THE JOURNAL FOR ADVANCED MICROCOMPUTING

$2.95

Microsystems

Volume 5/Number 3

March, 1984

MS-DOS
Overview, Part 1
MSPRO: MS-DOS on S-100

CP/M 2.2
Modifying Fortran-80
Mass Renaming

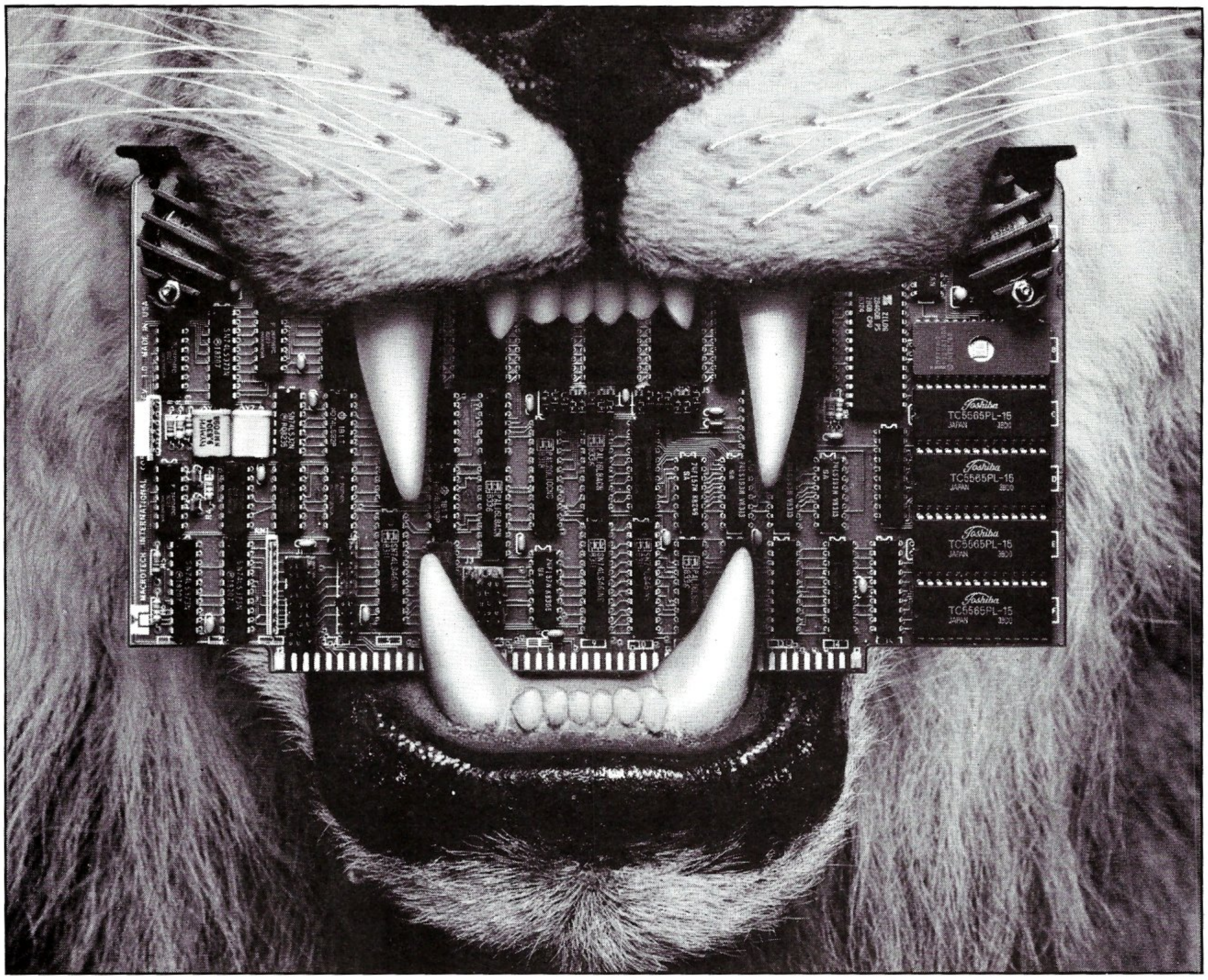
MP/M II
MP/M, Part 2
Date & Time Functions

Hardware Reviews
Lomas LDP-2
Dual Systems SIO4/DMA

Software Reviews
Magic Bind
SAL/80

MS-DOS- A new support for files and peripherals

0 14024 14352 03



KING OF THE S-100 JUNGLE

You'll reign supreme when your S-100 system is equipped with Macrotech's high performance products. Our full megabyte high speed dynamic RAM and 16 channel intelligent DMA I/O boards provide the edge you need in today's competitive jungle.

Our one slot MAX dynamic RAM is field expandable from 256K to one megabyte. You get far more than just the low price per kilobyte of $2.39. You get:

• IEEE 696 full parity detection • Virtual disk support in all popular Digital Research operating systems • High Speed Z80, 8085—6 MHZ; 8086, 8088, 68,000—8 MHZ No Wait States
• On-board refresh • 16-24 bit Memory Mapped Addressing option
• DMA fully supported in strict compliance with IEEE 696

Our ADIT 16-channel serial I/O board is no pussycat either. An on-board 6 MHZ Z80B provides intelligence for a large array of commands. Plus important things like:

• Resident virtual disk interface • Memory to memory DMA
• 24- bit DMA addressing • Field expandable from 4 to 16 channels
• Full compliance with IEEE 696

Macrotech makes every slot in your computer really count! To get the rest of our story on the ADIT or the MAX, write or call us. In the S-100 jungle Macrotech products can be your key to success.

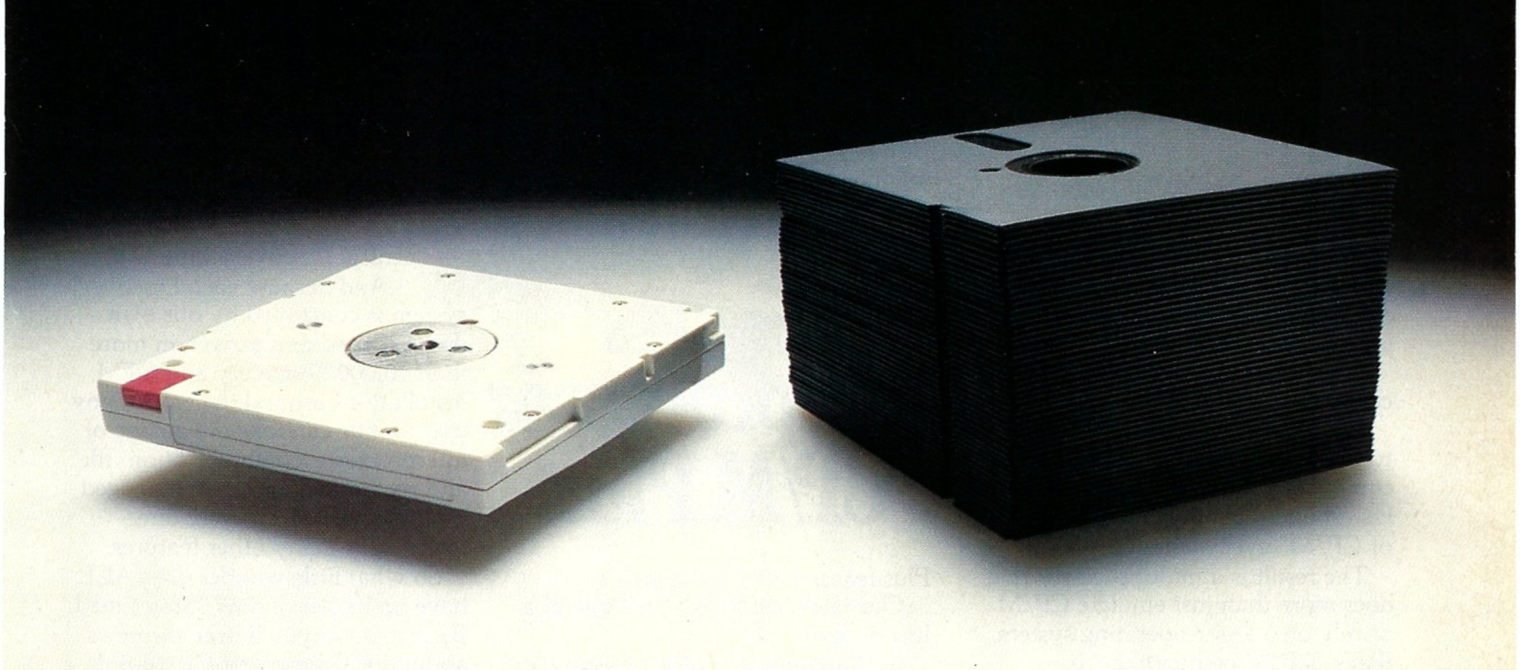
MACROTECH INTERNATIONAL CORP.
9551 Irondale Ave., Chatsworth, CA 91311, (213) 700-1501

Dealer/Distributors: Priority One Electronics (800) 423-5922, (213) 709-5111
Soft Machines (217) 351-7199; In England: Fulcrum (Europe) (0621) 828-763

CIRCLE 74 ON READER SERVICE CARD

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

Score another first for ACE.

Multiprocessor CP/M Plus.™

A lot of people who've never heard of ACE's impressive record of firsts in multiprocessing are about to sit up and take notice. Why?

It's our latest wonder: Multiprocessor CP/M Plus*.

ACE has developed a linkup between its powerful multiuser microcomputer, the Discovery, and the fastest and most powerful version of CP/M* ever marketed.

The result is a multiprocessor that does more than just emulate CP/M or rely on a lesser operating system that's "CP/M compatible." It actually delivers pure CP/M (either CP/M-86* or CP/M Plus) to any of Discovery's up to 16 on-line users.

With easy command line editing, quick file accessing through user-defined automatic search paths and comprehensive "help menus," CP/M Plus is as friendly as they come — and with big performance extras.

Like large files, directory hashing, data buffering, time and date stamping, and an extensive utility set that can accept English words.

All of which means that ACE's linkup with CP/M Plus will continue to give Discovery owners access to the largest selection of applications software in the world for years to come.

That's something to think about. Because who wants a multiuser system that's friendly and powerful today but unfriendly tomorrow when new software using CP/M Plus features hits the scene?

Our latest triumph is just one of a line of firsts.

ACE pioneered multiprocessing for microcomputers back in 1979 when we introduced a Discovery featuring what was then the revolutionary concept of dedicating a CPU to each on-line user.

Two years later, we moved further ahead of the pack with our next singular feat: the first user-processor on a single board, the dpc-180*. Everything was there: memory, CPU, serial I/O.

Nice.

Which made our 1982 breakthrough — the amazing 16-bit dpc-186*, mated with CP/M-86 — seem like business-as-usual to us even though it caught the competition flat-footed.

With room to grow from 128K to 1MB of expansion memory, and designed for systems also using our 8-bit board, this breakthrough let individual users tap into either 8- or 16-bit computing power with one central Discovery unit.

And because each Discovery system is controlled by our own dpc/os*, tried and proven in more than 10,000 Discovery and OEM installations around the world, any user on line can take advantage of interprocessor communication, file and record locking, fully managed multiple printers, private directories, plus many other features.

So what little wonder does ACE have up its sleeve now? Stay tuned. Because soon you'll hear the news about our Concurrent CP/M-86*, which will give the owners of our Discovery all the benefits of 16-bit computing power, CP/M's comprehensive software selection, and efficient multi-tasking capabilities. So each user can get several projects underway with a few quick taps on the keyboard.

Meanwhile, it'll be business as usual for ACE, keeping ahead of the crowd while turning out the best multiuser multiprocessor micro on the market, backed by one of the best nationwide service networks you could hope for.

So if you're a computer dealer, distributor, or systems house who wants to join the group that scores high with a multitude of users, just give us a call at **(800) 821-6596.** (In California, it's **(213) 351-5451.)**

And do us a small favor. Tell us you've heard of us.

CIRCLE 197 ON READER SERVICE CARD

ACTIONCOMPUTERENTERPRISE

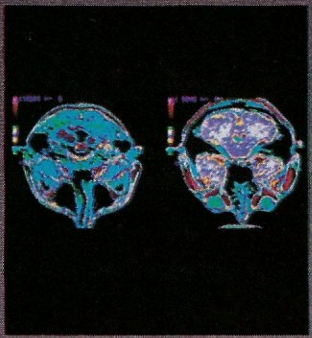
The Multiprocessing Company.

Action Computer Enterprise, Inc. (Corporate Headquarters): 430 No. Halstead St., Pasadena, CA 91107. TWX 910-588-1201 ACTION PSD
ACE/Europe: Boschdijk 189, Box 1275, 5602 BG, Eindhoven, Netherlands, Tel. 040-452658, Telex 51767 ACE E NL **ACE/Asia,** G/Floor Lee Wah Mansion, 171-177 Hollywood Rd, Hong Kong, Tel. 5-441692 or 5-442310, Telex 75332 PACIC HX **Canada:** Future Electronics, Inc., 237 Hymus Blvd., Pt. Claire, Quebec H9R5C7, Canada, Tel. (514) 694-7710, Telex 05-823554.

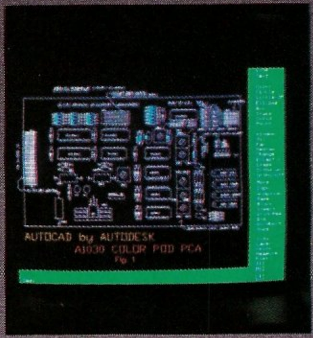
*CP/M Plus, CP/M, CP/M-86, and Concurrent CP/M-86 are either trademarks or registered trademarks of Digital Research, Inc. / *dpc-180 and dpc / os are registered trademarks of Action Computer Enterprise, Inc.

Serviced nationwide by Bell & Howell Company

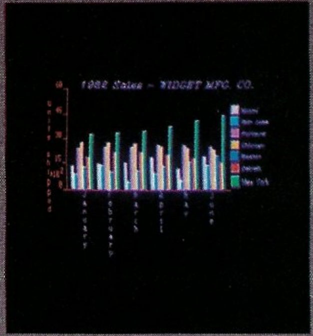
GDL's A-1000™
Why Pay More For Less?



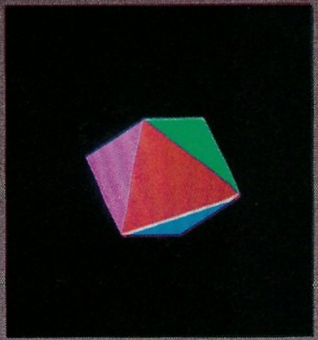
N.M.R. Brain Scan
J. Libove, Dual Systems



Autocad by Autodesk
R. H. Hymes, GDL

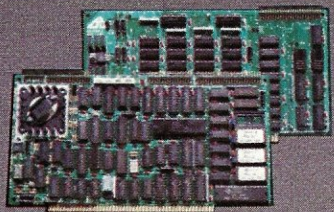


Supercalc Data Graph
Pacific Basin Graphics

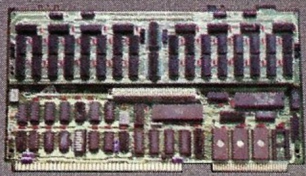


Octehedron With Hidden
Surfaces Removed
Dr. C.L. Morgan,
Calif. S.U. Hayward

- 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.**

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

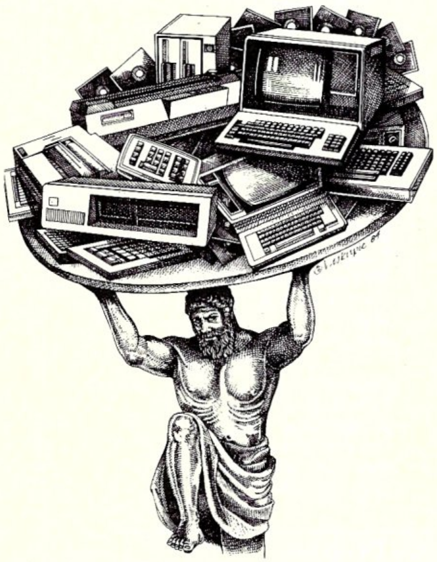
MOMS COMPUTING
Bldg. 1055 Fort Cronkite
Sausalito, Ca. 94965
(415) 331-2043

GRAPHICS DEVELOPMENT LABORATORIES
2832 Ninth St., Berkeley Ca. 94710
(415) 644-3551
CIRCLE 24 ON READER SERVICE CARD

Microsystems

Volume 5/Number 3
March 1984

March



MS-DOS-

A new support for files and peripherals.

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
Ilisa Goodheartz/James Buklarewicz/Deborah Barnett	art assistants
Jim Beloff	advertising director

ADVERTISING SALES OFFICES

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

Advertising Coordinator
Michele Fischetti, *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, Deanna D'Zamba
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* Paul H. Chook, *Executive Vice President, Marketing and Circulation* Phillip T. Heffernan, *Senior Vice President* Sidney Holtz, *Senior Vice President* Edward D. Muhlfeld, *Senior Vice President* Philip Sine, *Senior Vice President* Baird Davis, *Vice President* George Morrissey, *Vice President* Rory Parisi, William L. Phillips, *Vice President, Vice President* Selwyn Taubman, *Treasurer* Bertram A. Abrams, *Secretary*

MICROSYSTEMS (ISSN #0199-7955) is published monthly by Ziff-Davis Publishing Company, 3460 Wilshire Blvd., Los Angeles, CA 90010.

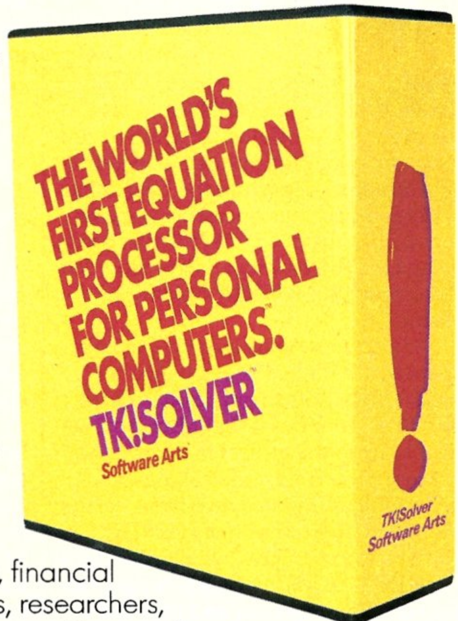
Second Class postage paid at Los Angeles, CA 90052 and at additional mailing offices. POSTMASTER: Send address changes to MICROSYSTEMS, PO Box 1987, Morristown, NJ 07960. Subscriptions are $26.97 for 12 issues. Canadian prices are $5.00 per year additional; other foreign $8.00 per year additional (U.S. 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.

EQUATIONS PROCESSED

NO PENCIL. NO PAPER. NO MANUAL LABOR.



The TK!Solver® program will take on your toughest problems—linear, quadratic, simultaneous equations, whatever. Then stand back. Because TK!Solver turns your personal computer into a simple, yet powerful, desktop equation processor.

Whether your problem is a simple formula or a model consisting of many equations, TK!Solver can help improve your productivity. Once the equations are written, enter the known values, press the ! key, and TK!Solver gives you the answer.

Engineers, scientists, architects, financial analysts and planners, educators, researchers, and other professionals who use equations and mathematical models can work more creatively with TK!Solver.

TK!SOLVER GIVES YOU:

BACKSOLVING

If the programs you use now require you to rewrite the same equation to solve for different unknowns, TK!Solver can dramatically improve your productivity. Enter your problem once and then solve for the unknowns no matter where they are in your equation.

ITERATIVE SOLVING

If TK!Solver can't solve an equation directly, take an educated guess at the answer. Type the ! key and the TK!Solver program starts with your guess and performs repeated approximations to converge on the answer.

LIST SOLVING

Given a list of input values, TK!Solver automatically calculates the equation for every value in your list. For example, if you want to know how different interest rates will affect monthly loan payments, enter a list of interest rates and let TK!Solver calculate the payment amount for each value.

UNIT CONVERSIONS

Any type of unit conversion—Fahrenheit to Celsius, meters to feet, dollars to deutschemarks, newtons to dynes—can be made without altering your equations. Just define the numerical relationship between two units of measurement and the TK!Solver program automatically converts the variable value to the unit you specify.

TABLES AND PLOTS

Quickly generate tables and plots of your results on your screen or printer.

AVAILABLE NOW

You can run the TK!Solver program on the IBM® PC and XT and compatible machines, the Digital™ Professional™ 350, the Digital™ Rainbow™ 100, the Wang Professional Computer, Apple® //e, and on the following personal computers using MS™-DOS: TI Professional Computer, GRiD Compass Computer,™ Canon AS-100, Eagle® 1600, Toshiba T300, and the Zenith Z-100.™

SEE IT TODAY

There's more. Lot's more. But you'll have to see it to believe it. And that's easy. Bring your own equations into your nearest computer retailer and ask to see the TK!Solver program in action. The world's first equation processor for personal computers.

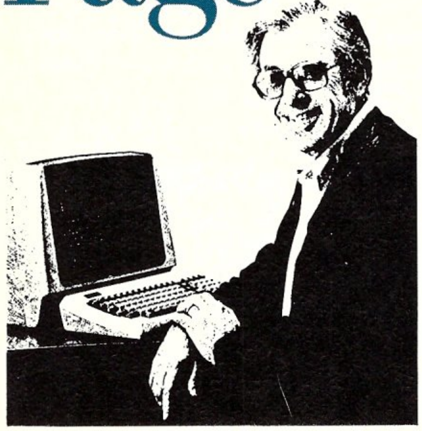
TK!Solver®

By Software Arts,™ creators of VisiCalc®

27 Mica Lane, Wellesley, Massachusetts 02181 617-237-4000

CIRCLE 9 ON READER SERVICE CARD

Editor's Page



Digital Research takes on UNIX

by Sol Libes

At the recent Comdex show Intel and Digital Research announced that they have signed a contract under which DRI is to port UNIX System V to the Intel 80286 and also provide a line of related software to run under the operating system. This represents a radical departure for DRI, since all of their previous operating systems were derived from CP/M. Further, only a year ago DRI had stated they intended not to bring out a UNIX operating system.

It also represents a radical change for Intel, which had previously contracted Microsoft to port System V to the 286. Rumor has it that Microsoft decided to drop the project in favor of other, more profitable undertakings. Microsoft has invested heavily in the development and refining of XENIX, their version of UNIX, which is basically UNIX V7 with the Berkeley and System III enhancements, and they have done more to popularize UNIX on microcomputers than anyone else has. Under the terms of the contract, Western Electric will own the operating system, and both Intel and DRI will market and support the software.

It is reported that the XENIX project has been only marginally profitable for Microsoft, and that the System V implementation turned out to be much more work than they had anticipated. Further, it is likely that Microsoft was not happy with Western Electric's insistence that the port adhere strictly to the System V standard they had developed.

Thus Intel was left hanging and turned to DRI to get them out of the hole. With DRI's extensive line of languages and software development tools, they represented the best alternative to Microsoft. Both Intel and Western Electric scrutinized DRI's capabilities before the contract was signed.

Although DRI does its software development on a UNIX system, they lack any experience in the UNIX area, compared to several other software houses (e.g., UniSoft and Human Computing Resources) and it should therefore be very interesting to see how long it takes them to get this implementation out. There are rumors that DRI may acquire a small software house experienced in porting UNIX in order to get the expertise in-house quickly.

It is interesting to note that at the last NCC show in May, Western Electric, Intel, Motorola and National Semiconductor held a joint press conference to announce their commitment to port UNIX System V to the 286, 68000 and 16032 microprocessors. In the nine months that have gone by, no one has announced such a product, and we now learn that Intel has changed suppliers and is probably still more than six months away from releasing a product. Interestingly enough, UniSoft and Human Computing Resources have announced System V ports for the 68000, and HCR also has one for the 16032.

DRI has extensive experience with the C language, with implementations for 8086 and 68000 products. It is likely that after finishing the 286 System V implementation they will do a 68000 implementation. After all, Motorola has yet to bring out such an implementation, despite their contract with Western Electric.

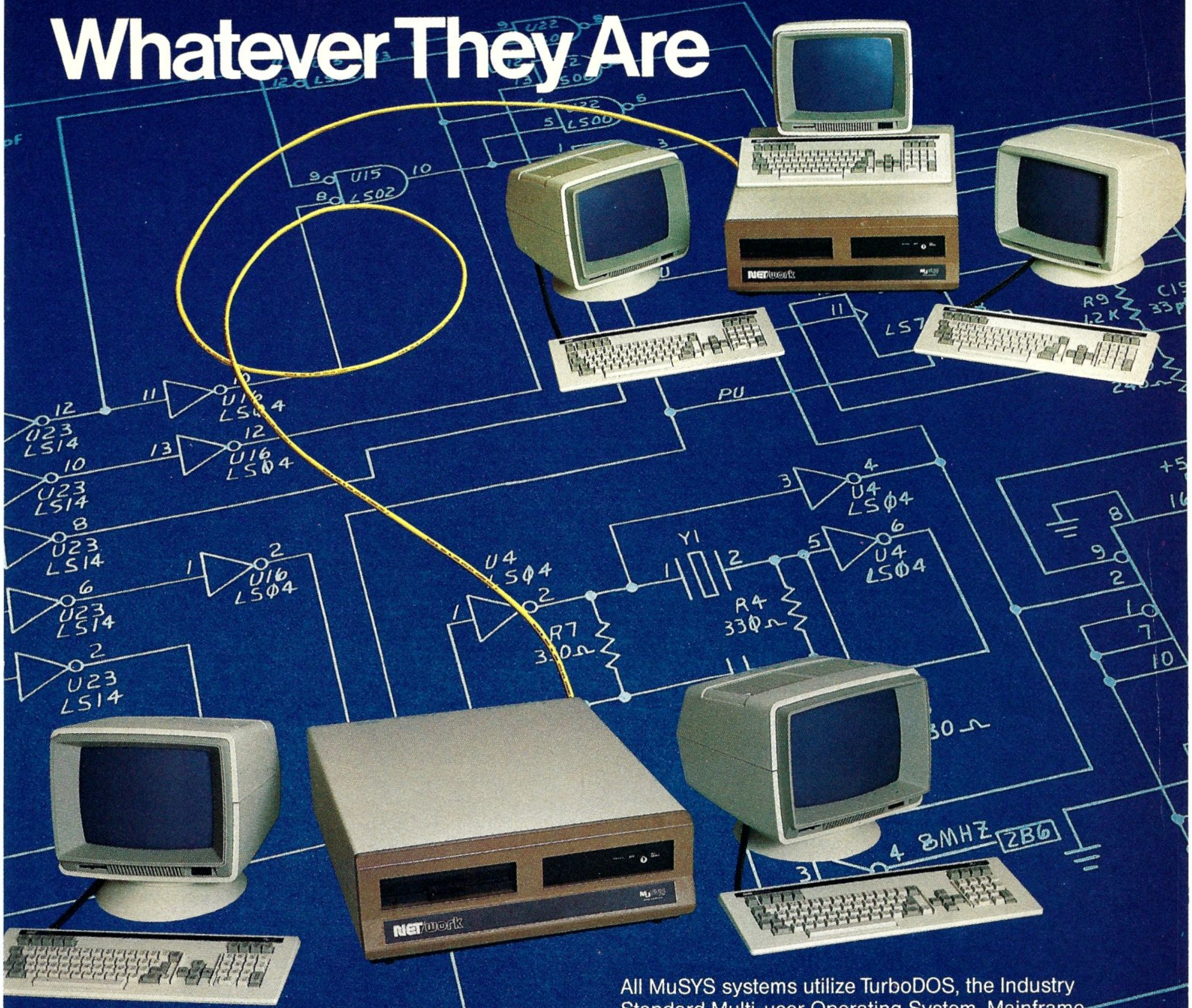
DRI is apparently attempting to become *the* vendor for the standard System V UNIX for the 286 and possibly for other microprocessors such as the 68000. However, there are questions as to how successful this effort will be. After all, Microsoft and UniSoft have virtually all of the current OEM market sewed up, and there is little likelihood that these OEMs will change. Further, Microsoft introduced an improved performance version of XENIX for the 286 back in May, and Bill Gates has indicated that Microsoft will do their own implementation of System V. Microsoft has proved to be a very aggressive competitor. It is probable that their version will not be a strict Western Electric System implementation, but will most likely include the enhancements that Microsoft added to XENIX to make it more commercially marketable (e.g., improved user interface).

IBM is rumored to be developing their own operating system for their upcoming generation of 286-based workstations. Further, Western Electric's UNIX-based systems are due shortly, and they are not likely to use DRI's implementation. However, the market for languages and software development tools for these IBM and Western Electric systems could prove lucrative.

It is likely that DRI will offer a utility for their UNIX System V implementation that will allow the running or porting of standard CP/M software on these systems. Considering the wealth of CP/M software currently on the market and in the public domain, this would be a definite plus for DRI and for the UNIX community.

It is estimated that about 85,000 UNIX systems were sold last year (of which about two thirds were XENIX). This was the first year in which large

It Meets Your Needs...
Whatever They Are



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.

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.

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.

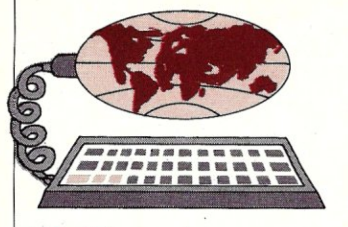
EDITOR'S PAGE

Continued from page 8

numbers of UNIX systems were sold. This certainly does not compare to the sizes of the CP/M and MS-DOS system markets, which are both about a million licenses each. It is estimated that the UNIX market will probably double this year and by the end of next year rise to well over 300,000 systems, particularly as IBM and Western Electric move into the marketplace.

The DRI version of UNIX now presents another area in which Microsoft and DRI are competing head-to-head. Considering Microsoft's previous successes in these battles, one can question the wisdom of DRI's decision to open up the war on yet another front. To a certain extent DRI may be forming an alliance with AT&T comparable to Microsoft's alliance with IBM.

For an in-depth discussion of Digital Research's entry into the UNIX marketplace and strategies being used by Microsoft, Intel, IBM and Western Electric, I refer the reader to an excellent monthly newsletter published by Yates Ventures, 4962 El Camino Real, Suite 111, Los Altos CA 94022, (415) 964-0130. The newsletter is titled *The Yates Perspective* and a yearly subscription is $450.



News & Views

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

by Sol Libes

Commodore is rumored to be negotiating with Mark Williams Company, Chicago IL to use the latter's UNIX-like Coherent operating system on Commodore's Z8000-based 16-bit microcomputer, now in development. The minimum system is expected to include 128K of RAM and a 320K disk drive and sell for under $1,000. An 8088 plug-in card option is also expected, with versions of MS-DOS and CP/M-86 to run on it. . . . Commodore is also rumored to be readying a portable computer employing a 16-line by 80-character liquid crystal display and wafer-tape drive. . . . Texas Instruments is rumored to be working on a 68000 processor card for the desktop IBM PC-compatible system. TI is expected to include a UNIX-type operating system with the card.

New public domain software

SIG/M (Special Interest Group for Microcomputers, Amateur Computer Group of New Jersey, Inc.) has issued one new volume of public domain software, bringing their total up to 152 volumes. The new volume contains a disk drive alignment program for the 1793 FDC; it does not require an oscilloscope, but you do have to purchase a $30 test disk from Dysan. Also on the disk are updates for the DU, LDIR, SWEEP, TYPE and USQ utilities.

For complete information on SIG/M software, send $2.50 ($4 foreign) for printed catalog to: SIG/M, Box 97, Iselin, NJ 08830.

P/D Software, 4691 Dundas St. West, Islington, Ontario, Canada M9A 1A7 (416/239-2835) is distributing CPMUG and SIG/M software libraries in 25 different disk formats including 8″, Apple, KayPro, Osborne, IBM and others. The software is available in prices ranging from $10-$20, depending on the disks required.

The New York Amateur Computer Club, Box 106, Church Street Station, NY, NY 10008, has released two 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:

Volume	Description
43	dBASE II order entry & inventory control system (converted from SIG/M library)
44	Demographics display system for IBM-PC

DRI announces CP/M-86 version 3.1

Digital Research has announced that it expects to start shipping an enhanced version of CP/M-86 this month. The new version, designated 3.1, will add window-like structures to terminal screens. Further, it will be capable of running MS-DOS and IBM PC-DOS based software. The question now is: will Microsoft add the capability to run CP/M-86 software to MS-DOS and PC-DOS? The Microsoft MSX operating system for low-cost Z80-based home computers can run CP/M-80 software.

3M announces new magnetic medium

3M Corporation has announced a new, low-cost magnetic medium that they claim will allow "semi-rigid" 5.25″ disks capable of storing 5 MB per side (10 MB total). The disks are expected to retail for about $10 in quantity and, like floppies, do not require a sealed environment. Commercial availability is still expected to be a year away.

It should be pointed out that Kodak's Spin Physics Group had previously announced a product with similar capabilities. However, the cost is expected to be higher.

In any event, we can expect that 1985 will see systems with removable-media disk drives storing 5 and 10 MB.

S-100 product directory in works

The annual *Microsystems* directory of IEEE-696/S-100 products is now being put together for publication in the May issue. In May of last year this directory listed 157 suppliers of approximately 500 IEEE-696/S-100 products. We have already discovered 33 new suppliers who have entered the marketplace and are not aware of any that have forsaken it. This is an increase of over 20% and it is likely that the number of products has increased by the same percentage, so that there are probably about 600 IEEE-696/S-100 products now being manufactured. This indicates that the S-100 marketplace is still active and growing.

If you are a supplier of such products and have not received a directory questionnaire form from us, please call us immediately and we will send one out to you.

Intel announces improved iAPX432

Intel has announced an improved version of its 32-bit iAPX432 microcomputer chip set. The chip set, which represents a radical departure in CPU architecture from Intel's 16-bit devices, has never gained market acceptance. There have been reports of disappointing performance, and no major computer manufacturer has adopted the devices.

SIMPLE.

Instant networking, independent of computer and peripheral brands.

Creating a micro and mini computer network has been costly and confusing. Communications hierarchy has prevented many types and various kinds of equipment from sharing information. Until now.

Introducing StationMate. One unit that joins mini computers, micro computers, terminals, and printers. It's transparent. So different brands and types of computers and peripherals can easily interface. StationMate allows you to construct a comprehensive communications network. *Inexpensively.*

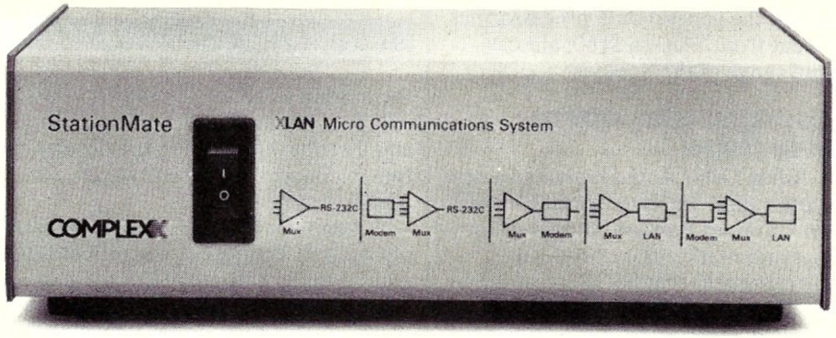
Users can select the information they need from the unit they want. StationMate can address any port in the network either by its assigned digital code or by its common identifying name. So everyone has access to all the local or remote computers, mass storage files, and peripheral devices connected in the network. *Immediately.*

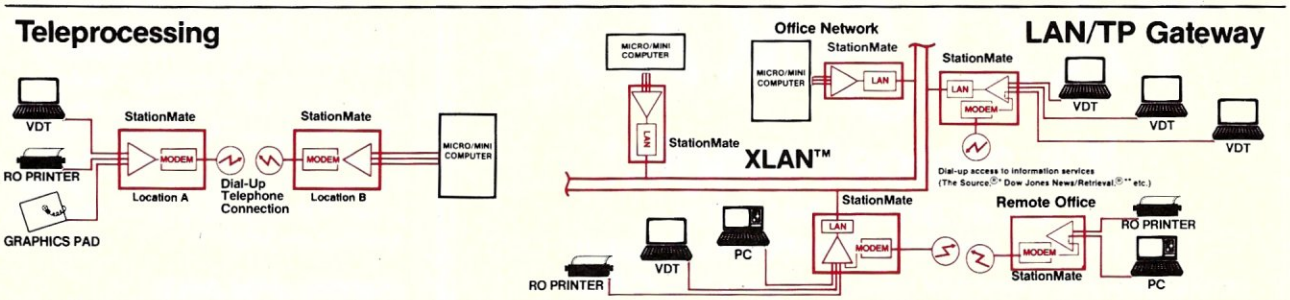
StationMate provides local area networking with XLAN™, the simple communications concept from Complexx. All you need is StationMate and an inexpensive shielded-pair cable.

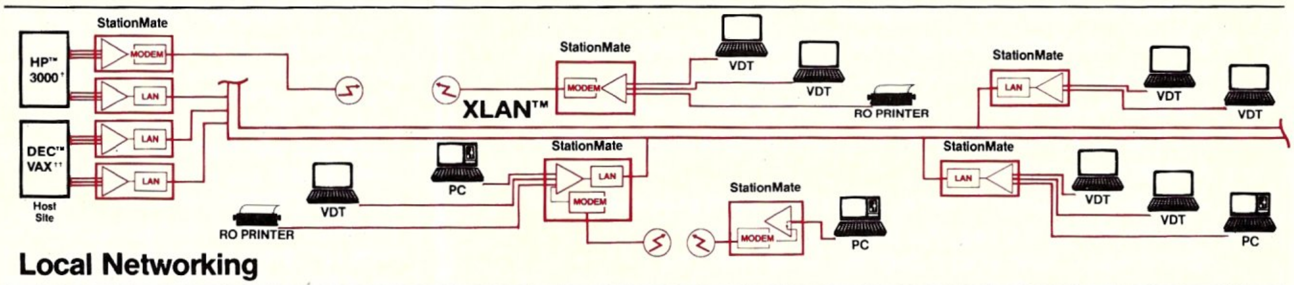
Besides local area networking, StationMate permits access to remote workstations via an internal dial-up modem. And it serves as a gateway for teleprocessing access to all LAN resources. You couldn't get all of this in one unit. Until now. And the suggested retail price is only $1,450.00. StationMate makes communications simple. *Incredibly.*

StationMate. The link you've been missing.

See your local dealer or contact Complexx for the name of the dealer or distributor nearest you. *Complexx Systems, Inc., 4930 Research Drive, Huntsville, AL 35805. 205/830-4310.*







COMPLEXX™

*The Source is a registered trademark of the Source Telecomputing Company, a subsidiary of the Reader's Digest Association. **Dow Jones News/Retrieval is a registered trademark of Dow Jones & Company, Inc. †HP 3000 is a registered trademark of Hewlett Packard. ††DEC VAX is a registered trademark of Digital Equipment Corporation.

CIRCLE 51 ON READER SERVICE CARD

News & Views

Continued from page 12

But Intel has made a commitment to the 432 architecture, which they consider a major step forward for knowledge-based systems. The new release of the device operates at a higher speed and handles instructions more efficiently. Intel claims it is twice as fast as the previous version. Additional enhancements have been made to reduce I/O overhead, and a virtual operating system has been developed that anticipates memory usage and supports 1 trillion bytes of virtual memory. As an additional incentive Intel has reduced the prices of the chips. For example, the price for the general data processor has been cut from $450 to $160; almost within range of the hobbyist.

When you're a little guy you have to pay your own bills, but then you get more successful . . .

Adam Osborne forms software start-up venture

Adam Osborne has launched a new venture called "Software Seed Capital Corp." to fund start-up software companies and market their products. Adam promises that SSCC, of which he is chairman and chief executive, will invest $10 to $15 million in these start-ups and hopes to have 50 to 60 programs on the market in about 12 months. SSCC will do the manufacturing, marketing and distribution, leaving the developers free to concentrate on software development.

Adam is still chairman of the board of Osborne Computer, which is still operating under Chapter 11 of the bankruptcy law. Adam claims that his day-to-day involvement in OC ended last February and "after that I really wasn't certain of what was going on." Further, he has stated that "with the damage done from March to September there's really nothing left of the company I created." And that "there's not much point in me going in to pick up the pieces." He described the OC situation as "a tragedy" that was "totally unnecessary" and that OC "went down the tubes when I brought someone else in to run it." Here he was referring to Robert Jaunich, former president of Consolidated Foods, whom Adam hired to be president of OC.

OC is looking for someone to purchase the company. In the meantime, lawsuits have been filed against Adam Osborne, founder, and Robert Jaunich, president, charging misrepresentation of financial status when selling securities.

Personal CP/M comes to Z80

As rumored previously in this column, a chip has been announced by American Microsystems, Santa Clara CA that places the kernel for the Personal CP/M operating system in the same chip with a Z80 processor. AMI has a second-source agreement with Zilog and expects to start sampling by the end of next month. The chip is expected to be used in low-cost home computer systems.

Random news

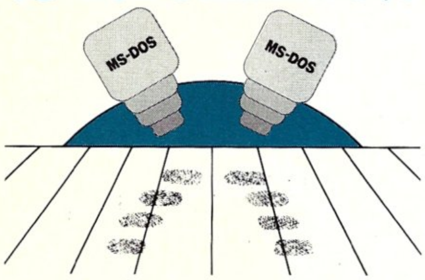
Tandy has finally begun shipping CP/M for its Model IV computer, fully nine months after it first announced it. . . . Wayne Technology, Anaheim CA (714/772-5757) has introduced the "CoCo Coupler One" which makes it possible to run CP/M on the Radio Shack Color Computer. It is inserted between the computer's cartridge port and disk controller.

Quotation of the month

"When you're a little guy, you have to pay all your own bills. But then you get a little more successful, you get a line of credit, and the bank starts paying your bills. Then you get a little more successful, and the bank won't pay anymore, so you pay your bills slower. . . . 30, 60, 120 days late. . . . and now your vendors are paying your bills. Now you're even more successful, you get venture capitalists, and they're paying your bills. And it used to be when you reached the highest level of success, you could go public and have the public paying your bills. But nowadays, when you finally reach the true pinnacle of financial success, like Chrysler, the government starts paying your bills."

George Morrow
President
Morrow Designs, Inc.

From the Sidelines



Subtle differences among the new releases give you more possibilities to choose from

by Hank Kee

The latest releases of PC-DOS are versions 1.10 and 2.1. PC-DOS 1.10 is essentially the same as version 1.1, but has the GRAPHICS option available in version 2.1, along with some differences between 1.10 Basic and 1.00 Basic. The change in designation is very subtle. Most IBM compatibles have only implemented variations of MS-DOS 2.0. Practically all IBM clones have version 1.1 or an equivalent. Most of the popular commodity software systems are currently available on either PC-DOS 1.10 or 2.1, albeit dependent on the IBM screen attributes. Version 2.1 requires greater RAM.

PC-DOS 2.1 is version 2.0 with the added functions of addressing the dynamic color video on the IBM PC*jr*. PC-DOS 1.10 is intended for systems with smaller memory sizes, whereas 2.1, with its requirement for over 24K of RAM, is intended for systems that have larger memory. The paradox in all this is that the PC*jr* can only be used with version 2.1: its memory limit is 128K, the reason being the PC*jr*'s dependence on the relocatable color video.

IBM PC-DOS 1.10 uses a 320K double-sided double-density format. Version 2.1 can reference either 320K or 360K of double-sided and double-density disk storage as well as fixed disks.

Like CP/M, MS-DOS/PC-DOS has implicit or internal commands as well as explicit or external commands. They are similar in function and format to CP/M in many respects. This month we will compare PC-DOS 1.10 to CP/M 2.2. This is strictly an arbitrary selection: they are similar in many respects, but different in others. In CP/M, destination-file is specified before source-file. To rename a file in CP/M, the proper command is:

```
REN (new-name)=(original-name)
```

In MS-DOS/PC-DOS it is just the reverse. An equivalent example is:

Table 1. Internal commands

PC-DOS	CP/M 2.2 equivalent	
COPY	PIP	copies one or more files (*)
DATE	—	displays and/or resets date
DEL	ERA	same as ERASE
DIR	DIR	lists directory
ERASE	ERA	deletes file
PAUSE	—	suspends system processing
REM	—	displays remarks within a batch file
REN(AME)	REN	changes the name of a file
TIME	—	displays and/or resets time
TYPE	TYPE	displays the contents of a file
—	SAVE	memory save

(*) PIP is an external CP/M command.

Table 2. External commands

PC-DOS	CP/M 2.2 equivalent	
CHKDSK	STAT	check disk
COMP	—	compare files
DISKCOMP	—	compare disk
DISKCOPY	—	copy disk
EXE2BIN	—	converts .EXE files to .COM (*)
FORMAT	—	initializes disk
GRAPHICS	—	screen print
MODE	STAT	sets mode of operation
SYS	SYSGEN	transfers operating system
.BAT	SUBMIT	batch file processing
.BAT	XSUB	extended batch processing
—	DUMP	file dump

(*) EXE2BIN converts relocatable binary to absolute binary code.

Unlock the Power of your IBM PC with MDBS III

If you're like most of us, you bought your IBM PC/XT to perform a few simple functions. But with the right software, the PC can do much more.

Mainframe-quality solutions for your PC

With MDBS III and your PC, you can build mainframe-quality application systems. Integrated accounting systems. Order entry. MRP. Job costing. Library management. Banking. Logistics. To name a few.

MDBS III is the most advanced data base management system running under PCDOS, MSDOS, CP/M-86 and MP/M-86. In fact, many have said it's the **only authentic** DBMS available on 8 and 16 bit microcomputers. MDBS III provides many facilities otherwise available only on mainframe DBMSs.

MDBS III allows application developers to define data base structures in the most natural and logical way, without resorting to redundancy to describe data relationships. Its truly innovative data structuring capabilities surpass but also accommodate the older relational, hierarchical and CODASYL-network architectures. That means power for your PC and flexibility for you.

How to get the most from your PC

Get MDBS III and get more of what you need to get the results you demand:

- Post-relational, extended network modeling of real-world data relationships for truly **integrated application systems**
- English-like non-procedural query language for **spur of the moment questions**
- Report generator for quickly specifying **customized reports**
- Automatic guarantees of data and relationship **integrity**
- Recovery and restart capabilities for **physical data protection**
- Active and passive locking down to the record level, supporting up to **127 simultaneous users**
- Redundancy and chaining eliminated, plus full data compression for **optimum efficiency**
- Data independence for **easy maintenance**
- Compatibility with all major programming languages for **flexibility and convenience**
- Encryption and access protection down to the field level for **unparalleled data security**
- Fine-tuning features for **optimizing performance**
- Compatibility with SCREEN MASTER, the PC's most **comprehensive screen management system**
- Interface to KnowledgeMan, the first system to integrate **third generation spreadsheet and relational data management**

So if you want to improve your productivity and your PC's capabilities, get MDBS III.

Call 800-323-3629 today.
Please send me the following:

☐ Set(s) of documentation, including MDBS III manual and Data Base Primer and Guide for $90.00 each plus $10.00 shipping and handling per set.* (Regular price is $100.00)

☐ Set(s) of "How to Evaluate and Select a DBMS" for $5.00 plus $1.00 shipping and handling.*

☐ Professional Training Course Information.

☐ Please have a MDBS/Application Development Products Account Representative contact me.

☐ MDBS III information.

*Prices subject to change without notice. Illinois residents please add 6% sales tax.

Name ______________________

Company ______________________

Address ______________________

City ______________________

State ______________ Zip ________

Phone ( ) ______________________

We accept VISA and MASTERCARD

Card No. ______________________

Expiration Date ______________________

Bank No. if MC ______________________

Signature ______________________

Micro Data Base Systems, Inc.
Application Development Products

85 W. Algonquin Road-Suite 400
Arlington Heights, IL 60005

CIRCLE 44 ON READER SERVICE CARD

IBC HIGH PERFORMANCE MIDDI CADET™	
Maximum Users	10
Disk Storage	40 MB
Disk Access Speed	40 MS
Memory	512 KB
CPU Speed	8 MHz
List Price	$10,995.00

SORRY! NO COMPARISON	
Maximum Users	
Disk Storage	Our competition is now a computor generation behind us.
Disk Access Speed	
Memory	
CPU Speed	
List Price	

The **IBC MIDDI Cadet** is better, faster and less expensive than computers from ALTOS™, ONYX™, DYNABYTE™, and others. And now the new high-performance **MIDDI CADET** is so powerful that it is in a class by itself.

