) 773-1521.*

CIRCLE 306 ON READER SERVICE CARD

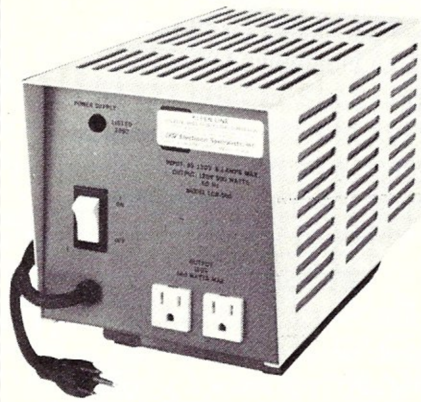
Kleen-Line conditioner

Electronic Specialists announces the Kleen-Line® conditioner, their newest entry for complete microcomputer protection and interference control. Ruggedized construction assures heavy load startup, crucial for disk and printer operation. Other features of this line conditioner include sine wave output, ultra-quiet operation, added input spike suppression, and wide-band prefiltering.

Regulation is to 117 VAC ± 4% output for 90 to 140 VAC input. Standard and intersocket Isolator models available in 250, 500, 1,000, and 2,000 watt ratings.

Price: $291.95 to $1051.95, depending on model.

Electronic Specialists, Inc., *171*



South Main St., P.O. Box 389, Natick, MA 01760; (800) 225-4876 or (617) 655-1532.

CIRCLE 307 ON READER SERVICE CARD

AC line filter and transient surge suppressor

PMC Industries, Inc., has announced its new Superguard AC line filter/transient surge suppressor for protection of personal and business computers, microprocessor-based scientific instrumentation, and sensitive electronic and audio equipment. It is designed to protect these units and their data from AC line noise, transients, and high-voltage surges that cause much damage.

The unit eliminates EMI/RFI common mode and differential mode noise to 70 db, as well as high voltage and energy spikes to 70 joules (6,500 amps; 780,000 watts). The unit comes with six three-wire grounded outlets, a 6′ three-wire shielded AC line cord, plus a 15-amp resettable circuit breaker, main on/off switch, and indicator light. It is made with all UL listed components, and is for 120 VAC, 15 amp use.

The Superguard's off-white painted case is all steel for additional sheilding and measure 5″ × 4½″ × 9½″; its weight is 7 lbs.

Price: $249

PMC Industries, Inc., *9353 Activity Rd., San Diego, CA 92126; (610) 695-3520.*

CIRCLE 308 ON READER SERVICE CARD

NEW PRODUCTS

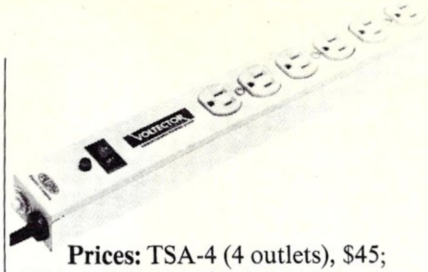
Continued from page 120

Conditioned power strip prevents unexplained data loss

Pilgrim Electric Company has introduced a new "TSA" economy line of conditioned power strips. These provide cost-effective protection against high-voltage spikes, surges, glitches, and transients that affect microcomputers, word processors, and other microprocessor-based office equipment.

The unique TSA design meets the latest industry Surge Voltage Standard and provides reliable normal and common mode protection. It assures electronic equipment of clean incoming AC power and prevents disturbances generated by equipment from being conducted back into the powerline. Hence Model TSA conditioned power strips supply the positive surge protection that many computer manufacturers leave out of their equipment.

Available with 4, 6, or 8 outlets, TSAs plug into any 120-volt receptacle and handle up to 15 amps. All units are circuit-breaker protected against overload and equipped with a 6′ heavy duty three-conductor-line cord, master on/off switch, and red pilot light.



Prices: TSA-4 (4 outlets), $45; TSA-6, $47.50; TSA-8, $49.95.