Because the **MIDDI** is completely software compatible with competitive systems using CP/M™ 2.2, MP/M™ II or OASIS™, you can transport your applications software to the **MIDDI** without modification. So why not take the benchmark test yourself.

If you are an OEM, system integrator, multiple end user, or dealer for any of our competitors, send a copy of your application program to **IBC.** We will run your software on the **MIDDI** without modification and give you the elapsed time in minutes. You be the judge. If it really is faster than your current hardware <u>and it is</u>, then you owe it to yourself and your customers to switch to IBC.

So remember! When you want a heavyweight performer at a low price, contact:

OUTSIDE THE USA

IBC/Integrated Business Computers

21592 Marilla Street
Chatsworth, CA 91311
(213) 882-9007 Telex No. 215349

WITHIN THE USA

IBC/ DISTRIBUTION

1140 36th Street, Suite 212
Ogden, Utah 84403
(801) 621-2294

Circle Reader Service number 29 for Dealers
Circle Reader Service number 30 for End-Users

ALTOS is a trademark of ALTOS Computor Systems; ONYX is a trademark of Onyx Systems, Inc.; DYNABYTE is a trademark of Dynabyte Business Computers; CP/M & MP/M are trademarks of Digital Research; and OASIS is a trademark of Phase One Systems.

SIDELINES

Continued from page 16

RENAME (original-name) (new-name)

Switching between similar systems can cause confusion in syntax. It is probably easier to work with two completely different systems.

A comparison of the available commands is given in Tables 1-3.

MS-DOS/PC-DOS 1.0 has a richer total command set than CP/M 2.2.

Hardware-specific CP/M support programs are normally provided on the system distribution disk. These normally include programs to format disks and to do fast disk copying. Some manufacturers include selected commodity software systems in their packages.

SIDELINES

Continued from page 19

Table 3. Supplied systems

PC-DOS	CP/M 2.2 equivalent	
DEBUG	DDT	dynamic debugger
EDLIN	ED	line editor
—	ASM	assembler
—	LOAD	converts .HEX to .COM
—	MOVCPM	generates CP/M sizing
BASIC	—	Basic interpreter
BASICA	—	advanced Basic interpreter
LINK	—	link editor

(*) LINK binds .EXE files.

The total command set of MS-DOS/PC-DOS 1.10 is richer than that of CP/M 2.2. However, MS-DOS/PC-DOS 1.10 and CP/M 2.2 require the lay person to understand the primitive structures of an operating system. MS-DOS/PC-DOS 2.1 and CP/M Plus provide for icon-driven formats, which in turn make them far easier to use. This will be discussed in future issues.

MS-DOS/PC-DOS, as offered, has file attributes different from those of CP/M. Each file under MS-DOS/PC/DOS has associated information in byte size, and date and time stamp of creation or last modification. An example of this is as follows:

```
FORMAT   COM  3816  5-07-82 12:00p
CHKDSK   COM  1720  5-07-82 12:00p
DISKCOPY COM  2008  5-07-82 12:00p
BASICA   COM 16768  5-07-82 12:00p
```

Under CP/M, a STAT command can be used to display information relevant to the number of blocks, size of file, number of extents and file attributes. With CP/M it is possible to display as well as alter information on file status, such as system or directory file, and read/write or read-only file status.

```
Recs   K   Ext  Acc
 14   2K    1   R/O A:BOOT.Z80
 38   6K    1   R/O A:(DDT.COM)
 33   6K    1   R/W A:DUMP.ASM
  4   2K    1   R/W A:(DUMP.COM)
```

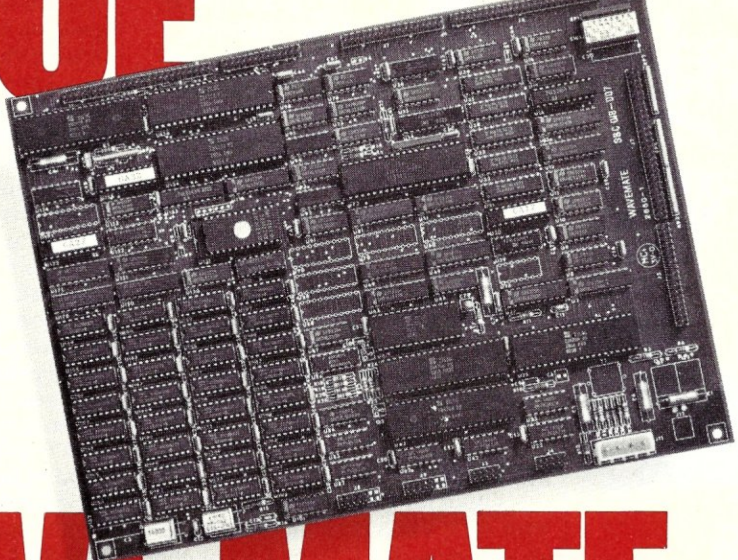
If only it were possible to have the file attribute options of CP/M with the full complement of MS-DOS/PC-DOS commands, including date and time stamp as part of the standard operating system!

Two Basic interpreters are supplied with PC-DOS, as well as demonstration programs written in Basic. These programs do make a good foundation for a better understanding of the Basic functions, and most people prefer them over the availability of an assembler as part of the distribution of system software. CP/M, on the other hand, supplies an assembler along with the sample source programs (.ASM) on their distribution disk.

However, under CP/M, one will find that there is not 100% portability of code between the interpretive MBasic and the BASic COMpiler. Microsoft did not discriminate between systems. This inconsistency exists under MS-DOS/PC-DOS as well. What makes it more difficult to understand is that the Basic compilers for MS-DOS are not compatible with those available under PC-DOS.

Hank Kee, 42-24 Colden St., Flushing, NY 11355

UNBEATABLE VALUE



WAVE MATE SUPER BULLET

8MHz Z-80H Single Board Computer. Outperforms All Micros & Most Minis

DON'T PAY MORE TO GET LESS
You get more for your money with the Wave Mate Super Bullet. Much more than any micro on the market. It outperforms most minis, too. You just can't find a better value. OEMs and systems integraters are finding that they pay *less* for Super Bullet and offer *more* to their customers. This makes better business sense, and bigger margins, too!

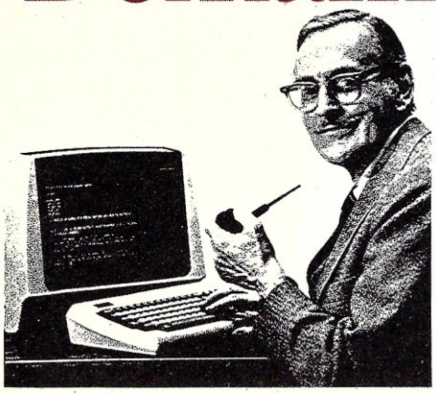
THE ONLY 8MHz Z-80H MACHINE OF ITS KIND
Super Bullet is unique. Wave Mate's exclusive enhancements of the basic Z-80H architecture offers 8MHz operation with flexible DMA facility, enhanced C-BIOS, fully interrupt-driven I/O and high-speed floppy disk controller. The CPU is utilized with a full complement of Zilog-compatible peripheral chips. Never so much sophistication and flexibility on a single board.

POWER/FLEXIBILITY
For years, the Wave Mate Bullet SBC has been recognized as the most sophisticated and cost-effective Z-80A, CPM-based single-board microcomputer on the market. A truly unbeatable price-performance package for the single user. Now, the "Super Bullet" adds a new dimension to Wave Mate's state-of-the-art technology. Our new 8-bit, 8MHz machine beats every 16-bit micro we've tested it against and there's documentation to prove it. "Super Bullet" has been designed especially for multi-user systems.

MULTI-USER ORIENTED
You get both CP/M 3.0 and MP/M II operation. You get an exclusively enhanced Z-80H-based CPU at a full 8MHz. You get 256K RAM and four serial ports. High-speed floppy disk with track-buffered controller. Plus SCSI port and LAN option. All of the above and more with the tested and proven Super Bullet. It's an unbeatable value.

WAVE MATE INC. • 14009 S. Crenshaw Blvd. • Hawthorne, CA 90250 • Tel: (213) 978-8600 • Telex: 194369
WAVE MATE EUROPE • 159 Chee de Vleurgat • 1050 Brussels, Belgium • Tel: (02) 649 10 70 • Telex: 24050

In the Public Domain



Several BIOS enhancements—from the early, quite basic, to the more sophisticated

by Chris Terry

The BIOS source code provided on the Digital Research CP/M distribution disk is set up for the Intel MDS (Micro Development System), and is seldom of any use to the purchaser. This same BIOS is built into the MOVCPM utility that allows system relocation to suit the available memory size. Further, the standard BIOS "tolerates, but does not use" the IOBYTE at location 0003.

Thus, in the early volumes of the CPMUG and SIG/M libraries, at a time when CP/M 1.3 was still in use, we find quite a number of BIOS enhancements. Some of these are quite basic; others are more sophisticated and include hard disk support.

Basic BIOS

CPMUG Vol. 1 contains VBIOS31, VBOOT31, and ASSIGN, all written by Jeff Kravitz for a system running an IMSAI disk controller, a ProTech VDM-1 memory-mapped video board, and an IMSAI SI02-2 serial I/O board. Its main value is for study, particularly of a simple video driver and the routines that use the codes in the IOBYTE to attach different physical devices to the four logical devices of CP/M (CON:, RDR:, PUN:, and LST:). All character I/O routines inspect the corresponding bits of the IOBYTE and despatch the character to the physical device currently specified. The ASSIGN transient program sets the codes in the IOBYTE according to keyboarded commands. These routines remain among the clearest and most effective for use of the IOBYTE under CP/M 1.3, 1.4 and 2.2, and well repay close study. CP/M Plus, of course, has built-in provision for far more elaborate I/O control.

CPMUG Vol. 25 contains early CBIOS, BOOT, COPY and FORMAT programs written by Tarbell staff for their single-density disk controller with Shugart 800 or Persci Dual 8" drives. All of these programs have been superseded by later versions, but they are well commented and again are worth close study if you want to learn the basics of writing a disk driver. Note, however, that the BIOS does not include access to the disk parameter tables, which were not brought out into the BIOS until the advent of CP/M 2.0.

Advanced BIOS

CPMUG Vol. 38 is interesting for two items. The first of these is a set of routines, with instructions, on how to integrate your own BIOS into the MOVCPM utility. You need MAC and SID to do this—it's a complex procedure for versions 1.3 and 1.4, and may need modification if you are using 2.2. It probably could be done nowadays much more easily with RMAC, which can generate both Relocatable (.REL) and System Page Relocatable (.SPR) code.

The second item of interest is DFOCO, a disk formatting and copy program for Tarbell or Delta double-density disk controllers. It can format or copy individual tracks in many different formats. Well commented and worth close study.

Until recently, disk controller manufacturers who also made CPU boards had a nasty habit of supplying their own operating system and putting all I/O routines in PROM on the CPU board to lock you into their system. SDS was one of these; thus we find several BIOS entries to allow the VersaFloppy II (a good, reliable controller) to run standard CP/M 2.2. One of these, for 8"and 5.25" floppy disks only, is to be found in SIG/M Vol. 26. Another, with support for XComp and SEAGATE-compatible hard disk, is in SIG/M Vol. 42. A BIOS that supports the DTC hard disk is to be found in SIG/M Vol. 50.

Anyone with a North Star Horizon system should look at CPMUG Vol. 82, which contains two North Star BIOS files, one of which has built-in drivers for the PMMI modem. The volume also contains various utilities. Cromemco users may be interested in SIG/M Vol. 41, which contains a BIOS for the Cromemco 16FDC and 4FDC disk controllers in a system running ACDOS, together with support utilities for CDOS.

Integrating your BIOS into MOVCPM is a complex procedure.

Finally, an interesting BIOS submitted to SIG/M (but not released at the time of writing) implements Bob Lurie's scheme for increasing the capacity of a single-density disk from the nor-

PUBLIC DOMAIN

Continued from page 22

mal 243K to 354K ("Single-Density Disk Formatting" *Microsystems,* October 1983). Willis Howard has writtten a formatter to create standard tracks of 26 128-byte sectors, or system tracks of 29 128-byte sectors with data tracks of two 2,432-byte sectors. A modified SYSGEN and patch to MOVCPM are also included. The BIOS reads and writes either standard disks or dense disks in either drive, adjusting itself automatically. An article on this system will appear shortly in *Microsystems.*

Expand single-density storage to 354K.

The UNIX File

"UNIX is a well-designed, usable operating system, but it can't protect users from ill-conceived attempts to produce enhancements." —*Banahan & Rutter*

by Ian F. Darwin

This month The UNIX File reviews another in the non-terminating series of UNIbooks, looks at writing programs in the shell, and at using your small UNIX system for typesetting.

The UNIX Book

For want of a nail,
the horse was lost.
For want of a horse,
the battle was lost.
For want of a battle,
the kingdom was lost.

And so it has been, all through history. Alas, what we have here is another valiant entry in the UNIbook arena which will probably be lost for want of a seemingly minor detail. *The UNIX Book* by Mike Banahan and Andy Rutter (Wiley, 1983) is one of the few non-American books on UNIX, and its British flavor adds to, rather than detracts from, the book's appeal. The authors seem to know the system quite well. They demonstrate use of the common system commands, the standard editor, shell programming, and the like. And this book conveys some sensible philosophy about UNIX.

"UNIX is a well-designed, usable operating system, but it can't protect users from ill-conceived attempts to produce 'enhancements.' Some suppliers feel duty-bound to include their own souped-up editors with the UNIX systems they sell. Are they really doing you a favor?

"If you feel that you *must* use the flashy editor supplied with your particular system, give some careful thought to portability—both its and yours . . . [The UNIX portable software] means that you don't have to worry nearly so much about details of hardware that differ from one installation to the next. If you start using nonstandard software and don't have a copy of its source code, you could be in trouble when you move from one piece of hardware to another. Users who know only a nonportable editor (by 'portable' we mean one that *has* been ported, not one that vendors say *could* be) are at a big disadvantage."

In this remark the authors convey much significant advice for anyone using UNIX or looking around the UNIX marketplace (or any computer market, for that matter). Portability has both a theoretical and a practical side; if the code could be ported, but won't be because the software vendor has some stake in a particular hardware vendor or system, then how portable is the program really? This points out the advantage of buying software that is known to be widely ported, not just portable.

The book is organized in a fairly natural manner. There is a brief introduction, a chapter on files, one on the standard editor, the filesystem, filters, text preparation, a chapter on C, one on UNIX processes, system libraries, and system administration. Their terminology will seem quaint; they call a filesystem a "filestore," for example. And they use "SIO" for "stdio," the standard I/O library. There is some advice which is interesting but not useful, such as "be wary of the UNIX 'expert,' who may have learned the system so long ago that his knowledge is out of date." Fine advice, but how does a novice tell whose knowledge is out of date and whose is in? And, at odds with the spirit of the above advice, "devour the listings of as many real programs as you can" to learn more about C. Folks, many of the "real programs" are real crocks. What you want to do to learn C is to find *good* program source code examples to learn from. The same page contains the admonition to "read the other books," which is probably good advice, since some of the chapters in this book do little more than whet the appetite.

One of the things that UNIX is about is not retyping manuscripts once they've been proofread. (Although my column gets retyped once at present, this is a temporary—I hope—imposition caused by a change in typesetting services). All the UNIX tools assume that their files are text, so that you can process most any kind of text with the same tools. The phototypesetter software is no exception, and most other UNIX books have been prepared under UNIX using either the **troff** program or some other package (see below). The Banahan and Rutter book, alas, was apparently not typeset under UNIX. Somebody seems to have laboriously retypeset the entire book from the printouts produced under UNIX. The result, predictable where technical text is concerned, is a typographical disaster.

In this book you are expected to infer a lot from the examples given. But many of the examples are typeset poorly or incorrectly. Spaces appear and disappear; command names change, wrong quotes are used, and so on. There is sloppy typesetting of examples throughout, which leads me to believe that the book was typeset by somebody with no real understanding of UNIX. This is, I hate to say, the worst typesetting job I have seen in any technical book. Ever.

The hand typesetting is reflected in other areas as well. *The UNIX Program-*

UNIX FILE

Continued from page 26

ming Environment, which I reviewed last month, was prepared and typeset completely under UNIX, and has an excellent index (nine pages, three columns per page) although somebody renumbered the Roman numbers in the front matter after the index was done. *The UNIX Book* index is but two pages, two columns, large type. Many useful references are omitted from this index. But the appendices more than make up for the poor index. Tables of all the common commands, editor subcommands, shell syntax, a well-organized listing of the system calls and standard libraries, and so on—this is the reference material that you need when learning UNIX.

The authors claim that "very few UNIX sites have access to a typesetter," which is probably more accurate in Great Britain than in North America, where most large UNIX sites either own a typesetter or have access to one. So it's too bad that the authors of *The UNIX Book* were not able to make better arrangements for typesetting. Of course, *a reputable publishing house such as Wiley should have come up with better arrangements.* If they do so, and come out with a second edition that corrects the poor typesetting, then this book could serve as a good although slightly superficial introduction to UNIX. As it is, *The UNIX Book* has some very good points, but these are sunk by some very poor visual presentation.

Shell programming considered beneficial

In previous columns I gave a few examples of "shell programming." This time I'd like to comment on the shell as a programming language comparable to Basic, C, or PL/1 in its utility. The shell provides an interactive programming environment like that of Basic, with program structuring comparable to C or PL/1, variables (not limited to two-character names!), immediate or stored-file execution, file access, etc.

As one example, I needed a prompter for some data entry I am doing on another editing project. It collects bibliographic data (citation, author, date, etc.) on magazine articles. I first wrote it as a C program. This worked, but took lots of time to write and debug. Fortunately, I had to do some of the entry on a UNIX system which (temporarily) did not have enough memory to run the C compiler. Being forced to replace my C program with a shell file made me think a lot about shell files as programs. I was able to replace more than a hundred lines of C with about 30 lines of shell command file. It worked the first time (unlike the C program, which took several tries to get right). Not only that, but after running it for a while I decided to change some parameters. This involved simply editing the shell file; no recompilation step was needed to try out my changes.

I recommend this exercise for any aspiring UNIX programmer. Forego use of the C compiler for two weeks, and do everything as a series of shell files. During this time, learn some of the finer points of **make, awk, sh, pr, nroff** and **adb** (or **sdb**). During the time my home system was without C, I developed a (nonspooled) version of **lpr** which converted backspaces to overstrikes and did tab expansion (using only **awk** and **stty**), several versions of the bibliography prompter mentioned above, an **awk** file to generate another **awk** file to read in and check a database, and other programs.

The use of existing tools in new ways enriches UNIX without building up new mythologies.

As well, at work we have replaced several C programs by shell files. To some programmers, this sounds like a step backwards (although it inarguably results in fewer lines of code to maintain!). Part of the UNIX philosophy consists of using the right tool for the right purpose. To me, this includes *not* inventing a new tool for which a perfectly good one exists. Of course there are tradeoffs between building new tools and using existing ones. My **awk**-based **lpr** would not be used on an overloaded timesharing system (but overloaded timesharing systems are supposed to be the ebb tide of the past). Part of the problem with some currently available UNIXes is that their maintainers did not familiarize themselves fully with the existing UNIX tools. Instead, they built a bunch of new tools or disfigured existing ones, rather than building tools out of combinations of standard utilities. The "new tools" approach complicates UNIX both for the end user and for subsequent developers; the use of existing tools in new ways enriches UNIX without building up new mythologies.

Typesetting from UNIX

UNIX has traditionally been used for preparing material to be typeset directly from computer-readable copy. This text can be checked by various spelling checkers, processed by equation, table, bibliographic, diagram and other preprocessors, formatted for a character printer with **nroff,** and finally typeset with a similar program called **troff.** But the **troff** program assumes that you have a particular typesetting machine, a Graphic Systems (WANG) C/A/T. They're not made anymore, and most people don't have one. However, if you have a small UNIX system you probably wish to get the benefits of the typesetting software. Here's how you can.

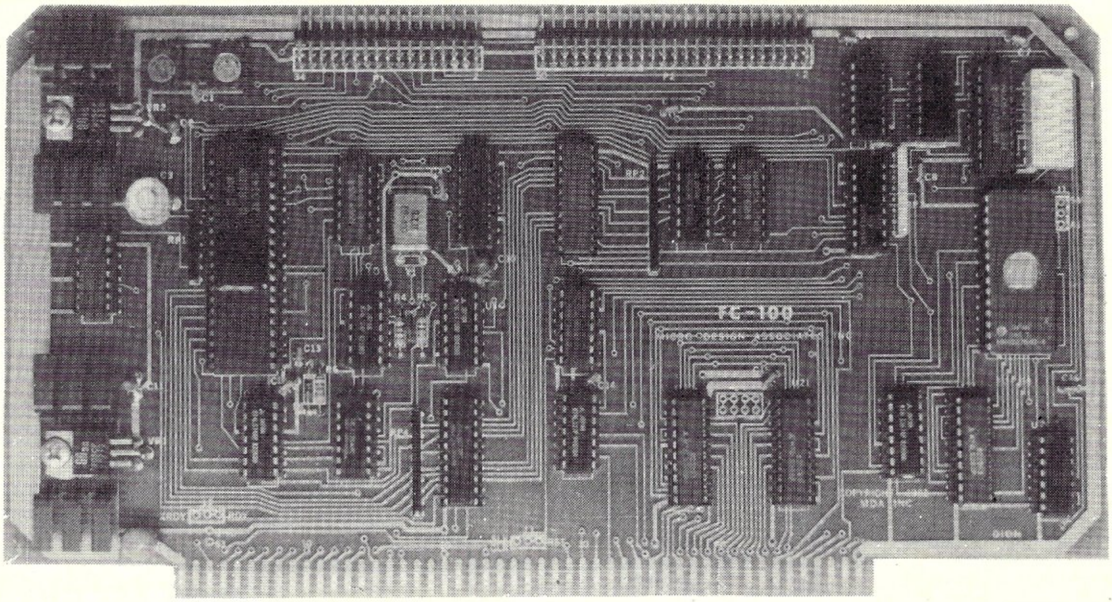
First decide if you need real phototypesetting, or if *good* proportionally spaced daisywheel printer output will do. Before you reject the latter, look at some printouts. Remember that this is what most micro-based word processors use. If the former, decide if you want to do the typesetting yourself or send it out to a professional typesetting firm.

If you want to buy a typesetter, you might consider buying a used C/A/T. They are somtimes available on the used equipment market. There also exists a serial interface which you'll probably need unless you have a Unibus VAX-11 or PDP-11 system. You'll be investing in really obsolete technology, but the **troff** which comes with your UNIX will know how to talk to it. You'll also need a photochemical processor to develop and fix the paper output—this will probably cost a few thousand more. And you'll need to develop a whole new realm of expertise to get good results.

Instead of a C/A/T, you might buy a more modern typesetter. Brent Byer of Textware in Cambridge, MA, sells a couple of packages for small UNIX systems. Both take the output from the standard **troff** and recast it for particular devices. **Tpost** transforms the CAT files into a format suitable for newer machines such as the Compugraphic MCS8400 with a serial interface. One of our customers at work has **Tpost** and an 8400. While it was not simple to set up, the results have seemed satisfactory to date. Textware also sells **Tplus,** which makes **troff** drive a daisywheel printer such as a Diablo, NEC Spinwriter, or the like. **Tplus** represents the output as best it can on a device with fixed type size but variable motion. For normal

THE ULTIMATE DISK CONTROLLER

STATE OF THE ART



ALL SIGNALS IEEE 696 (S-100)

$299
Assembled and Tested

- Runs 5¼″ and 8″ drives simultaneously
- Double Side, Double Density (1.4 MB 8″)
- On board extended address bus driver (16 Meg)
- On board shadowable ROM (up to 8K)

This controller was designed using the very latest in state-of-the-art technology. The board is made of highest quality epoxy-fiberglass and all components used are available "off the shelf".

- 8 bit or 16 bit data bus
- Low power consumption
- Compatible with all 1771, 179X & 279X software
- I/O mapped (does not take up any memory space)
- Very fast - runs at maximum transfer rate (.5 Meg B)
- Comes with free bios & instructions for your CP/M
- Reads & writes formats of the following systems: IBM 3740, System 34, IBM PC, Cromemco, Televideo, Morrow, Osborne, Kaypro, Lobo, Xerox, TRS 80 models 1, 2, 3, 4, 12, 16 and most other soft sectored systems
- Comes with format program & other utilities including Bios for CP/M 2.2
  CP/M 2.2 w/Bios - $100.00
  CP/M 3.0 w/Bios - $250.00

MasterCard

VISA

TO ORDER, PHONE:

MICRO DESIGN ASSOCIATES, INC.
25 S. 8th St.   (314) 874-2777
Columbia, MO 65201

CIRCLE 85 ON READER SERVICE CARD

1 YEAR WARRANTY!

After warranty service will never cost more than ½ the price of a new board. Replacements shipped within 24 hours.

MANUAL — $15.00 — Deductible From Purchase of Board

UNIX FILE

Continued from page 28

text such as letters, the sample outputs look quite good. For representing very large or very small-sized type, the quality drops off a little as the daisywheel letters get very far apart or very close together. But check it out yourself: **Textware International,** *Box 14, Harvard Square, Cambridge, MA 02238.* Their phone number is nice: *716-UNI-TEXT.*

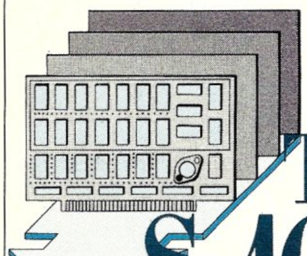
Another approach involves Bell Lab's new **troff,** also called Typesetter Independent Troff or Device Independent Troff (the latter is pronounced "dit-roff"; the former is better not pronounced). This is a major revision of **troff** that can be adapted to many new devices, including both typesetters and daisywheel printers. Unfortunately for the small-system owner, AT&T (Bell, Western Electric) only licenses **ditroff** in source form, and it's several thousand dollars. They have a binary sublicense, which would allow OEMs to sell **ditroff** to end users in binary form for $200 in royalties, but the major porters of UNIX have so far passed over this opportunity (hint, hint).

If you are not taken with **troff,** you might look at a couple of other packages. Knuth's TeX package can be had for UNIX, as can the SCRIBE package from UniLogics. Both are rather newer designs than **troff,** and have about the power of **troff** with one of its standard macro packages. Both drive a range of devices, including daisywheel typewriters and phototypesetters, from the same input. However, they are much more verbose, and generally offer you much less freedom of action. They are difficult to write preprocessors for. And because they do not expect preprocessors, because they try to be all things to all people, they are quite large programs. I am familiar with all three, but have only used **troff** seriously. Perhaps I'll revisit the topic if and when I've made some real use of them. I doubt I shall use them, though, since SCRIBE babbles on so, while doing even the simplest thing, that it gives new overtones to the term "full-screen software."

There are thus several approaches if you want to do your own typesetting. And if you want somebody else to do it, you have a variety of other ways to proceed. Some typesetting shops will accept **troff** input files; others will accept output files. There are important issues regarding typefaces and different widths; I suggest you talk these over with your local typesetting firm. Many large cities now have typesetting houses which know about **troff,** so look in the yellow pages for typesetting firms and phone around until you find somebody who knows what you're talking about. If you can't find anybody in your home town, drop me a line and I'll try to put you in touch with somebody nearby.

Watch for my comments on the UNIX conference in January in the next column, as well as more on "shell programming" and some information on networking. Also, next month's issue of *Microsystems* will have a directory of UNIX software. Watch for it!

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 to me at the University of Toronto Computing Services (UTCS), 255 Huron St., Toronto, Ontario, Canada M5S 1A1. If you have UNIX mail access to the USENET network, you can contact me at "decvax!utcsstat!ian". The opinions presented here are my own, and not necessarily those of the University of Toronto or of UTCS.—Ian F. Darwin



The S-100 Bus

Simple power failure backup systems for the S-100

by Dave Hardy

No matter how dependable your S-100 machine is, it is still susceptible to power failures and line dropouts. A well-designed power supply can save you if the dropout is short enough, but no filter can protect your system for more than a few seconds.

The best solution, of course, is an Uninterruptible Power Supply, but a UPS is also the most expensive. You could also use a good-sized DC-to-AC power converter, then buy a bunch of car batteries and then an AC-powered battery charger and hook them all up, but if you're like me, you'd probably have to sell your S100 frame to get enough money and make enough room for all that stuff. Plus, most "wet" batteries produce hazardous gasses when they are charged, and no one wants a room full of hydrogen gas and sulfuric acid fumes.

My own S-100 system runs 24 hours/day as an RCPM or RMPM, and suffers from line dropouts once or twice each week which, in the past, have caused a lot of grief. The worst problem was that the dropouts were often just long enough to cause the system to "glitch," but not quite long enough for the system to sense a power failure and automatically re-boot.

The solution, which has worked well for the past several months, was to replace the S-100 power supply with batteries. Surprisingly, it is easy to get inexpensive high-capacity sealed batteries in exactly the right voltages for an S-100 system. Five medium-sized "GEL-CELL" type 8-volt batteries, with a capacity of about 2 AMP/hours each, were used in my system: two batteries each for +16V and -16V, and one for the +8V line.

Because power regulation is performed on each individual circuit card in an S-100 system, the power supply voltages to the S-100 bus are not nearly as critical as they are in most other systems where the power supplies usually drive sensitive TTL devices directly. Therefore the battery voltages can vary significantly, as long as they are high enough to drive the S-100 cards' regulators.

Installing the batteries is easy, too, since they can just be attached to the S-100 bus power inputs. Because of their inherent low impedence, they also provide some additional filtering for the power supply. A three-pole power switch must also be added to allow the machine to be powered on and off.

After the batteries are installed, the machine's original power supply is still needed, but now it functions both as a primary S-100 power source *and* as a battery charger. Should the AC power line fail or drop out for a short time, the batteries will continue to run the system for as long as they are capable.

Of course, there are some parts of the system that can't be run by batteries, such as cooling fans and AC floppy disk drive motors, but many systems (including mine) have DC drive motors and use convection cooling. If your system has to have AC power to run these things, and you still want it to be able to run during a power failure, then you can always use a UPS or try a DC-to-AC power converter, but they will cost a great deal more than simple batteries.

If you want to be able to run the system entirely on batteries instead of just having it stay "alive" during a power failure, you will probably have to add additional batteries to supply the +12V or +24V lines for the disk drives. All my system does during a blackout is interrupt to a power line test routine and stay there until line voltage is restored, so the disk drives don't need to operate until the AC comes back on.

Replacing S-100 power supplies with batteries is surprisingly easy.

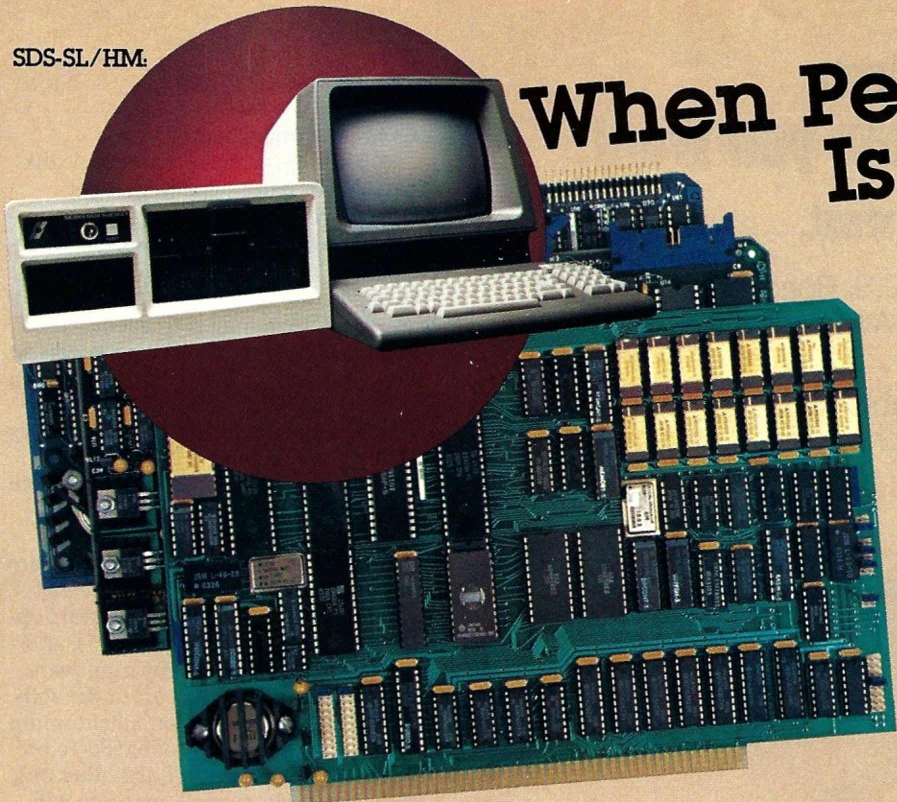
Phantoming and banking

Several readers have written asking the difference between "phantoming" and "banking" memory. Next month's S-100 Bus will discuss these two S-100 phenomenon as they exist in the IEEE-696 standard, and will include some easy ways to add them to any S-100 system. But to quickly answer the readers' questions, phantoming, in terms of the S-100 bus, is performed by just asserting a bus line (called PHANTOM*) that disables one memory block so that another may temporarily exist in that block's address space.

Many disk-based systems use phantoming to switch in a boot-up ROM to load an operating system, then switch the ROM back out so that the operating system can have 64K of RAM.

SDS-SL/HM:

When Performance Is A Factor...



Winchester/SMD Hard Disk Controller

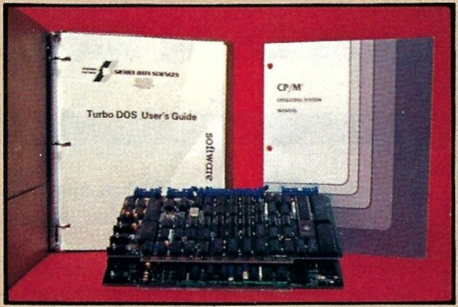
(Shown with Sierra's new 8MHZ Boards)

This is an amazing high-speed companion for your single-or multiuser system. Features NEC's new hard disk controller chip, on-board Z80 processor, buffered sector heads, full ECC supported, SASI or SMD versions available, and optional high-speed DMA pass-thru port for use on Sierra Data's new 8MHZ Master/Slave.

8MHZ-SBC Master and Slave Boards

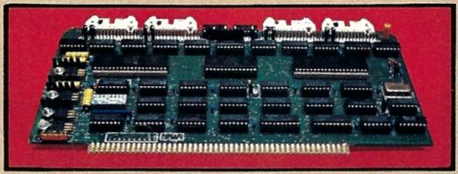
Now, from a Z80 processor, the most incredible throughput available today in the S-100 industry! As the powerful nucleus of a Sierra Data-supported CP/M 2.2* or 3.0* system, our master makes any system look good. With networking TurboDOS* and our new slave, the results are absolutely unbelievable. Check these features (M-master, S-slave):

- Z80H (M & S) Z80A (M optional)
- Dart, 2 serial ports (M & S)
- PIO, 2 parallel ports (M & S)
- CTC, 4 counter timers (M & S)
- From 64 to 512K on-board RAM (M & S)
- 8232 Math Processor (M & S)
- 4K/8K/16K EPROM (M & S)
- Meets IEEE-696/S-100 Standards (M & S)
- NEC 765 Floppy Disk Controller-controls 8 in. and 5¼ in. floppy drives simultaneously (M)
- Multiple Master Buss Arbitration (M)
- DMA Controller (M)
- High-speed Winchester DMA port (M)
- 4K static RAM 8/16 bit independent bus transfer circuit (S)
- Either one or two users per slave under TurboDOS* (S)



SBC-100 and SBC-100S Boards

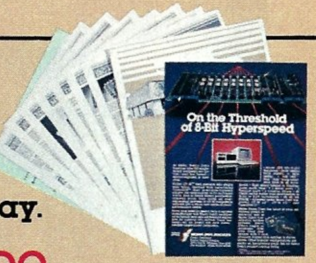
The workhorses of the single-board computer industry. At 4MHZ these totally IEEE-696 compatible champs out-perform even the 6MHZ imitators. In standalone or slave-satellite configurations you get dedicated Z80A processing with 64K RAM, 4K EPROM, 2 serial RS-232 channels, 2 parallel channels, 4 counter timers and a host of floppy and hard disk interfacing options. Teamed up with Sierra-supported CP/M 2.2* or networking TurboDOS* you get the most computer power and upgradability for the money.



ZSIO-4

A powerful four-port RS-232 serial I/O and real time clock board. Zilog SIO-chips provide ideal links to CP/M-MPM-and AlphaMicro-based systems for multiuser processing with high speed data communications.

Out-perform your competition with a company that stays a step ahead. Ask for our free "Performance Building Kit" today.



216-892-1800

HARDWARE
SOFTWARE

SIERRA DATA SCIENCES

Fresno, CA
Product Support Division • 25700 First Street • Westlake, OH 44145
(216) 892-1800 • TELEX: 980131 WDMR

*TurboDOS is a Registered Trademark of Software 2000 Inc.
*CPM is a Registered Trademark of Digital Research

CIRCLE 23 ON READER SERVICE CARD

S-100 BUS

Continued from page 32

Banking, or, more properly, bank selecting or bank switching, is performed by (usually) writing a value to a control port to cause one block of memory to be disabled, and another to be switched into its address space.

Bank switching is frequently used in multiuser or multitasking systems so that each user, or task, can have its own large bank of RAM. MP/M II, for example, allows up to 16 banks of memory of up to 48K in size. CP/M Plus uses bank switching for buffering things like directories and disk files to greatly increase its processing speed.

Bank selecting and phantoming are similar in their functions, except that phantoming is caused by a hardware operation (asserting PHANTOM*), and bank selecting is caused by a software operation (writing to a port under software control). In addition, phantoming allows only a single set of blocks to be switched (one turned off, the other on), while bank selecting allows up to 256 blocks to be selected.

CP/M Plus uses bank switching for buffering, which greatly increases processing speed.

The IEEE-696 S-100 bus also allows extended addressing (that is, up to 24 bits of addressing) that could be used by an 8-bit processor like the Z80 only if it were treated as bank-selected memory. Unfortunately, because the Z80 can only directly control 16 address bits, extended addressing is of little use unless some nonstandard provision is made for a block of "common" memory—in other words, bank selecting. It is interesting to note that 8-bit operating systems like CP/M Plus and MP/M that require multiple banks of memory cannot operate with extended addressing for the very same reason.

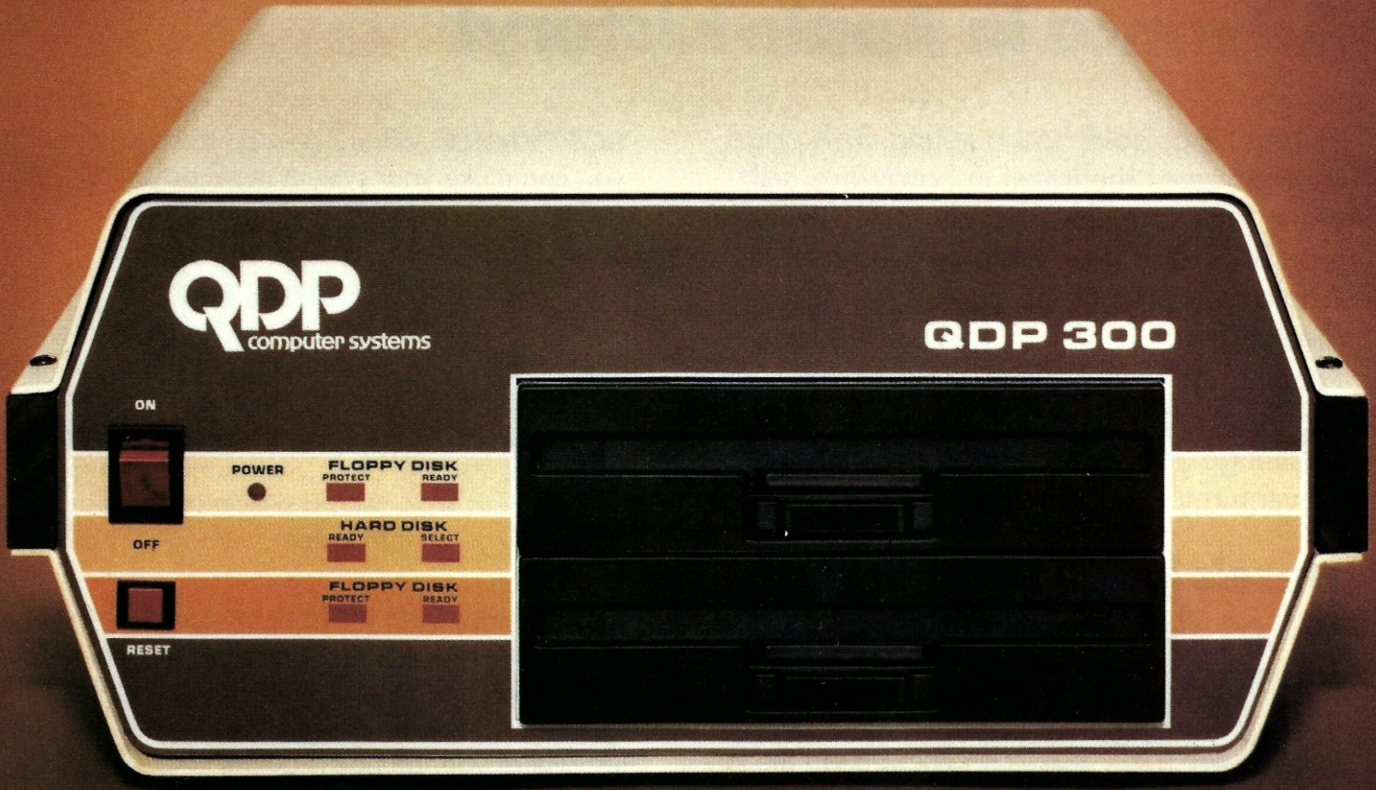
Bank selecting, phantoming, and extended addressing can all live comfortably together in the same S-100 system, although it is unlikely that there

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

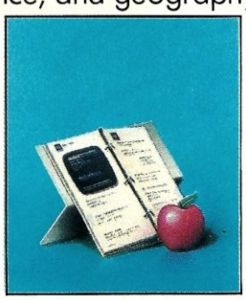
Specifications subject to change

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

PROGRAMMING TOOLS: Speed up program development with Apple text editors, testing and debugging aids and data base management systems. Since your Apple can speak many languages—from Cobal to Basic and Pascal—**A+** gives you a wide range of utilities for each language so you can create your own exciting programs.

HOME/PERSONAL PROGRAMS: covers it all...from basic processing...to helping you draw with Apple graphics...to speed reading...music theory and composition... even a course on cooking with your Apple... and then how to monitor your diet easily and accurately...along with many educational programs to learn math, science, and geography.

DATA FILES: Let **A+** probe new data base management systems so you'll be able to organize, store and retrieve information in ways you never thought possible.



YOUR TURN: Speak up with your ideas and innovations and network with other Apple users near and far.

Plug into Savings of Up to 33% with Our Introductory Offer!

YES! I want to subscribe to **A+** for:

- ☐ One year (12 issues) only $19.97—20% off!
- ☐ Two years only $36.97—26% off!
- ☐ Three years only $49.97—33% off!

Savings based on full one-year subscription price of $24.97.

Name____________________

8H179

Address____________________

City____________________

State__________ Zip__________

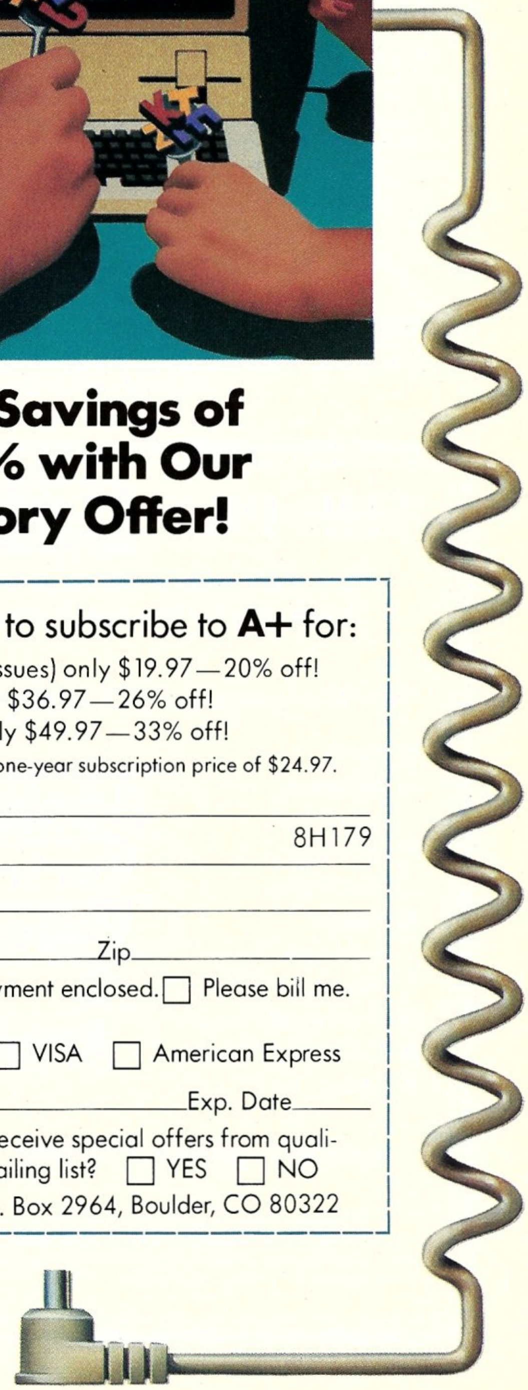
Check One: ☐ Payment enclosed. ☐ Please bill me.

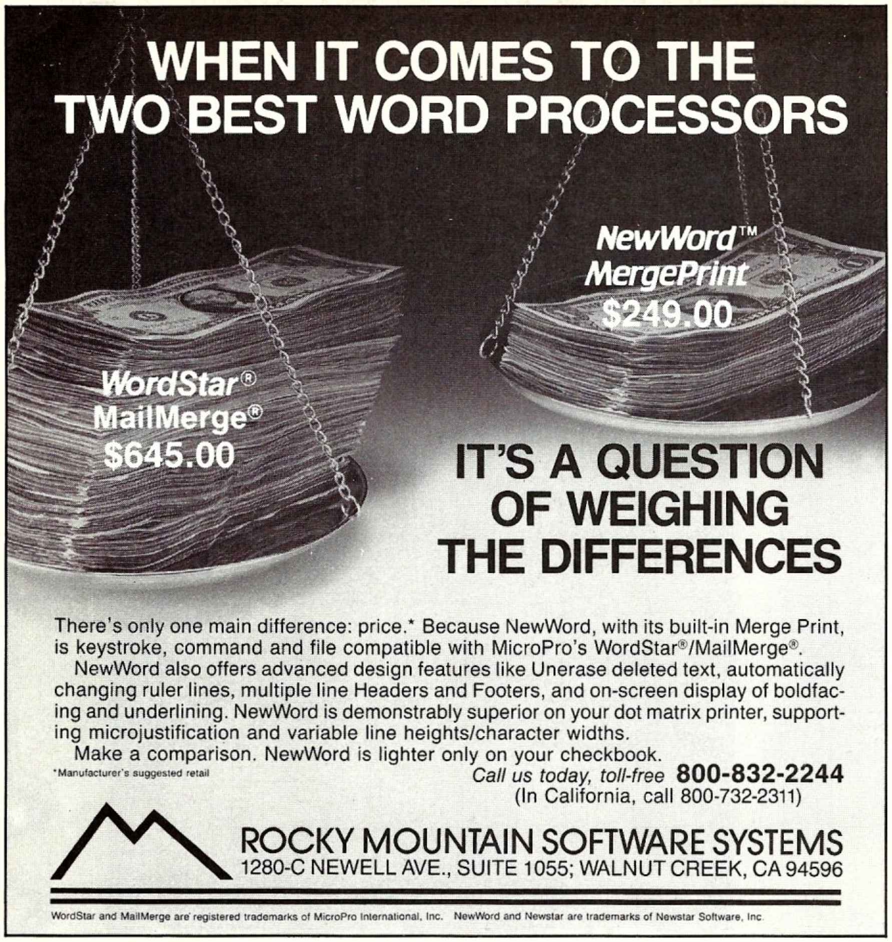
Please charge my:
☐ MasterCard ☐ VISA ☐ American Express

Card No.________________ Exp. Date______

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

MAIL TO: **A+**, P.O. Box 2964, Boulder, CO 80322





CIRCLE 20 ON READER SERVICE CARD

S-100 BUS

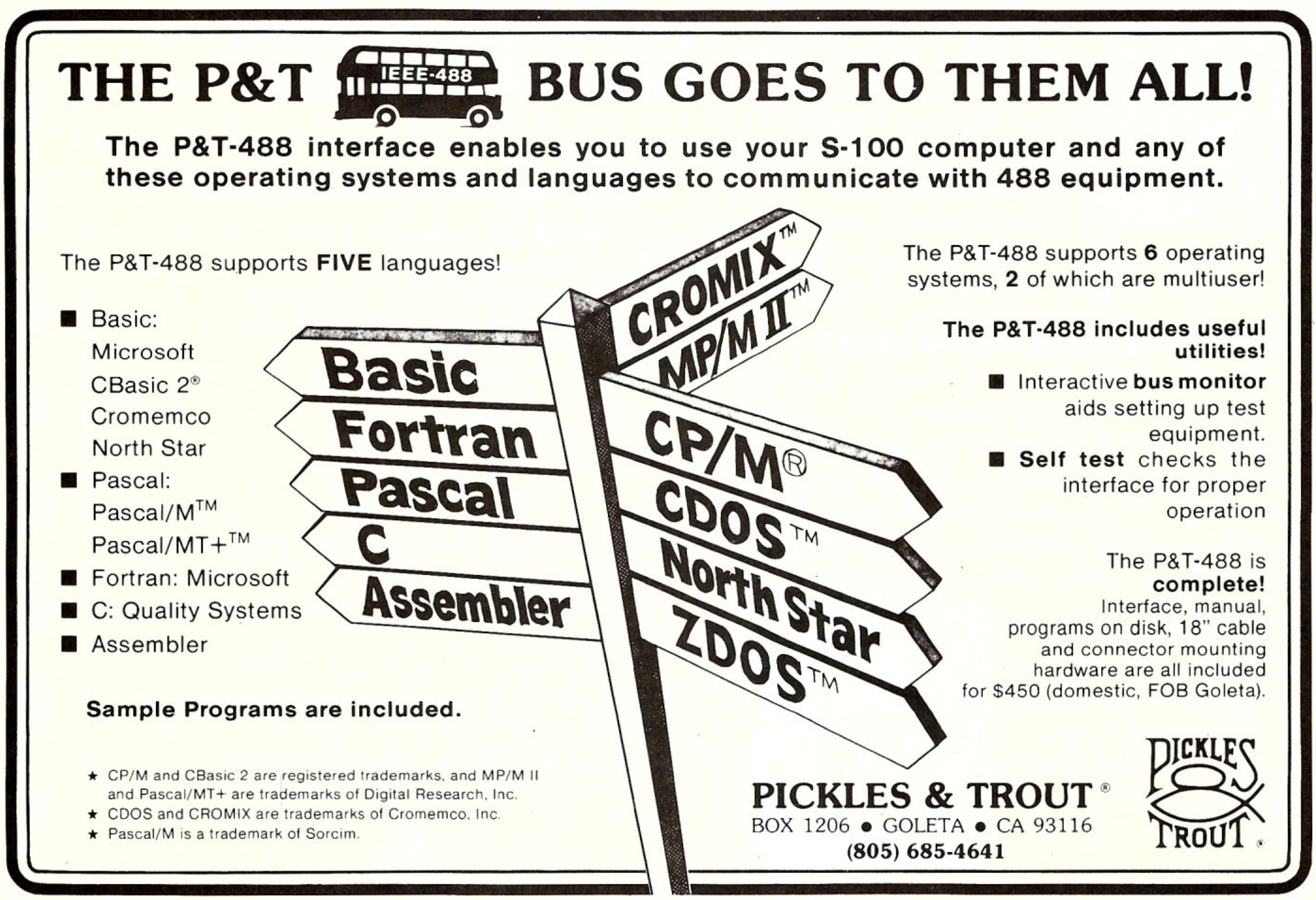
Continued from page 34

would ever be a need for both bank switching *and* extended addressing at the same time, since either method will allow more memory than any single S-100 frame could ever hold (at least until next year).

Future topics

Along with next month's phantoming and bank selecting examples, future S-100 Bus columns will include the much promised North Star bus pin-out comparison, more IEEE-696 standard "tutorials," and simple S-100 interfaces, like an S-100 to SASI or SCSI controller interface. However, I depend on reader feedback to set the direction of this column. I encourage reader participation in "The S-100 Bus," and look forward to your mail and calls. If you would like to see any particular facet of the S-100 bus discussed here, please let me know.

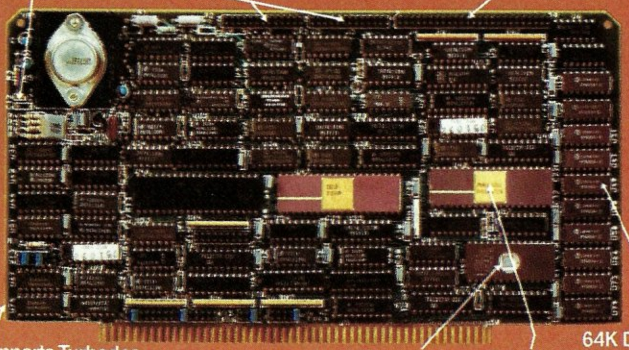
This column is intended as a forum on S-100 bus topics. Readers are encouraged to send in questions on the S-100 bus, which I will attempt to answer. Please write to: Dave Hardy, 736 Notre Dame, Grosse Pointe, MI 48203.



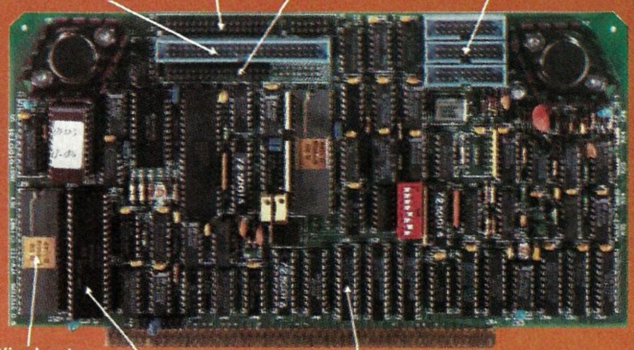
CIRCLE 89 ON READER SERVICE CARD

THE ULTIMATE MULTI-PROCESSOR BUILDING BLOCKS

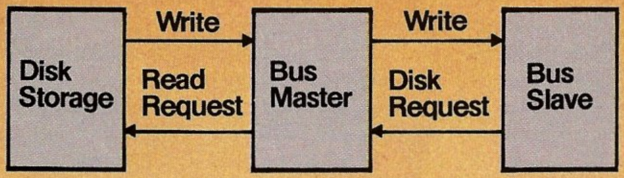
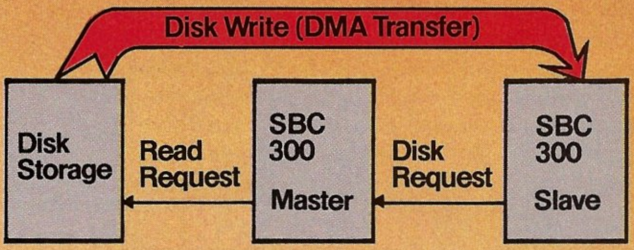
SBC 300
single board computer



VFW-III
floppy/Winchester disk controller



AND HERE'S WHY!

Our Competitors:	The SDSystems Advantage:
	

The SBC 300 offers you 64K bytes of dual ported parity RAM. Dual porting allows DMA transfers from disk storage *directly* to an individual slave processor. This maximizes multi-processor system performance.

Performing the DMA transfers is the VFW-III peripheral controller. It is quite possibly the only S-100 disk controller you'll ever want to buy. With the VFW-III you'll have simultaneous control of up to three 5¼″ Winchester drives and up to four 5¼″ and 8″ floppy drives.

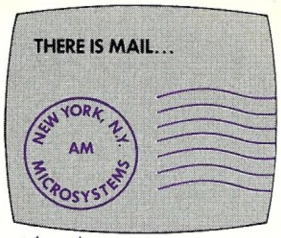
Together they are the ultimate multi-processor building blocks. Systems integrators appreciate our strict adherence to the IEEE-696 specification and our complete line of high quality S-100 bus products. Evaluation hardware is available for qualified OEM's. Call, write, or circle the number below to receive technical information on these and other fine SDSystems products.

Z80 product Zilog Corp. Turbodos is a trademark of Software 2000, Inc. Oasis is a trademark of Phase One Inc. CP/M Plus is a trademark of Digital Research, Inc.

P.O. Box 28810 Dallas, Texas 75228 214-340-0303 Telex 682 9016

SDSYSTEMS

CIRCLE 3 ON READER SERVICE CARD



Letters to the Editor

This month... problems plaguing the PIP utility, differing views on Nevada Fortran, and a closer look at memory management in older S-100 systems

Dear Mr. Libes,

Steven Fisher's article "PIP Data Between Computers" (*Microsystems,* July 1983) performs a great service to CP/M users and provides insight into an important, but poorly documented, program. He indicates that his approach will work on machines whose CBIOS (Customized Basic Input/Output System) does not implement the reader or punch BDOS functions. Unfortunately, he does not fully describe the procedure for using the console port to transmit files. His suggestion to change BDOS function numbers 3 (reader) and 4 (punch) to 1 (console input) and 2 (console output), respectively, does not work properly. There are three major problems whose discussion may save other users many frustrating hours.

1. BDOS function 1 (console input) in CP/M version 2.2 echos the byte received from the "console" port. When this port is attached to a CRT, this echo permits the user to see what was typed. However, when this port is attached to a computer running a program with Mr. Fisher's sequential read/write handshaking protocol, the echo produces undesirable side effects. In a situation with a CompuPro/Godbout Interfacer-4 board transmitting to a Teletek Systemaster, the side effect is for the first character of the original file to be propagated throughout the received file as every other character. This can be explained by the echo of BDOS function 1 and the I/O buffers in the Godbout and Teletek.

The easiest solution to this problem is to use BDOS function 6 (Direct Console I/O) for reading. This function does not echo and uses the value of register E to set reading, as is shown in Listing 1.

2. The second problem is more difficult to repair. When PIP completes its file manipulation, it performs a warm boot and returns control to the CCP. Under normal circumstances, this condition is conveyed to the user with the familiar CP/M prompt: A>. What could be more natural? However, when another computer running Mr. Fisher's handshaking protocol is attached to the "console" port, the prompt is transmitted and duly echoed by the receiving computer's CCP, or executing PIPIO program. The sending computer's CCP receives this, fails to find a program (COM file) by that name and sends a response intended for a CRT but which becomes more input to the receiving

LISTING 1

```
RCV:      MVI E,0FFH          ;LOAD FF TO READ
          MVI C,06            ;BDOS DIRECT I/O FUNCTION
          CALL BDOS
          STA  0109           ;A HAS CHAR, STORE IT FOR PIP
```

LISTING 2

```
LOCATION          INSTRUCTION

0100H             JMP 0200            ;JUMP TO TIME DELAY
0103H             JMP 010A            ;JUMP TO RECEIVE
0106H             JMP 0150            ;JUMP TO SEND
0109H             NOP                 ;CHAR FOR PIP.COM
:
:                                     ;READ AND WRITE
:                                     ;ROUTINES
:
0200H             LXI H,FFFF          ;HL HAS # OF WAITS
0203H             DCX H               ;16 BIT DECREMENT
:
:                                     ;READ
:                                     ;COMPARE H AND L
:
024DH             JNZ 0203            ;LOOP UNTIL WAIT DONE
0250H             MVI A,0CE           ;REPLACE PIP'S
0252H             STA 0101            ;ORIGINAL JUMP LOCATION
0255H             MVI A,04            ;
0257H             STA 0102            ;
025AH             JMP 0100            ;GO TO IT
```


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
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.
CIRCLE 215 ON READER SERVICE CARD

LETTERS

Continued from page 41

computer. These transmissions of "A", ">", "?", etc., continue ad infinitum.

One solution is to disconnect the cables connecting the two computers, but, aside from the inelegance of this approach, we found that often the receiving computer had failed to close its file at that point and the transmitted file was lost. Our solution was to check every character sent for Control-Z (end of file) and, following its transmission, to alter the appropriate memory locations in the CBIOS of the sending computer so as to route subsequent console output to a harmless device such as a printer. Two problems with this are (a) the sending computer hangs, and (b) the memory locations to alter are specific to a particular CBIOS, and therefore the program cannot be used on other systems.

3. Using the console means changing cables, but the sending computer begins transmission as soon as PIP.COM and the ASCII file are loaded into memory. On disks with short directories, this requires a high degree of hand-eye coordination to complete the cable change in time. We used a time delay which, because of line noise due to cable changes, incorporated Direct I/O reads. Forcing PIP to perform an initial wait requires the code shown in Listing 2.

The problem of implementing PIP's INP: and OUT: functions using a console turned out to be much more difficult than we anticipated. Moreover, our solutions required system-specific manipulations that defeated our attempts to produce transportable code. Perhaps other users have produced better solutions to these problems; if so, we certainly hope they publicize them.

James W. Haefner and Scott E. Kelso
Department of Biology
George Mason University
Fairfax, VA 22030

Dear Mr. Libes,

Having read the article on MP/M 8-16 in *Microsystems* (January 1984), I thought you might be privy to the solution for implementing WordStar 3.24 on the Altos 8600-12 under MP/M-86 2.11F0. Every 10 minutes or so the machine locks up—no error message or anything; more frequently with more users; sometimes with just *one!* I have sent away for updates to WordStar and MP/M-86. Meanwhile, Altos doesn't know how to fix it and MicroPro swears it works. I've had my hardware checked out and I've used new copies of my originals. I've tried making the memory partitions larger (and smaller). Among others, I've made the following GENSYS responses:

```
total character control works: 16
enable compatability attributes:
                              yes
(I then SET * .OVR & WS.CMD
[F1=ON,F2=ON,F3=0,F4=ON])

Temporary File drive B (my hard
     disk)
Maximum locked records per
     process = 14
Total locked records in system
     = 40
Number of extra process
     descriptions = 60
Maximum paragraphas per process
     = FFFF
Number of Queue Control Blocks
     = 60
Size of Queue Buffer Area in bytes
     = 600
Number of extra memory description
     = 60

2000  1800  3800  1800  4800  0800
5000  0800  5000  0800  6000  0800
6800  0800  7000  0800  7800  0800
```

With the exception of item #1 and the 0800 memory segments, the numbers above are all larger than the default numbers. Any help you can give me would be worthy of publication, for sure—I've heard similar complaints from other people.

Jeb Bonsteel
Data Manager
Southern Maryland Health
Systems Agency
9131 Piscataway Rd., P.O. Box 85
Clinton, MD 20735

Editor's note:

If any reader has encountered similar problems, or has suggestions for a solution, a letter would be very much appreciated.

Dear Mr. Libes,

I would like to take issue with the glowing review of Nevada Fortran, written by David L. Dupuy, which appeared in the November 1983 issue. My own experience with that compiler (I have version 2.2) is less than satisfactory. Besides its limitations, which the reviewer notes in passing, it has far more serious problems that were left unmentioned. The most glaring of these is the total lack of context in any error messages. A recent compilation of one of my programs produced this:

```
*** ERROR *** Unidentifiable
              Statement
```

No context, no line number, nothing. In order to debug my programs, I have been forced to introduce *deliberate er-*

rors in order to isolate the location of the real ones. Perhaps Mr. Dupuy is beyond the stage of writing code which contains errors; for me, this one problem makes the compiler essentially unusable.

There are other irritations in Nevada Fortran—for example, the fact that when reading in formatted data, input of characters which are inappropriate (i.e., letters when digits are expected) often causes the program to crash, spewing several lines of incoherent error messages to the screen before it dies. Also, Mr. Dupuy reports cheerfully that Nevada Fortran actually beats *interpretive* Microsoft Basic for speed. What a relief that should be to prospective buyers!

There are other annoyances, but of the kind I expect in a $29.95 product. It is the major error-handling problems which are so devastating, especially given that many buyers may know no other language but Basic and may attempt to use this inexpensive opportunity to learn Fortran. Good luck to them.

Lastly, I found it distressing that the review, which ended with instructions on how to purchase Nevada Fortran, followed by Ellis Computing's response to the few criticisms Dupuy did mention (they sure respond quickly to a review), was then followed on the next page by a full-page ad for Nevada Fortran and other compilers. While computer journals have traditionally tread the fine line between product promotion and fair criticism, this was too much for my stomach. *Microsystems* owes more to its readership than the gleeful pushing of mediocre products. Meanwhile, I am spending real money on a real compiler. It's worth it.

Rex E. Bradford
3 Bush Hill Road
Ipswich, MA 01938

Dr. Dupuy replies:

(1) Obscure error messages: *the error handling in the compiler could certainly be better. In retrospect, that would have been worth mentioning. I did not have significant problems tracking down errors, so I did not consider it a problem. But for $29 I did not expect it to measure up to a DEC compiler. My usual technique for debugging is to write a "five-liner" if I am unsure of the effects of a given few lines of source code, and debug these small modules separately.*

(2) Reading formatted data: *as far as I am aware,* any *Fortran will have problems if a formatted read statement encounters inappropriate data. Some Fortrans (Microsoft?) assign zeros and keep going, and that's worse! You then have a problem and don't know it. I got two error messages with Nevada Fortran while entering alphabetic data on an F format:* (a) Runtime error: ILL CHAR, *and that seems like an informative error;* (b) Program was executing line ???? in routine MAIN. *In this error message, I presumed there was a provision in the setup options that I overlooked to have the question marks replaced by the actual line number, but I didn't attempt to search this out. For most data input, the free form ACCEPT works well. In general, I found the error messages were adequate, especially with the details in the manual nearby.*

I personally have no objections to an ad near a review article. That's a question that could be included on a questionnaire to subscribers, if Microsystems *puts out another questionnaire.*

David L. DuPuy
Assoc. Professor
Virginia Military Institute
Lexington, VA 24450

Dear Mr. Libes,

Andrew Bender's article on adapting older S-100 machines (November 1983) for extended memory was timely and well presented. However, he fell a little short by assuming that the Z80 is limited to a byte at a time when moving data between banks.

It is true that the Z80 cannot move data directly between banks without some extra hardware, but data can be moved more swiftly by than a single byte at a time. The method I used when implementing CP/M Plus was to set aside a 128-byte buffer in the common area of memory. This allows me to use the Z80 block move instruction to transfer blocks of data at a time, and since I needed exactly 128 bytes of memory for auto disk selection, its use wasn't wasted. Although, since CP/M Plus is relocated to 1K boundaries, the average system probably has at least this much memory wasted anyway.

The procedure necessary to transfer data between banks goes like this: *select the source bank, block move 128 bytes into the common buffer, select the destination bank, and finally move the 128-byte buffer to the destination.*

If a large amount of memory is to be transferred between banks, such as the CCP, using this method will greatly speed the operation.

Robert Blum
5536 Colbert Trail
Norcross, GA 30092

Dear Mr. Libes,

I am somewhat disturbed at the number of people who present, in a public fashion, an overly simplistic view of the memory management problem in an S-100 system, particularly in connection with CP/M Plus or MP/M. I was prompted to take action, in the form of this letter, by the last-straw effect when I read Andrew Bender's article, "Extended Memory Management for Older S-100 Computers," in the November 1983 issue of *Microsystems.* While the article is well written and well presented as far as it goes, it covers only the surface of the problem; and the solution presented, in the form of an actual circuit, is thereby inadequate except in the most advantageous of circumstances.

There are two problems which were not addressed in this particular article, and in others I have seen. The first is relatively simple and easily solved, but is nonetheless extremely important in all but the simplest of systems. This problem relates to the use of DMA (Direct Memory Access) devices.

When it has control of the bus, a DMA device must place the addresses it wishes to access onto the bus; this includes, for IEEE-696 compatible devices, the extended address lines. However, many extended addressing circuits presented do not take this into account and constantly assert the upper address lines. There will be no problem in systems with no DMA devices, but since there is a simple solution, it should be incorporated from the beginning to allow for the future addition of DMA devices. The S-100 bus provides a signal known as ADSB* (Address Disable). This signal is used in the transfer of control to a temporary master (DMA device) in order to disable the address lines of the permanent master (CPU). The output of an extended address latch should be gated by this signal. In Bender's Figure 1, the following changes should be made:

1) DS1 and MD should be tied to ground.
2) DS2 should be attached to ADSB*.
3) OS6* should go through an inverter then to STB.

The second, and more serious, problem is that of the "common memory" required by CP/M Plus and similar operating systems. The existence of this problem was only implied by the Bender article.

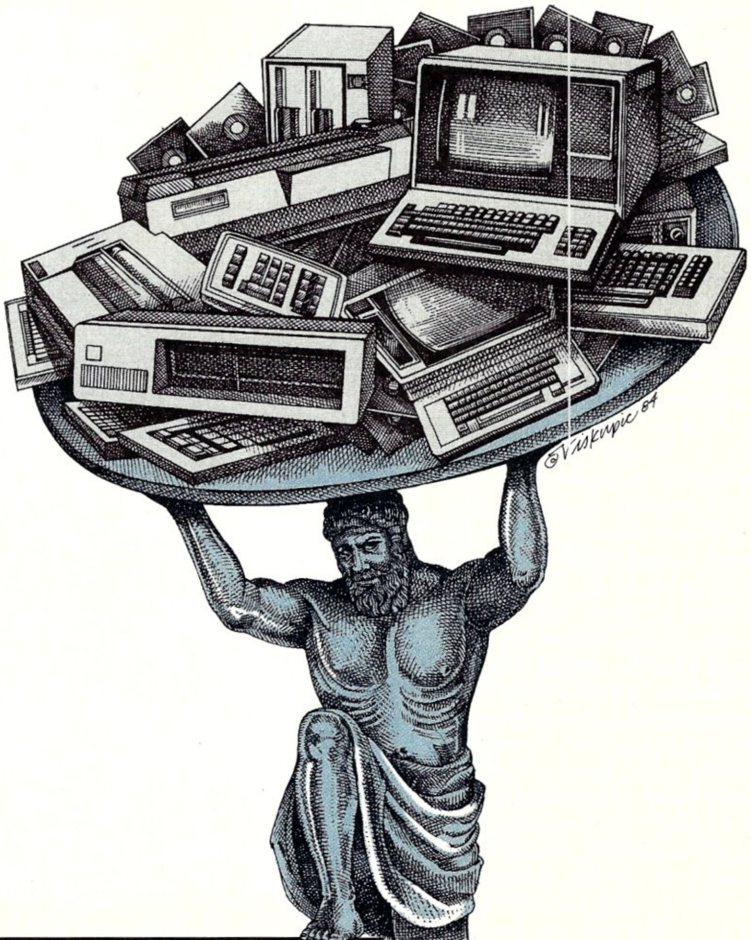
It was also implied that the problem would be dealt with by the memory boards themselves. While this may sometimes be the case, it will not always be so, especially with 64K and larger boards. In any event, it needs to be considered, so that it may be dealt with *somewhere.*

The problem is basically this: switching between banks of memory affects not only access of data, but access of program instructions as well. A solution to this problem is to have a section of the 64K address space refer to the same physical memory no matter which

(Continued on page 138)

Microsystems

MS-DOS: An Overview Part 1



MS-DOS 2.0, the enhanced version of 1.1, brings significant improvements in speed and functionality.

It also corrects some problems . . .

by William G. Wong

MS-DOS 2.0 is Microsoft's enhanced version of MS-DOS 1.1, which was made famous on the IBM PC as DOS 1.1. The new version is available on the IBM PC as DOS 2.0, along with a number of other 8088/8086 based machines. DOS 2.0 provides a significant number of improvements over its predecessor in terms of speed and functionality. It also corrects a number of problems encountered in 1.1.

This series on MS-DOS 2.0 will be divided into three articles in order to describe the inner workings in more detail. Part 1 is a general overview, discussing the major improvements and support programs. The new tree-structured file directory system is also explored. Part 2 will touch on the more intimate details of MS-DOS 2.0, including the programmer's interface and program file structure. Part 3 will cover field-installable device drivers.

Practically speaking, MS-DOS 2.0 is a superset of MS-DOS 1.x. For this reason, this review treats MS-DOS 2.0 as the set of enhancements added to the existing DOS 1.x. Some programs will run only under MS-DOS 1.x, and not under 2.0. This is usually because they access specific locations within the operating system that have changed or are eliminated in the new version. Copy-protected programs are typically in this class, since they bypass the normal system function call procedures.

PC-DOS 2.0 is IBM's rendition of MS-DOS 2.0. This article addresses the generic aspects of MS-DOS, but does not extend to a number of additional programs available with PC-DOS 2.0, such as GRAPHICS, which allows screen graphics to be sent to the printer, or CLS, which is used to clear the screen. These programs tend to be hardware specific.

This article is divided into three sections. The first section describes the new support programs supplied with DOS 2.0. The second section addresses the UNIX-style command line I/O redirection facility. The third section contains details on the multilevel file system also derived from UNIX, including the programs used to support it.

Batch files replace a lot of typing with a single command line.

New support programs

VERSION. When starting out it is always nice to know where the starting point is. To accomplish this, MS-DOS provides the version command, which simply prints out what version of MS-DOS is running. This will become more important as the number of revisions increases. Right now it just prints out version 2.00.

CONFIG.SYS. Although the new configuration file is not really a program in the conventional sense, it is probably the most important new support feature in MS-DOS 2.0. The configuration file is names CONFIG.SYS and is read by MS-DOS when the system is first initialized. The following is a sample configuration file.

```
BREAK=OFF
BUFFERS=2
FILES=8
DEVICE=ANSI.SYS
SHELL=A:COMMAND.COM
```

The BREAK option tells MS-DOS whether to check from the control-break key only during console I/O calls (OFF, default) or during all MS-DOS calls (ON). The latter is useful in stopping programs such as compilers, which do little console I/O. The BUFFERS option sets the number of 512-byte disk buffers to be used by the system. Data which is read or written to a disk passes through these buffers. MS-DOS is intelligent enough to keep track of what is in the buffers so that it will not have to access a disk if the information already resides in one of these buffers. Dramatic speedups can occur when using more buffers, especially when database programs are used. Of course, each buffer added reduces the amount of program space by a corresponding amount.

The FILES option specifies the maximum number of file "handles" that can be used at one time. Handles are 16-bit binary values that refer to a file accessed using the new MS-DOS UNIX-style system calls. These will be described in more detail in Part 2. The FILES option has no effect on the number of files that can be accessed using the conventional file control block (FCB).

The DEVICE option is used to load device drivers as part of the operating system, thereby making them available to the user. The new drivers can be used to access new peripherals or enhance access to existing peripherals. ANSI.SYS is a sample device driver supplied with PC-DOS 2.0, the IBM version of MS-DOS 2.0, which replaces the standard console driver. It enhances the console support by adding ANSI-compatible control functions, but the biggest advantage of the DEVICE option is that hardware vendors can now supply (with their hardware such as hard disks or local area networks) device drivers that do not modify the operating system directly, as was done with 1.x. Also, changing device drivers is now simply a matter of changing the CONFIG.SYS file.

Finally, there is the SHELL option. This allows the default shell, COMMAND.COM, to be replaced with a new user interface. The new shell can be more or less powerful than the default shell, depending upon the designer. It is now possible to present a menu-driven system to the user, who may never know that the base system is MS-DOS.

BACKUP. Although hard disk support with tape or cartridge disk backup has been available for MS-DOS systems, floppy disks are still the backup method for most systems. MS-DOS 2.0 now includes two new programs to help standardize this backup process. These programs are appropriately called BACKUP and RESTORE. As with most 2.0 software, there are a host of options using cryptic switches. For example, the following command line:

```
BACKUP C:WORKFILE\SOURCE.*
       A: /S /M /A /D:02-01-83
```

will use floppy drive A: to back up all files from drive C: in the directory WORKFILE with the filename SOURCE and any extension. All subdirectories (/S) will be included. Only files that have been modified (/M) will be copied, and the date associated with the file must be equal to or greater than February 1st, 1983 (/D:02-01-83). The new backup files will be added (/A) to any that already exist on the floppy. The syntax for RESTORE is similar.

RECOVER. Although directory corruption is rare when a system runs properly, there are always times when the system may crash—for example, when lightning strikes a nearby powerline. The RECOVER program gives some help in the event that the directory has been zapped. It will read the disk directory and try to reconstruct files. The resulting files are a mulitple of the block size used with the disk, so there may be additional garbage at the

MS-DOS

Continued from page 47

end of a file. Text files restored in this fashion can usually be fixed up with the aid of a word processor. The recovery process is not always complete, but at least there is now a prayer.

ASSIGN. Speaking of prayers, it has always been frustrating to use application programs that have been written to run only on specific disk configurations, and to find that these do not match the one on hand. The ASSIGN command has been added to alleviate some of these problems. ASSIGN allows logical drive names to be assigned to any physical drive. It is even possible to have multiple logical drives assigned to one physical drive. What use is this? Well, consider a home finance program that is written in assembler so it cannot be modified. It assumes that support programs will be on drive A: and data files on drive B:. This works great with two floppy drives, but we want to use drive E: which is a brand new hard disk. What to do? Enter:

```
ASSIGN A=E,B=E,C=A,D=B
```

and presto, the hard disk looks like drives A: and B: and the floppies can still be accessed as C: and D:. The application now runs fine, and the whole process was done without too much heartache.

VERIFY. The VERIFY is another command which helps to reduce the problems encountered with disk-based systems. Setting VERIFY ON makes MS-DOS read back all data which is written to a disk to make sure that the information really has been written. Although most disk systems are very reliable, many organizations require the more dependable operation provided by VERIFY. Most applications do not have this type of option, but, fortunately, MS-DOS 2.0 does. It can also be turned on and off at anytime as needed.

CTTY. Microsoft's enhancements are not confined to the disk area. The CTTY program has been added for those people who like to use more than one console—e.g., a CRT display for normal text processing and a printing terminal for recording the output of a particular program. It is even possible to communicate with a remote terminal through a modem, since CTTY accepts any character-oriented device as a parameter.

PRINT. The PRINT program is another nondisk enhancement that everyone has been waiting for. It provides a background print spooling facility to MS-DOS. Without it, you can watch your $5,000 computer sit and print for hours. Now it is possible to let it print and still use the machine for other purposes. The current incarnation allows up to 10 files to be placed into the print queue. Just make sure that you do not delete, rename or alter any files to be printed, remove the disk containing the files, or use the printer via a program—but what did you expect out of a single-tasking system? Even so, PRINT is an extremely useful option and probably the one which will get the most use.

BATCH. Finally, there is an enhanced batch processing facility, again akin to that of UNIX. Batch files are text fiels that can be created by most word processors. They are essentially programs with a limited vocabulary. The following batch program is an example of what the new batch file support can do.

```
REM This is a sample batch file
     which will print on the
REM console each file on drive A:
     with a designated
REM file type. The command line
     may contain any number
REM of file types or the command
     SWITCH. The latter
REM will allow you to change the
     diskettes. The last
REM name in the list should be
     DONE.
ECHO OFF
BREAK ON
:LOOP
IF %1==DONE GOTO END
IF NOT %1==SWITCH GOTO SKIP
REM Skip past SWITCH parameter
SHIFT
PAUSE Please put the next disk
     into DriveA:
:SKIP
REM Print all files with the
     appropriate file type
FOR %%F IN (*.%1) DO TYPE A:%%F
REM Shift in next parameter
SHIFT
GOTO LOOP
:END
```

The example is contrived, but it does show the various batch commands in action. The REM command is used to add remarks or comments to the file. These will not be executed, and removing them will not change the way the batch file runs except to speed up the execution; however, it helps to know how and why a program works. The comments in this file should give some idea of what is going on.

The ECHO command controls the

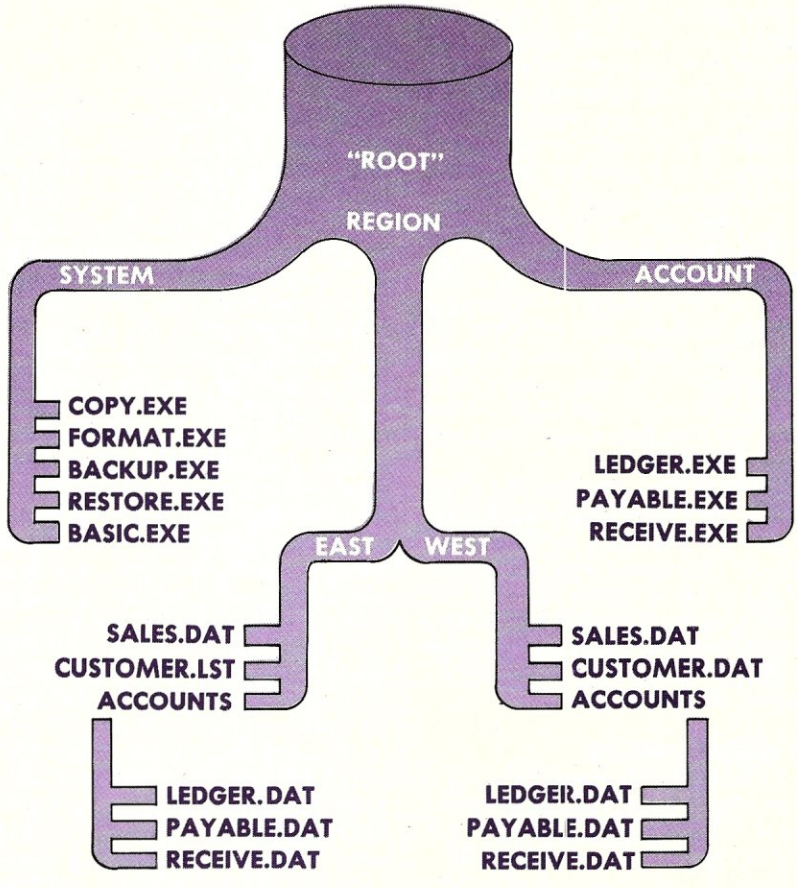


FIGURE 1.

display of the command file when it is run. In this case, turning ECHO off will let the system print all the lines up to and including the ECHO OFF line. The remaining portion of the batch file will not be printed; however, data printed by the PAUSE and TYPE programs will be displayed. ECHO can also be used to print any string (except ON or OFF) on the screen, too.

Moving down the program, the BREAK command performs the same operation as in the CONFIG.SYS file mentioned earlier. Next there are the IF and GOTO/label commands. These are the major additions to the batch file support. IF allows various conditions to be tested, the command following being executed only if the condition is true. The current conditions are limited to single string comparisons, checking to see if a file exists, and checking the result code from the previous program. In this case, the program is checking the first parameter (%1) to see if it is *not* the string SWITCH, in which case the GOTO command is executed. The GOTO command is followed by a "label" that must appear somewhere in the batch file preceded by a colon (:) which must be in the first column. Program execution will then shift to the line after the one containing the label. If the first parameter (%1) is SWITCH, the GOTO will not be executed and processing will continue on the line after the IF command; otherwise, the PROGRAM will GOTO label SKIP.

Next there is the PAUSE command, which prints the text following it and waits for a character. This allows the user to control the execution of the batch file. The PAUSE command in the example is used to let the user change disks before proceeding. The SHIFT command is used to logically move all the parameters on the initial command line one place to the left, throwing away the one initially on the left end. For example, assume the sample batch file is named LISTEXT and the following command line was used to start the batch file.

```
LISTEXT ASM LST SYM SWITCH BAK END
```

Table 1 shows the values of the parameters initially and after executing a number of SHIFT commands. Note that the name of the batch file is initially parameter 0 (%0).

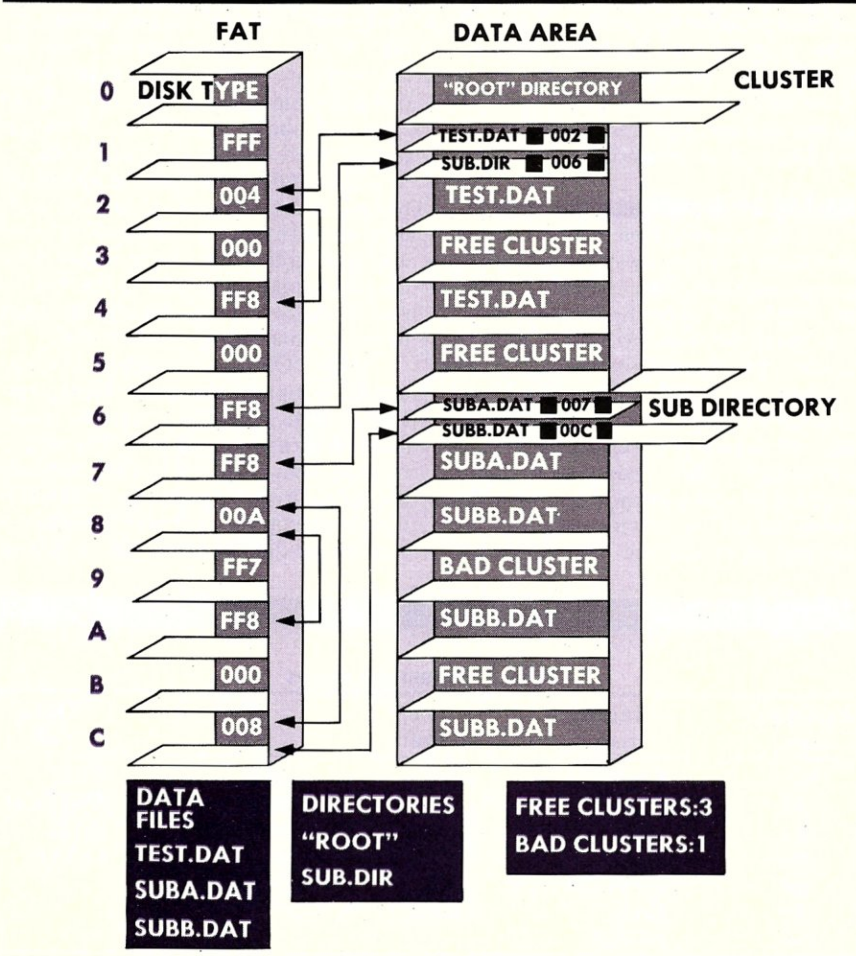


FIGURE 2. FILE ALLOCATION TABLE (FAT) EXAMPLE.

Note how the SHIFT function is used in this example. The first parameter (%0) is used through each iteration of the loop to hold either the filetype or one of the commands (SWITCH or DONE). The SHIFT function is used later to get the next command or filetype. The SHIFT function can also be used to access parameters in excess of the 10 (%0-%9) allowed by MS-DOS. For example, the sample batch program can take any number of arguments as long as they fit in one command line.

The FOR function is another very useful tool that is used in the sample program. The third parameter in parentheses can be a set of ambiguous filenames using the wildcard characters "*" and "?" separated by commas. FOR assigns the batch file variable (%%F), a matching unambiguous filename from the directory which matches a name in the set and then executes the command following DO, which usually contains the variable in one or more positions. This process is repeated until all matching names in the directory have been found and used. In this particular example, the batch file program will type all the files whose extension matches one of the parameters.

Batch files are extremely useful, and the MS-DOS implementation is very powerful. It can replace quite a bit of typing with one command line.

Command line I/O redirection

Command line I/O redirection is a facility used in UNIX which has been moved to MS-DOS. It allows a user to specify the source or destination of character data from a program when it is started from a command line. By default the source is the keyboard and the destination is the display screen; however, MS-DOS now allows these to be any text file.

There are a number of things to keep in mind, because the scheme is not foolproof. For example, programs must use the standard system input and output routines and must examine the command line with care. Also, MS-DOS will hang if a program is waiting for input after the end of the file has been reached. However, these are not always major concerns. The programs supplied with MS-DOS support the I/O redirection, as will many others. Now we'll talk about how and why it works.

The redirection is specified in one of two ways. The first is to place in front of a filename one of three prefixes: " < ", " > ", or " > > ". The first (<) indicates that the file is to be used as the standard input source. The other two indicate that the file is to be used as the standard output. These differ in the way that the

Continued from page 49

output file is manipulated. The ">" prefix will create a new file or overwrite an existing one. The ">>" prefix will append the new information to the end of an existing file; otherwise it creates a new file. The following are some examples using these prefixes. Note that, since MS-DOS allows devices as filenames, it is possible to say "direct output to the printer (PRN)."

```
DIR >PRN
DIR >DIRLIST
SORT >UNSORTED.TXT >SORTED.TXT
SORT >EXTRA.DAT >>SORTED.TXT
TYPE FOOTER.TXT >>SORTED.TXT
```

The second way to specify redirection is to use what has been called a "pipe." In this case, more than one program is started with one command line. UNIX does this in parallel, but MS-DOS does it sequentially. The character "|" is used to separate the commands. For example:

```
DIR | FIND ''.ASM'' | SORT >
          LIST.ASM
```

In this example the three programs are DIR, FIND, and SORT. This one command line is equivalent to the following five command lines, which are much more confusing:

```
DIR >%PIPE0.$$$
FIND ''.ASM'' >%PIPE0.$$$
              >%PIPE1.$$$
ERASE %PIPE0.$$$
SORT <%PIPE1.$$$ >LIST.ASM
ERASE %PIPE1.$$$
```

Essentially, "|" connects the standard output file of the program to its left to the standard input file of the program to its right. In MS-DOS, this file is an actual file placed on the default drive. In UNIX, the programs are connected directly together and no disk space is required. Even so, using pipes with MS-DOS can save a lot of typing and make commands much easier to understand.

Another thing to note is that MS-DOS does not support multipole designations for the standard input and output files. Therefore, the ">" and ">>" file prefixes should not be used to the left of a pipe ("|") and the "<" file prefix should not be used to the right. Also, there should not be multiple occurrences of any redirection characters within a command.

Programs written especially for use with pipes are called "filters." A filter program is one which uses the standard input to generate data for the standard output, using parameters on the command line. MS-DOS comes with three

Command I/O redirection— a UNIX facility— has been moved to MS-DOS.

filters: FIND, SORT, and MORE. FIND takes a parameter string in double quotes. Any text line containing the string will be sent to the standard output file. SORT reads each input line and sends these lines to the standard output in ascending order. MORE reads the standard input file and prints out one screenfull at a time, waiting for keyboard input after typing "—MORE—" at the bottom of the screen.

I/O redirection and filters are tools that programmers will like to use. They can be applied to applications especially when coupled with the batch file facility. Most people using general applications will not need or understand the power of these features; however, they are nice to have around.

MS-DOS 2.0 hierarchical file system

Although the idea behind the hierarchical file system has been around for quite some time, there have been a limited number of implementations on micro-based systems. MS-DOS now includes the support for all disk-based systems. Each disk drive has a "root" directory that has no name. This "root" directory can contain either data files or subdirectory files. Data files are those we have all come to know and love. The subdirectory files are the enhancement. Instead of containing data, these files are directories just like the "root." They can contain data files as well as subdirectories.

Figure 1 shows a graphic view of a multilevel file system. Note that each logical disk drive has a "root," which means that a multidrive system does not have just one tree but a whole forest. In fact, hard disks which are partitioned into many logical disks have their own grove of trees.

Although there is no theoretical limit on the number of drives or the depth of a tree, MS-DOS 2.0 does place some practical limits on both. MS-DOS 2.0 limits the number of logical disk drives to 63, and it limits the "path" name to 63 characters. A path name is the list of directory names and the name of the file in the lowest subdirectory. The names are separated by a backslash (\). The following are examples of path names.

```
\SYSTEM\GRAPHIC\DEMOS
\COMPILER\PASCAL\PASCAL.EXE
\ACCOUNT.NEW\SUMMARY.LST
\SALES\REPORTS\REGION\EAST.BAK
```

All filenames have the same format as in MS-DOS 1.x—i.e., a filename of up to eight characters, followed by an optional period (.) and file extension up to three characters long, for a total of 12 characters. This means you almost get four levels of subdirectories if all the filenames are of maximum length. Since they are typically much shorter, you can count on about seven levels, which is usually more than anyone ever uses.

MS-DOS also allows the parent directory and the current directory to be used in path names. These directories are referred to using two periods (..) for the parent and one period (.) for the current directory. These names are used just as normal subdirectory names.

MS-DOS has a number of commands to support the hierarchical directory system. MKDIR is used to create subdirectories, while RMDIR is used to remove them. RMDIR can be used only if the subdirectory is empty. CHDIR is used to change the current default directory on the default drive. TREE is used

Table 1. Parameter values

Parameter name	Initial value	One shift	Two shift	Three shift
%0	LISTEXT	AMS	LST	SYM
%1	ASM	LST	SYM	SWITCH
%2	LST	SYM	SWITCH	BAK
%3	SYM	SWITCH	BAK	DONE
%4	SWITCH	BAK	DONE	
%5	BAK	DONE		
%6	DONE			

to display the directory at multiple levels, whereas DIR only displays one level.

The multilevel directory system does complicate matters slightly, since programs are often partitioned in a fashion other than by the way they are used. For example, all the compiler files may be in one subdirectory, while general programs are in another and linker programs in a third. Trying to use all three at the same time means that the default directory must be changed when accessing a different program, which can be quite tedious. MS-DOS supplies the PATH command to solve this problem. The parameters to PATH are a list of directories to search if a program is not found in the current directory. The following is an example to fix the aforementioned situation.

```
PATH A:\SYSTEM,B:\COMPILER\PASCAL,
     B:\LINKER
```

Note that drive names may be included with the PATH parameters, allowing programs to be spit over a number of different drives. This is especially useful for floppy based systems, and it also comes in handy for hard disk based systems.

Although MS-DOS 1.x disks can be used with MS-DOS 2.0, the reverse is not true, because 1.x does not support the multilevel file structure. The remaining portion of this section deals with the new data structures on the disk used by MS-DOS 2.0 to support subdirectories and also partitioned hard disks. Note that the latter implementation is taken from the IBM PC-DOS 2.0 implementation, which may differ from other implementations. However, it is a good reference point to work from.

A physical disk may be partitioned into one or more logical disks. Each logical disk contains one logical multilevel directory system. This is implemented by partitioning the disk into the variable size sections, shown below.

MS-DOS disks are allocated in 512-byte blocks, and each section is

Boot Section
File Allocation Table 1
File Allocation Table 0
"Root" Directory
Data Area

Implementing MS-DOS 2.0 on the IBM PC has made it a de facto standard.

made up of one or more of these blocks. All sections are a fixed size for a particular disk implementation and do not vary. Data files and subdirectories are located in the Data Area and allocated dynamically. This process is discussed later in more detail.

The Boot Section on the boot disk is used only when the system is first initialized, and is accessed by the boot program normally located in read-only memory (ROM). This section usually contains an extended boot program which knows the MS-DOS disk structure and how to load the main resident parts of MS-DOS contained in the files DOS.COM (Disk Operation System) and BIO.COM (Basic I/O system). These, in turn, check the configuration file, CONFIG.SYS, and load the shell program, which is usually COMMAND.COM. In theory, only one disk need have the Boot Section. However, in practice, all disks have the Boot Section allocated for the sake of consistency.

There are two File Allocation Tables (FATs) for redundancy. They are normally identical except when they are being updated or if they have been corrupted in some fashion. Keeping a redundant copy of this table allows the RECOVER program to be more efficient when trying to restore a corrupted disk. Only one copy of the FAT is kept in memory by MS-DOS for each disk, and the two copies on the physical disk are updated one after the other, as required.

The FAT is referenced by the "root" directory and its subdirectories. Each file has a directory entry in either the "root" directory or one of the subdirectories. This entry refers to the FAT, which has a one-to-one mapping between it and the data area. Each directory entry is 32 bytes in length and has the format shown in Table 2.

The status flag in the filename field is used to indicate whether the directory entry can be used for a new file (00 and E5) or whether it is allocated as a directory entry (2E) or as a normal file entry. There are two indicators for unused entries for efficiency reasons. The directory is initially filled with zeros (00 is one of the flags) and the directory is always filled from the front to the back, making first use of entries flagged with E5 for new files. Therefore, directory searches can terminate whenever an entry with a zero flag is found, since no files can be allocated past this point. The references to the parent directory are used to support the ".." path facility.

All files are dynamically allocated, and the FAT is used to keep track of the allocation. The FAT is organized as an array of 12-bit entries (1.5 bytes/entry). The first two entries (3 bytes) are reserved for the disk description. In these entries, the last two bytes are always 0FFFF hex; the first byte is described as shown in Table 3.