Pilgrim Electric Co., *29 Cain Drive, Plainview, NY 11803; (516) 420-8989.*

CIRCLE 309 ON READER SERVICE CARD

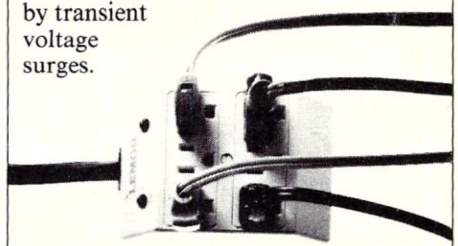
Transient voltage surge suppressor

A new AC power surge protector introduced by Electronic Protection Devices comes with a unique insurance program underwritten by Lloyds of London to accentuate and guarantee product performance.

The Lemon™ AC surge protector prevents computer hardware, software, peripherals, and other sensitive electronic devices from being damaged by transient voltage surges. Unlike conventional surge protectors, instantaneous voltage clamping includes "ground," which prevents flashover to other ports and damage to equipment.

The Lemon plugs into a wall outlet like an extension cord and accepts plugging for six devices, providing protection in excess of 6,000 V and 200 amps. It is housed in a high-impact plastic case and has two LEDs that light, verifying line or ground protection.

Price: $59.95 (list). The user's hardware is protected to $2,500 for repair or replacement due to damage by transient voltage surges.



Electronic Protection Devices, Inc., *217 Bear Hill Rd./ P.O. Box 673, Waltham, MA 02154; (800) 343-1813; (617) 890-2518.*

CIRCLE 310 ON READER SERVICE CARD

Surge Sponge protects RS-232 interface

The Model 21 Surge Sponge from Remark Datacom Inc. protects RS-232 interfaces from high-voltage transients produced by inductive coupling of interface cables with high power cables. The Surge Sponge also offers protection

NEW PRODUCTS

Continued from page 122

against lightning strikes that may hit the interface cable or somehow be conducted to the interface.

The Model 21 Surge Sponge incorporates fast MOV devices to protect pins 2, 3, 4, 5 and 7 of the RS-232 interface. The respective signal names for these pins are: TRANSMIT DATA, RECEIVE DATA, REQUEST TO SEND, CLEAR TO SEND and SIGNAL GROUND. All voltages appearing on any of these pins that exceed approximately 27 volts are clamped to FRAME GROUND, pin 1. All pins of the RS-232 interface are wired through the Model 21, so it appears totally transparent and has no effect on standard RS-232 levels.

The Surge Sponge is packaged in a small plastic case measuring 2.125 × 2.23 × 0.625 with a male DB25 connector on one side and a female DB25 connector on the other side. Both connectors are fitted with standard locking hardware to secure the Model 21 to computers, terminals, printers, and cables. The Surge Sponge incorporates PC board construction.

Price: $39.95 in single quantity.

Remark Datacom Inc., *148 New York Ave., Halesite, N.Y. 11743; (516) 423-3237.*

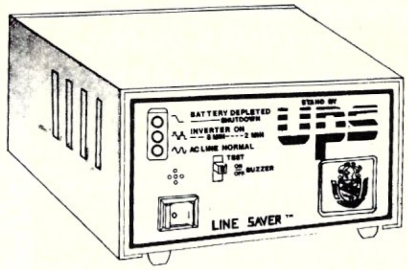
CIRCLE 311 ON READER SERVICE CARD

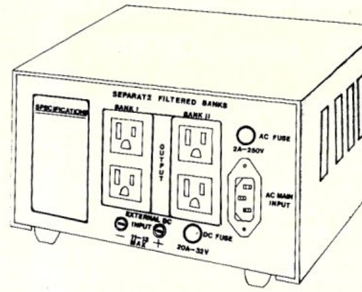
Line-Saver uninterruptible power system

Kalglo Electronics Co. has added a new standby uninterruptible power system (UPS) to its Aegis™ line of power conditioning equipment. Designated the Line-Saver™, it represents a breakthrough in uninterruptible power systems for use in the home and small business market.