All remaining FAT entries are used to designate free space or allocation for a particular file. Actually, reserving the first two entries wastes no space, since this is used for the "root" directory area. Each FAT entry corresponds to one "cluster" in the Data Area. Ownership of an FAT entry by a file indicates that the matching "cluster" in the Data Area is part of that file. The FAT index in the directory entry for a file indexes the first FAT entry for the file. This entry contains the index of the next FAT entry for the file and so on. The last FAT entry in this chain contains a value from FF8 hex to FFF hex. Any data record in a file can be found by computing the "cluster" index containing the data and then searching the file's alloction chain in the FAT to find the appropriate "cluster." The data is contained in the sectors of the corresponding disk area.

MS-DOS keeps track of unallocated clusters by placing a 000 hex in the entry. Bad sectors that will generate hardware errors are flagged with an FF7 hex so they will not be used. Figure 2 shows a portion of a sample FAT with allocated, unallocated and bad sector entries.

The term "clusters" has been used in the previous discussion but not really defined. Essentialy, a cluster is a fixed-size block of disk sectors. A cluster may be one or more disk sectors, depending upon the implementation. Hard disks tend to have larger clusters than floppy disks because of their greater storage capacity. The disk capacity comes into play because the maximum number of entries in an FAT is 4,096 (less reserved locations 000, 002 and FF0-FFF hex). The cluster size is therefore usually larg-

MS-DOS

Continued from page 51

er than the disk capacity divided by 4,096. The maximum size of a FAT is 6,144 bytes, or 12 512-byte sectors assuming all 4,096 entries are needed.

Increasing the size of a file entails finding a free cluster and adding it to the end of the file's FAT chain. A file is deleted by changing each entry in its FAT chain to zero and placing an E5 hex in the directory entry.

Note that the cluster size is essentially the minimum file allocation size. Even if a file uses only one byte for its data, it is allocated one cluster, and the unused space cannote be accessed by another file. Even so, the ability to dynamically extend the size of a file greatly outweighs any possible waste of space.

One interesting point about the hierarchical file system under MS-DOS 2.0 is that there no longer seems to be any restriction on the number of files a disk may contain, assuming there is sufficient space. Although the "root" directory is a fixed size, all subdirectories are normal files whose length can be dynamically extended. Therefore there is no limit on the number of files a subdirectory can contain; however, a limit is imposed by the FAT structure; this limit is not always mentioned. Remember that the FAT is limited to 4,096 entries, and any file must use at least one FAT entry. Since 18 are reserved for various purposes, that leaves 4,078 entries. This means that you can have 4,078 files that each use one cluster, one file that uses all 4,078 clusters, or soemthing in between. In any case, this limit will be encountered only on a hard disk with a large Data Area—but beware!

MS-DOS 2.0 can be told to use a number of buffers for disk caching by using parameters in the CONFIG.SYS file as mentioned before. It uses these buffers for data and directory information as well as the FAT tables. This means that a system with few buffers can still access a hard disk with a large directory and FAT. Increasing the number of buffers means that more of the directory and FAT can be resident. This makes MS-DOS 2.0 fast when accessing sequential files because the location of subsequent clusters can be found by looking at the memory-resident FAT. Unfortunately, the FAT access method falters under random access use, especially when a number of disk drives are in use at the same time and few buffers are available.

The random access mode means that subsequent file references can be made at any point in the file. Finding this point must always be done by sequentially searching the FAT. Since a large file usually requires a large FAT, randomly accessing the FAT means that a large number of buffers are needed or a number of additional disk accesses are needed to find the desired cluster. This time/space tradeoff can be critical in some large database applications. In these instances, it may be helpful to buy more memory and increase the number of buffers available to MS-DOS.

Overall, the new file system represents a major improvement over MS-DOS 1.x. It is more flexible and easier to use than its predecessor and offers a number of tradeoffs that can be made by the user. We will have to wait and see if MS-DOS 3.0 removes some of the limitations in MS-DOS 2.0.

Summary

MS-DOS 2.0 as a whole is a major improvement over 1.x. Its implementation on the IBM PC has made it a de facto standard to which most major manufacturers conform. Most of the enhancements will be exploited only by the designers and programmers, but users will benefit by the improvements in the applications provided by 2.0. Part 2 will discuss how to gain access to these enhancements at the program level.

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

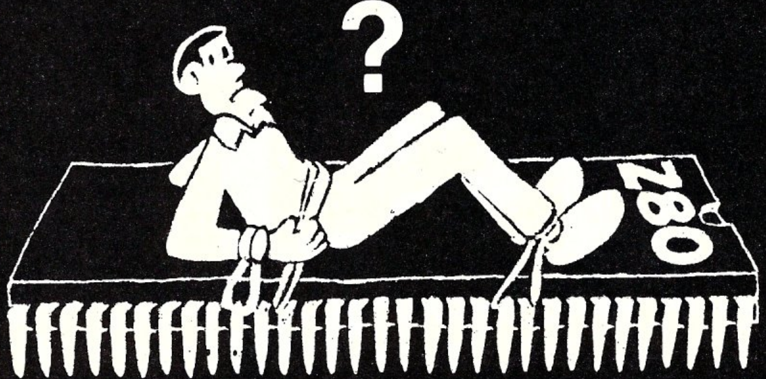
Table 2. Directory entry formats

Offset	Size	Description	
0	8	Filename, of which the first byte is a status flag	
		Byte 0	*Description*
		00 hex	Entry never used
		E5 hex	Entry erased
		2E hex	Directory entry; if the next byte is also a 2E hex, the FAT index points to the parent directory
		other	First character of the filename
8	3	File extension	
11	1	File attribute	
		Bit	*Description*
		0	Read only
		1	Hidden file
		2	System file
		3	Filename is volume label (only in "root" directory)
		4	Subdirectory
		5	Not archived
		6-7	Reserved
12	10	Reserved	
22	2	Time of creation or last update	
		Bits	*Description*
		0-4	Two-second increments
		5-10	Minutes (0-59)
		11-15	Hours (0-23)
24	2	Date of creation or last update	
		Bits	*Description*
		0-3	Day (1-31)
		5-8	Month (1-12)
		12-15	Year offset from 1980
26	2	FAT index	
28	4	File size in bytes (least significant byte first)	

Table 3. Disk description byte

Hex value	Description
FF	Double-sided 8 sectors/track disk
FE	Single-sided 8 sectors/track disk
FD	Double-sided 9 sectors/track disk
FC	Single-sided 9 sectors/track disk
F8	Fixed hard disk

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 35 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.

MSPRO: MS-DOS on the S-100 Bus

MSPRO brings some UNIX-like features to the CompuPro 8085/88 CPU

by Andrew L. Bender

Recently MS-DOS has been honored by Digital Research in an indirect fashion: they have made their software compatible with MS-DOS. This is not a bad idea because MS-DOS is to the 8086/88 what CP/M is to the 8080/85/Z80. Recently I had the opportunity to test the Godbout 8086/87 CPU board. I brought up CP/M-86 under that system and tested it. A few weeks later I got a review copy of MSPRO, an MS-DOS adapted for an S-100 system using the CompuPro 8085/88 CPU. The adaptation is by Computer House. It includes a new PROM for the Godbout Disk-1 Controller, a modified Lomas Data Products LDP-72 Disk Controller Board, a 5¼″ disk drive with a cable to connect it to the LDP-72, as well as Microsoft's MS-DOS and documentation. In case you don't already own one, the GO 86 PROM used to initiate operation of the 8085/88 board is also included.

System considerations

The Lomas disk controller board (DP-72) can drive both 8″ and 5¼″ disks at the same time. Thus it is possible to use only a Lomas LDP-72 in a system with mixed disk sizes. Computer House has modified the LDP-72 so that it works properly with MS-DOS. The modifications consist of some new wire jumpers on the board and a new IC in a spare socket. The new PROM for the Godbout Disk-1 board allows you to boot with either MS-DOS or CP/M. I could cross boot any system I wanted to without problems: CP/M-86, CP/M-80 and MS-DOS all came up properly with the new PROM.

You are expected to supply the CompuPro 8085/88 system with enough memory (at least 192K if you want to do anything serious with the system) a System Support Board for console I/O, and a clock as well as the address for the GO-86 PROM. If you want to use a printer—and who doesn't?—you will need an Interfacer 3 or 4 card, which I do not own.

That is a lot of hardware. I had done business with Computer House before and they sent all of this stuff to me for evaluation, which they expected me to do in three weeks. Alas, I make my living doing other things, so I couldn't possibly do an evaluation properly in that time.

I thought it would take me at least one week to get the hardware up and running but I was wrong—it took 20 minutes. One thin manual contained step-by-step instructions for getting the system up and running without a nervous breakdown. Not only did the system run after the steps were followed, but the manual didn't read like "Advanced Topics in Brain Surgery."

One thing you must do is set the switches on your Disk-1 controller card to indicate that you are using the System Support Board serial port as the console device. The manual doesn't say so, but the first thing you should do when your MSPRO system is up and running is to make a backup disk. I almost forgot this basic safety procedure and might have been left with a non-bootable system.

MS-DOS 2.0 has many features not present in the first 1.0 system. It offers multilevel named directories, a rudimentary system of pipes and filters, I/O redirection and what will probably be a very powerful queueing mechanism for I/O requests. None of the I/O dispatching routines are yet implemented—but they will be, and when they are MS-DOS will have the capability for multiprogramming and concurrent operations. MS-DOS 2.0 is a bit like XENIX, Microsoft's version of UNIX.

The system did exactly what the manual said it would, and I had a running MSPRO/MS-DOS version 2.0, except I couldn't print anything. I looked for the assembly language driver routines and installation material which, according to the manual, were supposed to be included and—you're right—they were not there.

There was a file on the disk which described the device drivers and how they work under MS-DOS. There were instructions for writing the drivers, and I listed them out over and over on my CRT, reading them until I had an idea on how to write a printer driver for my Interfacer 1. I must admit there were some unclear areas which I resolved with Chris Cochrane, the person who wrote the MSPRO drivers. Once these areas were cleared up, the print driver was installed in several more marathon debugging sessions lasting at least nine hours.

I was glad to have this experience because suffering builds character! I had to read the manuals carefully. There are no instructions in any of the manuals as to how you are to interface with MS-DOS from a user program. I know that in MS-DOS 1.2 you use interrupt 21 and some other interrupts, but that is not even mentioned in the 2.0 manuals. Absent also are the necessary data structures to communicate with MS-DOS.

MS-DOS is a bit like XENIX, Microsoft's version of UNIX.

Later, the folks at Computer House confided that this was the very first version 2.0 MSPRO system that they had shipped and that the manuals were not in "sync" with the system. This is not great, but if you want the latest working system, it is tolerable.

The editor and assembler

EDLIN, a thinly disguised version of ED, was equally irritating. There are lofty descriptions of how EDLIN is powerful (translates to "almost unusable") and useful. I suppose that when it is the only editor you have, it is all of these things. There are better editors, but I didn't have any of them, so I did all my work with EDLIN and was not above the muttering of an occasional obscenity. EDLIN works with some mysterious function keys which are not described. The MS-DOS manual suggested trying the keys on the keyboard to find those keys that subserved the special functions. Lots of luck! Why didn't they just give the sequences so that I wouldn't have to waste my time? So much for EDLIN.

MASM is the 8086 version of M-80 and is well documented and implemented. It works well and provides a wide range of useful directives (pseudo-operations). There is a macro facility that works like the Intel assembler macro generator in MAC and M-80, the 8080 assemblers. A wide range of listing directives allow for convenient debugging of not only macro-generated code but also the variable-length instructions that can cause problems if the operands are not defined properly. Strict adherence to the Intel mnemonics makes this assembler a proper development tool. Digital Research's assemblers (RASM86 and ASM86) do not support macros nor the complete set of Intel mnemonics—they cheat on FAR and NEAR calls and returns.

Writing MS-DOS device drivers

I assembled my driver with MASM. I was surprised at the speed of MASM. It was running under MS-DOS and was at least 20% faster than ASM86 running under CP/M-86. It is *big*! A lot bigger than ASM86, but who buys a "small" 8086 system? The system requires that the I/O drivers be in .EXE format so you have to process them with MSLINK—the linkage editor. LINK converts relocatable .OBJ files into .EXE files. An .EXE file is an absolute memory image of a program with enough information to allow for some positioning in memory when it is loaded. This was fast, too, but then again there were no other relocatables being collected.

After creating my print drive in .EXE format, I just needed to link it into the device driver chain by mentioning it in the CONFIG.SYS file. I did this and to my surprise (and I am sure to yours, too) it didn't work. Because CONFIG.SYS is processed during booting, the system would not come up. Thanks to the miracle of backup copies I could boot my backup disk, rewrite my CONFIG.SYS and continue my development.

```
THE MICROSOFT MACRO ASSEMBLER          10-03-83      PAGE     1-1
INTERFACER 1 DRIVER

                                 TITLE   INTERFACER 1 DRIVER
                          ; COMPU-PRO INTERFACER 1 DRIVER FOR
                          ; DIABLO 1643 PRINTER ON CHANNEL 2
                                 PAGE    60,132
                          ;
                          ;PROGRAMMED BY A. L. BENDER, M. D.
                          ;SEPTEMBER 27, 1983
                          ;RUNS UNDER MSDOS 2.0
                          ;
                          ;DEVICE = PRT
                          ;
                                 ASSUME  CS:CSEG
        0000              CSEG   SEGMENT                       ;ALL SEG REG'S SET =

                          ;THIS CODE IS ILLUSTRATIVE MORE THAN FUNCTIONAL
                          ;TO ALL YOU PURISTS, TAKE HEART-IT IS PROVIDED AS
                          ;A SKELETON SO THAT YOU CAN CODE YOUR OWN ROUTINES
                          ;BETTER AND IN LESS TIME THAN I DID THIS ONE.

                          ;SERIAL PORT ASSIGNMENTS
                          ;FOR INTERFACER-1 (WE USE THE SECOND PORT FOR THE PRINTER)
                          ;
```

MSPRO

Continued from page 55

Adding any additional I/O drivers to MS-DOS 2.0 means that you will have a separate file for every additional driver or complex of drivers installed. This could mean a lot of driver files in a complex system. For example, if you add a hard disk, a semiconductor drive and another printer, you will need three more files. There is no BIOS as in CP/M, so that customizing your existing hardware is not trivial. As an example, I wanted to bring up my console on the Interfacer 1 rather than the System Support Board. Forget that, it's built into MS-DOS. Each nonresident I/O driver is installed during booting by the system initialization function. This program reads a configuration file and installs each I/O device as specified in CONFIG.SYS, which is an ASCII directive file. If a device is installed from CONFIG.SYS that has the same name as a resident device, the device configured in CONFIG.SYS overrides the resident device.

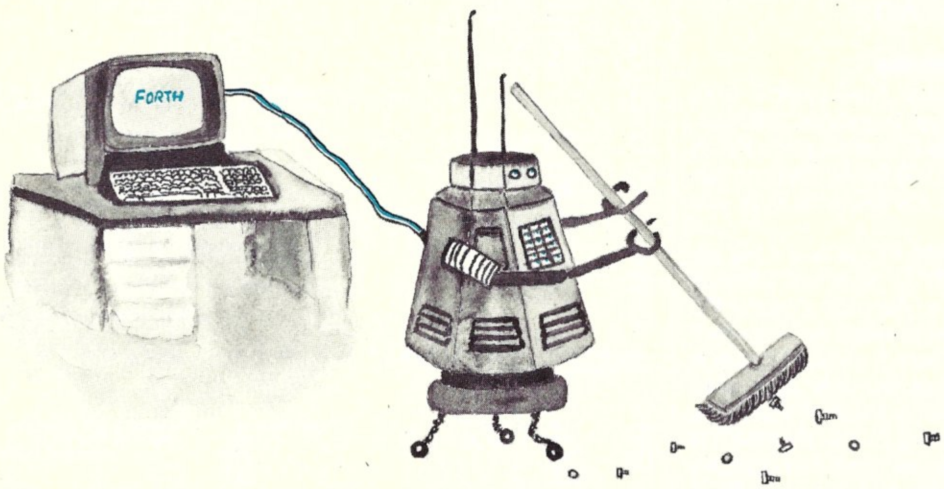
Set the switches on the Disk-1 controller card to indicate you are using the System Support Board serial port.

Reading the instructions in DEVDRIV.DOC—a file of documentation on the disk—is supposed to impart sufficient knowledge to allow anyone to write a driver and install it. There is one drawback—if a driver is installed, it is hard to test. If it cannot be installed, the system either bypasses the driver or just hangs up. For those of you seeking sleepless nights, read no more; just try it on your own. If you want to save time, I have included a skeleton driver (Listing 1) and the following advice:

1. Code the entire driver in the code segment bank.
2. ASSUME CS:CSEG.
3. Pay little attention to the drivel

```
        = 0002                       DATAPORT EQU     2        ;PORT 2=DATA PORT ON INTERFACER1
        = 0003                       SERSTAT EQU      DATAPORT+1

                                     ;        DRIVER HEADER TABLE
        0000  FF FF FF FF                     DD      -1       ;POINTER TO NEXT DEVICE IN CHAIN
        0004  8000                            DW      08000H   ;CHARACTER DEVICE IOCTL BIT
        0006  002F R                          DW      CHREQ    ;STRATEGY ROUTINE ENTRY
        0008  003A R                          DW      TRAP     ;INTERRUPT ENTRY
        000A  50 52 54 20 20 20               DB      'PRT     ' ;DRIVER NAME
              20 20

                                     ;
        = 0000                       SRHBASE EQU      0        ;SRH PACKET BASE
        =                            SRH      EQU     0[BX]    ;SR HEADER
        =                            UNITCOD EQU      1[BX]    ;UNIT CODE
        =                            FUNC     EQU     2[BX]    ;FUNC CODE
        =                            STATUS   EQU     3[BX]    ;16 BIT STATUS CODE
                                     ;LOW SEVEN BITS OF STATUS CODE ERROR CODES:
                                     ;                00=WRITE PROTECTION VIOLATION
                                     ;                01=UNKNOWN UNIT
                                     ;                02=DEVICE NOT READY
                                     ;                03=UNKNOWN FUNC
                                     ;                04=CRC CHECK
                                     ;                05=BAD DRIVE STRUCTURE REQUEST LENGTH
                                     ;                06=MEDIA SEEK ERROR
                                     ;                07=UNKNOWN MEDIA
                                     ;                08=SECTOR NOT FOUND
                                     ;                09=PRINTER PAPER FAULT
                                     ;                0A=WRITE FAULT
                                     ;                0B=READ FAULT

 THE MICROSOFT MACRO ASSEMBLER          10-03-83      PAGE    1-2
INTERFACER 1 DRIVER

                                     ;                0C=GENERAL FAILURE

        =                            MDESC    EQU     13[BX]   ;MEDIA DESCRIPTOR
        =                            XFRADR   EQU     14[BX]   ;TRANSFER ADDRESS
        =                            COUNT    EQU     18[BX]   ;COUNT OF BYTES TO TRANSMIT
        =                            NRUNITS EQU      13[BX]   ;NUMBER OF UNITS RETURNED BY INIT
        =                            BRKADR   EQU     14[BX]   ;BREAK ADDRESS
        =                            SEGADR   EQU     16[BX]   ;SEGMENT ADDRESS
                                     ;
                                     ;DATA STRUCTURES FOR DRIVER IN CONTROL SEGMENT
                                     ;

        0012  00A7 R                 MEMLOC: DW       ENDRVR   ;LOCATION OF END OF DRIVER
        0014  ????                           DW       ?        ;SEGMENT INFORMATION
        0016  ????                   PACKET: DW       ?        ;STATIC REQUEST HEADER ADR
        0018  ????                           DW       ?        ;SET TO ES: ON ENTRY -> DS: ON TRAPPING

                                     ;I/O WAIT (STATUS) SUBROUTINE

        001A  8A E0                  IOWAIT: MOV      AH,AL    ;SAVE AL FROM RUIN
        001C  E4 03                          IN       AL,SERSTAT ;GET SERIAL PORT STATUS
        001E  86 C4                          XCHG     AL,AH    ;FLIP REGISTERS
        0020  C3                             RET               ;RETRN (AH=STATUS BITS)

                                     ;SUBROUTINE TO OUTPUT A CHARACTER TO THE PRINTER

        0021  E8 001A R              OUTPUT: CALL     IOWAIT
        0024  80 E4 41                       AND      AH,041H ;CC BIT AND TBMT BIT
        0027  80 EC 01                       SUB      AH,1     ;TBMT BIT
        002A  75 F5                          JNZ      OUTPUT
        002C  E6 02                          OUT      DATAPORT,AL
        002E  C3                             RET

                                     ;CHAIN REQUEST FOR PRINTING (SOMEDAY)

                                     ;ENTER WITH ES:[BX]->SRH STRUCTURE

        002F                         CHREQ   PROC     FAR
        002F  2E: 89 1E 0016 R               MOV      WORD PTR CS:PACKET,BX ;SAVE STATIC REQUEST HEADER
        0034  2E: 8C 06 0018 R               MOV      WORD PTR CS:PACKET+2,ES ; SAVE SEGMENT TOO
        0039  CB                             RET               ;RETURN TO MSDOS
        003A                         CHREQ   ENDP

                                     ;INTERRUPT PROCEDURE FOR FUNCTIONS

        003A                         TRAP    PROC     FAR      ;FAR INTERRUPT PROCEDURE
        003A  50                             PUSH     AX       ;SAVE EVERYTHING
        003B  53                             PUSH     BX
        003C  1E                             PUSH     DS       ;SAVE IT FROM RUIN (=ES:)
        003D  2E: C5 1E 0016 R               LDS      BX,DWORD PTR CS:PACKET  ;RETRIEVE PACKET ADR
        0042  8A 47 02                       MOV      AL,BYTE PTR FUNC ;GET FUNCTION BYTE FROM SRH
        0045  84 C0                          TEST     AL,AL    ;CHECK FOR INITIALIZATION
        0047  74 5E                          JZ       ENDRVR   ;INITIALIZATION CODE BEYOND DRIVER
        0049  51                             PUSH     CX       ;PUSH FLAGS AND CX ONTO STACK

 THE MICROSOFT MACRO ASSEMBLER          10-03-83      PAGE    1-3
INTERFACER 1 DRIVER

        004A  9C                             PUSHF
        004B  FC                             CLD               ;SET TO AUTO INCREMENT
        004C  3C 08                          CMP      AL,8     ;CHECK FUNCTION
        004E  74 07                          JZ       WRITE    ;OUTPUT DATA TO PRINTER
        0050  73 20                          JNC      NEXT1
        0052  B8 8003                UNKCOM: MOV      AX,8003H        ;SET UNKNOWN COMMAND STATUS
        0055  EB 45                          JMP      SHORT RETURN
        0057  8B 4F 12               WRITE:  MOV      CX,WORD PTR COUNT ;NUMBER OF BYTES TO TRANSFER
        005A  8B C1                          MOV      AX,CX
        005C  85 C0                          TEST     AX,AX    ;CHECK FOR NULL CASE
        005E  74 0D                          JZ       WRRET    ;WRITE RETURN
        0060  56                             PUSH     SI
        0061  06                             PUSH     ES       ;SAVE EXTRA SEGMENT
        0062  C4 77 0E                       LES      SI,DWORD PTR XFRADR ;ADDRESS OF DATA BUFFER
        0065  AC                     WRITE1: LODSB             ;GET BYTE TO TRANSMIT
        0066  E8 0021 R                      CALL     OUTPUT   ;SEND BYTE TO DEVICE
        0069  E2 FA                          LOOP     WRITE1   ;LOOP TILL ALL BYTES SENT
        006B  07                             POP      ES
        006C  5E                             POP      SI
        006D  B8 0080                WRRET:  MOV      AX,80H   ;DONE, NO ERRORS
        0070  EB 2C                          JMP      SHORT RETURN+2 ;RETURN TO MSDOS
        0072  3C 0A                  NEXT1:  CMP      AL,10    ;10=OUTPUT STATUS
        0074  72 DC                          JC       UNKCOM   ;UNKNOWN COMMAND
        0076  74 15                          JZ       OUTSTAT ;OUTPUT STATUS ROUTINE
        0078  3C 0B                          CMP      AL,11
        007A  74 F1                          JZ       WRRET    ;FLUSH OUTPUT BUFFER (FLUSHED ALREADY)
                                     ;IOCTL OUTPUT ROUTINE
        007C  8B 4F 12                       MOV      CX,WORD PTR COUNT
```

FORTH GIVES YOU TOTAL CONTROL



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

FORTH: for Z-80®, 8080, 8086, 68000, and IBM® PC

(Complies with the New 83-Standard)

- ***FORTH* programs are instantly portable across the four most popular microprocessors.**
- ***FORTH* is interactive and 20 times faster than BASIC.**
- ***FORTH* programs are highly structured and easy to maintain.**
- ***FORTH* provides 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 distributed as turnkey COM files with no license fee.**
- ***FORTH* Cross Compilers are available for ROM'ed or disk based applications on most microprocessors.**
- **Custom programming, consulting, and educational services available.**

FORTH Application Development Systems include interpreter/compiler with virtual memory management and multi-tasking, assembler, full screen editor, decompiler, utilities, and detailed technical 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

FORTH Cross Compiler allows you to customize 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, \$300.00.

FORTH Native Code Compilers

For Z80 FORTH and CP/M	\$100.00
For 8086 FORTH and CP/M-86	\$200.00
For IBM PC and PC-DOS	\$200.00

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; B-Tree index manager, \$125.00; B-Tree index and file manager, \$200.00; QTF + Screen editor for IBM PC, \$100.00; Quad Integer Math Pack, \$25.00.

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

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 Sage. Most other formats can be special ordered. Dealer inquiries invited.

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; PC/FORTH and PC/GEN are trademarks of Laboratory Microsystems Inc.

LMI

Laboratory Microsystems Incorporated

4147 Beethoven Street, Los Angeles, CA 90066

*Phone credit card orders to **(213) 306-7412***

VISA® MasterCard

CIRCLE 27 ON READER SERVICE CARD

MSPRO

Continued from page 56

in the initialization section about returning the last address in the driver in DS:DX.

System performance

I made some speed tests with MSPRO. It is fast. It is about 30% faster than MS-DOS 1.2 running on a PC. It is at least this much faster than concurrent CP/M for the PC with only one task.

Reading IBM PC disks is a big advantage of MSPRO. So what? Most of this software is so dependent on the PC that it won't run properly on any other machine. Yes, there is quite a bit that is not PC-dependent, but then again most of that is in multiple formats. I used the little disk only a few times. My MSPRO disk was only single sided, so I couldn't read any of my files from my office PC. I understand that Computer House offers a double-sided drive for a few bucks more.

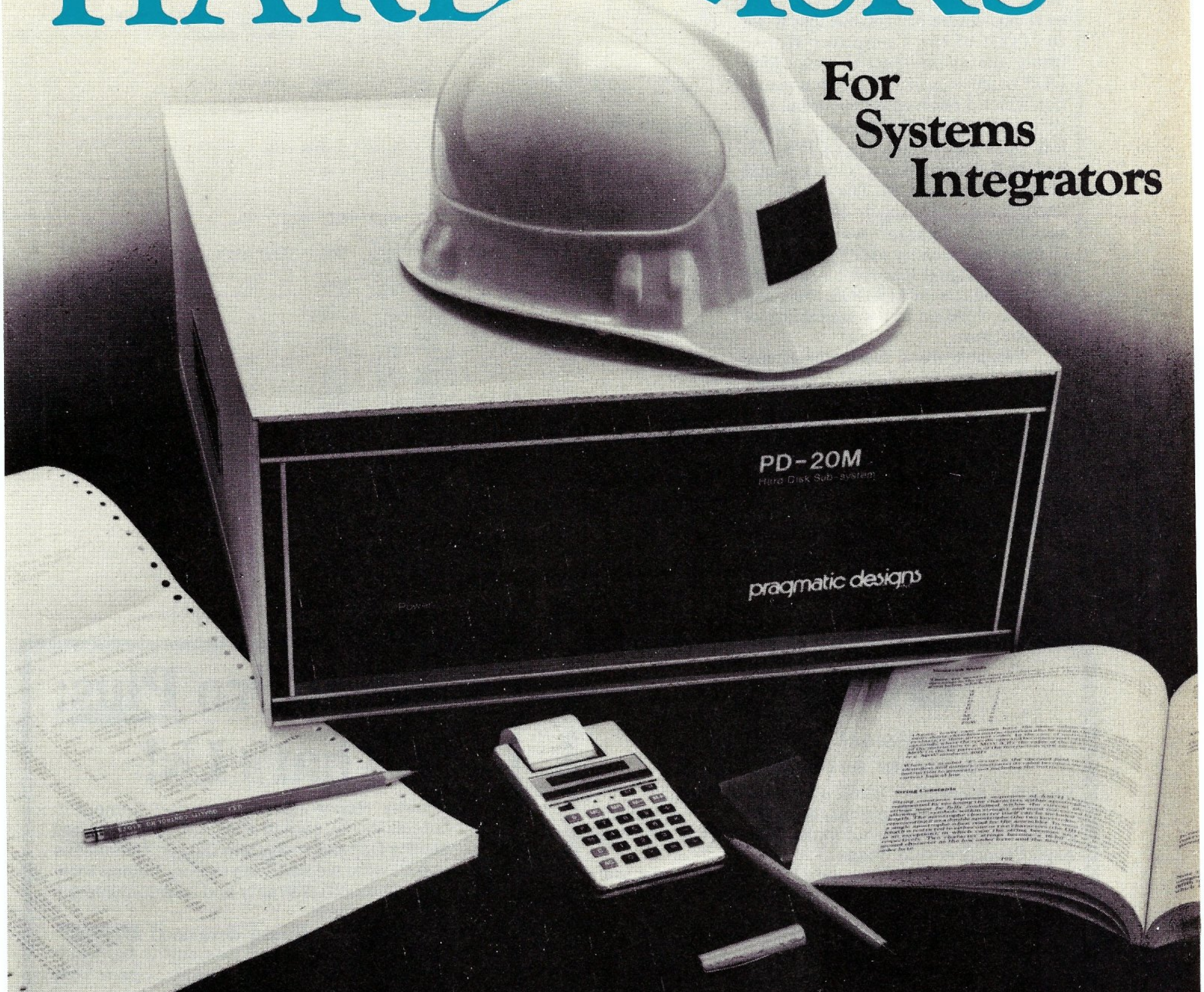
Conclusion

In summation I rate the implementation of MSPRO (MS-DOS by Computer House) as good. It would be much

```
        007F  56                             PUSH    SI
        0080  06                             PUSH    ES
        0081  C4 77 0E                       LES     SI,DWORD PTR XFRADR
        0084  AC                    IOCOUT:  LODSB
        0085  E6 03                          OUT     SERSTAT,AL
        0087  E2 FB                          LOOP    IOCOUT
        0089  07                             POP     ES
        008A  5E                             POP     SI
        008B  EB E0                          JMP     SHORT WRRET

                                    ;OUTPUT STATUS ROUTINE - RETURNS 1 IF DEVICE BUSY, 0 IF READY
                                    ;USES BSY BIT IN SRH STATUS BYTE FOR THESE BITS. (BIT 9)

        008D  E4 03                 OUTSTAT: IN      AL,SERSTAT
        008F  24 41                          AND     AL,041H
        0091  2C 01                          SUB     AL,1
        0093  74 05                          JZ      RDY
        0095  B8 0100                        MOV     AX,100H
        0098  EB 02                          JMP     SHORT RETURN
        009A  33 C0                 RDY:     XOR     AX,AX

                                    ;GENERAL RETURN FROM TRANSMISSIVE OR STATUS OPERATIONS
                                    ;SETS COMPLETION BIT (BIT 8) IN STATUS WORD OF SRH

                                    ;RESTORES ALL USED GENERAL REGISTERS

                                    ; DOES A FAR RETURN TO ROUTINE CALLING TRAP

 THE MICROSOFT MACRO ASSEMBLER           10-03-83       PAGE     1-4
INTERFACER 1 DRIVER


        009C  0C 80                 RETURN:  OR      AL,80H  ;SET COMPLETION BIT
        009E  89 47 03                       MOV     WORD PTR STATUS,AX      ;SET STATUS IN SRH
        00A1  9D                             POPF            ;RESTORE FLAGS
        00A2  59                             POP     CX
        00A3  1F                             POP     DS
        00A4  5B                             POP     BX
        00A5  58                             POP     AX      ;RESTORE ALL
        00A6  CB                             RET             ;RETURN TO USER CODE (FAR)

                                    ;THIS CODE IS NON RESIDENT (CODE BEYOND THE ABOVE RETURN FAR)
                                    ;IT IS USED ONLY DURING INITIALIZATION AND IS THEN DESTROYED
                                    ;BY THE CONFIG.COM INITIALIZATION AT BOOT TIME
                                    ;ANY NON-RESIDENT FUNCTIONS BELONG IN HERE

        00A7                        ENDRVR:                  ;END OF DRIVER (BREAK ADR)
        00A7  2E: A1 0012 R                  MOV     AX,WORD PTR CS:MEMLOC
        00AB  C6 47 0D 00                    MOV     BYTE PTR NRUNITS,0
        00AF  89 47 0E                       MOV     WORD PTR BRKADR,AX
        00B2  8C 4F 10                       MOV     WORD PTR SEGADR,CS
        00B5  C7 47 03 0080                  MOV     WORD PTR STATUS,0080H
        00BA  1F                             POP     DS
```

HARD DISKS

For Systems Integrators



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

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

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

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

pragmatic designs INC.

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

™ *CompuPro is a registered trademark of Godbout Electronics*

CIRCLE 186 ON READER SERVICE CARD

MSPRO

Continued from page 58

better if the user were not constrained by the hardware drivers, but that is *not* the fault of the Computer House implementation—just the way MS-DOS is put together. The documentation which I received was poor but was very preliminary. Performance was at least 30% better than under CP/M-85, and the disk allocation was less wasteful of space than in CP/M. The UNIX-like features are a nice touch. Frankly, the assembler and linker are worth the $395 charged for the entire MSPRO package. Now that Digital Resarch has brought out its products for MS-DOS, there will be an acceleration in the availability of 8086 software.

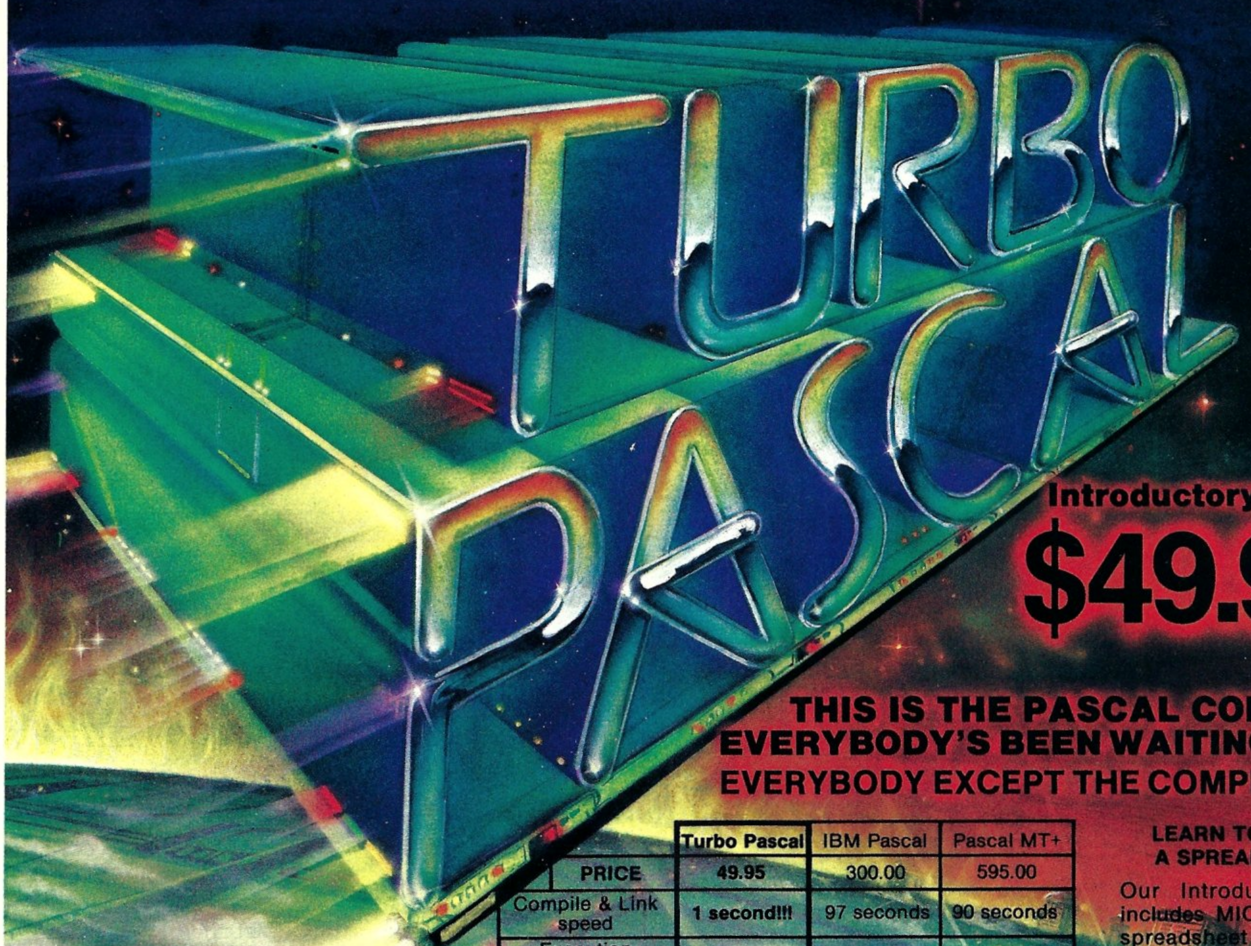
Available from: **Computer House** *722 B Street, San Rafael CA 94901 (415) 453-0865.* Price: $745 plus shipping. Includes Disk Controller Card, single-sided 5¼" drive, ROM, MS-DOS version 2 utilities, MASM assembler and documentation. For double-sided 5¼" drives, add $395 each.

CIRCLE 306 ON READER SERVICE CARD

Andrew L. Bender, M.D., 641 Victoria Terrace, Paramus, NJ 07652

```
        00BB  5B                      POP     BX
        00BC  58                      POP     AX
        00BD  CB                      RET             ;RETURN FROM INITIALIZATION
        00BE                  TRAP    ENDP    ;END OF PROCEDURE
        00BE                  CSEG    ENDS            ;END OF SEGMENT
                                      END

 THE MICROSOFT MACRO ASSEMBLER             10-03-83     PAGE    SYMBOLS-1
INTERFACER 1 DRIVER


SEGMENTS AND GROUPS:

                N A M E                     SIZE    ALIGN   COMBINE CLASS

CSEG . . . . . . . . . . . . . .            00BE    PARA    NONE

SYMBOLS:

                N A M E                     TYPE    VALUE   ATTR

BRKADR . . . . . . . . . . . . .            TEXT    14[BX]
CHREQ. . . . . . . . . . . . . .            F PROC  002F    CSEG    LENGTH =000B
COUNT. . . . . . . . . . . . . .            TEXT    18[BX]
DATAPORT . . . . . . . . . . . .            NUMBER  0002
ENDRVR . . . . . . . . . . . . .            L NEAR  00A7    CSEG
FUNC . . . . . . . . . . . . . .            TEXT    2[BX]
IOCOUT . . . . . . . . . . . . .            L NEAR  0084    CSEG
IOWAIT . . . . . . . . . . . . .            L NEAR  001A    CSEG
MDESC. . . . . . . . . . . . . .            TEXT    13[BX]
MEMLOC . . . . . . . . . . . . .            L NEAR  0012    CSEG
NEXT1. . . . . . . . . . . . . .            L NEAR  0072    CSEG
NRUNITS. . . . . . . . . . . . .            TEXT    13[BX]
OUTPUT . . . . . . . . . . . . .            L NEAR  0021    CSEG
OUTSTAT. . . . . . . . . . . . .            L NEAR  008D    CSEG
PACKET . . . . . . . . . . . . .            L NEAR  0016    CSEG
RDY. . . . . . . . . . . . . . .            L NEAR  009A    CSEG
RETURN . . . . . . . . . . . . .            L NEAR  009C    CSEG
SEGADR . . . . . . . . . . . . .            TEXT    16[BX]
SERSTAT. . . . . . . . . . . . .            NUMBER  0003
SRH. . . . . . . . . . . . . . .            TEXT    0[BX]
SRHBASE. . . . . . . . . . . . .            NUMBER  0000
STATUS . . . . . . . . . . . . .            TEXT    3[BX]
TRAP . . . . . . . . . . . . . .            F PROC  003A    CSEG    LENGTH =0084
UNITCOD. . . . . . . . . . . . .            TEXT    1[BX]
UNKCOM . . . . . . . . . . . . .            L NEAR  0052    CSEG
WRITE. . . . . . . . . . . . . .            L NEAR  0057    CSEG
WRITE1 . . . . . . . . . . . . .            L NEAR  0065    CSEG
WRRET. . . . . . . . . . . . . .            L NEAR  006D    CSEG
XFRADR . . . . . . . . . . . . .            TEXT    14[BX]

WARNING SEVERE
ERRORS  ERRORS
0       0
```

ENTER . . . THE FUTURE!



Introductory offer

$49.95

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

	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.

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.

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!

[redacted]

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.

Enhancing MP/M II: A Batch Processor

Your one-task system can let several programs run at once.

by Tom Clodfelter

In Part 1 of this series (January 1984), I discussed some of the features and faults of MP/M II as a multiuser system. Two of my major complaints were the fact that there was no multiuser login facility, and that MP/M II does not have a good batch processor. I will address these two points here.

The first project that I tackled under MP/M II was the problem of a user login and accounting system. My CP/M system had such a feature, since it had its own phone line. It was an RCPM-like system, but it was not public. MP/M is for a nonhostile environment, but a good multiuser system should still have a means to keep out unwanted users, yet allow those who have logged in to find out who else is on the system and communicate with them.

The password facility that does exist under MP/M II gives protection to disks and files. If password protection is enabled, you cannot access a file (or program) unless you know the password. You can assign different passwords to each file. You can also set a default password for yourself that the system will attempt to use with any files that are password protected. There are many situations in which this type of file protection is good, but it is no substitute for keeping unwanted people out of the system entirely. Since you cannot password protect resident system processes, MP/M allows them to be run without any form of protection. There is also the danger that the default password will still be set to that of the previous user, leaving the system wide open. I desired a system that could keep all unwanted users from logging in. It would assign to the user the correct user number, on the correct disk drive, with the correct default password for his application. Finally, it would even have the ability to start an application program associated with the login.

For my first attempt at providing this service, I wrote a resident system process that attached to all consoles after system boot. Under normal circumstances the Terminal Message Process (TMP) is attached to each console when the system boots. The TMP is the process that accepts your command line and sends it to the system. If you want a program to take control instead, it is only necessary to write a Resident System Process (RSP) that has a higher run priority than the TMP. This could then be made to detach from a console only after you logged into the system and to attach to your console when you logged out.

While this method works, it has several drawbacks. In the first place it takes up additional space in the common memory area. It also gives you no control over what the user does once he is logged in. I realized that the best place for this function was within the TMP itself. This way the TMP would have some control over the commands that the user issues, allowing different levels of privileged operation.

If you are not logged in, the TMP should accept only a login request. Once you are logged in, the TMP would have the first crack at checking your command line. If it saw something it did not like, it just wouldn't pass it on. This is a simple concept. That is, it is simple if you have the source to the TMP. Needless to say, I didn't have the source, so I wrote Digital Research to ask if I could buy it. Now, I won't say that they were rude (well, maybe close), but the answer was NO.

The MP/M II manual, although much better than that of 1.0, does not tell how to write a TMP. I had to know how the TMP was connected to the system. It took several months of investigation before I learned enough about how the TMP works to be able to write a replacement. My special TMP, soon to be a UNIX-like shell, is not the main topic of this installment. It is, however, important that you be aware of it when demonstrating BATCH. You are about to see MP/M do things that it is not supposed to do. If you are a MP/M user, you may not even recognize your operating system.

If you are familiar with CP/M's SUBMIT and XSUB, you may be surprised to find out that MP/M's submit facility is nowhere near as elaborate. There are several SUBMIT replacements available now, but most will not work correctly under MP/M. Even the ones that would work under MP/M fell short of what I wanted. I wanted a true batch facility with which I could run a job or series of jobs in background without tying up a terminal. I also wanted to be able to save the output from that job in a way that would show me when it ran, how long it took, and what it did.

The big secret of the solution to my problem came from a trick that I learned from several DEC operating systems. The cleverness lies in the concept of a virtual console. A virtual console is a software console that the system cannot tell from a hardware one. It does not really exist, but there is a driver and a TMP supporting it just as if it were. Tie in the virtual console concept with MP/M's very nice message queue data structures, and you already have half of a background batch system.

The idea is that the system has a console driver that does I/O to and from queues instead of a physical device. A program can then read the desired input for the virtual console from a disk file, and it can record the consoles output to another file or device. There are a few catches, but that is the basic idea, and we will cover the problems as we get to them.

In order to provide virtual console drivers, it is necessary for you to modify your XIOS. The XIOS is the BIOS equivalent for MP/M. The modifications are simple, so there is nothing to be afraid of. You should be able to accomplish them even if you have not done much assembly level programming. It is necessary, however, to have the source to your XIOS. I am working on a new method, that will allow virtual consoles even if you do not have the XIOS source. To modify your XIOS, just follow the examples.

Listing 1 shows an example of the MP/M console driver jump tables. Simply find the jump tables in your XIOS and add the additional data statements with the labels of your virtual consoles. Remember also to change the constant that is the maximum number of con-

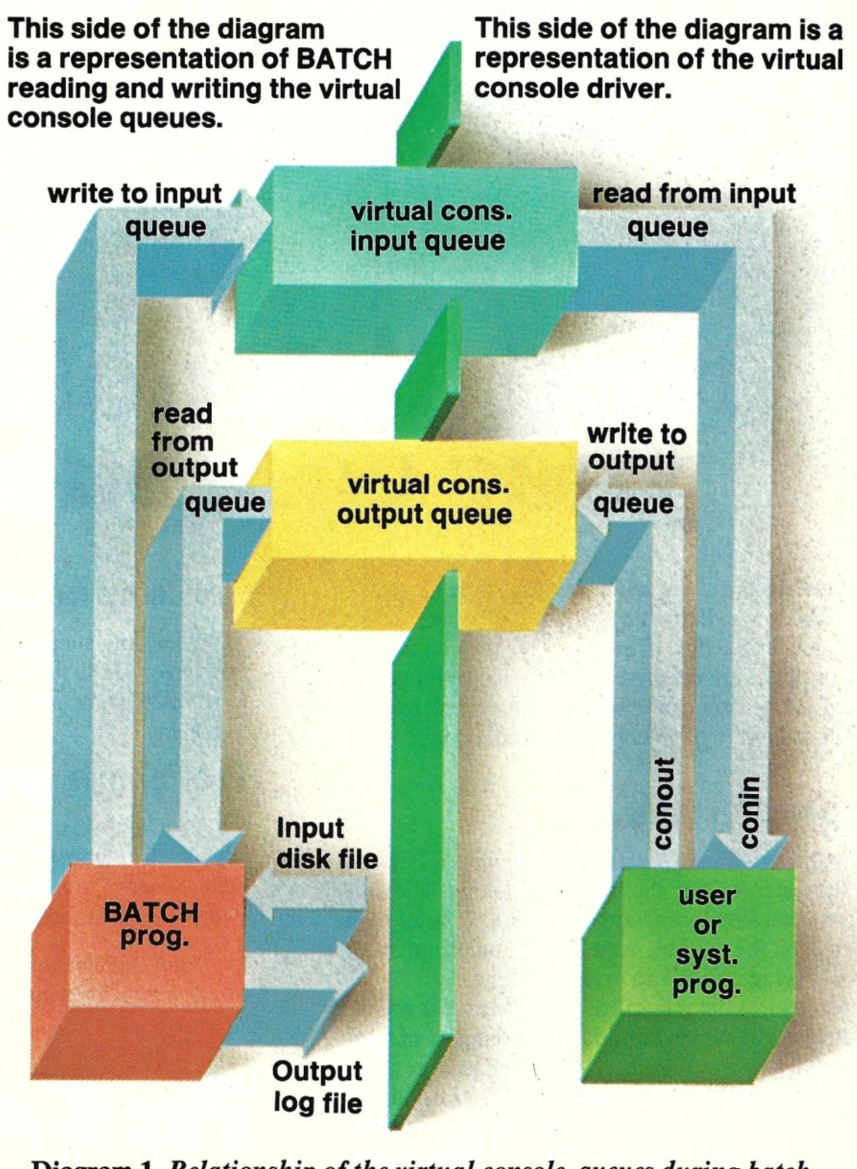
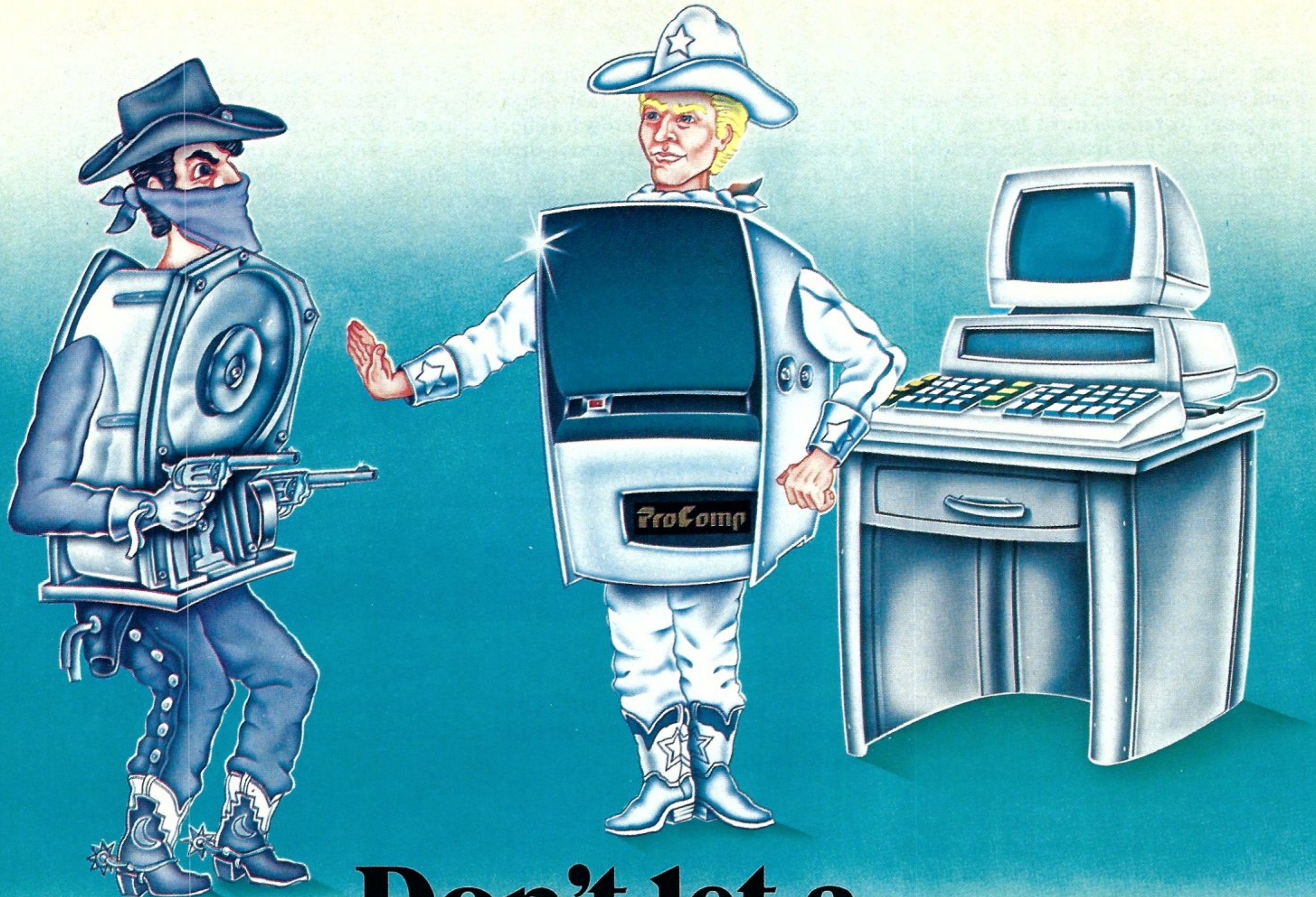


Diagram 1. *Relationship of the virtual console queues during batch operation.*



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

Not all hard disks are the same. In fact, some hard disks are just plain bad news. Behind a mask of low price, all they really offer is low efficiency. That's because they do not have the capacity or performance drive to do the job of modern software. The wrong disk can hold up your computer system. And, rob your operation of valuable time and money!

ProComp to the rescue.

ProComp brings your system up-to-date with reliable high speed subsystems using the latest technology. The average access time of a ProComp hard disk is 800% faster than the slowest disk on the market. And, among our competitors, ProComp still averages 200% more access time.

ProComp ensures the reliability of our hard disks with a thin film plated media which eliminates the dangers of fatal head crash through power failure...shipping damage...or even someone dropping the computer. Our drives use direct motors and thermal compensation so that they give you the same data on hot days *and* cold days. We even take the same care with our cabinets by providing you with oversize power supplies, fans and sturdy steel construction.

ProComp can take on the big guys.

ProComp subsystems support up to 36MB (unformatted) per drive. These drives are available NOW, not next month or six months from now. And, we have bigger and faster drives to follow.

ProComp is easy to have around.

ProComp has more experience writing disk software than anyone else. Our software can be installed by even the most novice user in less than 10 minutes. And, we still provide you with all the features that you could ever want.

ProComp is the most reliable hard disk that you can buy. So, don't get held up by any 'ole hard disk...call us—FAST.

ProComp
systems inc.
333 West 52nd Street, New York, NY 10019 (212) 246-0074
Dealer Inquiries Invited

CIRCLE 91 ON READER SERVICE CARD

MP/M II

Continued from page 63

soles your XIOS supports to reflect the new number. This constant is usually called NMBCNS. Remember that the system is not to know that this virtual console doesn't actually exist. In fact, it does "exist" now.

Listing 2 is an example of the actual driver code for the virtual console. You may insert it directly into your XIOS in the same area as your normal console drivers. The required data structures for the queues are also included. It is necessary that these data structures reside above the common base point in a banked system. Please note the one

MP/M II is for a nonhostile environment. Its file protection cannot keep unwanted users out entirely.

tricky thing that is different about these drivers: The virtual console status routine should always return flagged so that no character is available.

The status routine is employed by user programs and the operating system to check for special control (^) characters like ^S, ^Q, or ^C. CP/M and MP/M usually do not give you type-ahead in an interrupt-driven system because of this routine. A program checking for these special break codes will usually throw away any noncontrol characters. Since the input for the virtual console comes from a file, we need true file-sized type-ahead, and we don't want the program to throw away our input.

Having the external status routine return not ready will allow most programs to run unchanged on the virtual console. A screen editor might not run correctly though, and the system will not respond to a ^D (detach) or ^C (abort) in the same manner. These limitations will not impede the performance

Listing 1. MP/M console driver jump tables

```
;;      This is a normal MP/M console I/O jump table.  There is
;;      nothing unusual here.  Just add the jumps to your additional
;;      (Virtual) consoles.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;       console status **
const:
        call    ptbljmp         ; computand jump to handler
        dw      polci0
        dw      polci1          ; console #1
        dw      polci2          ; console #2
        dw      polci3          ; console #3
                                ; ** add new entries here **
        dw      polci4          ; Virtual console #4
        dw      polci5          ; Virtual console #5

;;      console input **

conin:                          ; console input character tabel
        call    ptbljmp         ; compute and jump to handler
        dw      pt0in           ; console #0 input
        dw      pt1in           ; console #1
        dw      pt2in           ; console #2
        dw      pt3in           ; console #3
                                ; ** add new entries here **
        dw      pt4in           ; Virtual console #4
        dw      pt5in           ; Virtual console #5

;;     console output **

conout:                         ; console output character tabel
        call    ptbljmp         ; compute and jump to handler
        dw      pt0out          ; console #0 output
        dw      pt1out          ; console #1
        dw      pt2out          ; console #2
        dw      pt3out          ; console #3
                                ; ** add new entries here **
        dw      pt4out          ; Virtual console #4
        dw      pt5out          ; Virtual console #5
```

Listing 2. Driver code for the virtual console

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;      Console #4 is a Virtual console.
;;
;;      Virtual consoles are transmit and receive queues that feed the
;;      data into the system as if they were actually console devices.
;;      They are generally feed by programs reading from and writing
;;      to disk files.  For the present the console status always
;;      returns not ready so that input is only there if the program
;;      request it.  This should allow most programs to run in batch
;;      mode.

pt4in:                          ; return character in <a>.
        push    d               ; dont destroy <de>.
        mvi     c,readque       ; get data from input que.
        lxi     d,c4inuqcb      ;  for console #4.
        call    xdos            ; have system do it.
        lda     ch4in           ; get the character.
        ani     07fh            ; strip any parity bit.
        pop     d               ; restore <de>.
        ret                     ; return with char in <a>.

;;      Console #4 input status
;;
;;      Many CP/M and MP/M programs check the console status routine
;;      to see if the user is trying to abort the program.  This
;;      presents a problem in batch operation because there is a
;;      character waiting in most cases.  Having this routine always
;;      return as if no character was available allows programs to
;;      run in batch mode and only get input when they request it.
;;      This allows the type of type ahead that is needed for batch.

polci4:                         ; poll console #4 input
        mvi     a,0
        ret                     ; always return not ready.

;;      Console #4 output
;;

pt4out:                         ; reg <c> = character to output
        push    d
        mov     a,c
        sta     ch4out          ; save the character
        mvi     c,writeque
        lxi     d,c4outuqcb
        call    xdos
        pop     d
        ret

;
;       input que structure

c4inq:  dw      0               ; que link
        db      'Vc4in          ; name
        dw      1               ; msg length
        dw      1               ; number of msgs
```

MP/M II

Continued from page 65

of a background process, however.

The modifications to this console driver make it much more powerful than the CP/M SUBMIT and XSUB method. Under CP/M you can feed input to a program run under SUBMIT, but there are severe limitations. With XSUB the program will only receive the input from the SUBMIT file if the program is doing line-buffered inputs through CP/M BDOS call #10. MP/M also has a SUBMIT program, but it has no XSUB facility at all.

The last modification to the XIOS is in the SYSTEMINIT routine. This simple modification, shown in Listing 3, ensures that the system will properly create the queues required. Now assemble your new XIOS and regenerate your system as per the MP/M II manual instructions. Remember that you have more consoles now. When all of this is done, a status display for your system should look something like Listing 4.

Now it is time to write a batch handler program to feed the virtual consoles. The listing with this article is a simple one written in C. Its only purpose is to demonstrate how the BATCH processor works. A similar batch processor could be written in almost any language. I chose C because I am familiar with it, and it relates very well to system data structures and pointers to structures.

This simple program does some translation of character sequences into

If you want to pass control to a program instead of to the TMP, simply write an RSP with a higher run priority.

```
         dw      0
         dw      0
         dw      0
         dw      0
c4incnt:
         ds      2                  ; msgcnt
         ds      1                  ; buffer

c4inuqcb:                           ; input user qcb
         dw      c4inq              ; pointer
         dw      ch4in              ; pointer to character
ch4in:   db      0                  ; input character

;        Mutual Exclusion que for pc4.

mxvc4:   dw      0                  ; que link
         db      'MXVc4             ; name
         dw      2                  ; msg length
         dw      1                  ; number of msgs
         dw      0
         dw      0
         dw      0
         dw      0
pc4adr:
         ds      2                  ; buffer

;
;        output que structure

c4outq:  dw      0                  ; que link
         db      'Vc4out            ; name
         dw      1                  ; msg length
         dw      80                 ; number of msgs
         dw      0
         dw      0
         dw      0
         dw      0
c4outcnt:
         ds      2                  ; msgcnt
         ds      80                 ; buffer

c4outuqcb:                          ; input user qcb
         dw      c4outq             ; pointer
         dw      ch4out             ; pointer to character
ch4out:  db      0                  ; output character
```

Listing 3. SYSTEMINIT routine modification

```
;;       You should insert this (or similar) code in at the tail end
;;       of your system init routine in the XIOS.  This is necessary
;;       so that the system will create the Virtual Console I/O queues.
;;       I have two Virtual consoles in my system (4 & 5).

         mvi     c,makeque          ; create special queues
         lxi     d,c4inq            ; Vc4:
         call    xdos

         mvi     c,makeque          ; create special queues
         lxi     d,c4outq           ; Vc4:
         call    xdos

         mvi     c,makeque          ; create special queues
         lxi     d,mxvc4            ; MXVc4:
         call    xdos

         mvi     c,makeque          ; create special queues
         lxi     d,c5inq            ; Vc5:
         call    xdos

         mvi     c,makeque          ; create special queues
         lxi     d,c5outq           ; Vc5:
         call    xdos

         mvi     c,makeque          ; create special queues
         lxi     d,mxvc5            ; MXVc5:
         call    xdos
```

Listing 4. Status display

```
1C>MPMSTAT<cr>

****** MP/M II V2.0 Status Display ******

Top of memory = FFFFH
Number of consoles = 06
Debugger breakpoint restart # = 06
Stack is swapped on BDOS calls
Memory is bank switched
BDOS disk file management is bank switched
Z80 complementary registers managed by dispatcher

Ready Process(es):
  MPMSTAT [1] Idle
Process(es) DQing:
 [Sched   ] Sched    [0]
 [SPOOLQ  ] Spool    [0]
 [CliQ    ] cli
 [ATTACH  ] ATTACH
 [Vc5in   ] Tmp5     [5]  <---- TMP for virtual console #5
 [Vc4in   ] Tmp4     [4]  <---- TMP for virtual console #4
Process(es) NQing:
Delayed Process(es):
Polling Process(es):
```

```
  Tmp0     [0] Tmp2     [2] Tmp3     [3]
Process(es) Flag Waiting:
  01 -  Tick
  02 -  Clock
Flag(s) Set:
Queue(s):
  MPMSTAT  Sched     STOPSPLR SPOOLO    CliQ      ATTACH    MXList
  MXDisk   MXVc5     Vc5out   Vc5in     MXVc4     Vc4out    Vc4in <--- queues
  MXProc
Process(es) Attached to Consoles:
  [0] - Tmp0
  [1] - MPMSTAT
  [2] - Tmp2
  [3] - Tmp3
  [4] - Tmp4         <------- virtual console #4
  [5] - Tmp5         <------- virtual console #5
Process(es) Waiting for Consoles:
  [1] -  Tmp1       [1]
Process(es) Attached to Printers:
  [0] - Unattached
  [1] - Unattached
Process(es) Waiting for Printers:
Memory Allocation:
  Base = 5E00H  Size = A200H  Bank = 00H  Allocated to MP/M-80 [0]
  Base = 0000H  Size = C000H  Bank = 01H  * Free *
  Base = 0000H  Size = C000H  Bank = 02H  * Free *
  Base = 0000H  Size = 5D00H  Bank = 00H  * Free *
  Base = 0000H  Size = 4000H  Bank = 03H  * Free *
1C>
```

Listing 5a. Example of BATCH program

```
/*****************************************************
 *                                                   *
 *      Background Batch Processor for MP/M II.      *
 *                                                   *
 *              By  Tom Clodfelter                   *
 *                  The Arecibo Observatory          *
 *                  Arecibo, PR 00612                *
 *                  11-Feb-82                        *
 *                                                   *
 *      This program is to demonstrate how a batch   *
 *      processor can work through virtual consoles  *
 *      under MP/M II.                               *
 *                                                   *
 *****************************************************/

#include "a:stdio.h"
#include "mpm.h"

#define TRUE  1
#define FALSE 0

FILE    *fi, *fo, *fopen();

main(argc, argv)
int     argc;
char    *argv[];
{
        register char *buffin, *buffout;
        register int k, c, ch, lstch;
        char    *s;
        int     flagl, llog, logit, pk;

        struct  _uqcb inque;            /* console input que control block */
        struct  _uqcb outque;           /* console output que control block */
        struct  _tod  time;             /* time buffer */

/* handle any value initiazation */

        inque.msg = buffin;
        outque.msg = buffout;
        flagl=logit=llog=FALSE;
        c = 0;
        pk = 4;

        printf("\nMP/M II (+) Background Batch Processer V1.0");

/* build the Queue control blocks and try to open the queues */

        strncpy(outque.name, "Vc4out  ", 8);
        if (bdos(OPENQUE, &outque) != 0) {
                printf("\nbatch: Error can't open output.");
                exit(0);
        }
        strncpy(inque.name, "Vc4in   ", 8);
        if (bdos(OPENQUE, &inque) != 0) {
                printf("\nbatch: Error can't open input.");
                exit(0);
        }

/* process option switches */

        while (--argc > 0 && (*++argv)[0] == '-')
                for (s = argv[0]+1; *s != '\0'; s++)
                    switch(toupper(*s)) {
                    case 'D':
                        flagl = TRUE;   /* detach Batch option */
                        break;
                    case 'L':
                        llog = TRUE;    /* log option set */
                        break;
```

control codes. When it finds a tilde (~) in a control file, it will gobble it up and convert the next character into a control character. This was done to make it easier to edit the control file used to drive the batch stream. It does not yet have variables or flow control, but it still serves me well. By the time you read this I should have a batch language processor in place of this demonstration program. Remember that the program in Listing 5a is only a simple example to show how BATCH works. Listing 5b shows the include file that contains the necessary system data structure for the BATCH program.

I wanted a true batch facility to run jobs in background without tying up the terminal.

Now turn your attention to Listing 6 for an example of a batch control file. This one is named TEST.CTL—an original enough name to suit our present needs. The first part of the control file consists of the necessary character strings to log into my system. A ^G is necessary to get the system's attention; it then expects a user name and password. These steps would not be necessary in normal MP/M. Also note some of the other commands like WHO, TIME, DATE, etc. Many of these are new intrinsic commands. They reside in my own TMP. Some of my other MP/M systems works will also be briefly demonstrated during the course of the batch run.

The Basic shown is my own interpreter designed around MP/M realtime and multiuser capabilities. Though it is still a very simple interpreter at this point, it is developing well. My goal has been to make it compatible with DEC Basic Plus with added realtime features.

PS (Process Status) is another in-

MP/M II

Continued from page 67

teresting utility that I have written for MP/M II. It can serve as a replacement for MPMSTAT, although I continue to use them both. PS offers more information about running programs than MPMSTAT does.

Listing 7 is an example of how to run the BATCH program. I use a WHO command before and after the batch invocation in order to show you how the batch user gets logged into the system. You will note that the BATCH program itself still belongs to me, but the things it runs belong to the virtual console. Afterwards, I check my PS program to find that BATCH is no longer running. I may now check the output from the batch operation. It can be found in the file TEST.LOG. A listing of the output may be seen in Listing 8.

When a program wants service from the operating system it executes a system call. This system call causes the system to dispatch the process. When a process is dispatched, this means that the operating system determines what type of system service the program requested and what resources it needs, then places it on a processing list for those resources. This has the effect of blocking the program from executing until the requested resource is available. The system then checks the status of all processes on its resource list to see if any are ready to be moved to the ready list. If any processes are due to be moved to the ready list, it does so and then picks the program with the greatest priority from the ready list and executes it.

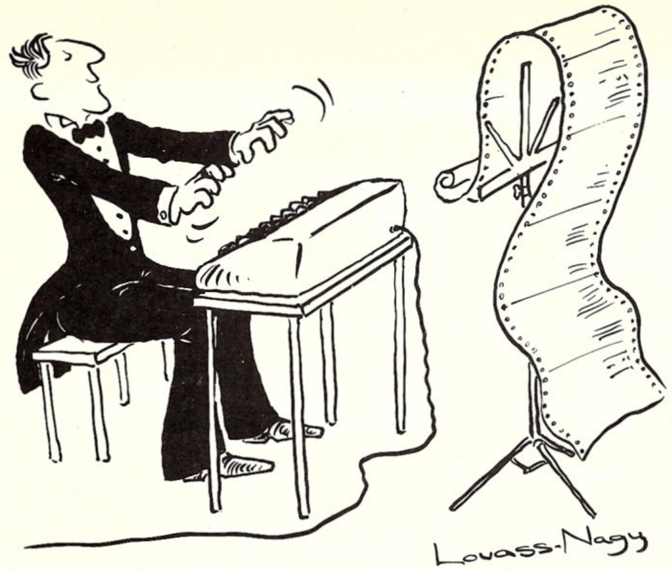
For example, let's say that our program wants an input character. First it

Tie in a virtual console with a message queue data structure, and you have half of a batch processor.

```
                default:
                    printf("\nusage: batch -ld pname.ctl pname.log");
                    printf("\n       -d detach and lower priority.");
                    printf("\n       -l log output requested.");
                    printf("\n       -? print this message.");
                    exit(0);
                }
/*      Act on options requested */

        if ((fi = fopen(*argv++, "r")) == NULL) {
                printf("\nbatch: can't open Ctl file.");
                exit(1);
        }
        if (llog == TRUE) {
                if ((fo = fopen(*argv, "w")) == NULL) {
                        printf("\nbatch: can't open log file.");
                        exit(1);
                }
                bdos(GETTOD, &time);
                fprintf(fo, "\n%02x:%02x:%02x $JOB %s\n",time.hour
                        ,time.minute,time.second, *argv);
        }
        if (flagl == TRUE) {            /* detach */
                printf("\nbatch: Detaching.....\n");
                bdos(SETPRI, 220);      /* raise priority */
                bdos(DETACH, 0);        /* detach from console */
        }
        while (TRUE) {
                ch = getc(fi);          /* get from input */
                if (ch == '^') {        /* test for control character flag */
                  ch = getc(fi) - '@';  /* convert next char to control */
                }
                if (ch == 0x0a) ch = getc(fi);
                if (ch == '\n') ch = 0x0d;
                *buffin = ch;
                while (bdos(CONDWRQUE, &inque) == 0xff) {
                        while (bdos(CONDRDQUE, &outque) == 0) {
                                c = *buffout;
                                if (c == 0x0a && lstch == 0x0d) logit = TRUE;
                                if (llog == TRUE) putc(c, fo);
                                if (logit == TRUE && llog == TRUE) {
                                        bdos(GETTOD, &time);
                                        fprintf(fo, "%02x:%02x:%02x ",
                                            time.hour,time.minute,time.second);
                                }
                                logit = FALSE;
                                lstch = c;
                        }
                }
                if (ch == 0x1a) {
                        while (bdos(CONDRDQUE, &outque) == 0) {
                                c = *buffout;
                                if (llog == TRUE) putc(c, fo);
                                if (c == 0x0a && lstch == 0x0d) logit = TRUE;
                                if (logit == TRUE && llog == TRUE) {
                                        bdos(GETTOD, &time);
                                        fprintf(fo, "%02x:%02x:%02x ",
                                            time.hour,time.minute,time.second);
                                }
                                logit = FALSE;
                                lstch = c;
                        }
                        fclose(fi);                     /* close input */
                        if (llog == TRUE) {
                                bdos(GETTOD, &time);
                                fprintf(fo, "\n%02x:%02x:%02x $EOJ\n",
                                    time.hour,time.minute,time.second);
                                fclose(fo);     /* and output */
                        }
                        exit(0);
                        }
                }
        }
```

Listing 5b. Include file for BATCH program

```
/*********************************************************************
 *                                                                   *
 *      Special structures and defines for MP/M II system programs.  *
 *                                                                   *
 *              By      Tom Clodfelter                               *
 *                      The Arecibo Observatory                      *
 *                                                                   *
 *********************************************************************/

#define FLAG char

/* tod data block */

struct _tod {
        unsigned date;          /* 16 bit days since Jan. 1, 1978 */
        char    hour;           /* BCD hours */
        char    minute;         /* BCD minutes */
        char    second;         /* BCD seconds */
};

/* lqcb linked que control block */

struct _lqcb {
        struct  _lqcb *nextque;/* pointer to next linked que */
        char    name[8];        /* que name */
```



•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.

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.

<table>
<tr><th>Avocet Cross-assembler</th><th>Target Microprocessor</th><th>CP/M-80 Version</th><th>•CP/M-86 IBM PC, MSDOS** Versions•</th></tr>
<tr><td>•XASMZ80</td><td>Z-80</td><td rowspan="2"></td><td rowspan="10">$250.00 each</td></tr>
<tr><td>•XASM85</td><td>8085</td></tr>
<tr><td>XASM05</td><td>6805</td><td rowspan="10">$200.00 each</td></tr>
<tr><td>XASM09</td><td>6809</td></tr>
<tr><td>XASM18</td><td>1802</td></tr>
<tr><td>XASM48</td><td>8048/8041</td></tr>
<tr><td>XASM51</td><td>8051</td></tr>
<tr><td>XASM65</td><td>6502</td></tr>
<tr><td>XASM68</td><td>6800/01</td></tr>
<tr><td>XASMZ8</td><td>Z8</td></tr>
<tr><td>XASMF8</td><td>F8/3870</td><td rowspan="2">$300.00 each</td></tr>
<tr><td>XASM400</td><td>COP400</td></tr>
<tr><td>XASM75</td><td>NEC 7500</td><td colspan="2">$500.00</td></tr>
<tr><td colspan="2">Coming soon: XASM68K...68000</td><td colspan="2"></td></tr>
</table>

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

CIRCLE 201 ON READER SERVICE CARD

MP/M II

Continued from page 68

will make an input character system call. At this point the operating system takes over. The program's state is saved in the process descriptor block. The system then scans various lists to determine which programs are due for a change of status, then choses a program and executes. When that program makes a system call, or when a tick (system time interval interrupt) occurs, the system suspends the program and scans its lists to see what needs checking. When it is our program's turn to be first on the ready list again, the system will restore the process state of our program and allow it to continue from where it stopped. This will now cause the program to actually execute the system service routine for character input. Part of the character input routine is a system call to poll the I/O device. When this call is executed, the process will again be

The system's console driver does I/O to and from queues instead of to and from a physical device.

blocked from execution if no character is available. If a character is available, it will be returned to our program. Our program will now run until it makes another system call or a system tick occurs.

With our special virtual consoles there are a few more steps to go through, but the procedure is similar. As before, our program wants an input character, so it must make an operating system call for the system to dispatch it. When it is time for the system to execute the program call again, it will cause the program to execute the read queue call in the console input routine. This will again cause the system to dispatch our process and force it to wait for execution

```
        unsigned msglen;           /* message length */
        unsigned nmbmsgs;          /* number of messages */
        unsigned dqph;             /* DQPH */
        unsigned nqph;             /* NQPH */
        unsigned mh;               /* MH */
        unsigned mt;               /* MT */
        unsigned bh;               /* BH */
};

/* memseg memory segment table structure */

struct _memseg {
        char     base;             /* memory segment base page */
        char     size;             /* memory segment size in pages */
        char     attributes;       /* memory segment attributes */
        char     bank;             /* memory segment bank number */
};

/*  pdb  Process Descriptor Block */

struct _pdb {
        struct   _pdb *nextproc;/* link to next process */
        char     status;           /* status byte */
        char     priority;         /* process run priority */
        unsigned stack;            /* pointer to stack top */
        char     name[8];          /* process name */
        char     cons_lst;         /* console and list numbers */
        char     memseg;           /* memory segment */
        unsigned dparam;           /* dparam */
        unsigned thread;           /* process thread */
        unsigned dma;              /* dma address */
        char     dsk_usr;          /* disk and user number */
        unsigned dcnt;             /* dcnt */
        char     searchl;          /* searchl */
        unsigned searcha;          /* searcha */
        unsigned pdextent;         /* extent */
                                   /* the register storage area follows */
        unsigned _HL;              /* HL' */
        unsigned _DE;              /* DE' */
        unsigned _BC;              /* BC' */
        unsigned _AF;              /* AF' */
        unsigned IY;               /* IY */
        unsigned IX;               /* IX */
        unsigned HL;               /* HL */
        unsigned DE;               /* DE */
        unsigned BC;               /* BC */
        unsigned AF;               /* AF */
        unsigned scratch;          /* scratch space reserved for MP/M II */
};

/* intdata XDOS internal data segment structure */

struct _intdata {
        struct   _tod time;        /* time of the day */
        struct   _pdb *ready;      /* ready list root */
        struct   _pdb *delay;      /* delay list root */
        struct   _pdb *disp;       /* dispatched ready list */
        struct   _pdb *poll;       /* poll list root */
        struct   _pdb *swap;       /* swap list root */
        struct   _lqcb *qroot;     /* que list root */
        unsigned thread;           /* thread root */
        char     nmbcons;          /* number of consoles */
        unsigned attachtbl[16];    /* console attach table */
        unsigned consque[16];      /* console que */
        char     nmbflags;         /* number of flags */
        unsigned sysflags[32];     /* system flags list */
        char     numbseg;          /* number of memory segments */
        struct   _memseg mem[8];   /* memory segment table */
        struct   _pdb pdtable[8];/* pdb table for memory segments */
        char     nmblpt;           /* number of list devices */
        unsigned lstattach[2];     /* list attach table */
        unsigned lstque[2];        /* list que */
};

/* sysdata  System data page structure */

struct _sysdata {
        char     memtop;           /* top page of memory */
        char     nmbcns;           /* number of consoles */
        char     brkptrst;         /* breakpoint restart number */
        FLAG     systks;           /* add system call user stack boolean */
        FLAG     bnked;            /* banked switched boolean */
        FLAG     Z80;              /* z80 version boolean */
        FLAG     bnkbdos;          /* banked bdos, boolean */
        char     xiosjmp;          /* xios jump table page */
        char     resbdos;          /* resbdos page */
        unsigned cpnetadr;         /* cp/net config table address */
        char     xdospage;         /* xdos page address */
        char     rsppage;          /* RSP's (bnkxios top+1) base page */
        char     bnkxiospg;        /* banked xios page address */
        char     bnkbdospg;        /* banked bdos page address */
        char     maxmemseg;        /* maximum memory segment number */
        struct   _memseg mem[8];   /* initial memory segment table  */
        char     bkpointvec[16];/* breakpoint vector table */
        char     reserved[16];     /* reserved for MP/M II */
        char     stktbl[16];       /* stack pointer table */
        char     reserved2[24];    /* reserved for MP/M II */
        unsigned numbrecs;         /* number of records in MPM.SYS file */
        char     nmbticks;         /* number of ticks per second */
        char     systemdrive;      /* system default drive */
        char     commonpage;       /* common memory base page */
        char     numbrsps;         /* number of resident system processes */
        unsigned listcp;           /* listcp array address */
```

```
        char      submitflg[16]; /* submit flag array */
        char      reserved3[43]; /* reserved for MP/M II */
        char      maxlocked;     /* max locked records/process */
        char      maxopen;       /* max open files/process */
        unsigned  nmblocked;     /* number of locked list items */
        char      *locktbl;      /* pointer to lock table free space */
        char      totallocked;   /* total system locked records */
        char      totalopen;     /* total system open files */
        FLAG      dayfile;       /* dayfile logging boolean */
        char      tempdrive;     /* temporary file drive */
        char      numblpt;       /* number of printers */
        char      reserved4[44]; /* reserved for MP/M II */
        char      bnkxdospg;     /* banked xdos page address */
        char      tmpbase;       /* tmp pdb base */
        char      cnsdatbase;    /* console.dat base */
        unsigned  bdosentry;     /* bdos/xdos entry point */
        char      tmpsprbase;    /* tmp.spr base */
        char      nmbbnkrsp;     /* number of banked RSP's */
        char      brspbase;      /* BRSP base address */
        struct    _pdb *nonres;  /* link to non-resident rsp */
        struct    _intdata *internal; /* pointer to internal data segment */
        struct    _pdb *resproc; /* link to resident system processes */
};

/* UQCB user que control block */

struct _uqcb {
        struct _lqcb *lnkque;        /* pointer to actual que */
        struct _qmsg *msg;           /* pointer to message area */
        char         name[8];        /* que name */
};

/* apb Abort Parameter Block */

struct _apb {
        struct _pdb *pdb;            /* process descriptor address */
        unsigned code;               /* termination code */
        char     name[8];            /* process name */
        char     cons;               /* console number used by process */
};

/* dpb dph blocks from CP/M Interface Guide Sect 6.5 */

struct _dpb {                        /* CP/M Version 2 Disk Parameter Block */
        unsigned spt;                /* sectors per track */
        char     bsh;                /* block shift factor */
        char     blm;
        char     exm;                /* Extent Mask */
        unsigned dsm;                /* Highest block number on this disk */
        unsigned drm;                /* total number of directory entries -1 */
        unsigned al;                 /* bit field corresponding to direc blocks */
        unsigned cks;                /* size of the directory check vector */
        unsigned off;                /* number of reserved tracks on this disk */
};

struct _dph {                        /* MP/M Version 2 Disk Parameter Header */
        char     **xlt;              /* logical to physical xlat vector */
        int      ooo;
        char     *dirbuf;            /* 128 byte scratchpad for directory use */
        struct   _dpb *dpb;          /* disk param block for this type of disk */
        char     **csv;              /* scratch area for detecting changed disks */
        char     *alv;               /* pointer to bit vector alloc map for disk */
};

struct _fcb {                        /* MP/M Version 2 fcb AS SEEN BY THE USER */
        char     dr;                 /* drive number */
        char     fname[8];           /* fname[1] used by TAG2 */
        char     ftype[3];           /* ftype[1] 8th bit set for $SYS */
        char     ex;                 /* file extent normally 0 */
        char     s1;                 /* reserved for bdos's benefit */
        char     s2;                 /* likewise, =0 on call to open,make,search */
        char     rc;                 /* record count for extent[ex]  0...128 */
        char     dmap[16];           /* group pointers
        char     cr;                 /* current record, initialized to 0 by usr */
        unsigned recn;               /* highest record number */
        char     recovf;             /* overflow of above */
};

/* following are BIOS calls */

#define CONST   2                    /* bios console char ready */
#define CONIN   3                    /* bios cons char input */
#define CONOUT  4                    /* bios cons char output */

#define UFNSIZE     15               /* d:filename.typ\0 is 15 chars */

/* following are BDOS calls */

#define SELDSK      14               /* bdos select disk 0=default disk */
#define SRCH        17               /* bdos search for file pattern*/
#define SRCHNXT     18               /* search for next occurrence */
#define GETDEFDISK  25               /* get current disk  (0-15) */
#define SETDMA      26               /* set address for read, write, etc. */
#define GETALLOCP   27               /* get address of allocation vector */
#define SETATTRIB   30               /* update file attributes */
#define GETDPBP     31               /* get DPB address for disk */
#define SETGETUSER  32               /* set or get user number */
#define COMPFILSIZ  35               /* compute file size into recn and recovf */

#define ABSMEMREQ   128              /* absolute memory seg request */
#define RELMEMREQ   129              /* relocatable memory seg request */
#define MEMORYFRE   130              /* memory free request */
```

until a message is available in the queue. At this point the system picks another program from the ready list and executes it. One of the programs on the ready list will eventually be BATCH. When it is time for BATCH to execute, it will make a system call that will write a character message to the queue. BATCH is then suspended, and the system finds out that our program now has the queue message it needs, so it again places it on the ready list. When it is our turn to run again, the system returns from the read queue routine, and then the I/O routine returns with the character it read from the queue. Our program

Having the external status routine return a not ready status will allow most programs to run unchanged on the virtual console.

then gets to run until it makes another system call or a system tick occurs.

Simple? Fortunately the system keeps track of all of this, and you don't have to. All this was meant to give you an idea of how a system that allows only one task at a time to actually execute through time sharing lets two or more programs interact concurrently. Diagram 1 shows the relationship of the virtual console queues during batch operation.

Future versions of BATCH will contain many advanced features. For instance, with a few simple modifications to the XDOS, it will be possible to use program return codes to make decisions during a batch process.

There are many more interesting things to be done with MP/M. I hope that I have stimulated your interest in some of them.

Tom Clodfelter, C/O Microsystems, One Park Ave., NY, NY 10016

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	$99.50	$99.50	$99.50
Z-8 XASM	99.50	500.00	$99.50	$99.50	$99.50
6502 XASM	99.50	500.00	$99.50	$99.50	$99.50
6800,2,8 XASM	99.50	500.00	$99.50	$99.50	$99.50
6801,03 XASM	99.50	500.00	$99.50	$99.50	$99.50
6805 XASM	99.50	500.00	$99.50	$99.50	$99.50
6809 XASM	99.50	500.00	$99.50	$99.50	$99.50
8748 XASM	99.50	500.00	$99.50	$99.50	$99.50
8051 XASM	99.50	500.00	$99.50	$99.50	$99.50
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 $ __________

Check one:
☐ 8″ Single Density
☐ 5¼″ Osborne
☐ IBM P.C.
☐ Cartridge Tape
☐ Apple (Softcard)

shipping/handling ($6.50 per unit) $ __________

Total Order $ __________

Signature ____________________

2500AD SOFTWARE INC.

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

MP/M II

Continued from page 71

```
#define POLLDEV      131          /* Poll device request */
#define FLAGWAIT     132          /* Flag wait request */
#define FLAGSET      133          /* Flag set request */
#define MAKEQUE      134          /* Make queue request */
#define OPENQUE      135          /* Open Que request */
#define DELETEQUE    136          /* Delete Que request */
#define READQUE      137          /* Read que request */
#define CONDRDQUE    138          /* conditional read que request */
#define WRITEQUE     139          /* Write que request */
#define CONDWRQUE    140          /* conditional write que request */
#define DELAY        141          /* delay system ticks */
#define DISPATCH     142          /* release time slice */
#define TERMINATE    143          /* terminate current process */
#define CREATEPROC   144          /* create process */
#define SETPRI       145          /* set process priority */
#define ATTACH       146          /* attach console */
#define DETACH       147          /* detach console */
#define SETCONS      148          /* set console device */
#define ASSIGN       149          /* assign console to program */
#define SENDCLI      150          /* send command line to cli */
#define CALLRSP      151          /* call resident system procedure */
#define PARSEFN      152          /* parse file name */
#define GETCONS      153          /* return console number */
#define GETSYSDATA   154          /* return system data page address */
#define GETTOD       155          /* get time and date */
#define GETPDA       156          /* get process descriptor address */
#define ATTACHLST    158          /* attach list device */
#define DETACHLST    159          /* detach list device */
#define SETLST       160          /* set list device */
#define CONDATLST    161          /* conditional attach list */
#define CONDATCNS    162          /* conditional attach console */
#define MPMVERS      163          /* return MP/M version number */
#define GETLST       164          /* return list number */

/* end of mpm.h */
```

Listing 6. Example of a batch control file

```
0C>type test.ctl<cr>
~GDEMO
SECRET
;
; Now we are logged into my system on a Virtual Console.
; Lets see who else is on the system.
; Also NOTE the Time Stamp to the left of every line.
; This is because we are running as a background job. What you
; are reading is the log file that the batch processor created.
; The batch processor stamps everything so that you will know
; when a job executed and how long it took.
;
who
;
; Now find out what time it is.
;
time
;
; So far all of the commands that I have used are handled
; in my special TMP.
;
; Lets take a look at PS, my status program.
;
ps
;
; Note: the default mode for PS was to only show processes
;       that were associated with my console.
; Now I will do a PS for all processes.
;
ps -a
;
; And now for all processes in long format.
;
ps -al
;
; Now I will use PS to dump the Process Descriptor block of
; a specified process.
;
ps -p -n spool
;
; Now I will ask PS what options it has.
;
ps -?
;
; Lets try the Que option.
;
ps -q
;
; Thats enough of PS for Now.
; Now lets take a quick look at my Basic interpreter for MP/M.
;
basic
10 rem we will enter a short program into basic from batch.
20 detach : rem this will cause the program to run detached.
30 for x = 1 to 1000
40 next x
50 print "the answer is ";x;"."
60 end
list
run
```

TURN-KEY, MULTI-USER BUSINESS SYSTEM
Running the Turbo-Dos Operating 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 tc 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 FORMATTED H.D.
- 832k BYTES OF MEMORY
- ATTRACTIVE ENCLOSURE
- TURBO-DOS 1.30
- TURBO-PLUS 1.04
- CCD CONFIG 1.05
- ALL CABLES
- MANUALS
- PREPAID SHIPPING

SAVE OVER $3000.00

The system software is **"TURBO-DOS"**, using the **"TURBO-PLUS"** extension. 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...$16,499.95

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

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"

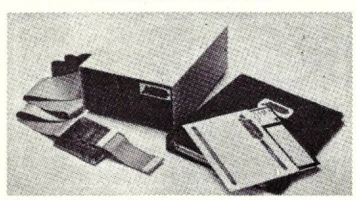
CCD09201-08 11 Megabyte, CP/M	$1699.95
CCD09202-08 22 Megabyte, CP/M	$2199.95
CCD09203-08 44 Megabyte, CP/M	$3099.95

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, Teletek	$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

MP/M II

Continued from page 74

```
;
; Now we are detached from the basic program. Lets take
; a quick look with PS and then with MPMSTAT.
;
ps -l
;
mpmstat
;
; Now I will attach to the basic program and find out what
; the answer was.
;
attach basic
rem now i will exit basic
bye
;
; Now we are back in TMP mode.
;
logoff

0C>
```

Listing 7. How the BATCH program runs

```
User-Name : TOM CLODFELTER
Password  :

Welcome to MP/M II V2.1  at  17:59:50 Tue 01-Mar-1983


0C>dir<cr>
Directory for User  0:
C: TEST     CTL
System Files Exist
0C>time<cr>
Time: 18:00:30 Tue 01-Mar-1983
0C>who<cr>
****** MP/M II V2.1 Login Summary ******

Console [0] is logged in  as [ OPERATOR ]
Console [1] is logged in  as [ TOM CLODFELTER ]

0C>batch -dl test.ctl test.log<cr>

MP/M II (+) Background Batch Processer V1.0
batch: Detaching.....