The Line-Saver is engineered to give trouble-free standby backup power available in 120/240 V, 60/50 Hz with 250 VA at 0.6 pF with 150 watts capacity. It uses the latest Pulse Width Modulation (PWM) technology to regulate the RMS AC output voltage for greater efficiency to various load conditions. The PWM AC output will also increase battery efficiency to increase backup time: 5-10 min. at full load, 20-25 min. at half load, 35-40 min. at one-third load.

In addition, the unit is furnished with an internal 12 V sealed rechargeable battery, four Spike-Spiker™ voltage-surge protected and EMI/RFI filtered AC outlets, audible and visual power failure warning system, test mode indicator and switch, and replaceable external fuses. It is compact in size, with external 12 VDC





battery connectors to allow for mobility and extended holdup time. The unit comes with a 6′ detachable three-prong heavy-duty grounded cord set with a CEE-22 connector.

Price: $395

Kalglo Electronic Co., Inc., *Dept. LS, 6584 Ruch Rd., E. Allen Twp., Bethlehem, PA 18017; (215) 837-0700.*

CIRCLE 312 ON READER SERVICE CARD

Microsystems Mart

S-100 COLOR
GRAPHICS CONTROLLER BOARD

Using NEC 7220 Graphics Display Controller

One board, single jumper configurable for 8-color mode or monochrome mode. For color, configures to 3 planes of 32K bytes for output to R-G-B TTL monitors. For monochrome to a single large 96K image plane.

For use with standard TTL monitors with separate syncronization; R-G-B or monochrome, 15.75 KHz horiz., 60 Hz vert. A composite output is available on the board for no-interlace monochrome monitors.

Select positive or negative sync - Each bit (pixel) is individually addressable - Built in algorithms for arcs, lines, rectangles, area fill - Light pen input - 1x to 16x zoom - Occupies two contiguous addresses on the system I/O bus - Interlace or non-interlace modes.

Programmable display window: 640x408 interlaced color, or 512x512 interlaced color, or 640x240 non-interlaced color and monochrome, or 640x480 interlaced monochrome, and others.

Using pan techniques enables viewing image areas larger than the display window most likely in non-interlace or monochrome applications, or when zoomed.

$595 This is an extremely versatile, cost effective graphics controller for personal, home, and small business computers.

$460 for partially stuffed board with less memory for monochrome - upgrade later.

PIXELTRONICS

1050 Tulip Way
Palatine, Illinois, 60074
(312) 359-1442

BOBCAT

THE FINEST CP/M CATALOGUING SYSTEM YET!

only $25.00

- creates, adds, updates, and deletes a filename catalog
- provides four report formats and three search routines
- numbers disks and provides titling in catalog
- prints disk contents for disk jacket
- 24 page MICROGUIDE* documentation
- 8" CP/M SSSD or Popular 5¼"

REQUIREMENTS
- CP/M 1.4 and up
- 56K and up memory
- two or more drives

U.S. residents	$25.00 U.S.
Canadian residents ..	$25.00 Cdn (Ont. residents add $1.75 pst)
Other countries	$30.00 U.S. plus $2 P & H

NAME, NUMBER, EXP DATE
(MC FOUR EXTRA DIGITS)

Bank drafts; cert. checks; money orders; company checks

R&L Micro Consulting Services
Box 15955, Station F
Ottawa, Ont.
K2C 3S8 (613) 225-7904 THE HOME OF THE BOBCAT

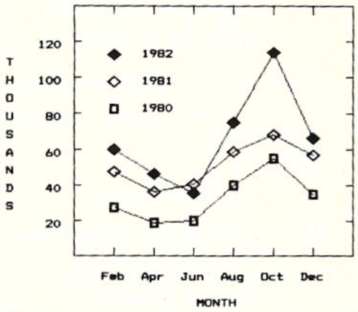
*MICROGUIDE is a (tm) or R & L Micro Consult. Svs. CP/M is a tm of Digital Research.