0C>who<cr>
****** MP/M II V2.1 Login Summary ******

Console [0] is logged in  as [ OPERATOR ]
Console [1] is logged in  as [ TOM CLODFELTER ]
Console [4] is logged in  as [ DEMO ]

0C>mpmstat<cr>

****** MP/M II V2.0 Status Display ******

Top of memory = FFFFH
Number of consoles = 06
Debugger breakpoint restart # = 06
Stack is swapped on BDOS calls
Memory is bank switched
BDOS disk file management is bank switched
Z80 complementary registers managed by dispatcher

Ready Process(es):
  MPMSTAT [1] BATCH    [1] Idle
Process(es) DQing:
 [Sched    ] Sched    [0]
 [SPOOLQ   ] Spool    [0]
 [CliQ     ] cli
 [ATTACH   ] ATTACH
 [Vc5in    ] Tmp5     [5]
Process(es) NQing:
 [Vc4out   ]
Delayed Process(es):
Polling Process(es):
  Tmp0     [0] Tmp2     [2] Tmp3     [3]
Process(es) Flag Waiting:
  01 -   Tick
  02 -  Clock
Flag(s) Set:
  03
Queue(s):
  MPMSTAT  Sched     STOPSPLR SPOOLQ    CliQ      ATTACH    MXList
  MXDisk   MXVc5     Vc5out   Vc5in     MXVc4     Vc4out    Vc4in
  MXProc
Process(es) Attached to Consoles:
  [0] - Tmp0
  [1] - MPMSTAT
  [2] - Tmp2
  [3] - Tmp3
  [4] - PS        <---- BATCH is running PS on console #4
  [5] - Tmp5
Process(es) Waiting for Consoles:
  [1] -   Tmp1      [1]
  [4] -   Tmp4      [4] <--- The TMP is waiting for PS to finish on #4
Process(es) Attached to Printers:
  [0] - Unattached
  [1] - Unattached
```

```
Process(es) Waiting for Printers:
Memory Allocation:
  Base = 5E00H  Size = A200H  Bank = 00H  Allocated to MP/M-80 [0]
  Base = 0000H  Size = C000H  Bank = 01H  * Free *
  Base = 0000H  Size = C000H  Bank = 02H  * Free *
  Base = 0000H  Size = 5D00H  Bank = 00H  Allocated to PS      [4] <---
  Base = 0000H  Size = 4000H  Bank = 03H  Allocated to BATCH   [1]

0C>time<cr>

Time: 18:02:24 Tue 01-Mar-1983
0C>ps<cr>

   PID    TTY     TIME  CMD
 164615   tty1      0   PS
 164701   tty1      0   BATCH   <--- BATCH still running
 177000   tty1      0   Tmp1
 165704   tty1      0   cli

0C>
0C>time<cr>

Time: 18:03:10 Tue 01-Mar-1983
0C>ps<cr>

   PID    TTY     TIME  CMD
 164615   tty1      0   PS
 177000   tty1      0   Tmp1    <--- BATCH is gone now
 165704   tty1      0   cli

0C>
0C>who<cr>          <---- Check to see that BATCH logged off
****** MP/M II V2.1 Login Summary ******

Console [0] is logged in  as [ OPERATOR ]
Console [1] is logged in  as [ TOM CLODFELTER ]
                                          <--- DEMO is off
0C>
0C>dir<cr>
Directory for User  0:
C: TEST     CTL    TEST     .LOG <--- The log file is now ready
System Files Exist
0C>
```

Listing 8. Output of the batch operation

```
18:01:01 $JOB TEST.LOG
18:01:02 MP/M II V2.1
18:01:02 Copyright (C) 1981, Digital Research
18:01:02
18:01:02 User-Name : DEMO
18:01:02 Password  :
18:01:02 Welcome to MP/M II V2.1  at  18:01:02 Tue 01-Mar-1983
18:01:03
18:01:03 4C>;
18:01:03 4C>; Now we are logged into my system on a Virtual Console.
18:01:03 4C>; Lets see who else is on the system.
18:01:04 4C>; Also NOTE the Time Stamp to the left of every line.
18:01:04 4C>; This is because we are running as a background job. What you
18:01:05 4C>; are reading is the log file that the batch processor created.
18:01:06 4C>; The batch processor stamps everything so that you will know
18:01:06 4C>; when a job executed and how long it took.
18:01:06 4C>;
18:01:06 4C>who
18:01:07 ****** MP/M II V2.1 Login Summary ******
18:01:07
18:01:07 Console [0] is logged in  as [ OPERATOR ]
18:01:07 Console [1] is logged in  as [ TOM CLODFELTER ]
18:01:07 Console [4] is logged in  as [ DEMO ]
18:01:08
18:01:08 4C>;
18:01:08 4C>; Now find out what time it is.
18:01:08 4C>;
18:01:08 4C>time
18:01:08 Time: 18:01:08 Tue 01-Mar-1983
18:01:09 4C>;
18:01:09 4C>; So far all of the commands that I have used are handled
18:01:09 4C>; in my special TMP.
18:01:11 4C>;
18:01:12 4C>; Lets take a look at PS, my status program.
18:01:12 4C>;
18:01:12 4C>ps
18:01:12
18:01:12     PID    TTY     TIME  CMD
18:01:13  164615   tty4      0   PS
18:01:13  176500   tty4      0   Tmp4
18:01:13  165704   tty4      0   cli
18:01:13 4C>;
18:01:15 4C>; Note: the default mode for PS was to only show processes
18:01:15 4C>;       that were associated with my console.
18:01:16 4C>; Now I will do a PS for all processes.
18:01:16 4C>;
18:01:18 4C>ps -a
18:01:18
18:01:18     PID    TTY       TIME  CMD
18:01:19  164615   tty4        0   PS
18:01:19  164701   tty1        0   BATCH
18:01:19  177100   console     0   Tmp0
18:01:20  177000   tty1        0   Tmp1
18:01:20  176700   tty2        0   Tmp2
```

MP/M II

Continued from page 77

```
18:01:20  176600   tty3      0  Tmp3
18:01:21  176500   tty4      0  Tmp4
18:01:21  176400   tty5      0  Tmp5
18:01:21  146002   tty1      0  MPMSTAT
18:01:22  145402   console   0  Sched
18:01:22  145002   console   0  Spool
18:01:22  166630   console   0  Tick
18:01:23  166740   console   0  Clock
18:01:23  165704   tty4      0  cli
18:01:23  167050   console   0  ATTACH
18:01:23 4C>;
18:01:25 4C>; And now for all processes in long format.
18:01:26 4C>;
18:01:26 4C>ps -al
18:01:26
18:01:26  F  S UID     PID   PPID CPU PRI NICE   ADDR  SZ WCHAN TTY     CMD
18:01:27  0  R   4  164615      0   0 310   24      0 135     0 tty4    PS
18:01:27  9  S   0  164701 164615   0 334   24      0 100     0 tty1    BATCH
18:01:28  3  P   0  177100 164701   0 306   24  57000 242     0 console Tmp0
18:01:29  3  P   0  177000 177100   0 306   24  57000 242     0 tty1    Tmp1
18:01:30  3  P   2  176700 177000   0 306   24  57000 242     0 tty2    Tmp2
18:01:31  3  P   3  176600 176700   0 306   24  57000 242     0 tty3    Tmp3
18:01:32  10 W   4  176500 176600   0 306   24  57000 242     0 tty4    Tmp4
18:01:32  1  Q   5  176400 176500   0 306   24  57000 242     0 tty5    Tmp5
18:01:33  1  Q  14  146002 176400   0 276   24  57000 242     0 tty1    MPMSTAT
18:01:34  1  Q   4  145402 146002   0 144   24  57000 242     0 console Sched
18:01:35  1  Q  14  145002 145402   0 311   24  57000 242     0 console Spool
18:01:36  4  W   0  166630 145002   0  12   24 140000  77     0 console Tick
18:01:37  4  W   0  166740 166630   0  24   24 140000  77     0 console Clock
18:01:37  1  Q   4  165704 166740   0 310   24 140000  77     0 tty4    cli
18:01:38  1  Q   0  167050 165704   0  24   24 140000  77     0 console ATTACH
18:01:39 4C>;
18:01:39 4C>; Now I will use PS to dump the Process Descriptor block of
18:01:39 4C>; a specified process.
18:01:42 4C>;
18:01:42 4C>ps -p -n spool
18:01:42
18:01:42     ==========================================================
18:01:42     Process Descriptor Block for : Spool        PID 145002
18:01:44       Process link ... 0000H 0000000
18:01:44       Status Byte .... 01H 0010
18:01:45       Priority ....... 201 3110
18:01:45       Stack Top ...... 6a5cH 0651340
18:01:45       Process Name ... Spool
18:01:46       Console ........ tty0
18:01:46       Printer ........ lp0
18:01:46       Memory Seg ..... 00H 0000
18:01:46       Dparam ......... 0000H 0000000
18:01:47       Process Thread . ed98H 1666300
18:01:47       DMA Address .... 0421H 0020410
18:01:47       Default Disk ... B:
18:01:48       User number .... 14
18:01:48       DCNT ........... 77b6H 0736660
18:01:48       SEARCHL ........ 3aH 00720
18:01:49       SEARCHA ........ 1ea8H 0172500
18:01:49       PDEXTENT ....... 3200H 0310000
18:01:49     Saved Register Values for : Spool
18:01:50       IY (0730H)  HL' (1ea4H)  DE' (a83aH)  BC' (fe1eH)AF' (c213H)
18:01:50       IX (cfcdH)  HL  (ca44H)  DE  (ca36H)  BC  (6a70H)AF  (004cH)
18:01:51       SCRATCH ......... 0001H
18:01:51 4C>;
18:01:53 4C>; Now I will ask PS what options it has.
18:01:53 4C>;
18:01:53 4C>ps -?
18:01:53
18:01:53 usage: ps -hablpq? {-n} {pname}
18:01:54         -h show system data page header.
18:01:54         -a show all processes.
18:01:54         -b short (brief) display [default].
18:01:54         -l long display format.
18:01:55         -p show process descriptor block.
18:01:55         -n show processes matching {pname}.
18:01:55         -q shows que list and statistics.
18:01:55         -? print this message.
18:01:55 4C>;
18:01:57 4C>; Lets try the Que option.
18:01:58 4C>;
18:01:58 4C>ps -q
18:01:58
18:01:58  Qname    Len  Max  Lqcb  Link   DQPH  NQPH  Type          Cnt/lnk
18:01:59  MPMSTAT  2    1    cc36H cb36H  cc02H 0000H Circular_Que  0
18:01:59  Sched    69   1    cb36H caceH  cb02H 0000H Linked_Que    cb4eH
18:02:00  STOPSPLR 0    1    caceH ca36H  0000H 0000H Circular_Que  0
18:02:01  SPOOLQ   62   2    ca36H ec2aH  ca02H 0000H Linked_Que    ca4eH
18:02:01  CliQ     129  1    ec2aH ee78H  ebc4H 0000H Linked_Que    ec42H
18:02:02  ATTACH   10   1    ee78H f050H  ee28H 0000H Linked_Que    ee90H
18:02:02  MXList   0    1    f050H f034H  0000H 0000H Circular_Que  1
18:02:03  MXDisk   0    1    f034H c4adH  0000H 0000H Circular_Que  1
18:02:03  MXVc5    2    1    c4adH c440H  0000H 0000H Circular_Que  0
18:02:04  Vc5out   1    80   c440H c422H  0000H 0000H Circular_Que  55
18:02:05  Vc5in    1    1    c422H c379H  fd00H 0000H Circular_Que  0
18:02:05  MXVc4    2    1    c379H c391H  0000H 0000H Circular_Que  0
18:02:06  Vc4out   1    80   c391H c35bH  0000H 0000H Circular_Que  80
18:02:07  Vc4in    1    1    c35bH e76eH  0000H 0000H Circular_Que  1
18:02:08  MXProc   0    1    e76eH 0000H  0000H 0000H Circular_Que  1
18:02:08 4C>;
18:02:09 4C>; Thats enough of PS for Now.
18:02:09 4C>; Now lets take a quick look at my Basic interpreter for MP/M.
18:02:11 4C>;
18:02:12 4C>basic
18:02:12
```

```
18:02:12 MP/M II RealTime BASIC V1.5  Cons[4] User[4] Drive C:  23101 free.
18:02:12
18:02:12 Ready
18:02:12 10 rem we will enter a short program into basic from batch.
18:02:13 20 detach : rem this will cause the program to run detached.
18:02:13 30 for x = 1 to 1000
18:02:14 40 next x
18:02:14 50 print "the answer is ";x;"."
18:02:14 60 end
18:02:14 list
18:02:14 10 REM we will enter a short program into basic from batch.
18:02:15 20 DETACH : REM this will cause the program to run detached.
18:02:15 30 FOR X = 1 TO 1000
18:02:15 40 NEXT X
18:02:15 50 PRINT "the answer is ";X;"."
18:02:15 60 END
18:02:16 Ready
18:02:16 run
18:02:16
18:02:16 4C>;
18:02:16 4C>; Now we are detached from the basic program. Lets take
18:02:16 4C>; a quick look with PS and then with MPMSTAT.
18:02:19 4C>;
18:02:19 4C>ps -1
18:02:19
18:02:19  F  S UID     PID    PPID CPU PRI NICE    ADDR   SZ WCHAN TTY      CMD
18:02:20  0  R   4  164615       0   0 310   24       0  135     0 tty4     PS
18:02:21  10 W   4  164445  164615   0 310   24       0  300     0 tty4     BASIC
18:02:22  10 W   4  176500  176600   0 306   24   57000  242     0 tty4     Tmp4
18:02:23  1  Q   4  165704  166740   0 310   24  140000   77     0 tty4     cli
18:02:23 4C>;
18:02:23 4C>mpmstat
18:02:23
18:02:23 ****** MP/M II V2.0 Status Display ******
18:02:24
18:02:24 Top of memory = FFFFH
18:02:24 Number of consoles = 06
18:02:24 Debugger breakpoint restart # = 06
18:02:24 Stack is swapped on BDOS calls
18:02:25 Memory is bank switched
18:02:25 BDOS disk file management is bank switched
18:02:25 Z80 complementary registers managed by dispatcher
18:02:25
18:02:25 Ready Process(es):
18:02:25   MPMSTAT [4] BATCH    [1] Idle
18:02:26 Process(es) DQing:
18:02:26  [Sched   ] Sched    [0]
18:02:26  [SPOOLQ  ] Spool    [0]
18:02:26  [CliQ    ] cli
18:02:27  [ATTACH  ] ATTACH
18:02:27  [Vc5in   ] Tmp5     [5]
18:02:27 Process(es) NQing:
18:02:27 Delayed Process(es):
18:02:27 Polling Process(es):
18:02:27   Tmp1     [1] Tmp0     [0] Tmp2     [2] Tmp3     [3]
18:02:28 Process(es) Flag Waiting:
18:02:28   01 -  Tick
18:02:28   02 -  Clock
18:02:28 Flag(s) Set:
18:02:28   03
18:02:28 Queue(s):
18:02:29   MPMSTAT  Sched    STOPSPLR SPOOLQ   CliQ     ATTACH   MXList
18:02:29   MXDisk   MXVc5    Vc5out   Vc5in    MXVc4    Vc4out   Vc4in
18:02:29   MXProc
18:02:29 Process(es) Attached to Consoles:
18:02:30   [0] - Tmp0
18:02:30   [1] - Tmp1
18:02:30   [2] - Tmp2
18:02:30   [3] - Tmp3
18:02:30   [4] - MPMSTAT
18:02:30   [5] - Tmp5
18:02:30 Process(es) Waiting for Consoles:
18:02:31   [4] -  Tmp4     [4] BASIC    [4]
18:02:31 Process(es) Attached to Printers:
18:02:31   [0] - Unattached
18:02:31   [1] - Unattached
18:02:31 Process(es) Waiting for Printers:
18:02:32 Memory Allocation:
18:02:32   Base = 5E00H  Size = A200H  Bank = 00H  Allocated to MP/M-80 [0]
18:02:32   Base = 0000H  Size = C000H  Bank = 01H  Allocated to BASIC   [4]
18:02:33   Base = 0000H  Size = C000H  Bank = 02H  * Free *
18:02:35   Base = 0000H  Size = 5D00H  Bank = 00H  * Free *
18:02:35   Base = 0000H  Size = 4000H  Bank = 03H  Allocated to BATCH   [1]
18:02:36 4C>;
18:02:36 4C>; Now I will attach to the basic program and find out what
18:02:36 4C>; the answer was.
18:02:38 4C>;
18:02:38 4C>attach basic
18:02:38
18:02:38 Attach:BASIC
18:02:38 the answer is  1001
18:02:39 Ready
18:02:39 rem now i will exit basic
18:02:39 Ready
18:02:39 bye
18:02:39
18:02:39 4C>;
18:02:39 4C>; Now we are back in TMP mode.
18:02:40 4C>;
18:02:40 4C>logoff
18:02:40 DEMO  logged off console [4] at 18:02:40 Tue 01-Mar-1983
18:02:40 $EOJ
```

Julian Date Conversion Functions

Convert to and from Julian format under CP/M Plus, MP/M II and TurboDOS

by Ron Fowler

In the November 1983 *Microsystems,* reader Arthur Zatarain bemoaned the difficulty in making use of the time and date formats provided by Digital Research's MP/M and Software 2000's TurboDOS in applications software. Mr. Zatarain's letter suggests that the awkward, modified Julian date provided by both systems may be responsible for the absence of time and date support in languages and programs that run under both of these operating systems, and concludes with a request for source code illustrating conversion techniques between the "tight" operating system formats and the more readable (at least by human beings) character-string formats.

Mr. Zatarain is right; there is indeed very little software around, public domain *or* commercial, that makes use of these functions. If this lack is indeed due to the difficulty in format conversions, perhaps the routines presented in this article will be of some use to systems and applications programmers who would like to exploit the time and date functions that these operating systems provide.

Format of the time/date specification

There have been variations in the function number as well as the data structure and parameter passing conventions among the different releases of both operating systems. For example, MP/M release 1 used a single system call (number 155), and the time and date returned by the system included seconds. This system call was retained in MP/M release 2, but a new system call (number 105) was added, whose only difference was that the "seconds" field was omitted. This same system call was finally added to the single-user CP/M with the release of CPM Plus. In all three releases, a pointer is passed (in DE) to a data structure in memory ordered as follows:

```
2-byte Julian date field
1-byte hours field
1-byte minutes field
1-byte seconds field (ignored
   in system call #105)
```

The 2-byte Julian date field is defined as the number of days elapsed since 1 January, 1978; hours, minutes and seconds are specified as binary-coded decimal (BCD) values.

Meanwhile, early releases of TurboDOS were totally incompatible: a different function call was used (number 84), the values for hours, minutes

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

EXPECT A LOT *from* VEDIT

Newest Features:

- Horizontal Scrolling
- Single Key and Conditional Search and Replace with Pattern Matching
- Easier to Use than Ever

Widely acclaimed as an editor, VEDIT has evolved to be much more. VEDIT is not only a versatile full screen editor, its command mode has become a powerful text-oriented programming language based on TECO. For the first time you'll be able to perform complex text manipulations that are impossible for any word processor. Just a few examples of VEDIT'S command macro power are:

- Hundreds of tedious search/replace operations, once performed by hand, can now be made automatically on dozens of files without your intervention.
- VEDIT macros are indispensible with source code translations. (Example ZILOG to/from INTEL translator macros are included). A complete line of translators will be available soon.
- An on-line tutorial and mail merge program, completely written with VEDIT macros, (source included) is available as an inexpensive option --imagine what you can write for your own applications.

Word processing is a snap with paragraph formatting, word wrap, print functions, and VEDIT's unmatched 'cut and paste' capability.

VEDIT cuts programming time in half - with multiple file handling and special features for Pascal, PL/1, 'C', Assembler and other languages.

VEDIT is completely customizable -- you configure your own keyboard layout. Use any function or cursor key you wish. Plus our total commitment to hardware support ensures that VEDIT will be perfectly matched to any microcomputer you are ever likely to own.

Reviewers' Comments

'The performance of this product is nothing short of outstanding!'
'VEDIT is a 'virtual' text editor. The amount of memory in your computer does not limit the size of the file you can edit.'

Tim Daneliuk, InfoWorld - May '83

'At present there are three major entrants into the screen editor category for the PC...VEDIT is the front-runner for the definitive quality screen editor.'

Todd Katz, PC Magazine - October, '83

'VEDIT's 'visual' mode, where you will spend most of your time, has a multiplicity of valuable one-or-two-keystroke commands which make it very easy to enter and modify text.' *Frederick Zimmerman, Sextant - Fall, '83*

'VEDIT's by all odds the most flexible programming editor I've ever seen.'
'One of its best features is multiple buffers; that is, you can store chunks of programs in various places and pull them out into your main file when needed.'

Jerry Pournelle, Byte - September, '83

True Full Screen Editing
Edit Files One Disk in Length
Automatic Disk Buffering
Compact (Only 16K) and Fast
Display of Line and Column
Set/Goto Text Markers
'Undo' Key to Restore Line
Automatic Indent/Undent
Adjustable Tab Positions
Repeat Function Key

Text Move and Copy
10 Scratchpad Buffers
Load/Save Buffers on Disk
Powerful Command Macros
Directory Display
Edit Additional (small) Files Simultaneously
Insert Another Disk File
Unlimited File Handling
Recovery from 'Full Disk'

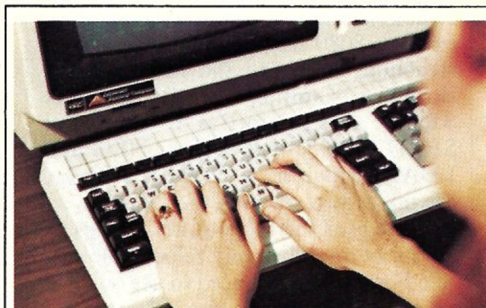
Change Disks While Editing
Word Wrap, Format Paragraph
Simple Printing
150 Page Indexed Manual
Startup Command File
Menu Driven Installation
Program CRT Function Keys
Support Newest CRT terminals
Flexible Memory Map Support
Customizable Keyboard Layout

VEDIT is available for virtually every 8080, Z80 or 8086 microcomputer made. To order please specify your computer, operating system and disk format.

VEDIT - Disk and Manual

VEDIT CUSTOMIZABLE FULL SCREEN EDITOR

8080, Z80 or IBM PC . $150
CP/M-86 or MSDOS . $195
Manual only $18
On-Line Tutorial . . . $15
Mail Merge $15



CIRCLE 26 ON READER SERVICE CARD

CompuView
PRODUCTS, INC.

1955 Pauline Blvd., Suite 200 Ann Arbor, Michigan 48103 (313) 996-1299 Telex - 701821
Orders: P.O. Box 1349, Ann Arbor, Michigan 48106

JULIAN DATES

Continued from page 80

and seconds were binary rather than BCD, and the Julian date was based on 1 January 1948. Finally, the parameters were all returned in the registers, rather than in a date block held in memory.

TurboDOS release 1.2 attempted to gain some MP/M compatibility. Among the changes was the addition of a fully MP/M-compatible get date/time call: number 105. No support was provided for MP/M's call 155 (the one that supports the "seconds" field); however, but the system did keep its MP/M-incompatible function 84 in order to retain support for the software developed for early TurboDOS releases.

Finally, TurboDOS release 1.3 effects a "split" of system calls: those which are MP/M-compatible are called in the usual way (via the jump instruction at location 5), while TurboDOS-only system calls are available by calling location 50 (hex) with a new set of function codes. Luckily, the MP/M-compatible function #105 is still available via a call to location 5; this is the only system call that is available through both operating systems (and CP/M Plus) with identical calling procedures (excluding the early TurboDOS releases, of course), and will be the one employed here.

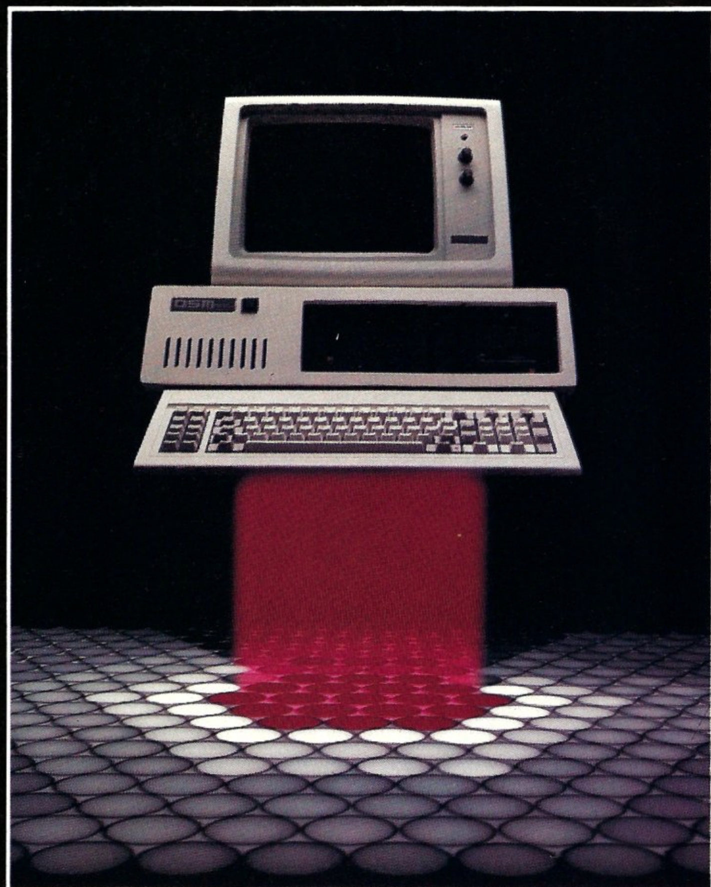
Some notes about the listings

All of the example listings presented here are written for the old TDL/XITAN assembler. This assembler still lives (although TDL and XITAN are long gone), and is now called the CDL (Computer Design Labs) assembler. This assembler is fully Z80-compatible, but uses a mnemonic set that bears much more resemblance to Intel mnemonics than to the Zilog Z80 set. Most of the mnemonics will be fairly self-explanatory to anyone familiar with the Intel set, and the Z80 instructions that add addressing modes to existing 8080 instructions have a construction similar to the Intel convention. For example, store-BC-direct is SBCD; this is similar to the Intel SHLD. Instructions such as NEG and LDAR, which have no 8080 analogy, are taken directly from the Zilog set.

Each of the subroutines resides as an object module in my relocatable system library. When I need to use one in a program, I simply declare it "external," and call it freely. When I link the program, my linkage editor searches the system library and includes those subroutines referenced in my main program, freeing me from the tedium of rewriting a subroutine each time I use it in a program. Hence, you'll see some ex-

```
----------------------------------------------------------------
                        *** LISTING #1 ***
----------------------------------------------------------------
                        ;
                                .IDENT  PRDATE          ;05/17/82
                        ;
                        ;
                        ; ROUTINE TO PRINT DATE IN FORM OF
                        ; MM/DD/YY ON CONSOLE VIA TYPE#. ON
                        ; ENTRY, HL = @DATE SPEC (WITH YEAR
                        ; FIELD BASEYR# RELATIVE)
                        ;
                                .ENTRY  PRDATE
                        ;
0000'   D5              PRDATE: PUSH    D               ;SAVE WORKING REGS
0001'   E5                      PUSH    H
0002'   7E                      MOV     A,M             ;FETCH THE YEAR
0003'   23                      INX     H
0004'   66                      MOV     H,M
0005'   6F                      MOV     L,A
0006'   11 002A'                LXI     D,YEAR          ;BUFFER TO HOLD DATE
0009'   CD 0000:04              CALL    FMJUL#          ;CONVERT FROM JULIAN
000C'   3A 002B'                LDA     MONTH           ;PRINT MONTH AND SEP
000F'   CD 0021'                CALL    DATSEP
0012'   3A 002C'                LDA     DATE            ;THEN DATE
0015'   CD 0021'                CALL    DATSEP
0018'   3A 002A'                LDA     YEAR            ;FINALLY, YEAR
001B'   CD 0000:05              CALL    DECOTB#
001E'   E1                      POP     H               ;RESTORE WORKING REGS
001F'   D1                      POP     D
0020'   C9                      RET
                        ;
                        ; SUBROUTINE TO PRINT A IN DECIMAL FOLLOWED BY "/"
                        ;
0021'   CD 0000:05      DATSEP: CALL    DECOTB#         ;PRINT IN DECIMAL
0024'   3E2F                    MVI     A,"/"           ;FOLLOWED BY SEPARATOR
0026'   CD 0000:06              CALL    TYPE#
0029'   C9                      RET
                        ;
002A'   00              YEAR:   .BYTE   0               ;YEAR
002B'   00              MONTH:  .BYTE   0               ;MONTH
002C'   00              DATE:   .BYTE   0               ;DATE
                        ;
                                .PRGEND
----------------------------------------------------------------
                        *** Listing #2 ***
----------------------------------------------------------------
                        ;
                        ;
                                .IDENT  PRTIME          ;05/15/82
                        ;
                        ;
                        ; PRINT TIME FROM DATE/TIME BUFFER @HL, FORM HH:MM
                        ;
                                .ENTRY  PRTIME
                        ;
0000'   23              PRTIME: INX     H               ;SKIP PAST DATE
0001'   23                      INX     H               ;TO GET TO TIME
0002'   7E                      MOV     A,M
0003'   23                      INX     H
0004'   CD 0000:04              CALL    PRTBCD#
0007'   3E3A                    MVI     A,':'
0009'   CD 0000:05              CALL    TYPE#
000C'   7E                      MOV     A,M             ;NOW WITHOUT COLON
000D'   CD 0000:04              CALL    PRTBCD#
0010'   C9                      RET
                        ;
                                .PRGEND

----------------------------------------------------------------
                        *** LISTING #3 ***
----------------------------------------------------------------
                        ;
                        ;
                                .IDENT  PARSDT          ;05/20/82
                        ;
                        ;
                        ; PARSE DATE FROM AN INPUT STRING @HL
                        ; (FORM=MM/DD/YY) RETURN JULIAN DATE
                        ; (BASEYR# RELATIVE) IN HL
                        ;
                        ; THE FOLLOWING ERRORS ARE TRAPPED AND RETURNED
                        ; IN A REG (A=0 AND CY=0 IF NO ERRORS):
                        ;
                        ;       1) DATE OUT OF RANGE
                        ;       2) BAD DATE DELIMITER
                        ;       3) COMPONENT OUT OF RANGE (MTH, YR OR DATE)
                        ;       4) BAD DECIMAL #
                        ;
                                .ENTRY  PARSDT
                        ;
                        ;
0000'   D5              PARSDT: PUSH    D               ;SAVE DATE POINTER
0001'   CD 000B'                CALL    ..PARS
0004'   EB                      XCHG                    ;MOVE JUL DATE TO DE
0005'   E1                      POP     H               ;FETCH STRUC PNTR
0006'   D8                      RC                      ;RETURN IF PARSE ERROR
0007'   73                      MOV     M,E             ;ELSE STORE DATE INTO STRUC
0008'   23                      INX     H
0009'   72                      MOV     M,D
000A'   C9                      RET
000B'   ED73 005D'      ..PARS: SSPD    SPSAVE          ;SAVE STACK LEVEL
000F'   11 005F'                LXI     D,DATBUF        ;POINT TO DATE BUFFER
0012'   0603                    MVI     B,3
0014'   D5              ..LOOP: PUSH    D
0015'   C5                      PUSH    B
0016'   CD 0000:04              CALL    DECIN#          ;FETCH DECIMAL DIGIT
0019'   382F                    JRC     ERROR4          ;SKIP IF NO INPUT ERROR
001B'   CD 0052'                CALL    DELMCK          ;CHECK DELIMITER
001E'   2022                    JRNZ    ERROR2
0020'   14                      INR     D               ;TEST HI BYTE=0
0021'   15                      DCR     D
0022'   2022                    JRNZ    ERROR3          ;MUST BE ZERO
0024'   7B                      MOV     A,E
0025'   C1              ..STOX: POP     B
0026'   D1                      POP     D
0027'   12                      STAX    D
0028'   13                      INX     D
0029'   10F9                    DJNZ    ..LOOP
002B'   2A 005F'                LHLD    DATBUF          ;TWEAK THINGS AROUND
```

What Makes The OSM Personal Computer Different Than Any Other IBM-Compatible PC ?



Completely IBM-Compatible

The OSM-PC™ is completely IBM™-Compatible. Software and hardware. That means you can choose from thousands of different programs and hardware options to meet your specific requirements.

A Variety of Configurations

The OSM-PC comes in three configurations tailored to fit your individual needs. The dual 320KB floppy system. The 10 or 20 Megabyte hard disk-based system, if you need more power for your business applications. And a unique hard disk-based system with tape backup *all* in the same unit.

Networking

With OSM's™ ZeusNet™ controller, you can link up to 16 IBM-Compatible personal computers together. Start with one, and as need arises, link additional units together for file sharing, electronic mail, and many other communications features found only in traditional Local Area Networks (LANs). ZeusNet allows your computing power to grow as you do... at a cost that will surprise you.

Service and Support

OSM has been manufacturing business computer systems for years, so you can depend on us for quality, service and support.

For more information, call your nearest OSM Dealer. Or call OSM at 1-800/538-5120, (In California call 415/961-8680), and we'll send you more reasons why the OSM-PC is different from other IBM-Compatible Personal Computers.

See us at Booth 127.

OSM

OSM Computer Corporation
665 Clyde Avenue
Mtn. View, CA 94043 USA
Telephone (415) 961-8680
TWX 910 379 5003

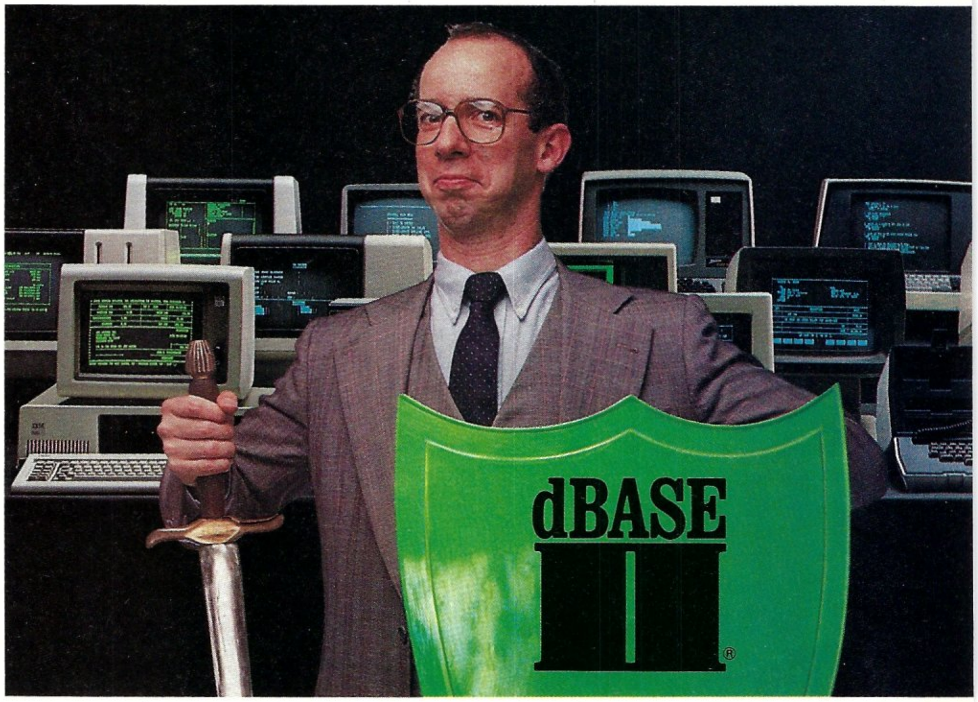
OSM, OSM-PC and ZeusNet are trademarks of OSM Computer Corporation.
IBM is a registered trademark of International Business Machines.

CIRCLE 254 ON READER SERVICE CARD

Self-dFENSE™ for EDP managers.

The micro invasion has begun. And, chances are, you've now got a lot of different people in a lot of different departments using a lot of different micros.

Now there's a way for you to control and maximize the benefits of all the different micros in your domain.



Fight back with dBASE II.®

dBASE II is the relational database management system from Ashton-Tate that enables you to manage your micro-based corporate data resources with the high level of consistency and sophistication you've enjoyed with mainframe and minicomputer systems.

Armed with dBASE II and the dBASE II RunTime™ program development module, you can write programs which will enable micro users in each department to "do their own thing" while creating complete database consistency throughout the company.

dBASE II is a powerful, flexible way for you to effectively manage the micro proliferation.

Help is here.

If you'd like to know more about how dBASE II and RunTime can help you win the micro management battle, contact Ashton-Tate today. 10150 West Jefferson Boulevard, Culver City, CA 90230. (800) 437-4329, ext. 217 . In Colorado (303) 799-4900. In the U.K. (0908) 568866.

ASHTON·TATE™

CIRCLE 32 ON READER SERVICE CARD

JULIAN DATES

Continued from page 82

ternal references in the listings—these are denoted by a pound sign at the end of the name of the reference.

These externals are defined within their own modules as globals (also called "public") using the .ENTRY pseudo-op.

For example, the program line

```
        CALL    DECOUT#
```

implies that the routine DECOUT resides outside of the current module. In its own module, DECOUT might begin like this:

```
          .ENTRY   DECOUT
DECOUT:   PUSH     H
```

The .IDENT statement at the beginning of each module gives the module a name. Some linkers use this module name to build module-load maps to assist the programmer in locating his code sections during debugging.

TurboDOS 1.3 allows both MP/M and TurboDOS system calls.

The TDL/XITAN series of assemblers allows more than one source module to be contained within a file. All modules except the last must end with a .PRGEND instead of the normal .END. This alerts the assembler that a logical assembly is complete, but more modules should be expected. Since my library takes advantage of this and is structured as a single file, only the last module (Listing 10) will be terminated by the .END pseudo-op; all others use .PRGEND.

A global used in several modules is defined in the module MTBL (Listing 7). This global is called BASEYR, and it contains the value 78 (for the year 1978, which is the Base Year of MP/M's Julian date numbering system). This was defined as an external to most modules in an attempt to allow them to be used with early releases of TurboDOS, which (as mentioned previously) use a base year of 1948.

```
002E'   3A 0061'                LDA     DATBUF+2        ;(TO FIT DATE ROUTINES)
0031'   32 005F'                STA     DATBUF
0034'   22 0060'                SHLD    DATBUF+1
0037'   21 005F'                LXI     H,DATBUF
003A'   CD 0000:05              CALL    TOJUL#
003D'   D0                      RNC                     ;RETURN IF GOOD DATE
003E'   3E01            ERROR1: MVI     A,1             ;RETURN ERROR #1
0040'   180A                    JMPR    ERROR
0042'   3E02            ERROR2: MVI     A,2             ;RETURN ERROR #2
0044'   1806                    JMPR    ERROR
0046'   3E03            ERROR3: MVI     A,3             ;RETURN ERROR #3
0048'   1802                    JMPR    ERROR
004A'   3E04            ERROR4: MVI     A,4             ;RETURN ERROR #4
004C'   37              ERROR:  STC                     ;FLAG THE ERROR
004D'   ED7B 005D'              LSPD    SPSAVE          ;RESTORE STACK LEVEL
0051'   C9                      RET
                        ;
0052'   FE2F            DELMCK: CPI     '/'
0054'   C8                      RZ
0055'   FE20                    CPI     ' '
0057'   C8                      RZ
0058'   78                      MOV     A,B             ;TEST FOR 3RD PASS
0059'   3D                      DCR     A               ;(ONLY TIME ZERO TERM ALLOWED)
005A'   C0                      RNZ
005B'   B7                      ORA     A               ;3RD PASS, TEST FOR ZERO
005C'   C9                      RET
                        ;
005D'   0000            SPSAVE: .WORD   0
005F'   000000          DATBUF: .BYTE   0,0,0
                        ;
                                .PRGEND
------------------------------------------------------------
                        *** LISTING #4 ***
------------------------------------------------------------
                        ;
                        ;
                                .IDENT  PARSTM          ;05/17/82
                        ;
                        ;
                        ; PARSE TIME FROM INPUT STRING HL TO TIME/DATE
                        ; BUFFER @DE. CY= ERROR
                        ;
                                .ENTRY  PARSTM
                        ;
0000'   13              PARSTM: INX     D               ;SKIP OVER YEAR
0001'   13                      INX     D               ;TO GET TO TIME
0002'   D5                      PUSH    D               ;SAVE IT
0003'   CD 0012'                CALL    ..PARS          ;PARSE TIME
0006'   D1                      POP     D               ;DESTINATION
0007'   D8                      RC                      ;NO XFER IF ERROR
0008'   21 006B'                LXI     H,BUFR          ;FETCH SOURCE
000B'   01 0003                 LXI     B,3
000E'   EDB0                    LDIR                    ;MOVE VALID TIME IN
0010'   B7                      ORA     A               ;INSURE CY CLEAR
0011'   C9                      RET
                        ;
0012'   11 006B'        ..PARS: LXI     D,BUFR          ;TEMPORARY HOLD
0015'   0624                    MVI     B,24H           ;PASS LIMIT FOR HOURS
0017'   CD 0044'                CALL    ..BCD           ;GET BCD NUMBER
001A'   D8                      RC                      ;RETURN IF ERROR
001B'   12                      STAX    D               ;STORE IT
001C'   13                      INX     D               ;POINT TO NEXT
001D'   CD 0059'                CALL    DEL1            ;CHECK DELIMITER
0020'   D8                      RC                      ;BOMB OUT IF BAD DELIMITER
0021'   0660                    MVI     B,60H           ;PASS LIMIT FOR MINUTES
0023'   CD 0044'                CALL    ..BCD           ;GET BCD NUMBER
0026'   D8                      RC                      ;PASS ERROR BACK
0027'   12                      STAX    D               ;STORE IT
0028'   13                      INX     D               ;POINT TO NEXT
0029'   CD 0059'                CALL    DEL1            ;TEST DELIMITER
002C'   3006                    JRNC    ..3RD           ;SKIP IF OK
002E'   B7                      ORA     A               ;EARLY DONE (HH:MM ONLY)
002F'   37                      STC
0030'   C0                      RNZ                     ;ERROR IF NOT
0031'   12                      STAX    D               ;STORE SECONDS=0
0032'   3F                      CMC                     ;TURN OFF CARRY
0033'   C9                      RET
0034'   0660            ..3RD:  MVI     B,60H           ;PASS LIMIT FOR SECONDS
0036'   CD 0044'                CALL    ..BCD           ;GET BCD NUMBER
0039'   D8                      RC                      ;RETURN BAD # ERROR
003A'   12                      STAX    D
003B'   CD 0059'                CALL    DEL1            ;TEST DELIMITER
003E'   D0                      RNC                     ;RETURN IF GOOD
003F'   B7                      ORA     A               ;ALLOW NULL TERMINATOR AT END
0040'   37                      STC
0041'   C0                      RNZ                     ;RETURN ERROR IF NOT NULL
0042'   3F                      CMC                     ;IT'S NULL, CY=0
0043'   C9                      RET
                        ;
0044'   D5              ..BCD:  PUSH    D
0045'   C5                      PUSH    B
0046'   CD 0000:04              CALL    BCDIN#          ;GET BCD VALUE
0049'   32 006A'                STA     DELIM           ;SAVE DELIMITER
004C'   C1                      POP     B
004D'   3808                    JRC     ..BACK          ;IF ERROR...
004F'   7A                      MOV     A,D             ;INSURE <100
0050'   B7                      ORA     A
0051'   37                      STC
0052'   2003                    JRNZ    ..BACK
0054'   7B                      MOV     A,E
0055'   B8                      CMP     B               ;TEST AGAINST PASSED LIMIT
0056'   3F                      CMC                     ;PERVERT CARRY
0057'   D1              ..BACK: POP     D
0058'   C9                      RET
                        ;
                        ; TEST VALID DELIMITERS (CY=1 IF INVALID)
                        ;
0059'   3A 006A'        DEL1:   LDA     DELIM
005C'   FE3A                    CPI     ':'
005E'   37                      STC
005F'   C8                      RZ
0060'   FE20                    CPI     ' '
0062'   37                      STC
0063'   C8                      RZ
0064'   FE2F                    CPI     '/'
0066'   37                      STC
0067'   C8                      RZ
0068'   3F                      CMC
0069'   C9                      RET
```

JULIAN DATES

Continued from page 85

Note that the ordering of the modules does not correspond to the order in which they are presented in the text. Since they are part of a relocatable library, I've ordered them such that all external references are forward, allowing the library to be linked in a single pass of the linkage editor.

Printing the date

Listings 1 and 2 are routines for printing the date and time, respectively, from a date/time structure pointed to by HL. This structure is identical to that returned by MP/M function #105. Thus, a typical calling sequence look like this:

```
LXI   D,DBLOCK   ;pass pointer to
                  structure
PUSH D           ;put 2 copies on
                  stack
```

Several modules can be contained in a single file.

```
PUSH D
MVI   C,105      ;function call
                  number
CALL BDOS        ;get date/time
POP   H          ;get a pointer to
                  date/time
CALL PRDATE      ;print date
                  (mm/dd/yy)
MVI   A,' '      ;separate date
                  and time
CALL TYPE        ;with a blank
POP   H          ;2nd copy of
                  struct. pntr
CALL PRTIME      ;print the time
```

Note that the routine TYPE is called from both modules (and from several others in the various listings). This global routine must reside in the calling program and should print the character passed in register A on the console, while preserving all registers other than A. Supplementary routines DECOTB (decimal output byte) and PRTBCD (BCD output byte) are provided in Listing 10.

The most noteworthy aspect of these routines is the conversion of the

```
                        ;
                        ;
                        ;
006A'   00              DELIM:  .BYTF   0
006B'   000000          BUFR:   .BYTF   0,0,0
                        ;
                                .PRGEND
------------------------------------------------------------
                        *** LISTING #5 ***
------------------------------------------------------------
                        ;
                        ;
                                .IDENT  TOJUL           ;5/20/82
                        ;
                        ;
                        ; ROUTINE TO CONVERT DATE @HL TO MODIFIED
                        ; JULIAN IN HL.  PASSED BUFFER LOOKS LIKE:
                        ;
                        ;       .BYTE   YEAR  (BCD)
                        ;       .BYTE   MONTH   "
                        ;       .BYTE   DATE    "
                        ;
                        ; RETURNED JULIAN DATE IS THE DAY NUMBER
                        ; USING A BASE DATE OF JAN 1 OF "BASEYR"
                        ; RETURNS CY=1 IF A PASSED PARM WAS ILLEGAL
                        ;
                                .ENTRY  TOJUL
                        ;
                        ;
                        ; STRUCTURE OF PASSED BUFFER
                        ;
0000                    YROFS   =       0               ;YEAR OFFSET
0001                    MNOFS   =       1               ;MONTH OFFSET
0002                    DTOFS   =       2               ;DATE OFFSET
                        ;
0000'   D5              TOJUL:  PUSH    D               ;SAVE WORK REGS
0001'   C5                      PUSH    B
0002'   DDE5                    PUSH    X
0004'   7E                      MOV     A,M             ;GET YEAR
0005'   E5                      PUSH    H               ;GFT USER BUFR PTR TO X
0006'   DDE1                    POP     X
                        ;
0008'   FE64                    CPI     100             ;PAST DECIMAL?
000A'   D2 005F                 JNC     ERROR           ;THEN GO ERROR OUT
000D'   E603                    ANI     3               ;IS THIS A LEAP YEAR?
000F'   3E1C                    MVI     A,28            ;(FEBRUARY 28-DAY MONTH)
0011'   2001                    JRNZ    ..SKIP          ;JUMP IF NOT
0013'   3C                      INR     A               ;IT IS, GET 29 DAYS FOR FEB
0014'   32 0000:04      ..SKIP: STA     MFEB#           ;FIX FEBRUARY
0017'   21 0000                 LXI     H,0             ;INIT JDATE
001A'   11 0000:05              LXI     D,MTBL#         ;GFT MONTH TABLF POINTER
001D'   DD7E01                  MOV     A,MNOFS(X)      ;GET CURRENT MONTH
0020'   3D                      DCR     A               ;MAKE IT 0-RELATIVE
0021'   FA 005F'                JM      ERROR           ;CAN'T HAVE A MONTH=0
0024'   FE0C                    CPI     12              ;TEST UPPER
0026'   D2 005F'                JNC     ERROR
0029'   47                      MOV     B,A             ;USE IT AS A COUNTER IN B
                        ;
                        ; SUM DAYS FROM MONTHS IN CURRENT YEAR
                        ;
002A'   78              ..SUM:  MOV     A,B             ;COUNT EXPIRED?
002B'   B7                      ORA     A               ;(MONTH CNTR=0)
002C'   2808                    JRZ     ..DAY           ;THEN JUMP OUT
002E'   05                      DCR     B               ;NO, COUNT DOWN FOR NEXT PASS
002F'   1A                      LDAX    D               ;FETCH # DAYS
0030'   13                      INX     D               ;POINT TO NEXT MONTH
0031'   CD 0065'                CALL    ADDHL           ;ADD IT IN
0034'   18F4                    JMPR    ..SUM
                        ;
                        ; NOW ADD IN THE # DAYS IN CURRENT MONTH
                        ;
0036'   DD7E02          ..DAY:  MOV     A,DTOFS(X)      ;GET DATE
0039'   3D                      DCR     A               ;FIX SO CARRY WORKS ON COMPARE
003A'   FA 005F'                JM      ERROR           ;CAN'T HAVE DATE=0
003D'   EB                      XCHG                    ;HL=PNTR INTO MONTH TABLE
003E'   BE                      CMP     M               ;TEST DATE AGAINST MAX FOR CURNT
003F'   EB                      XCHG                    ;RESTORE JUL DATE TO HL
0040'   D2 005F'                JNC     ERROR
0043'   3C                      INR     A               ;FIX PREV DCR A
0044'   CD 0065'                CALL    ADDHL           ;ADD IT IN
                        ;
                        ; HL HAS JULIAN DATE FOR CURRENT YR.  ADD IN DAYS
                        ; SINCE BASEYR TO GET SYSTEM DATE.
                        ;
0047'   DD7E00                  MOV     A,YROFS(X)      ;GET CURRENT YEAR
004A'   47                      MOV     B,A             ;SAVE FOR COMPARISONS
004B'   3A 0000:06              LDA     BASEYR#         ;FETCH BASE YEAR
004E'   4F                      MOV     C,A             ;BASE YEAR TO C REG
004F'   11 016D                 LXI     D,365           ;DAYS/YEAR
0052'   79              ..LOP:  MOV     A,C             ;GET WORKING YEAR #
0053'   B8                      CMP     B               ;ARE WE UP TO CURRENT?
0054'   280A                    JRZ     BACK            ;THEN FINISHED (AND CY=CLEAR)
0056'   0C                      INR     C               ;SET YEAR FOR NEXT PASS
0057'   19                      DAD     D               ;NO, ADD IN ANOTHER YEAR
0058'   E603                    ANI     3               ;DID WE ADD IN A LEAP YEAR?
005A'   20F6                    JRNZ    ..LOP           ;NO.
005C'   23                      INX     H               ;YES, ADD IN ANOTHER DAY
005D'   18F3                    JMPR    ..LOP
                        ;
005F'   37              ERROR:  STC                     ;RETURN ERROR
0060'   DDE1            BACK:   POP     X               ;RESTORE WORKING REGS
0062'   D1                      POP     D
0063'   C1                      POP     B
0064'   C9                      RET
                        ;
                        ; ROUTINE TO ADD A TO HL
                        ;
0065'   85              ADDHL:  ADD     L
0066'   6F                      MOV     L,A
0067'   D0                      RNC
0068'   24                      INR     H
0069'   C9                      RET
                        ;
                                .PRGEND
```

Drowning in a sea of solutions?

Help's within reach...

Applications backlog. Technology blur. Mounting user pressures. Staying abreast of changing prices. A tidal wave of problems!

Even if some answers *are* available, you have to go to too many sources and sift through so much information, there's not enough time to locate the solutions to your immediate and most important problems.

Well, help IS within easy reach—in the form of an information system that displays all your current software, hardware and communications options for instant reference.

That system is DATA SOURCES, and you're invited to make it work for you. Simply fill in the coupon and we'll send you DATA SOURCES' 4 quarterly editions at 33% off their regular price.

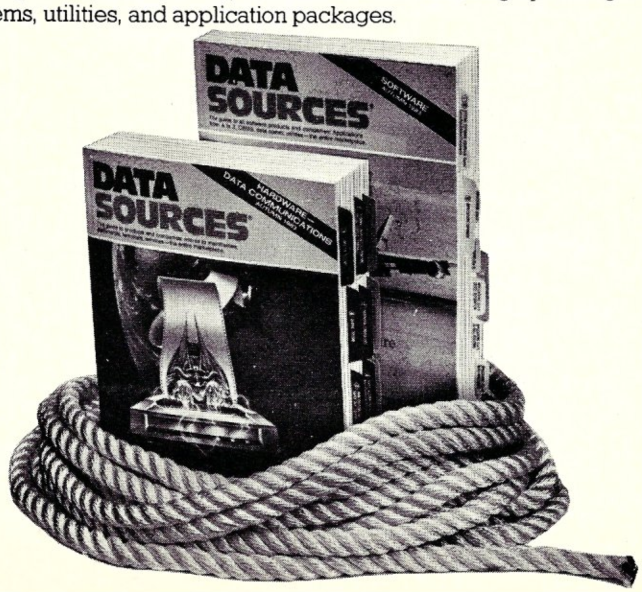
What DATA SOURCES does is organize your options. It guarantees that you've considered all relevant products by guiding you to those with the best possibility of success.

DATA SOURCES is a conveniently sized, extremely well-organized and skillfully indexed 2-volume reference...maintained on a data-base, constantly updated and published quarterly.

YOU GET...

HARDWARE—Over 16,000 products from Micros to Mainframes including peripherals—terminals, printers, memory devices, etc. ..organized by systems compatibility.

SOFTWARE—The largest available inventory of 18,000 business packages for micros, minis, and mainframes including operating systems, utilities, and application packages.



COMMUNICATIONS EQUIPMENT—From modems and multiplexors to local networks, carriers and services, plus all diagnostic and test equipment.

SERVICES, SUPPLIER, SUPPORT—Most complete organized listing of suppliers from installation design to maintenance and data center operations—over 8,000 companies.

NEW FEATURES...

- **Software Index—by package name.**
- **Quick-reference product comparison charts.**
- **Before-you-buy selection criteria and checklist.**
- **Product Hotline—the very latest information on products announced between issues of Data Sources.**

ORDER NOW AND SAVE $60

Yes! Rush me the latest two-volume edition of DATA SOURCES and enter my subscription at $120.00 for 4 quarterly editions. I save a full 33% off the regular price of $180.

Name____________________

Title____________________

Company____________________

Address____________________

City__________ State______ Zip______

Telephone____________________

Nature of Company's Business____________________

Please check appropriate box: ☐ END USER ☐ VAR/OEM

☐ DISTRIBUTOR ☐ DP MFG ☐ OTHER__________

An information product of Ziff-Davis Publishing Company. Satisfaction guaranteed.

SAVE TIME...CALL DIRECT **1-800-227-1617**

EXT. 251

(In Ca., 1-800-772-3545 Ext. 251)

Return coupon to:

DATA SOURCES

P.O. Box 5845, Cherry Hill, N.J. 08034.

T206

Residents of Ca., Co., Ct., D.C., Fl., Il., Ma., Mi., Mo., N.J., N.Y., Vt., please add applicable state taxes.

JULIAN DATES

Continued from page 86

MP/M-format Julian date to a string of the form mm/dd/yy. This tedious job is performed by the routine FMJUL (*from* Julian) in Listing 6. FMJUL is called with the Julian date in HL, and with DE pointing to a 3-byte output area that will hold the converted date in binary form. PRDATE then simply prints the decimal digits returned by FMJUL.

Another routine I'll describe shortly does the opposite of FMJUL (converts BCD dates to modified-Julian), and is called TOJUL. Both FMJUL and TOJUL use an external table of days per month (MTBL), which you'll find in Listing 7.

Date arithmetic

I'm including two other routines from my library, both of which do some calculations based on time specifications.

The library saved me from the tedium of rewriting a subroutine.

The first of these is ET (Elapsed Time; see Listing 8)—this routine calculates the difference between two date/time specifications. The "oldest" (earliest chronologically) is passed in the HL register pair, while the more recent is passed in DE. A pointer is passed in BC to a 3-byte area that will hold the result (see the listing for the format of this result area). This routine is valid for up to 9,999 hours and 59 minutes of elapsed time, and may be extended to even larger amounts of time with a little effort.

Incidently, this routine should provide some justification for the format chosen by Digital Research in the original MP/M. Arithmetic manipulation of dates is much easier with a numeric form of date. If the system provided the date in the form of an ASCII string, additional conversion work would be necessary to convert the date to a number.

The second routine is called ADDTIM (Listing 9), and is designed as a companion to ET. ADDTIM calcu-

```
------------------------------------------------------------
                    *** LISTING #6 ***
------------------------------------------------------------
                         ;
                         ;
                         ;
                                 .IDENT   FMJUL          ;5/17/82
                         ;
                         ;
                         ; ROUTINE TO CONVERT MODIFIED JULIAN DATE
                         ; PASSED IN HL TO YY/MM/DD, INTO BUFFER
                         ; PASSED IN DE. PASSED BUFFER LOOKS LIKE:
                         ;
                         ;       .BYTE    YEAR  (BCD)
                         ;       .BYTE    MONTH   "
                         ;       .BYTE    DATE    "
                         ;
                         ; RETURNED JULIAN DATE IS THE DAY NUMBER
                         ; USING A BASE DATE OF JAN 1 OF "BASEYR"
                         ; (DEFINED IN MTBL MODULE)
                         ;
                                 .ENTRY   FMJUL
                         ;
                         ;
                         ; STRUCTURE OF PASSED BUFFER
                         ;
0000                     YROFS    =       0              ;YEAR OFFSET
0001                     MNOFS    =       1              ;MONTH OFFSET
0002                     DTOFS    =       2              ;DATE OFFSET
                         ;
0000'   D5               FMJUL:  PUSH     D              ;SAVE WORK REGS
0001'   C5                       PUSH     B
0002'   DDE5                     PUSH     X
                         ;
0004'   D5                       PUSH     D              ;GET USER BUFR PTR TO X
0005'   DDE1                     POP      X
0007'   3A 0000:04               LDA      BASEYR#        ;FETCH BASE YEAR
000A'   47                       MOV      B,A            ;SAVE IT IN B
                         ;
                         ; SCAN THROUGH YEARS STARTING AT BASEYR
                         ; SUBTRACTING DAYS FROM OUR CURRENT DATE
                         ;
000B'   78               ..LOOP: MOV      A,B            ;FETCH CURRENT YEAR
000C'   E603                     ANI      3              ;TEST FOR LEAP YEAR
000E'   11 016D                  LXI      D,365          ;FIRST GET # LP YR # DAYS
0011'   2001                     JRNZ     ..NL1          ;AND SKIP IF NOT LP YEAR
0013'   13                       INX      D              ;IS LEAP YEAR, FIX UP DAYS
0014'   04               ..NL1:  INR      B              ;BUMP YR FOR NEXT PASS
0015'   B7                       ORA      A              ;CLEAR CY FOR DBL SUBTRACT
0016'   ED52                     DSBC     D              ;SUBTRACT DAYS IN CUR YR
0018'   2802                     JRZ      ..ENDY         ;JUMP IF LAST DAY OF YR
001A'   30EF                     JRNC     ..LOOP         ;CONTINUE TILL WE OVERSHOOT
001C'   19               ..ENDY: DAD      D              ;OVERSHOT, CORRECT DAYS
001D'   05                       DCR      B              ;AND YEAR
001E'   78                       MOV      A,B            ;FETCH CALCULATED YEAR
001F'   FE64                     CPI      100            ;REWIND EVERY CENTURY
0021'   3802                     JRC      ..PYR          ;(NOW THAT'S ACCURACY!)
0023'   0600                     MVI      B,0            ;REWIND YR 100 TO 0
0025'   DD7000           ..PYR:  MOV      YROFS(X),B     ;PUT AWAY THE YEAR
0028'   78                       MOV      A,B            ;NOW CHECK FOR LEAP AGAIN
0029'   E603                     ANI      3
002B'   3E1C                     MVI      A,28           ;28 DAYS FOR FEB
002D'   2001                     JRNZ     ..NL2          ;GO STORE IF NOT LP YEAR
002F'   3C                       INR      A              ;ADJUST FOR LEAP YEAR
0030'   32 0000:05       ..NL2:  STA      MFEB#          ;FIX MONTH TABLE
0033'   11 0000:06               LXI      D,MTBL#        ;POINT TO MONTH TABLE
0036'   0600                     MVI      B,0            ;INIT MONTH NUMBER
                         ;
                         ; SCAN THROUGH MONTHS IN CURNT YR, SUBTRACTING DAYS
                         ;
0038'   D5               ..MNTH: PUSH     D              ;SAVE MONTH TABLE POINTER
0039'   1A                       LDAX     D              ;GET DAYS IN CURNT MONTH
003A'   5F                       MOV      E,A            ;GET IN DE AS 16 BITS
003B'   1600                     MVI      D,0
003D'   B7                       ORA      A              ;CLR CY FOR DSBC
003E'   ED52                     DSBC     D
0040'   3807                     JRC      ..FND          ;EXIT LOOP WHEN OVERSHOOT
0042'   2805                     JRZ      ..FND          ;LAST DAY OF MONTH?
0044'   04                       INR      B              ;BUMP TO NEXT MONTH
0045'   D1                       POP      D              ;RETRIEVE MONTH TBL POINTER
0046'   13                       INX      D              ;BUMP TO NEXT MONTH
0047'   18EF                     JMPR     ..MNTH         ;AND CONTINUE
                         ;
0049'   F1               ..FND:  POP      PSW            ;CLEAR STACK
004A'   19                       DAD      D              ;FIX OVERSHOOT
004B'   04                       INR      B              ;MAKE MONTH 1-RELATIVE
004C'   DD7001                   MOV      MNOFS(X),B     ;PUT MONTH IN BUFFER
004F'   DD7502                   MOV      DTOFS(X),L     ;REMAINDER IS DATE
                         ;
0052'   DDE1                     POP      X              ;RESTORE WORKING REGS
0054'   C1                       POP      B
0055'   D1                       POP      D
0056'   C9                       RET
                         ;
                                 .PRGEND
```

```
------------------------------------------------------------
                    *** LISTING #7 ***
------------------------------------------------------------
                         ;
                         ;
                                 .IDENT   MTBL
                         ;
                         ;
                         ; TABLE OF MONTHS FOR TOJUL AND FMJUL
                         ; ALSO CONTAINS DEFINITION FOR BASE YEAR
                         ;
                                 .ENTRY   MTBL,MFEB,BASEYR
                         ;
0000'   1F               MTBL:   .BYTE    31                     ;JAN
0001'   1C               MFEB:   .BYTE    28                     ;FEB
0002'   1F1F1F1E                 .BYTE    31,30,31,30            ;MAR-JUN
0006'   1F1F1E1F1E1F             .BYTE    31,31,30,31,30,31      ;JUL-DEC
                         ;
000C'   4E               BASEYR: .BYTE    78                     ;BASE YEAR=1978
                         ;
                                 .PRGEND
```

LIFEBOAT™ Associates:
The full support software source.

Reach for the programming horizon of the 80's with Lattice™ C, the fastest C compiler.

Set the course of your next software project towards greater power and portability by selecting Lattice C, recognized as the finest and fastest 16-bit C compiler. Lattice C, the full implementation of Kernighan and Ritchie, continues to lead the way by announcing full memory management up to 1 Megabyte. Major software houses including Microsoft, MicroPro™ and Sorcim™ are using Lattice C to develop their programs.

Lattice C is available for a wide variety of 16-bit personal computers including IBM®, NCR®, Texas Instruments, Victor, Wang and other microcomputers running PC™-DOS, MS™-DOS and CP/M86.™

Call LIFEBOAT at 212-860-0300 for free information on the Lattice C family of software development tools.

LIFEBOAT
Associates

Lifeboat offers Lattice C with a tested set of software tools to provide a complete development system including:

- HALO™
- PANEL™
- PMATE™
- PLINK™-86
- C-FOOD SMORGASBORD™
- LATTICE WINDOW™
- FLOAT87™

- Color graphic primatives
- Screen design aid
- Customizable program editor
- Overlay linkage editor
- Screen and I/O utilities
- Multi-window functions
- 8087 floating point math

LIFEBOAT
Associates

1651 Third Avenue
New York, NY 10028
212-860-0300

Please send me free information on:

- ☐ Lattice and development tools
- ☐ Corporate purchase program
- ☐ Dealer program
- ☐ OEM agreements

☐ Send me the complete LIFEBOAT software catalog. $3.00 enclosed for postage and handling.

Name
Company
Address
City
State Zip
Telephone

CIRCLE 5 ON READER SERVICE CARD

JULIAN DATES

Continued from page 88

lates cumulative time and takes as input the result area from ET (the one that was passed to ET in the BC register pair). A pointer to this result area is passed to ADDTIM in the DE pair. Another parameter required by ADDTIM is a cumulative-time area; a pointer to this area is expected in HL. The elapsed time is added to the cumulative time, and the result is stored back into the cumulative-time area.

These routines might find typical application in a remotely accessible MP/M system, where usage statistics might be desired for cost-accounting purposes.

Input parsing

The final routines in my library perform parsing of date and time from strings; this might be useful in an application where date and time specs are taken from the console operator interactively and must be converted to the more compact MP/M format (perhaps to be written into tightly packed disk records, or simply to allow calculations using routines such as ET).

The ET routine is valid for up to 9,999 hours and 59 minutes.

These routines are rather limited in the format they will accept. Dates must be of the form mm/dd/yy (e.g., 5 December 1982 would be input as 12/05/82); times must be of the form hh:mm:ss (2:05 p.m. would be input as 14:05:00). Note that seconds are optional.

The routine PARSDT (Listing 3) parses a date from the input string passed as a pointer in HL and returns the modified Julian date (in the MP/M format) in HL. This routine calls on a secondary routine TOJUL (To Julian, Listing 5) that is the opposite of FMJUL described earlier. PARSDT calls on a decimal input routine, DECIN, included in the common subroutines module in Listing 10.

```
-------------------------------------------------------
                   *** LISTING #8 ***
-------------------------------------------------------
                  ;
                  ;
                         .IDENT  ET       ;RE-WRITTEN 04/21/82
                  ;
                  ;
                  ; ELAPSED TIME ROUTINE
                  ;
                  ; ON ENTRY: HL= "OLD" TIME SPEC          ("THEN")
                  ;           DE= "NEW" TIME SPEC          ("NOW")
                  ;           BC= ELAPSED TIME SPEC+3
                  ;
                  ; PERFORMS (ET @ BC) = (TIME @ DE) - (TIME @ HL)
                  ;
                  ;                 TIME SPEC FORMAT:
                  ;                 -----------------
                  ;      .WORD      DATE  (MODIFIED JULIAN)
                  ;      .BYTE      HOURS (BCD)
                  ;      .BYTE      MINS  (BCD)
                  ;
                  ;                 ELAPSED TIME FORMAT:
                  ;                 --------------------
                  ;      .BYTE      HOURS- THOUSAND,HUNDREDS \
                  ;      .BYTE      HOURS- TENS, UNITS        > ALL BCD
                  ;      .BYTE      MINS - TENS, UNITS       / (2 DIGS/BYTE)
                  ;
                  ;
                         .ENTRY  ET
                  ;
0000'   AF        ET:    XRA     A              ;ZERO OUT CARRIES
0001'   32 0076'         STA     MINCAR         ;MINUTES
0004'   32 0077'         STA     HRSCAR         ;AND HOURS
                  ;
0007'   23               INX     H              ;BUMP "THEN" TO MINUTES
0008'   23               INX     H
0009'   23               INX     H
000A'   13               INX     D              ;BUMP "NOW" TO MINUTES
000B'   13               INX     D
000C'   13               INX     D
000D'   03               INX     B              ;POINT TO MINUTES OF ET
000E'   03               INX     B
                  ;
000F'   1A               LDAX    D              ;FETCH "NOW" MINUTES
0010'   96               SUB     M              ;SUBTRACT "THEN" MINUTES
0011'   27               DAA
0012'   02               STAX    B
0013'   3009             JRNC    DOHRS          ;GOOD IF NO CARRY
0015'   C660             ADI     60H            ;CARRIED, BORROW 60 MINUTES
0017'   27               DAA
0018'   02               STAX    B
0019'   3E01             MVI     A,1            ;AND SHOW THE BORROW
001B'   32 0076'         STA     MINCAR
001E'   2B        DOHRS: DCX     H              ;BUMP DOWN TO HRS
001F'   1B               DCX     D
0020'   0B               DCX     B
0021'   1A               LDAX    D              ;FETCH "NOW" HOURS
0022'   96               SUB     M              ;SUBTRACT "THEN" HOURS
0023'   27               DAA
0024'   300B             JRNC    ..A            ;SKIP IF NO BORROW
0026'   C624             ADI     24H            ;BORROW A DAY
0028'   27               DAA
0029'   02               STAX    B
002A'   F5               PUSH    PSW            ;DON'T ALTER CALCULATION
002B'   3E01             MVI     A,1            ;BUT SET BORROWED DAY FLAG
002D'   32 0077'         STA     HRSCAR
0030'   F1               POP     PSW
0031'   E5        ..A:   PUSH    H              ;SAVE "THEN" POINTER
0032'   21 0076'         LXI     H,MINCAR       ;AND PREPARE TO...
0035'   96               SUB     M              ;SUBTRACT THE MINUTES BORROW
0036'   27               DAA
0037'   E1               POP     H              ;RESTORE "THEN" POINTER
0038'   02               STAX    B
0039'   3009             JRNC    ..DATE         ;AND CONTINUE IF NO MORE BORROWING
003B'   C624             ADI     24H            ;BORROW OUT OF HOURS
003D'   27               DAA                    ;SO ADD A DAY
003E'   02               STAX    B
003F'   3E01             MVI     A,1            ;AND SHOW THE BORROW
0041'   32 0077'         STA     HRSCAR
0044'   1B        ..DATE: DCX    D              ;"NOW" POINTER=>DATE
0045'   1B               DCX     D
0046'   2B               DCX     H              ;DITTO, "THEN" POINTER
0047'   2B               DCX     H
0048'   7E               MOV     A,M            ;LOAD "THEN"
0049'   23               INX     H
004A'   66               MOV     H,M
004B'   6F               MOV     L,A
004C'   EB               XCHG                   ;DE="THEN"
004D'   7E               MOV     A,M            ;HL="NOW"
004E'   23               INX     H
004F'   66               MOV     H,M
0050'   6F               MOV     L,A
0051'   B7               ORA     A              ;CLR CY FOR DSBC
0052'   ED52             DSBC    D              ;HL="NOW"-"THEN"
0054'   3A 0077'         LDA     HRSCAR         ;SEE IF WE BORROWED A DAY
0057'   B7               ORA     A
0058'   2801             JRZ     ..NOC          ;SKIP IF NOT
005A'   2B               DCX     H              ;ELSE TAKE ONE AWAY
005B'   EB        ..NOC: XCHG                   ;DE=# ELAPSED DAYS
005C'   60               MOV     H,B            ;MOVE ET HRS POINTER TO HL
005D'   69               MOV     L,C
005E'   2B               DCX     H              ;JUMP BACK TO 100'S OF HOURS
005F'   3600             MVI     M,0            ;INIT IT TO ZERO
0061'   23               INX     H              ;NOW BACK TO TENS/UNITS
                  ;
                  ; THIS LOOPS ADDS 24 HOURS TO THE ELAPSED TIME
                  ; @HL FOR EACH DAY COUNTED IN DE
                  ;
0062'   7A        DAYLOP: MOV    A,D            ;ANY DAYS LEFT?
0063'   B3               ORA     E
0064'   C8               RZ                     ;DONE IF NOT
0065'   1B               DCX     D              ;YES, COUNT IT DOWN
0066'   7E               MOV     A,M            ;FETCH TENS/UNITS OF ELAPSED HRS
0067'   C624             ADI     24H            ;ADD A DAY
0069'   27               DAA
006A'   77               MOV     M,A            ;UPDATE TENS/UNITS
006B'   30F5             JRNC    DAYLOP         ;CONTINUE IF NO CARRY
```

JULIAN DATES

Continued from page 90

The routine PARSTM (Listing 4) parses the time from the input string passed as a pointer in HL; also passed to this routine is a pointer in DE to an MP/M-format date/time buffer, which will hold the parsed time. This buffer is modified slightly from the MP/M function #105 format (in fact, it is more compatible with MP/M function #155): it must be one byte longer than normal in order to include the "seconds" field. Since this routine was originally written for MP/M function #155, I allowed the presence of seconds in the time string. In fact, if seconds are not specified (they are, as I said, optional), this byte will be zeroed.

PARSTM also uses a support routine, this one for BCD input (BCDIN). This routine may be found in the common subroutines module in Listing 10.

These routines might be useful in a remotely accessible MP/M system.

Strings attached?

A good deal of recently released semi-public-domain software has carried the stipulation that the user not sell the software, or otherwise profit from it. I don't believe that such things as reusable subroutines, such as those presented in this article, should be limited this way. For that reason, the routines presented in this article may be used with no strings attached. If you'd like to include this stuff in your next big commercial project, feel free; I'd be flattered (mind you, if you feel obligated to send me a complimentary copy of your package, I'll not object!). I'd like to see a lot more software, commercial and public domain, exploit the time and date capabilities of these new operating systems.

Mr. Zatarain, did I answer your question?

Ron Fowler, Rte. 1, Box 7, Fort Atkinson, WI 53538.

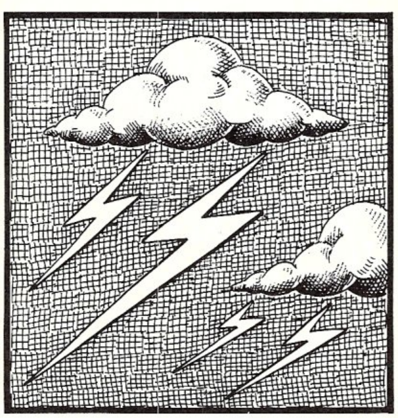
```
006D'   2B                  DCX     H               ;CARRY TO THOUSANDS/HUNDREDS
006E'   7E                  MOV     A,M             ;BUMP IT ONE
006F'   C601                ADI     1
0071'   27                  DAA
0072'   77                  MOV     M,A             ;PUT BACK
0073'   23                  INX     H               ;POINT PROPER.
0074'   18EC                JMPR    DAYLOP          ;AND CONTINUE
                  ;
0076'   00        MINCAR:   .BYTE   0               ;MINUTES CARRY
0077'   00        HRSCAR:   .BYTE   0               ;HOURS CARRY
                  ;
                            .PRGEND
------------------------------------------------------------
                    *** LISTING #9 ***
------------------------------------------------------------
                  ;
                  ;
                            .IDENT  ADDTIM          ;REWRITTEN 04/21/82
                  ;
                  ;
                  ; ADD TIME: ADD CUMULATIVE ELAPSED TIME SPEC
                  ;             @HL TO ELAPSED TIME SPEC @DE,
                  ;              RESULT @HL
                  ;
                  ; ELAPSED TIME FORMAT:
                  ;                          .WORD   HOURS- THOUSAND,HUNDREDS \
                  ;                          .WORD   HOURS- TENS, UNITS        > ALL BCD
                  ;                          .BYTE   MINS - TENS, UNITS       /
                  ;
                  ;
                            .ENTRY  ADDTIM
                  ;
0000'   01 0000   ADDTIM:   LXI     B,0             ;B=HRSCAR  C=MINCAR
                  ;
0003'   23                  INX     H               ;BUMP POINTERS TO MINUTES
0004'   23                  INX     H
0005'   13                  INX     D
0006'   13                  INX     D
                  ;
0007'   1A                  LDAX    D               ;FETCH ELAPSED MINUTED
0008'   86                  ADD     M               ;ADD TO ACCUM MINUTES
0009'   27                  DAA                     ;MAKE IT DECIMAL
000A'   3004                JRNC    ..SKIP          ;SKIP IF NO CARRY
000C'   C640                ADI     40H             ;ADD DIF BETWEEN 60 AND 100
000E'   0E01                MVI     C,1             ;BORROW FROM MINUTES TO HRS
0010'   FE60      ..SKIP:   CPI     60H             ;MINUTES OVERFLOWED?
0012'   3805                JRC     ..HRS           ;DONE IF NOT, GO DO HOURS
0014'   D660                SUI     60H             ;YES, CORRECT OVERSHOOT (Z80 ONLY!!)
0016'   27                  DAA
0017'   0E01                MVI     C,1             ;AND GET A BORROW
0019'   77        ..HRS:    MOV     M,A             ;STORE MINUTES SUM
001A'   1B                  DCX     D               ;BACK UP POINTERS TO HOURS
001B'   2B                  DCX     H
001C'   1A                  LDAX    D               ;FETCH ELAPSED HOURS
001D'   86                  ADD     M               ;ADD CUMULATIVE
001E'   27                  DAA                     ;MAKE IT DECIMAL
001F'   3002                JRNC    ..TENS          ;CONTINUE IF NO HUNDREDS OVFL
0021'   0601                MVI     B,1             ;SET OVERFLOW TO HUNDREDS
0023'   81        ..TENS:   ADD     C               ;ADD IN ANY MINUTES  OVERFLOW
0024'   77                  MOV     M,A             ;AND STORE THE SUM
0025'   3002                JRNC    ..TEN2
0027'   0E01                MVI     C,1             ;SET HUNDREDS OVERFLOW
0029'   2B        ..TEN2:   DCX     H               ;POINT TO HUNDREDS OF HOURS
002A'   1B                  DCX     D
002B'   1A                  LDAX    D
002C'   86                  ADD     M               ;ADD HUNDREDS DIGITS
002D'   27                  DAA
002E'   80                  ADD     B               ;ADD CARRY
002F'   77                  MOV     M,A
0030'   C9                  RET
                  ;
                  ; BUMP CUM HOURS BY ONE
                  ;
0031'   F5        ..ADHR:   PUSH    PSW
0032'   2B                  DCX     H               ;POINT TO CUM HOURS
0033'   7E                  MOV     A,M             ;FETCH CUM HOURS
0034'   C601                ADI     1               ;BUMP BY ONE
0036'   27                  DAA                     ;MAKE DECIMAL
0037'   77                  MOV     M,A             ;STORE SUM
0038'   3007                JRNC    ..BAK           ;SKIP IF NO HUNDREDS OVFL
003A'   2B                  DCX     H
003B'   7E                  MOV     A,M
003C'   C601                ADI     1
003E'   27                  DAA
003F'   77                  MOV     M,A
0040'   23                  INX     H
0041'   23        ..BAK:    INX     H
0042'   F1                  POP     PSW
0043'   C9                  RET
                  ;
                  ;
                            .PRGEND
------------------------------------------------------------
                    *** LISTING #10 ***
------------------------------------------------------------
                  ;
                            .IDENT  COMMON          ;COMMON SUBROUTINES
                  ;
                  ; THIS MODULE CONTAINS BCD AND DECIMAL INPUT AND
                  ; OUTPUT ROUTINES.  THESE ARE CALLED FROM SOME OF
                  ; THE PREVIOUS MODULES
                  ;
                            .ENTRY  DECOUT,DECOTB,PRTBCD,DECIN,BCDIN
                  ;
                  ;
                  ; ROUTINE TO INPUT A BCD VALUE INTO
                  ; DE FROM ASCII STRING AT HL
                  ; RETURN CY IF NO BCD CHARS FOUND
                  ; ON SUCCESSFUL RETURN, A= TERMINATOR,
                  ; AND HL=ONE PAST TERMINATOR
                  ;
0000'   11 0000   BCDIN:    LXI     D,0     ;INIT RETURNED VALUE
0003'   7E                  MOV     A,M     ;FETCH A CHAR
0004'   CD 0020'            CALL    ..HTST  ;TEST IT FOR HEX
0007'   3002                JRNC    ..LOOP  ;SKIP IF OK
0009'   7E                  MOV     A,M     ;BAD, RETURN BAD CHAR IN A
000A'   C9                  RET
000B'   7E        ..LOOP:   MOV     A,M     ;FETCH NEXT CHAR
000C'   CD 0020'            CALL    ..HTST  ;TEST IT
000F'   3004                JRNC    ..CONT
```

```
0011'  7E                 MOV    A,M        ;DONE, FETCH TERMINATOR
0012'  23                 INX    H          ;POINT PAST IT
0013'  B7                 ORA    A          ;RETURN CARRY CLEAR
0014'  C9                 RET
0015'  23        ..CONT:  INX    H
0016'  EB                 XCHG              ;GET VALUE TO HL
0017'  29                 DAD    H          ;MAKE ROOM FOR NEW DIGIT
0018'  29                 DAD    H
0019'  29                 DAD    H
001A'  29                 DAD    H
001B'  B5                 ORA    L          ;ADD IN NEW DIGIT
001C'  6F                 MOV    L,A
001D'  EB                 XCHG
001E'  18EB               JMPR   ..LOOP
                 ;
                 ; TEST FOR VALID BCD DIGIT IN A
                 ;
0020'  D630      ..HTST:  SUI    '0'              ;TEST DECIMAL LOWER LIMIT
0022'  D8                 RC                      ;TOO SMALL, RETURN
0023'  FE0A               CPI    10
0025'  3F                 CMC
0026'  C9                 RET
                 ;
                 ; ROUTINE TO INPUT DECIMAL VALUE INTO
                 ; DE FROM ASCII STRING AT HL
                 ; RETURN CY IF NO DEC CHARS FOUND
                 ; ON SUCCESSFUL RETURN, A= SEPARATOR
                 ;
                 ;
0027'  11 0000   DECIN:   LXI    D,0        ;INIT RETURNED VALUE
002A'  7E                 MOV    A,M        ;FETCH A CHAR
002B'  CD 004B'           CALL   ..DTST     ;TEST IT FOR HEX
002E'  D8                 RC                ;OOPS, BAD
002F'  7E        ..LOOP:  MOV    A,M        ;FETCH NEXT CHAR
0030'  CD 004B'           CALL   ..DTST     ;TEST IT
0033'  3004               JRNC   ..CONT
0035'  7E                 MOV    A,M        ;RETURN DELIMITER IN A
0036'  23                 INX    H          ;POINT PAST IT
0037'  B7                 ORA    A          ;RETURN CY CLEAR
0038'  C9                 RET
0039'  23        ..CONT:  INX    H
003A'  EB                 XCHG              ;GET VALUE TO HL
003B'  C5                 PUSH   B
003C'  44                 MOV    B,H
003D'  4D                 MOV    C,L
003E'  29                 DAD    H          ;MAKE ROOM FOR NEW DIGIT
003F'  29                 DAD    H
0040'  09                 DAD    B          ;(MULT BY 10)
0041'  29                 DAD    H
0042'  85                 ADD    L          ;ADD IN NEW DIGIT
0043'  6F                 MOV    L,A
0044'  3001               JRNC   ..NOOV
0046'  24                 INR    H
0047'  EB        ..NOOV:  XCHG
0048'  C1                 POP    B
0049'  18E4               JMPR   ..LOOP
                 ;
                 ; TEST FOR VALID DECIMAL DIGIT IN A
                 ;
004B'  D630      ..DTST:  SUI    '0'        ;TEST DECIMAL LOWER LIMIT
004D'  D8                 RC                ;TOO SMALL, RETURN
004E'  FE0A               CPI    10
0050'  3F                 CMC
0051'  C9                 RET
                 ;
                 ; PRINT A IN SPLIT BCD
                 ;
0052'  F5        PRTBCD:  PUSH   PSW
0053'  0F                 RRC
0054'  0F                 RRC
0055'  0F                 RRC
0056'  0F                 RRC
0057'  CD 005B'           CALL   ..TYP
005A'  F1                 POP    PSW
005B'  E60F      ..TYP:   ANI    0FH
005D'  C630               ADI    '0'
005F'  CD 0000:04         CALL   TYPE#
0062'  C9                 RET
                 ;
                 ;
                 ; ROUTINES TO PRINT A, HL IN DECIMAL VIA TYPE#
                 ;
                 ;
0063'  E5        DECOTB:  PUSH   H                ;SAVE HL
0064'  6F                 MOV    L,A              ;MAKE A 16 BITS IN HL
0065'  2600               MVI    H,0
0067'  CD 006C'           CALL   DECOUT           ;PRINT VIA DECOUT
006A'  E1                 POP    H
006B'  C9                 RET
                 ;
                 ; OUTPUT DECIMAL HL: FROM CPMUG
                 ;
006C'  C5        DECOUT:  PUSH   B
006D'  D5                 PUSH   D
006E'  E5                 PUSH   H
006F'  01 FFF6            LXI    B,-10
0072'  11 FFFF            LXI    D,-1
0075'  09        ..LOOP:  DAD    B
0076'  13                 INX    D
0077'  38FC               JRC    ..LOOP
0079'  01 000A            LXI    B,10
007C'  09                 DAD    B
007D'  EB                 XCHG
007E'  7C                 MOV    A,H
007F'  B5                 ORA    L
0080'  C4 006C'           CNZ    DECOUT           ;THIS IS RECURSIVE
0083'  7B                 MOV    A,E
0084'  C630               ADI    '0'
0086'  CD 0000:04         CALL   TYPE#
0089'  F1                 POP    H
008A'  D1                 POP    D
008B'  C1                 POP    B
008C'  C9                 RET
                 ;
                          .END                    ;END OF LIBRARY
-----------------------------------------------------------------
                        <END OF LISTINGS>
-----------------------------------------------------------------
```

MICROSYSTEMS REVIEWS

Lomas LDP-2 "Lightning One"

A faster, more powerful alternative to the IBM PC and its look-alikes



by Bruce Ratoff

The truck driver struggled single-handedly with a massive wooden crate. I at first mistakenly assumed it was destined for some nearby furniture store, the thought never crossing my mind for a moment that it might actually be intended for me. After all, 8086-based computers are supposed to be puny little things like the IBM PC, right? Well, I guess not, at least in this case, for the Lomas Data Products LDP-2 turns out to be neither PC-like nor puny.

Boards supplied

Lomas Data Products was one of the first vendors to advertise 8086 and 8088 CPU boards for the S-100 bus when those processor chips became available a few years ago. The system Lomas provided for evaluation was based on their "Lightning One" CPU board, equipped with an 8086 main processor and an 8087 math coprocessor. While the 8086 supplied is capable of running at an 8 MHz clock rate, the board came set up to operate with a 5 MHz clock, since that is the fastest available speed for the accompanying 8087 chip. I also received the Lomas LDP72 floppy disk controller, the "Hazitall" multifunction I/O card, and the RAM67 128K static RAM card. The board set was housed in an Integrand 800-series cabinet, along with a pair of Qume DT-8 8″ dual-sided floppy drives and a 10-slot S-100 cardcage. Since no terminal was provided, I plugged in a TeleVideo 925 which happened to be on hand at the moment.

CPU

The Lomas "Lightning One" is a rather interesting CPU card with a great many options. First of all, it will handle either an 8-bit 8088 or a 16-bit 8086 as the main processor chip. By selecting the proper chip, the user can opt for full compatibility with existing 8-bit memories and peripherals, or build a true 16-bit system with 16-bit memory for maximum speed. Jumper options are provided to run the CPU board with a 4, 5, 8 or 10 MHz main clock, so the board can be set up for all the currently available chip speeds. The "Lightning One" provides sockets for the addition of an 8087 math processor and an 8089 I/O processor. The board we tested had the 8087, but not the 8089, which Lomas uses only in their multiuser systems. In addition to all this CPU power, there is an 8259A interrupt controller and two 28-pin sockets capable of accommodating two 2716, 2732 or 2764 EPROMs. Using two 2764s, you could have up to 16K of on-board firmware.

Disk controller and I/O

The Lomas LDP72 floppy disk controller is based on the Intel 8272 dual-density floppy controller chip. With appropriate software, it will accommodate any mix of up to four 8″ and 5¼″ single- or double-sided drives. Connectors for both drives sizes are provided at the top edge of the card. Our test system had two 8″ drives, and the software was set up to allow for an optional 5¼″ drive as a third drive. Three formats are supported on the 8″ drives: single-sided single-density (SSSD), single-sided double-density (SSDD), and double-sided double-density (DSDD). The storage capacities for each format on CP/M-86 and MS-DOS are:

OS	SSSD	SSDD	DSDD
CP/M-86	243K	600K	1200K
MS-DOS	246K	610K	1221K

Performing the rest of the I/O chores in this system is the aptly named "Hazitall" system support card. It contains two serial ports with software-selectable baud rates up to 9600 baud. Each port supports the major RS-232 handshake lines for asynchronous devices, and one of the ports may also be operated in synchronous mode. There is also a strobed parallel output port (usually configured as a Centronics-style printer interface), a strobed parallel input port, a programmable realtime interrupt (no software provided), a clock/calendar chip with battery backup (supported in the supplied MS-DOS implementation, but not in CP/M-86), a Winchester disk controller support port (intended to interface with a Western Digital WD1000 series hard disk controller), and an empty socket for an Intel 8231 or 8232 math processor. This last item would not normally be used with an 8086 or 8088, since the 8087 does a better job. However, it might be useful if you were to run the Hazitall board with an 8080, 8085 or Z80 processor.

Memory

The Lomas RAM67 is a 128K memory card, built using low-power, high-speed CMOS static memory chips. Due to the low power consumption, it is possible to order a special version of this board with battery backup to yield a fairly large nonvolatile memory system. A half-populated (64K) version is also available. The 128K version can be addressed on any 128K boundary, while the 64K version can be addressed on any 64K boundary. Both versions respond to the S-100 phantom line. It is also possible to disable a section of the board to avoid conflict with another memory device. Eight-bit and 16-bit-wide data transfers are handled automatically using the S-100 sXTRQ signal. The claimed access time of 130 ns should be fast enough to run with any existing 8- or 16-bit processor.

Software and documentation

The software provided by Lomas consisted of evaluation copies of their implementations of CP/M-86 version 1.1 and MS-DOS version 2.0. In order to have some programming languages to try on the system, *Microsystems* contacted Digital Research, Inc., who graciously provided copies of PL/I-86, CB-86 (a native code compiler for CBasic), and their new Personal Basic interpreter. This is where the first snag appeared: the standard release format these days for all CP/M-86 and MS-DOS compatible software is the 5¼″ disk. While the Lomas documentation states that a 5″ drive may be added to the system, the test system contained only 8″ drives. Luckily, my lab also houses an 8″ CP/M-80 based system and an 8-bit 5″ system that can read PC disks. I was therefore able to download all the 5″ release disks to 8″ single-density, and then read those copies on the LDP2. Lomas offers another version of their system, which comes with 5″ drives. Potential buyers would probably do better to consider that model, unless conversion facilities are available.

Lomas supplied two large binders full of documentation for the system. One of these contained the hardware manuals of all their boards. The other binder contained a preliminary manual for MS-DOS version 2.0. A few weeks later, a third large binder arrived, containing a more complete and up-to-date MS-DOS manual. In addition to these large binders, we received the CP/M-86 manual set, in its standard IBM-style minibinders.

Performance

At first the system did not boot up, but a quick scan of the recommended jumper settings listed in the MS-DOS installation notes revealed one discrepancy on the Hazitall board. I'm not sure why, but fixing this jumper allowed the system to come up properly with both CP/M-86 and MS-DOS. One other note page which proved handy was the sheet at the front of the hardware manual, which described the correct way to wire terminal and printer cables. The two serial ports on the Hazitall are laid out as DTE devices, which means that they would connect directly to a modem (modems are DCE devices). Just to increase the confusion, the serial connectors mounted on the unit were female, which usually signifies DCE, not DTE. It would have saved a good deal of time and confusion if Lomas had laid out their connectors in such a way that special cables were not required for the terminal and printer.

My next step in getting the system up was to attempt to create some working disks. The operating systems were supplied on single-sided, double-density disks, which of course worked fine once I solved the jumper problems mentioned above. Since the system was supplied with double-sided drives, I decided to make some double-sided work disks. Well, it turns out that the utilities provided with both operating systems would not generate a double-sided system disk directly. One must instead use the direct hardware disk read and write commands in the system's PROM-resident monitor to transfer the system tracks of a single-sided disk to a double-sided disk. Other than this one anomaly, both CP/M and MS-DOS seemed to take the intermixing of single-density single-sided, double-density single-sided, and double-density double-sided pretty much in stride. Of course, switching formats usually required typing control-C when swapping disks around.

I later discovered one other strange quirk in Lomas' disk handling. Occasionally, when logging in a new disk, the system would just hang up with the heads loaded on the new disk. Whenever this occurred, the system reset button had no effect. The only way to regain control of the system was to switch the power off and on. Each time I did this I kept my fingers crossed, because on most machines removing power from a floppy drive with the heads still loaded usually results in a crunched disk. This

Table 1. Parameter values

Language	Lomas LDP2	PC clone
PL/I-86	7.1	13.2
CB-86	9.5	17.2
Personal BASIC	1:25.4	2:38.5

LIGHTNING ONE

Continued from page 95

did not seem to be a problem on the Lomas, however, since in the dozen or so hangups that I experienced, the disk in question was never harmed.

Once I actually started programming on the system, I was immediately impressed by its speed on disk operations. The disk drives employ a large sector size along with a track-buffering scheme which appears to work quite well. Its disk I/O ranks among the fastest of the many 8″ floppy systems I have used.

Computer-bound operations also seemed to be going faster. In order to see just how much faster, I entered and ran the "Sieve or Erastothenes" benchmark in all three languages provided by Digital Research. For those unfamiliar with this benchmark, it uses a simple process of elimination to determine all the prime numbers in a given range. Since only addition and comparison operations are used, it's a good test of raw CPU speed. Unfortunately, I had not obtained any programming languages that would take advantage of the 8087 math processor. Although it was not available at the time I wrote this, Digital Research has advised me that 8087-compatible versions of several of their language compilers are about to be released. By the time you read this, you should be able to obtain 8087-supporting versions of DR C, Pascal/MT+, Fortran, and PL/I-86. At the present time, all of my testing reflects the use of the 8086 by itself. For the sake of comparison, I ran the same benchmark programs on a PC clone. Since the PC look-alikes also use a 5 MHz clock, the huge timing differences must be attributed to the difference in bus width: 16 bits for the 8086 in the Lomas system, 8 bits for the 8088 in the PC clone. The results I obtained are shown in Table 1.

All times have been given as "seconds.tenths" or "minutes:seconds.tenths". As you can see, the difference in execution times was almost two to one. One can only guess what difference the 8087 might have made on each machine. Perhaps I can investigate and report this at a later date.

Summary

What conclusions can we reach about this machine? Well, the lack of IBM PC compatible graphics and hardware architecture means that many programs for the IBM PC won't run, even though the processor and operating system are the same. On the other hand, more generic MS-DOS and CP/M-86 programs that use only the normal I/O facilities in the operating system will run almost twice as fast, with about four times as much disk space available to them. Of course, with the Lomas machine you also have the easy expandability of the S-100 bus. Of the 10 slots provided in the chassis, only four are used by the basic system. This leaves plenty of room for additional memory and peripherals such as an S-100 modem card.

Maybe someday someone will design an S-100 graphics card that emulates the graphics of the IBM PC. Until then, we can conclude that while this machine is not for everybody, it does succeed in providing a faster, more powerful alternative to the IBM PC and its look-alikes for the serious user.

More information on this system and the rest of the Lomas line may be obtained from: **Lomas Data Products, Inc.,** *66 Hopkinton Road, Westborough, MA 01581, (617) 366-6434.*

CIRCLE 302 ON READER SERVICE CARD

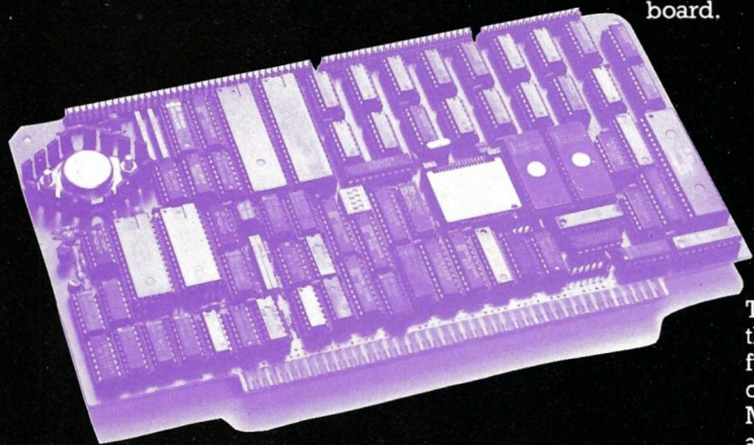
Bruce Ratoff, 26 Broad Street, Cranford, NJ 07016

INTRODUCING
THE ONLY S100 BUS 16 BIT
SINGLE BOARD COMPUTER!

LOMAS DATA PRODUCTS PRESENTS:

THUNDER 186

THUNDER 186 provides the earth shaking performance you have come to expect from LOMAS DATA PRODUCTS, now in a full 16 bit single board computer. THUNDER 186 utilizes a highly integrated 80186 microprocessor from Intel to bring all the features required of an S100 bus computer system together on a single height IEEE-696 board.



The Thunder 186 features:

- 8 Mhz 80186 micro-processor (10 Mhz 8086 performance).
- Floppy disk controller, controls both 5¼″ and 8″ disk drives simultaneously.
- Two RS232 serial ports with full handshaking.
- One parallel printer port.
- Either 128K bytes or 256K bytes of no waitstate dynamic RAM with parity.
- Full IEEE-696 (S100) bus compliance.

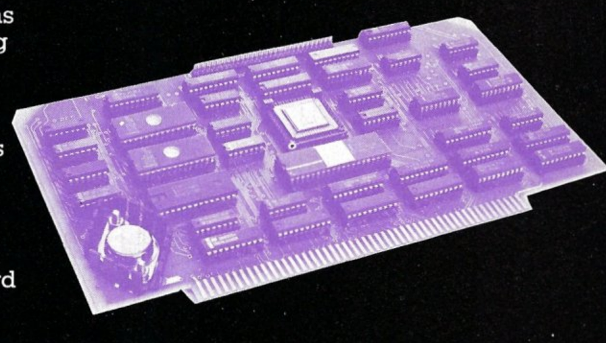
THUNDER 186 is the only single board 16 bit computer that requires no additional boards to complete a functional system. THUNDER 186 is fully supported by our full line of operating systems: MS-DOS**, CP/M-86*, MP/M-86*, and CONCURRENT CP/M-86*, which are all available immediately. Thunder 186 includes CONCURRENT CP/M-86 in the low introductory price.

128K Byte version **$1250.00**
256K Byte version **$1595.00**

LIGHTNING 286

*CP/M-86 and CONCURRENT CP/M-86 are trademarks of Digital Research. **MS-DOS is trademark of Microsoft. ***Lightning One is trademark of Lomas Data Products, Inc.

When we introduced our Lightning One*** in January of 1982 it was the fastest CPU board on the S100 BUS available. Now our Lightning 286 replaces the Lightning One as the fastest board on the S100 BUS. We have incorporated the Intel 80286 microprocessor on a CPU board that again brings new levels of performance to microcomputers. The 80286 is capable of supporting up to 16 megabytes of physical memory and up to 1 Gigabyte of virtual address space when utilized in virtual address mode. In real address mode, the 80286 is compatible with the vast library of 8086/88 programs already written. We have the three major operating systems available immediately on our Lightning 286, no need to wait for third party support to take advantage of this exciting new processor.



The LIGHTNING 286 features:

- Up to 64 Kbytes of onboard EPROM
- 9 Vectored interrupts expandable to 65 levels
- Full compliance with IEEE 696
- Multi-layer board for low-noise reliable operation

In addition to these features, a high speed memory bus expansion port has been included that will allow extension of the memory with a high speed dynamic memory board. The use of the companion memory board will allow use of an 8 Mhz 80286 without memory wait states. The board will contain 1 Mbyte of memory, and will allow cost efficient expansion of 80286 memory space.

Price **$1395.00**

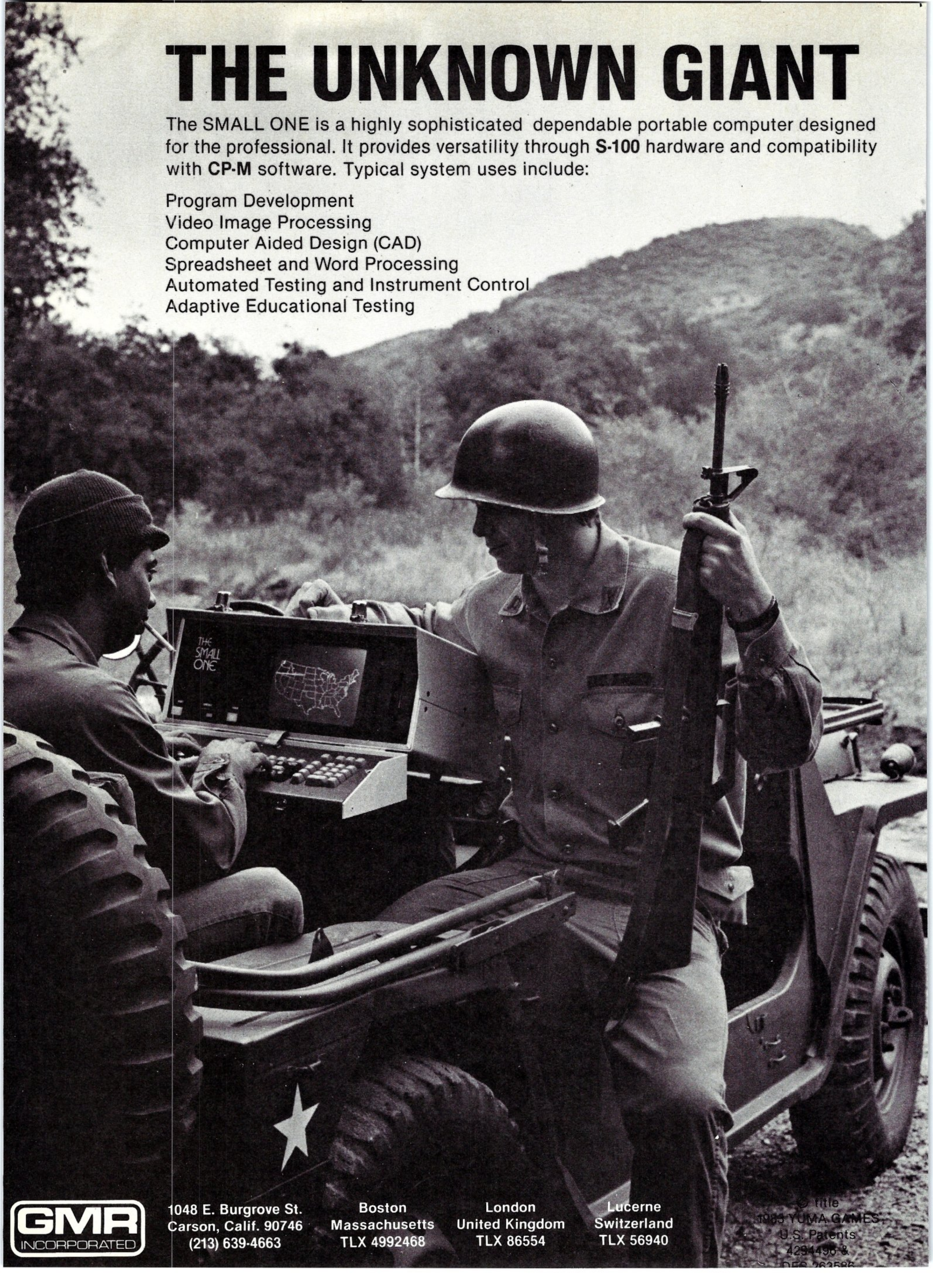
If your application requires 16-bit computing power and versatility, call Lomas Data Products today.

Dealer inquiries invited.

LDP

LOMAS DATA PRODUCTS, INC. □ 66 Hopkinton Road, Westboro, MA 01581 □ Tel: (617) 366-6434

CIRCLE 248 ON READER SERVICE CARD



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

Fortran-80: Simple Changes for I/O

How to eliminate problems with carriage returns and line feeds

by Maynard Brandsma

Microsoft's Fortran-80 has problems with formatted I/O to disk files, as pointed out by Robert S. Minnis in his article in the November 1983 issue of *Microsystems.* The first problem is that formatted writes to a disk file do not convert the carriage control characters of Fortran to the ASCII characters needed by some text editors and all printers to properly display the text. The second problem involves formatted input from a disk file. When an input file is prepared with a text editor that uses a carriage return-line feed combination to mark the end of a line, the presence of line feeds confuses the formatted input as described by Minnis. This technical note presents the code changes to the disk driver routine DSKDRV.MAC necessary to solve these problems. DSKDRV.MAC is the source code for the disk driver routines that Microsoft supplies as part of the Fortran-80 package. I have been using the modified DSKDRV for Fortran-80 version 3.44 with no problems.

One benefit of sending formatted output to a disk file instead of directly to the printer is that programs execute much faster if they don't have to wait for a slow printer. A second benefit is that input and output can be arranged in the same way as on an IBM mainframe. You can follow the IBM convention of reading input data from unit 5 and writing output to unit 6. When a program is through, you can review the output using your text editor. You can decide if you want to print all, or part, or none of the output file.

Modification for the formatted disk read routine

The addition of two extra lines of code to the routine that gets a character from the disk file (DSKIN) will cause this routine to throw away any line feeds it finds in an input record. Listing 1 shows the necessary changes.

Modification of formatted disk write routine

This is the more complicated situation mentioned by Minnis. I have modified the formatted disk write routine to convert the carriage control characters produced by Fortran in column 1 of formatted output to the ASCII characters

FORTRAN-80

Continued from page 99

for line feed or form feed needed by text editors and printers.

With this modification, you can open an output file with a statement like:

```
CALL OPEN(6,'OUTPUT DAT',0).
```

Direct your program's output to that file with program statements like:

```
   WRITE(6,15) A
15 FORMAT(///,5H A = ,F5.1).
```

You can view the output file using your text editor. To print your output you can copy it to the CP/M list device using:

```
PIP LST:=OUTPUT.DAT.
```

Multiple output files can be printed using a command like:

```
PIP LST:=OUT1.DAT,OUT2.DAT,
```

Sending formatted output to a disk file causes programs to execute faster.

The required changes to the formatted disk write routine DSKFWR are shown in Listing 2.

Maynard Brandsma, P.O. Box 374, Drango, CO 81301

LISTING 1. Modification to DSKDRV.MAC to ignore line feeds in formatted disk reads. The routine shown, DSKIN, is part of the lengthy DSKDRV.MAC file. The CPI and JZ instructions inserted near the end cause DSKIN to ignore line feeds in the input file.