CIRCLE 249 ON READER SERVICE CARD

PUBLISHABLE QUALITY GRAPHS on MOST DOT MATRIX PRINTERS

Requires no Graphics Screen
Line Graphs and Scatterplots
Easy to Use, Excellent Manual
CP/M 80, CP/M 86
Most disk formats.
Price: $50. Manual Alone: $10.

DataPlotter from **Lark Software**
7 Cedars Rd, Caldwell, N.J. 07006
(201) 226-7552 Visa, M/C



CIRCLE 250 ON READER SERVICE CARD

CP/M.D.

Diagnose and cure disk problems quickly and easily. Recover erased files, retrieve "bad sector" files, lock out bad sectors. CP/M.D. is a "doall" CP/M utility with dozens of menu-driven fast-acting functions. Requires 2 drives, 32K, CP/M 1 or 2.

SIMPL-SIMON SOFTWARE **$29.95**

Add $2.25 per order for shipping. Send check for prompt shipment. Ga. residents add 4% sales tax. Specify CP/M 8", Superbrain, or Xerox 5.25". CP/M is a trademark of Digital Research, Inc.

Write for information about CP/M database system, $89.95; word processing, $88.90; other inexpensive CP/M software.

TELEPRINT, INC.
PO Box 10B, Sylvania, Ga. 30467

CIRCLE 251 ON READER SERVICE CARD

CCS BRINGS YOU 6 UTILITY PROGRAMS TO WRITE TO YOUR COMPUTER ABOUT!

All 6 Programs for Only $75.00

Randit© - a random file editor
- edits random files (files with holes)
- edits files up to a full 8 megabytes

Hextran/Hexrecv™© Transposes binary files
- You can now download or upload executable files on to any communications network (i.e. The Source® Compuserve®

Hexit/UnHexit- Converts binary files to an editable form
- Now easily make patches to programs and convert back to an executable form

Cactus Computer Systems Inc.
2765 S. Highland, Suite #113
Las Vegas, N.V. 89109

Specify 8" single density or Kaypro 4/10 format
Please allow 3-6 weeks for delivery.

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 252 ON READER SERVICE CARD

NEED MORE $$$

If you make your living programming, we can make you richer!

The Programmer's Apprentice is a professional program development system that generates fully debugged MBasic source code with:

- B-Tree File Management
- Multi-level Menus
- Report and Program generators

Library driven, so you can modify it to write code to your specifications.

Contact: THE SOFTWARE GROUP
10471 Brookhurst
Anaheim, CA 92804
(714) 535-5274

ICs PROMPT DELIVERY!!
SAME DAY SHIPPING (USUALLY)

DYNAMIC RAM		
64K	200 ns	$ 5.97
64K	150 ns	6.09
64K	120 ns	6.95
16K	200 ns	1.56
EPROM		
27128	300 ns	$19.20
2764	250 ns	5.99
2732	450 ns	4.25
2716	450 ns	3.60
2532	450 ns	4.75
STATIC RAM		
6264P-15	150 ns	$38.00
6116P-3	150 ns	5.62

Prices 11 15 83

MasterCard VISA or UPS CASH COD
Factory New, Prime Parts μPco
MICROPROCESSORS UNLIMITED
24,000 South Peoria Ave.
BEGGS, OK. 74421
(918) 267-4961

Prices subject to change. Please expect higher prices on some parts due to world wide shortages. Call for volume prices. Subject to available quantities. Shipping & Insurance extra. Cash discount prices shown. Federal Express Standard Air @ $5.99! Orders received by 6 PM CST can be delivered to you by the next morning. We try harder.

SALES POWER

Sync not only sells itself at a nice profit, it helps sell the hardware and software in your store too. Because it goes home with customers and continues to influence buying decisions long after you're closed for the day. Let's talk now.

CALL COLLECT:
(212) 725-7679
Or write:
ZD Ziff-Davis Publishing
One Park Avenue
New York, NY 10016

Minimum order, 10 copies. We pay all shipping costs.

classified ads

CLASSIFIED RATES: Per Word, 15 Word Minimum. REGULAR: $1.35. EXPAND-AD: $2.03. 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 (212) 725-3927.

SOFTWARE

MCL—a music compiler. Runs Digital Research (of Texas) Stereo Sound Boards. Easy sheet music translation, sound effects, more. Requires 48k-TPA, CP/M-2.2. $25.00. 8-inch SSSD. Howard Peters, Box 3324, Rockford, IL 61106.

SPEAKER SYSTEM DESIGNERS! Analyze your woofer/cabinet/port/horn combinations and get response curves on your printer. CP/M diskette 8 inch SS/SD or 5 inch SS/SD for Kaypro, Osborne, Morrows, or Xerox. $39.00 diskette and booklet includes UPS 48 states. ($10.00 Foreign shipping). Prepay with check, Visa, MasterCard, or American Express. SONIX CO. INC., Rt. 2 Box 163-G, Indian Head, MD 20640 (301) 753-6432.

STRUCTURED ASSEMBLY LANGUAGES SAL/80 and SAL/86 do for assembler what RATFOR does for FORTRAN. Improves productivity by factor of two, maintainability by order of magnitude. Extensively documented, available all CP/M formats. $59.00, requires MAC/RMAC and 64K RAM. California residents add 6%. PROTOOLS® "Software Tools for the Professional", 24225 Summerhill Avenue, Los Altos, CA 94022. (415) 948-8007.

COMPUTER EQUIPMENT/SUPPLIES

FREE! Computer & Supplies Catalog—low prices—satisfaction guaranteed—DATA SYSTEMS, Box 99, Fern Park, FL 32730. (305) 788-2145.

COMPUTER PAPER—Save $$$; Top quality. Low single carton prices. Free samples. Shipping via UPS. Call A-1, (800) 628-8736 or (213) 804-1270.

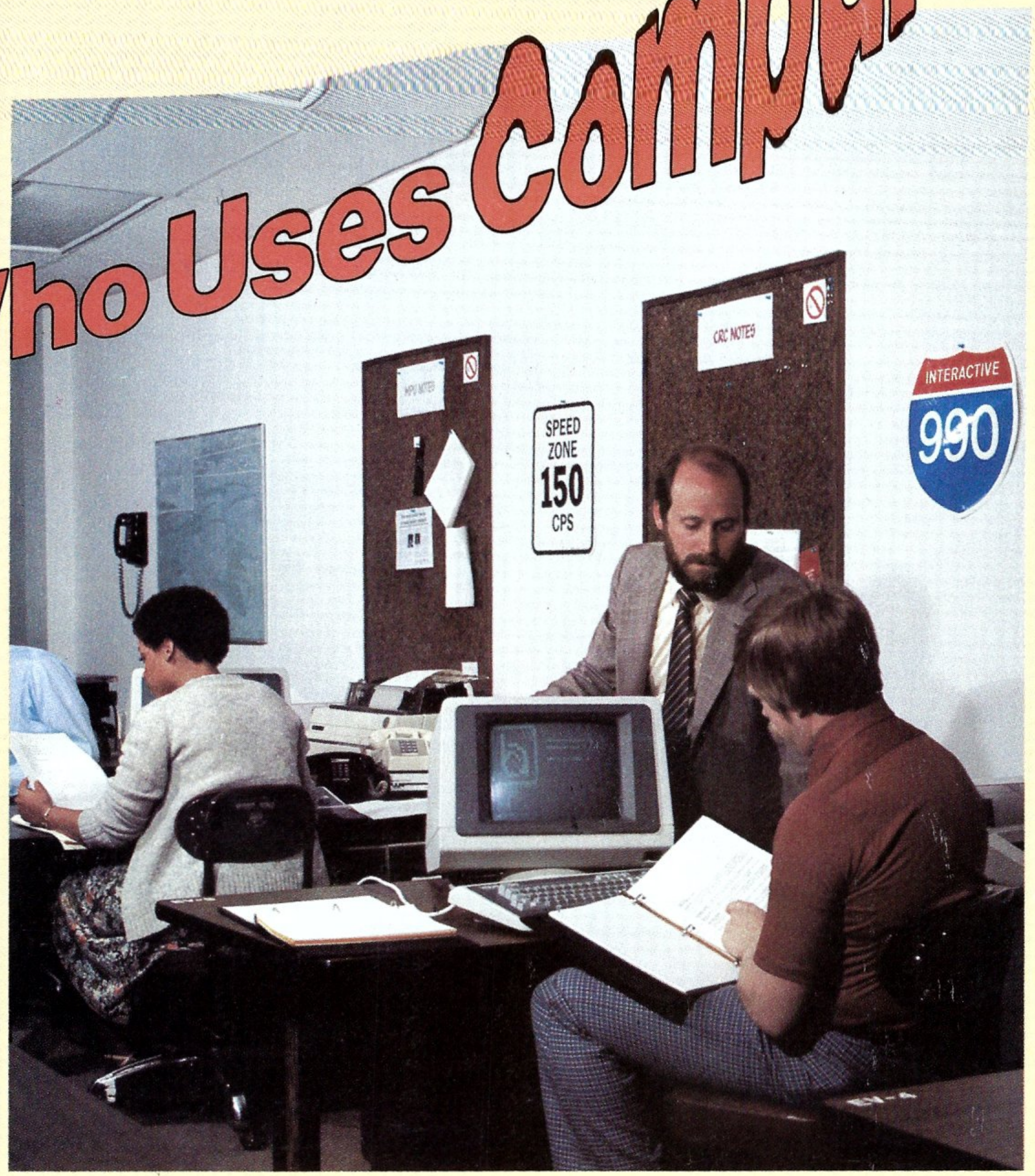
REPLACEMENT RIBBONS for computer printers and word processors. Fantastic savings! Thousands in stock. Quick delivery. Call or write: 1-(800) 292-6272. National Computer Ribbons Corp., 1114 Elbank Ave., Baltimore, MD 21239.

FOR SALE

WATCHES: REPLICAS OF WORLD FAMOUS watches: Rolex, Piaget, Concord, etc. Please send $3.00 (refundable with order) for color brochure. GLOBAL IMPORTS 1301 E. Oakland Park Blvd. Dept. BT, Ft. Lauderdale, FL 33334 or (305) 564-6557.

ADVERTISERS INDEX

Who Uses CompuPro?



Barton-Aschman Associates, Inc. is based near Chicago but its influence is felt on pavements across the nation. This traffic engineering firm uses eight **CompuPro System 816**™ computers to help keep millions of motorists moving smoothly.

Senior associate Dennis Strong manages the firm's interstate computer operation in addition to his traffic engineering responsibilities. He decided to use **CompuPro**'s sophisticated microcomputers to replace a time-sharing service after a three-week mainframe breakdown that paralyzed Barton-Aschman's dozens of consultants.

"The **CompuPro** hardware is solid," he says. "We haven't had a board failure for over a year."

CompuPro's advanced S-100 architecture and CP/M®-based software appealed to Mr. Strong, who is determined to keep his computers modular and standardized for ease of upgrading and maintenance.

For motorist management or manager enhancement, **CompuPro** delivers performance, quality and reliability. Call **(415) 786-0909** ext. 206 for the location of the **Full Service CompuPro System Center** nearest you.

Barton-Aschman's **CompuPro** systems were integrated by Lillipute Computer Mart of Skokie, Illinois.

CompuPro®