```
;-----------------------------------------------------
;
; DISK INPUT ROUTINE
;
DSKIN:
        CALL    GTBOFF          ;Buffer.offset(LUN)
        ORA     A               ;BUFFER EMPTY?
        CZ      REDBUF          ;YES, REFIL
        RC                      ;EOF
        DCR     M               ;DECR. # LEFT
        LXI     H,$FLBUF-2
        DAD     D
        DAD     D               ;GET $FLBUF PTR
        CMA
        ADI     129             ;128-#LEFT=OFFSET
        ADD     M
        MOV     E,A
        INX     H
        MVI     A,0
        ADC     M
        MOV     D,A
        LDAX    D               ;GET CHAR
                                ...patch to ignore line feeds in input
        CPI     12Q             ;line feed?
        JZ      DSKIN           ;yes, ignore it
                                ...end patch
        ORA     A
        RET
```

LISTING 2. Modification to DSKDRV.MAC to add line feeds and form feeds in formatted disk writes. The routine shown, DSKFWR is part of the DSKDRV.MAC file. Replace the section indicated (routine DSKFW2) with the patch shown in Listing 3.

```
;-----------------------------------------------------
;
; FORMATTED WRITE
```

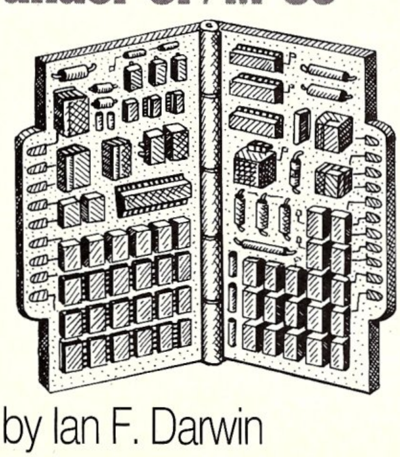
```
;
DSKFWR:
        MVI     C,MD.OUT
        CALL    OPNCHK          ;OPEN FILE IF NEEDED
        XRA     A
        STA     $DSKER          ;CLEAR COUNT
        LDA     $BL             ;GET # TO WRITE
        ORA     A
        RZ      ;IGNORE NULL BUFFERS
        LDA     $BL
        ORA     A
        JP      DSKFW0
        CALL    $ERR            ;Warn User of..
        DB      OBOVF           ; Output Buffer Limit Exceeded
        MVI     A,127           ; and truncate to 127 bytes.
DSKFW0:
        CALL    RNDCHK          ;If Rnd Mode, Set offset to 0.
        LHLD    $BF             ; Get buffer ptr
;
;************************************************************
;                               ;                       ***
DSKFW2:                         ;                       *** REPLACE
        PUSH    PSW             ; SAVE COUNT            *** THIS
        MOV     A,M             ;                       *** SECTION
        CALL    DSKOUT          ;SEND OUT BYTE          *** WITH
        INX     H               ; Increment buffer ptr  *** PATCH
        POP     PSW             ; Retrieve count        *** GIVEN IN
        DCR     A               ; Decrement count       *** LISTING 3
        JNZ     DSKFW2          ; One more time         ***
;                                                       ***
;************************************************************
;
DSKWDN: MVI     A,15Q
        CALL    DSKOUT          ;PUT OUT <CR>
        CALL    GTMODE          ;Mark Mode Byte with
        ORI     MD.WRT          ; Write-Data-in-Buffer.
        MOV     M,A
        LDA     $DSKER          ;GET $DSKEROR STATUS
        ORA     A               ;ERROR?
        RZ      ;NO
        STC     ;YES
        RET
```

LISTING 3. Patch to be inserted where shown in Listing 2.

```
;
; begin patch for formatted writes to disk -----------------------
; patch adds carriage control for formatted output to disk files
        DCR     A               ;decrement buffer length
        PUSH    PSW             ;and save it
        MVI     A,15Q
        CALL    DSKOUT          ;output carriage return
        MOV     A,M             ;GET 1ST CHARACTER IN BUFFER
        CPI     "+"
        JZ      DSKFW2          ;NO PAPER ADVANCE IF 1ST CHAR = "+"
        CPI     "1"
        JNZ     DSKFW1          ;NO PAGE ADVANCE IF 1ST CHAR. NOT "1"
        MVI     A,14Q
        CALL    DSKOUT          ;INSERT FORM FEED IF 1ST CHAR = "1"
        JMP     DSKFW2
;
DSKFW1: MVI     A,12Q           ;CARRIAGE CONTROL
        CALL    DSKOUT          ;ADVANCE PAPER 1 LINE
        MOV     A,M             ;GET CHARACTER BACK
        CPI     "0"             ;
        JNZ     DSKFW2
        MVI     A,12Q           ;LINE FEED
        CALL    DSKOUT          ;IF 1ST CHAR = "0", ADVANCE PAPER ANOTHER LINE
DSKFW2: POP     PSW             ;GET LENGTH BACK
        INX     H               ;INCREMENT POINTER
DSKLOP: RZ                      ;RETURN IF NO MORE CHAR.
        PUSH    PSW             ;SAVE CHARACTER COUNT
        MOV     A,M             ;FETCH NEXT CHARACTER
        INX     H               ;INCREMENT BUFFER POINTER
        CALL    DSKOUT          ;SEND OUT CHARACTER
        POP     PSW             ;RETRIEVE COUNT
        DCR     A               ;DECREMENT COUNT
        JNZ     DSKLOP          ;DO IT AGAIN
; end of patch for carriage control-----------------------
```

Dual Systems SIO-4 I/O Board

This intelligent four-line serial I/O board, first designed for UNIX, performs well under CP/M-80



by Ian F. Darwin

There's a new and improved way of connecting terminals and printers to your S-100 system. The Dual Systems "SIO4/DMA" intelligent four-line serial input/output board was designed from the ground up with UNIX in mind, but can be used quite well with CP/M-80, as this article shows.

In fact, if you have a Dual Systems System/83 UNIX computer (see review in the September 1983 issue) *and are technically inclined,* you could alternate between CP/M 80 and UNIX as I do. Pull the 68000 CPU card, install an 8085 or Z80 CPU board, and change disks. The listing describes the portion of BIOS for the console; a BIOS for the floppy disk controller can be ordered from CompuPro (the secret is out; Dual uses a CompuPro floppy disk controller). Here's how to use the SIO4/DMA with CP/M, but first—the hardware.

The SIO4/DMA board

Like most Dual boards, the SIO4 is very well designed and built. It uses 24-bit (or 16-bit) addressing, has vectored interrupt lines, is a temporary master (the TMA interface), can handle a CPU clock up to 8 MHz, and will insert from 1 to 16 wait states as needed. There is support on board for XON/XOFF protocol if you want it, for printers with a "buffer full" signal, and direct access to the USARTs. The serial line drivers are claimed to conform fully to the RS-232 signal standard. The SIO4 has a limited ability to do synchronous I/O (this could be used with high-speed intercomputer communications). Each line can be wired for a terminal or a modem (DTE or DCE). Vectored interrupts on the VIO lines can be sent for "character ready" in the FIFO and/or DMA output complete. The baud rate on each channel is settable under software control. In fact, for simple serial applications, that's all the programmer has to do, because the board itself is intelligent. It has an 8085 processor on board; the 8085 implements buffering of characters on input and DMA on output. This 8085, and its 2K of software in EPROM (2716), also takes care of initializing the USARTs. A USART is a chip that converts the parallel data from the CPU or system bus into a form suitable for transmission over serial cables to terminals, printers, and modems. The USARTS in the SIO4 are Motorola MC2661s—an extension of the Signetics 2651. And because the 8085 takes care of buffering input and doing DMA on output, the operating system

Dual Systems

Continued from page 103

can tell the board to output an entire line of characters, and have only a single interrupt when the entire line has been sent.

Considerable CPU overhead is freed up for productive use by application programs.

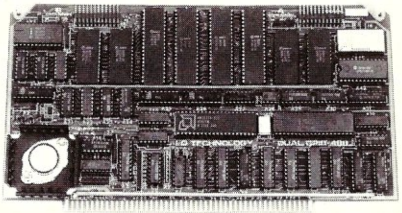
However, CP/M-80 does not know how to utilize such a powerful board, since there is no call to the BIOS to write a line to the console. The developer of the CP/M system assumed that serial I/O boards would never be smarter than they were in 1975 on the particular system he was using. So the BDOS part of CP/M forcibly breaks a call to "write a line on the console" into multiple BIOS calls to "write a character to the console." You could easily write a BIOS routine to send one character at a time to the DMA output processor. Fortunately, the board designer thought to allow you direct access to the USARTs, so the BIOS can just treat the output operation like any non-DMA I/O card. This is what my BIOS does.

The SIO4/DMA also has two modes for doing input. One buffers incoming characters in a FIFO queue (FIFO stands for "First-In, First-Out"). The other is a conventional polled I/O mode. The FIFO mode means that the operating system does not have to read characters as fast as you type them. Since the Godbout BIOS is not interrupt driven, you cannot "type ahead" when using the Godbout "Interfacer 1" serial card (which Dual used in their early systems as the "SIO2" card). If you have a Dual SIO4/DMA card, you can type ahead up to 256 characters while your application is executing. Although the characters are not echoed until they are actually read by the program or by CP/M, it's still quite an improvement! The usual editing characters work as you'd expect. But once you've typed RETURN, the command is on its way! A control-C on a subsequent line will *not* be seen by CP/M until it reads the line. Thus you might occasionally need to hit RESET if you enter a command and then realize that it will have deadly effects on your files.

The big difference between the SIO4/DMA and most other cards only shows up in the 'FIFO input' mode of operation. When operating with the FIFO turned on, *the SIO4/DMA provides a single input channel for all incoming characters!* This shows clearly how well the DUAL designer anticipated the needs of a multitasking, multiuser operating system such as UNIX, in which a single "device driv-

```
; SIO4.ASM--CP/M-8Ø BIOS routines for Dual Systems Corp SIO4-DMA
; with line Ø as the console.
; This version uses FIFO input, direct output to USART.
; Other ports on board can NOT be used for input with this simple
; interface, as there is but a single multiplexed input channel.
; This version by Ian Darwin, based on some examples in the
; Dual systems manual for the Sio4/DMA board.
; Dual Systems, 253Ø San Pablo Ave, Berkeley CA 947Ø2.

sio4base equ     2ØH            ;where their UNIX driver puts it.
DADR     equ     sio4base+Ø     ;address for DMA-addr
rctl     equ     sio4base+3     ;receiver control.
DBYC     equ     sio4base+4     ;Address for DMA byte count
STRTSTOP equ     sio4base+6     ;MSB toggles ^S/^Q onboard; LSB halts output
DCTL     equ     sio4base+7     ;address for DMA control
UDATA    equ     sio4base+8     ;USART data
USTAT    equ     sio4base+Ø9    ;USART status.
EPMØ     equ     sio4base+1Ø    ;USART mode address
EPCM     equ     sio4base+11    ;USART command address.
IPST     equ     sio4base+12    ;input fifo status
IPDT     equ     sio4base+13    ;input fifo data.
LNSEL    equ     sio4base+15    ;address for line selection

strt     equ     1              ;bit to start transfer in dtcl
rcven    equ     2              ;bit to enable receiver in rctl.
done     equ     4Øh            ;bit to indicate completion in dtcl
chrdy    equ     4Øh            ;bit in ipst to indicate char in fifo.
TxRdy    equ     1              ;bit in ustat says prev char still outbound

ConLin   equ     Ø              ;line zero on board is console.

;        C O N S O L E   I N I T I A L I Z A T I O N
coninit:                        ;initialize line Ø for console.
         mvi     a,ConLin       ;line zero
         out     lnsel          ;select line zero.
         mvi     a,rcven        ;enable fifo for
         out     rctl           ;this line.
         mvi     b,Ødh          ;code for 96ØØ baud
         in      epcm           ;ensure mode reg one
         in      epmØ           ;is next, and read it.
         mvi     a,Øfh
         ana     b              ;get baud rate only
         mov     b,a            ;save baud rate
         in      epmØ           ;read mode register 2
         ani     ØfØh           ;remove old baud rate
         ora     b              ;OR in new baud rate
         mov     b,a            ;save in B
         in      epmØ           ;read mode reg 1 again
         mov     a,b            ;get new data for mode
         out     epmØ           ;register 2, and write to it.
         ret
;
;        C O N S O L E   O U T P U T
;        call with character in C.
;        sends output directly to usart.
conout:
         mvi     a,ConLin       ;get line number
         out     lnsel          ;and select.
conota:  in      ustat          ;get current status of usart.
         ani     TxRdy          ;is previous char finished outputting?
         jz      conota         ;no, spin a while.
         mov     a,c            ;yes, get this output char into `A'.
         out     udata          ;and send it!
         ret

;        C O N S O L E   S T A T U S
;        return a=ØØ if no char ready, a=FF if char ready.
const:
;        do not select line here as we are reading from FIFO.
         in      ipst           ;see if char ready.
         ani     chrdy
         jnz     conavail
         mvi     a,Ø
         ret
conavail: mvi    a,255          ;say char ready
         ret

;        C O N S O L E   I N P U T
;        gets a char (waiting if need be); returns it in A
conin:
;        do not select line here since we are reading from FIFO.
conina:  in      ipst           ;get status
         ani     chrdy          ;is char ready yet?
         jz      conina         ;no, try again (and again...)
         in      ipdt           ;read the char into A
         ani     7fh            ;mask off parity bit (sorry...)
         ret
```

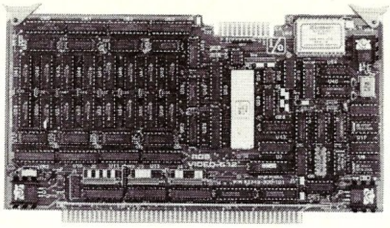
Uncompromising Additions to your S-100/IEEE-696 BUS



DUAL GPIB-488 INTERFACE BOARD

A Stand-Alone, Independently Controlled Dual Channel IEEE-488 I/O Processor. Interface Activity Modes for Controller-in-Charge, Controller Assigned or Terminal Bus Slave, and all Interface Functions are handled transparent to Host System CPU through an on-board CPU and DMA controller. User Friendly operation.

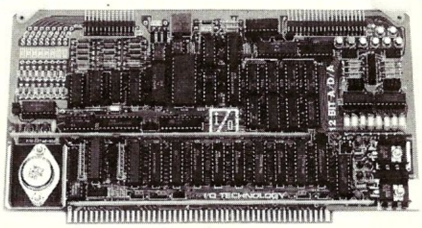
A&T, P/N 52748-800-102 $795.00



RGB VIDEO-512 COLOR GRAPHICS BOARD

Programmable resolution up to 512 x 512 pixels with 4 local video planes and on-board graphics processor. Color mapper allows 16 colors from a palette of 4096. Light pen input. Plus more ...

A&T, P/N 52748-300-101 $985.00



12-BIT A-D-A CONVERTER BOARD

8 Channel A-D: 12 microsec. Conversion, 50KHz Sample Rate, Programmable Gains, Offset and Diff./Single Modes. **8 Channel D-A:** 2 microsec. Settling, Bipolar V or Unipolar I Output. Programmable Reference levels, Dual-Ported Channel Refresh RAM. **16/8-Bit Data Transfers** via I/O or Memory Mapped

A&T, P/N 52748-900-101 $455.00

MULTI-FUNCTION I/O BOARD

A&T, P/N 52748-100-101 $325.00

SMART PROTOTYPING KIT

P/N 52748-400 $60.00

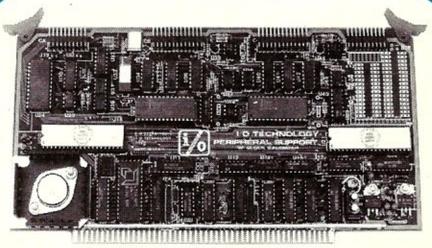
128K x 8/64K x 16 CMOS STATIC RAM MODULE

A&T, P/N 52748-650-128 $825.00

California residents add 6½% sales tax.
U.S. Domestic Price, FOB Factory.
Prices and Specifications subject to change without notice.

I/O TECHNOLOGY
POST OFFICE BOX 2119
CANYON COUNTRY
CA 91351
(805) 252-7666

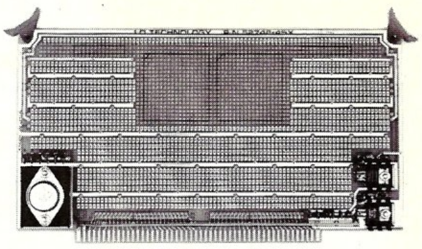
CIRCLE 11 ON READER SERVICE CARD



PERIPHERAL SUPPORT BOARD

Two Serial SYNC/ASYNC Ports with RS-232, TTL or Current Loop Outputs, three 8-Bit Parallel Ports, three Timers, Real Time Clock/Calendar and Response Programmable Interrupt Controller. Small Proto Area with +5 and ±12v.

A&T, P/N 52748-150-101 $395.00



MULTI-PURPOSE PROTOTYPING KIT

Industrial Quality with Plated-Thru holes for Wire-Wrap or Solder projects. Complete with +5, ±12v Regulators, Bus Bar, Filter Capacitors, and Manual.

A&T, P/N 52748-450 $72.00

Gifford has a lock on multiuser CP/M® 8-16.

It's 11:00 P.M. Do you know where your files are?

It's great when multiple local and off site users can run any 8- or 16-bit CP/M or MP/M™ program. It's even better when they can share expensive resources like printers, hard disks, and tape drives. Best of all is when they can share your most precious resource—data. Gifford has been delivering systems with all these features for over two years.

But sometimes data is sensitive. How do you keep people from taking more than their fair share?

Gifford adds a new dimension to CP/M security.

With our new security features, you can control what resources and data are shared.

Gifford's proprietary security enhancements include user login with encrypted passwords, control over access rights of modem users, secure electronic mail, and the ability to restrict users to specified terminals, programs, and directory areas. Plus, an audit log utility that keeps a permanent record of system activity. And you also get all the standard security features of Digital Research's MP/M-86.™

You select the level of security needed to get the best balance between file sharing and file safety.

Unleash productivity with Gifford's Virtual Terminals.

With our Virtual Terminals, each terminal on your system can monitor up to four different programs running concurrently. And at the touch of a key you can switch screens instantly from one program to another.

You could look up an address in dBASE II,™ jump over to SuperCalc™ to make some projections, then switch instantly to WordStar® to use this information to update a letter. If you forget what's on a screen, just touch a key to refresh your memory. You won't need to go through the distracting process of loading and unloading programs.

And since your Virtual Terminal can run any 8- or 16-bit CP/M or MP/M program, you can choose the best programs for your job from the biggest software library in the world. It's easier than 1, 2, 3!

The Gifford Security Blanket: Total Solutions.

Gifford delivers solutions. This means professional pre-sale consultation, expert system integration with 200 hour system burn-in, complete training, and full after sale support.

For example, our three user CompuPro® based system with a 21-megabyte hard disk costs just $9,990, and can be easily expanded for $500 per user. This includes MP/M 8-16, SuperCalc, and dBASE II.

Other Gifford solutions include systems with hard disks that range from 5 to 300 megabytes, 4 and 9 track tape backup, printers, plotters, and modems. Single- and multiuser 8086, 68000, and Z-80 based systems are available for immediate delivery, with 80286 and 16032 systems on the way.

Two year warranty protection.

In the unlikely event that you encounter a hardware related problem, we'll replace any defective S-100 part within 24 hours FREE for two full years. But chances are, it can be solved on the Gifford service hotline or diagnosed via modem. All at no cost to you.

Lock in on Gifford Security today.

If total support, training, on site service, obsolescence-proof upgradeable S-100 bus architecture, and complete system security sound appealing, cut the coupon or give us a call. We'll send you a free brochure that tells the whole story. Once you get it you'll see why Gifford has a lock on multiuser CP/M 8-16.

Gifford Computer Systems is a Full Service CompuPro® Systems Center.

The powerful Gifford System 321 shown with optional GCS-80 Virtual Terminals.

GIFFORD COMPUTER SYSTEMS
2446 Verna Court, San Leandro, CA 94577
(415) 895-0798 A division of G&G Engineering
I'D LIKE THE WHOLE STORY.
Please send me your brochure.
Name________ Title________
Organization________ M/S________
Address________
City________ State________ Zip________
Phone________
☐ Please have a representative call me. M-S

GIFFORD COMPUTER SYSTEMS ☐ San Leandro, CA (415) 895-0798 ☐ Los Angeles, CA (213) 477-3921 ☐ Houston, TX (713) 680-1944 Amherst, NY (716) 833-4758 ☐ Telex: 704521

CIRCLE 181 ON READER SERVICE CARD

Dual Systems

Continued from page 104

er" controls all occurrences of a particular kind of device (such as all serial lines, all 8″ disks, all 5″ Winchesters, and so on). Each device in such a group is usually numbered; for example, UNIX might use the names **ttyd0, ttyd1, ttyd2** and so on, for the serial lines. If one of the lines is a serial printer, well, to UNIX's device drivers it's still just a TTY line. This all turns out very well for UNIX, but doesn't work quite as neatly in a CP/M BIOS, where each device is a special case, and where all devices are assumed to be independent of each other. So the designers allow you either to use this feature or to access the USARTs directly. And what's more, this feature can be enabled on a line-by-line basis! So you could be doing FIFO input from lines 0 and 2, and polled input from lines 1 and 3, or any such combination. That, to me, is versatility.

This explains why the board is priced above conventional serial boards. This board was not intended to compete with nonbuffered serial I/O boards, but was designed as part of Dual's System/83 UNIX product. UNIX began life on DEC hardware, and the SIO4/DMA has a strong family resemblence to the preferred serial I/O board on DEC equipment, the DH-11.

When operating with the FIFO on, the SIO4/DMA provides a single input channel for all incoming characters!

The only problem I've had with the board is that it hangs my system when I erroneously refer to a nonconnected terminal. Handshaking is implemented using DTR on pin 20 (changeable), and the board will wait until the terminal is ready. The manual describes a way around this for applications in which full handshaking is not used.

Documentation

The manual for the board comes in two parts. Part 1 is written by Dual; it provides an overview, details the switch settings and options, and provides programming information. Part 2 is an extract from the Motorola document on the MC2661 USART; this section gives bit-twiddling details. I found the "programming" section of the Dual manual a little short of examples (only three on 8080 examples). A few details were not explicit; i.e., the baud rate and other parameters to which the on-board processor initializes the USARTs. However, a programmer accustomed to writing BIOS code should be able to handle it. And that's who this manual is written for.

The CP/M-80 BIOS

In the FIFO input mode, the Dual SIO4/DMA gives you a single input channel of (possibly typed-ahead) characters. The board returns the line number along with the character, and the BIOS has to sort them out. This version of my BIOS uses the FIFO for console

Dual Systems

Continued from page 107

input (because I like type-ahead), but doesn't have to sort out what character is from which line, because I've only enabled the FIFO from line zero.

To use multiple FIFO input lines, you could implement internal queuing within the BIOS, assigning a character to an input queue based upon its line number. A simpler approach, and one more in keeping with the CP/M BIOS way of doing things, would be to not enable the FIFO and access the USARTs directly. Then input is like any conventional multiline I/O card. And conventional coding techniques could be used within the CP/M BIOS.

Listing 1 contains the SIO4/DMA access routines for simple, console-only access using the FIFO for input, and sending one character at a time to the USART for output. The code for reading characters from the USART would not be much different; the input ports used by ConSt and ConIn become USTAT and UDATA, and the value of ChRdy has to be changed. The overall logic does not change much. The code for the LST: device is similar but contains only the output routines.

To use the code in Listing 1, replace the "Interfacer" access modules in the BIOS with this code. Add a call to "coninit" in the cold boot code (label CBOOT:). Then regenerate the BIOS and system disk using a working CP/M-80 system and the Godbout documentation. If, on a System/83, you want to keep the hard disk for UNIX use only, be sure to disable the code in the BIOS for hard disk access! Enable the floppy controller's BOOT EPROM (Dual UNIX runs with it disabled). Take out the 68000 CPU and put it in a *safe* place. Install a 8080 or Z80 CPU card with 24-bit addressing and boot the system. (The CPU must have 24-bit addressing, since this is used on all the other cards). You should now be running CP/M from the floppy disk.

I've discussed the SIO4/DMA serial interface card, and shown one way—the hard(ware) way—to run CP/M with Dual Systems' System/83 computer. There are other ways—but that's a topic for another article.

Note: If you're not experienced with S-100 hardware and CP/M BIOS software, then please *don't tear apart a Dual Systems' UNIX system yourself. Get your dealer to do the integration and show you how to make the changes. The instructions in this article are intended for the experienced "techie" only.*

At a glance

Name: SIO4/DMA intelligent four-port serial input/output board

Use: Interfacing the IEEE-696/S-100 bus to peripheral devices (terminals, modems, printers, etc.) over serial lines

Price: $650 assembled and tested, with on-board software

LSI used: 8085 processor, 4 MC2661B USARTS

System: IEEE-696 S-100

Features: Oriented to multiuser systems; optional FIFO buffering of inputs, optional output requests via DMA, optional on-board handling of XON/XOFF.

Dual Systems Corp.
2530 San Pablo Ave.
Berkeley CA 94702
(415) 549-3854

CIRCLE 303 ON READER SERVICE CARD

Magic-Bind Supersedes MailMerge

All the power of MailMerge with none of the complexity

by Roger C. Schlobin

Did you ever expect to hear the words: "MailMerge superceded"? Good old MailMerge, the indispensable add-on to WordStar that allows you to merge-print, make multiple copies, use command files, boilerplate, insert files, read variables, and all the other good features that have saved users so much tedium for a number of years. Yet, just as MicroPro's SpellStar has been challenged by superior spelling checkers such as The Word Plus, so too has MailMerge fallen under the shadow of MagicBind.

MagicBind offers all the features of MailMerge and much more—all at the same price! The first blessing MagicBind brings is a solution to that bitter insecurity that has always plagued WordStar lovers. We all know that dread moment: as you are raving to a wide-eyed audience about the joys of WordStar and all its support programs, a sardonic voice invariably carries from the back of the room: "What about proportional spacing?" Suddenly the WordStar lover, now in a cold sweat, is subject to all the smug vulgarities of Perfect Writer, SpellBinder, and Final Word users. Never mind the fact that these word processors accomplish proportional spacing by spreading out the spaces between the words and thus haven't a real typeset quality. They had it and we didn't!

Now, through both MagicBind and its more limited cousin, MagicPrint, the WordStar user can have *true* proportional spacing with either a justified or an unjustified right margin. The increments are spread throughout both the spaces and the words, and true centering is finally a reality (no more of one letter sticking out when two words of different lines have only a one-character difference). This comes at an important time: many users are now trying to decide whether or not to update their old MailMerges and WordStars to V3.3, and I expect that many are considering the MicroPro line or the addition of MailMerge. After reading this article, you may find that MagicBind is not only a more useful utility, but a real bargain as well.

Before getting into the many joys of MagicBind, let's take a quick look at the usual requirements. To run MagicBind, you need at least 48K of RAM, WordStar (or another standard ASCII text editor like T/Maker, WordMaster, P/Mate, Magic Wand, or Electric Pen-

The Image Solution

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

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

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

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

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

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

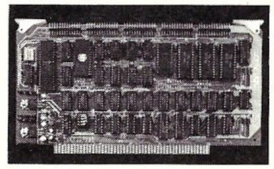
DIGITAL GRAPHIC SYSTEMS, INC.

What you see is what you get.

CIRCLE 33 ON READER SERVICE CARD

Extended Processing S100 Boards

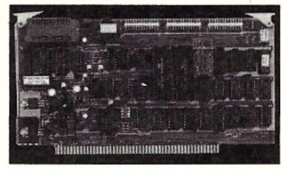
POWER I/O



High performance S100/IEEE-696 smart slave computer with 64K RAM, 3 serial ports, 1 centronic port, comprehensive 4K operating system in EPROM and 1 timer. Host access is through a high speed parallel I/O port. Accepts 256K RAMS when available. Optional ADD-ON board doubles I/O and RAM. Standard software and hardware supports 6 serial ports, 2 parallel ports and 512K of RAM. Entire board is software programmable including all I/O buffer sizes.

POWER I/O w/64K and 3S+P:	$375.00
64K RAM ADD-ON board:	$175.00
3S+P ADD-ON board:	$195.00
64K and 3S+P ADD-ON board:	$295.00

BURNER I/O II



Multifunction S100/IEEE-696 board. Complete EPROM programmer handles 5 volt EPROMS: 2508, 2758, 2516, 2716, 2532, 2732, 2732A, 2564, 2764, 27128, 27256. Fully I/O mapped. EPROM selected totally with software. No switches or program modules. Menu driven software supplied in 4K EPROM. 2 independent serial ports with baud rate to 19,200. 1 centronic type parallel port. Memory management for address lines A16-A23.

Option A: Full board	$355.00
Option B: Programmer	$220.00
Option C: I/O (2S+P)	$220.00
Option D: Programmer+I/O	$330.00
Option E: Memory management	$110.00
Memory management for B or C:	$ 25.00

All E.P. boards are built with quality components and are fully assembled and tested. Full documentation including schematics and source code listings.

ep Extended Processing 3861 Woodcreek Lane, San Jose, Ca, 95117 (408) 249-8248

CIRCLE 142 ON READER SERVICE CARD

MicroScript™ $99

State of the Art Text Formatter

- extensible generic markup
- multiline headers, footers, and footnotes
- automatic widow and orphan suppression
- automatic section numbering
- automatic bullet, number, and definition lists
- floating and inline figures
- left, center, right, or justify text alignment
- left and right indention with delay and duration
- bold, underscore, and proportional spacing
- macros and symbols
- multiple input files of unlimited size
- direct printer control
- format preview
- IDS, Qume, Diablo, NEC, C.ITOH, and all TTY
- table of contents, index
- multiple columns
- conditional processing

MicroEd™ $49

Customizable Full Screen Editor

- full cursor control by character, word, or line
- position to top or bottom of window or file
- scroll by line, half window, or full window
- global or selective find and replace
- delete by character, word, line, or block
- read external files into current file
- copy, move, and write blocks of text
- insert, overlay, or wordwrap text
- all cursor addressable VDTs

Postpaid within U.S. & Canada, outside U.S. add $5, CA residents add 6½%.
8″ SS/SD CP/M-86*, 5.25″ SS/DD PC-DOS.

VISA

MicroType™
6531 Crown Blvd., Suite 3A, San Jose, CA 95120
(408) 997-5026

Master Card

* CP/M-86 is a trademark of Digital Research, PC-DOS is a trademark of IBM Corporation.

CIRCLE 157 ON READER SERVICE CARD

MagicBind

Continued from page 111

cil), CP/M-80 (a CP/M-86 version is in the works) or SB-80, and a Diablo 630/1650 compatible printer or a NEC Spinwriter. If your equipment satisfies these requirements, you can produce copy that has over 20% more text per page, and many people will not be able to distinguish it from typesetting, especially with a Diablo metal wheel (a small tip: Diablo plastic wheels work well, too).

MagicBind is, as you have probably surmised, a print formatter, among many other things. The new version for WordStar (1.11/W) will run from WordStar's "R" (Run another Program) command. This is a major improvement over the earlier version, which would only run out of CP/M. However, it will not allow you to edit one file and print another at the same time. Given MagicBind's many features, this isn't a serious limitation at all and can easily be overcome with a software or hardware spooler.

What, then, can MagicBind do, in addition to true proportional spacing and centering? For starters, it can do everything MailMerge can: it can respond

MagicBind has all the features of MailMerge and much more—for the same price!

to all of WordStar's embedded print commands except that MagicBind has a more sensitive method of "kerning" or changing pitch, respond to all the standard dot commands (line height, etc.), and do it all easier and much faster. MagicBind can read data files created in WordStar's Document mode, not just the Non document mode, and it skips empty fields in data files automatically. MagicBind's merge commands are also much easier: instead of the ".RV" line and ampersand delimiters, simply identify the data file (created the usual way) and pop in the number of the item of each field wherever it appears. Control-PA, a colon, and the number of the item. That's all there is to it, and there are none of MailMerge's annoying pauses while printing.

MagicBind can also do columns (speaking of another WordStar insecurity), multiple-line headers and "footers," automatic chapter and paragraph numbering, print-time record selection while merging, print-time insertion of formatting commands, and automatic handling of widow/orphan lines. Most of this is accomplished by using double dot (..)commands, allowing a file to have both MailMerge and MagicBind commands in it. As if all this weren't enough, MagicBind also has a system for automatically producing up to 15 footnotes per page with a maximum of 500 characters per note. Since Footnote (Digital Marketing) at $125 is currently the only program that does this for WordStar, and since it does have limitations that aren't in MagicBind, many users may find this feature, by itself, enough justification to buy the program.

Believe it or not, there's even more! Two very nifty programs are included

MagicBind

Continued from page 113

with MagicBind: VERIFY and LABEL. VERIFY is used to check the accuracy of data files: no more of the hit-and-miss disasters discovered only at print time. LABEL is a real treasure. While installing it, the user specifies the type of labels most frequently used: four across, two across, whatever. This becomes the default (although there are other choices at print time). No more of the excessive dot commands that MailMerge demands for label formatting—just a simple control file for the merge with the data file, using the combination of Control-PA, a colon, and the item number.

It's rare to find a program as versatile and as easy to use as MagicBind. The manual is well written: the program's over 60 commands are clearly explained, and the menus are more than helpful. There are instructions for program modifications, and installation is a breeze. There is even room to get very creative with all the program's features.

MagicPrint is a smaller version of MagicBind, with all the print features, but none of the merge capabilities. Considering the $55 difference, you'd be better off buying MagicBind instead of MagicPrint, unless you don't need the merge program.

VERIFY checks data files—no more hit-and-miss disasters at print time!

Ben O. Jone, the creator of these programs, has also produced an indexing program for MagicBind and MagicPrint, called MagicIndex (necessary since both change the pagination, rendering such programs as StarIndex and Documate/Plus inaccurate in their pagination modes). When bought with MagicBind, MagicIndex is a real steal for only an extra $45 (as opposed to StarIndex at $195). Computer EdiType is also working on a communications' program, MagicLink, and on an editor, MagicEditor. If these programs are as well conceived and as well coded as MagicBind, they, too, will be splendid!

WordStar, MailMerge, and SpellStar are trademarks of **MicroPro International,** *33 San Pablo Ave., San Rafael, CA 94903; (415) 499-1200.*

Footnote is marketed by **Digital Marketing,** *2363 Boulevard Circle, Walnut Creek, CA 94595; (415) 938-2880.*

MagicBind, MagicPrint, MagicIndex, MagicLink (forthcoming), and MagicEditor (forthcoming) are registered trademarks of Computer EdiType Systems. MagicBind is $250; MagicPrint, $195; and MagicIndex, $150. The MagicBind-MagicIndex package is $295. All these programs are available through local dealers or **Computer EdiType Systems,** *509 Cathedral Parkway #10A, New York, NY 10025. (212) 222-8148*

CIRCLE 304 ON READER SERVICE CARD

Roger C. Schlobin, 802 Calumet Rd., Chesterton, IN 46304

SAL/80 from Protools

Keep your code under control and your programming understandable with SAL/80

by Mike Barker

On the one hand, structured languages often produce code that isn't too efficient in terms of space and time. On the other hand, assembly languages can be a haven for unstructured, unreadable, but efficient code. If you wanted to provide methods for writing structured assembly code and spent enough time with the macro capabilities of your favorite assembler, you might produce something like SAL/80. Of course, with SAL/80, you can just start writing the code for your program.

The macro library

SAL/80 version 2.1 is a macro library containing a variety of useful macros, including macros for selection (if-else, case), repetition (while, repeat, loop, do, for-until), flow-of-control (call, conditional calls), and various helpful utilities. It is intended for use with the MAC or RMAC assemblers from Digital Research, and requires a 64K 8080/Z80 system running CP/M 2.2.

The SAL/80 macro libraries contain a good collection of assembly language macros. The manual does not provide the easiest introduction to their use, but with the original source listing provided in the manual and some experimentation, they are usable.

The utility macros in the SAL/80 libraries are, in part, adapted from those in the MAC library on Volume 24 of the CP/M Users' Group, and from other sources. They provide several varieties of console interaction, including console status, character or buffer input/output, binary converted to ASCII hex for output, and ASCII hex input converted to binary. Binary/hex and binary/decimal conversions are also included.

Utility macros for register/stack and register/memory manipulations, double-register subtracts and compares, and 16×8-bit multiply and divide routines act as extensions to the assembly language. Block moves, string search within a block, lower-to-upper case conversion, and string comparison macros complete the set of utilities.

Although the utility macros are a kind of bonus in SAL/80, since the package focuses on structured assembly language programming, I would have liked a more complete set of utility macros, including interfacing to all the CP/M functions and a fuller set of string manipulations. The set of utility macros that is provided does not seem to have been as carefully designed and thought out as the set of structured lan-

SAL/80

Continued from page 116

guage macros. For example, the lack of file-I/O macros forces you to write new code every time, or to write your own macros, or to use macros from yet another source.

Documentation

The SAL/80 manual is (mostly) a hacker's explanation of an approach to programming and an example of that approach. The example uses the SAL/80 macros and assembly language. However, the description focuses on the approach, not the macros. I enjoyed this description as a working example of an approach to programming by one hacker—something that is often described abstractly, but rarely with the depth of concrete detail given here.

The manual includes a section describing the macros. This section is modeled on a syntactic description of a programming language that makes it difficult to see the macros for the syntax. I would like to see this section moved to an appendix (for those who like formal syntax descriptions). It could be replaced with a simple list or index of the macros, followed by single-page descriptions of the macros, like those often used in language reference manuals. Each page could describe a macro structure and its parameters. This type of description is present, but is difficult to see in the current section.

Why use SAL/80?

Programmer's tools like SAL/80 are measured by a simple question, "Why should I use it?"

SAL/80 can help assembly language programmers by providing code structures that encourage the use of structured programming techniques. Assembly languages can certainly be made friendlier by judicious use of macro libraries (some PDP-11 code seems to consist almost entirely of macros, using the well-developed libraries available for those systems).

If you have the MAC assembler (or RMAC), or if you have to write extremely fast or compact code, SAL/80 may be just the extra advantage you need. Or, if you want to learn about macros, the large collection in SAL/80 could be a useful introduction. Macros are a powerful, exciting feature in any language, and SAL/80 shows how helpful they can be.

Before you buy, decide what tools you need for the job you want to do. If you decide that MAC or RMAC is what you need, then try SAL/80. It can help to keep your code under control, and your programming understandable. However, if you're trying to pick a structured language for programming, I would recommend a C compiler.

A lengthy "single system license agreement & support contract" of relatively normal format for microcomputers is included. The license does not indicate whether or not a royalty is required if you sell code produced using SAL/80. If you intend to use SAL/80 for serious production of salable software, you will need to negotiate a contract aimed at that production work.

Protools, a division of Newberry Microsystems, is registered with the U.S. Patent Office.

SAL/80 is a trademark of **Newberry Microsystems** and is available from: **Protools,** *24225 Summerhill Ave., Los Altos, CA 94022; (415) 948-8007.*

Price: $59 for 8" SD disk, 235-page manual, and one free update.

CIRCLE 305 ON READER SERVICE CARD

Mike Barker, 1612 Berlin Rd., Cherry Hill, NJ 08003

C COMPILER

- FULL C
- UNIX* Ver. 7 COMPATABILITY
- NO ROYALTIES ON GENERATED CODE
- GENERATED CODE IS REENTRANT
- C AND ASSEMBLY SOURCE MAY BE INTERMIXED
- UPGRADES & SUPPORT FOR 1 YEAR

C SOURCE AVAILABLE FOR $2500⁰⁰

HOST	6809 TARGET	PDP-11*/LSI-11* TARGET	8080/(Z80) TARGET	8088/8086 TARGET
FLEX*/UNIFLEX* OS-9*	$200.00 WITHOUT FLOAT $350.00 WITH FLOAT	500.00	500.00	500.00
RT-11*/RSX-11* PDP-11*	500.00	200.00 WITHOUT FLOAT 350.00 WITH FLOAT	500.00	500.00
CP/M* 8080/(Z80)	500.00	500.00	200.00 WITHOUT FLOAT 350.00 WITH FLOAT	500.00
PCDOS*/CP/M86* 8088/8086	500.00	500.00	500.00	200.00 WITHOUT FLOAT 350.00 WITH FLOAT

*PCDOS is a trademark of IBM Corp. MSDOS is a trademark of MICROSOFT. UNIX is a trademark of BELL LABS. RT-11/RSX-11/PDP-11 is a trademark of digital Equipment Corporation. FLEX/UNIFLEX is a trademark of Technical Systems consultants. CP/M and CP/M86 are trademarks of Digital Research. OS-9 is a trademark of Microware & Motorola.

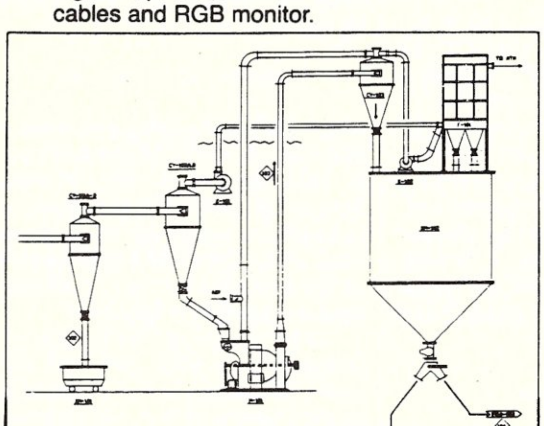
408-275-1659

TELECON SYSTEMS
1155 Meridian Avenue, Suite 218
San Jose, California 95125

CIRCLE 269 ON READER SERVICE CARD

YOUR S-100 PLUS OUR M-1000/S = CAD

M-1000/S CAD subsystem for S-100 includes 512 × 480 color graphics display, digitizer, plotter, all interface boards and cables and RGB monitor.



APPLICATIONS

P & ID
Process flow diagrams
Vessel drawings
Architectural drawings
Facilities planning
Plot plan and layout
Schematics
PC Board Layout

Two-dimensional drafting and design software from Autodesk Inc. is fully installed, configured and supported by MOMS.

MOMS

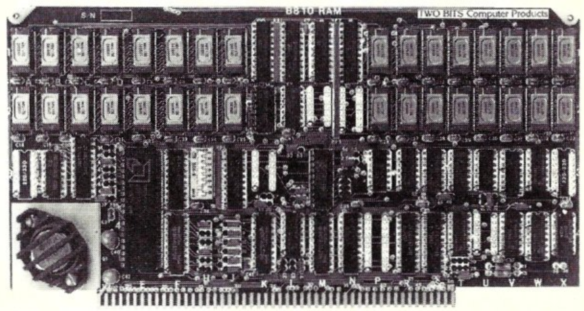
MOMS COMPUTING
Dealer inquiries invited.
Building 1055 Fort Cronkhite, Sausalito, California 94965 415 331-2043

CIRCLE 99 ON READER SERVICE CARD

OLD! EXPERIENCED AVAILABLE

The B810/A

256K Dynamic Memory System



FEATURES

- 8/16-Bit Operation
- 24-Bit Extended Addressing
- Port Selection
- FAST! 6MHz & Up
- Memory Mapping
- 5 48K User Banks/Board
- Virtual Memory & Semiconductor Disk Applications

The B810/A RAM is the most cost-effective solution for S-100 systems requiring reliability and speed at remarkably low cost. Qualified in a wide range of commercial, industrial and military applications, the B810/A has a 3-year history of field use in some of the harshest environments imaginable without a single failure. Optimized for Z80/Z8000* CPUs, the B810/A is, quite siimply, the fastest and most reliable dynamic memory available.

Multi-layer board with full ground and power planes assures noiseless, glitch-free operation. No buried signal traces.

Quality: Screw machine contact sockets. Workmanship per MIL-STD-454E. Quality that meets or exceeds MIL-I-45208. Packaging per MIL-81795, Type II. 100% AQL with dynamic testing in environmental chambers at 80°C.

Protection: One Year Warranty (three years for Validated Boards). Upgrades for revisions made during warranty protects the user's investment.

Compatibility: IEEE-696 standard, the B810/A operates with any Z80 S-100 CPU and supports Oasis*, MP/M*, CP/M 3.0*, TurboDos*, Cromix*, Unix* and other operating systems.

Support: Extensive applications engineering support is available for systems integrators and OEMs.

Pedigree: The B810/A is not a new product. Introduced in 1981, it has been field tested for three years — the end product of experience gained at over 200 beta test sites.

*Oasis is a trademark of Phase One; CP/M and MP/M are trademarks of Digital Research; TurboDos is a trademark of Software 2000; Cromix is a trademark of Cromemco; Unix is a trademark of Bell Labs; Z80 and Z8000 are trademarks of Zilog.

$895

Recommended retail
A&T with One Year Warranty
Technical Manual $25
OEM and dealer inquiries invited.
Further information on request.

TWO BITS
Computer
Products

(805) 489-6733

235 Tally Ho Road, Arroyo Grande, CA 93420

CIRCLE 74 ON READER SERVICE CARD

Mass Renaming by Filetype

Rename files of one type to a different type using a single command

by George M. Gallen

One major problem with the CP/M "REN" command is that you can't REName files of one type to a different type using a single command. For example, if you wanted to REName all of your .ASM files to be .LIB files, you would have to go and REName them one at a time. You couldn't just type a command like:

```
REN *.LIB=*.ASM.
```

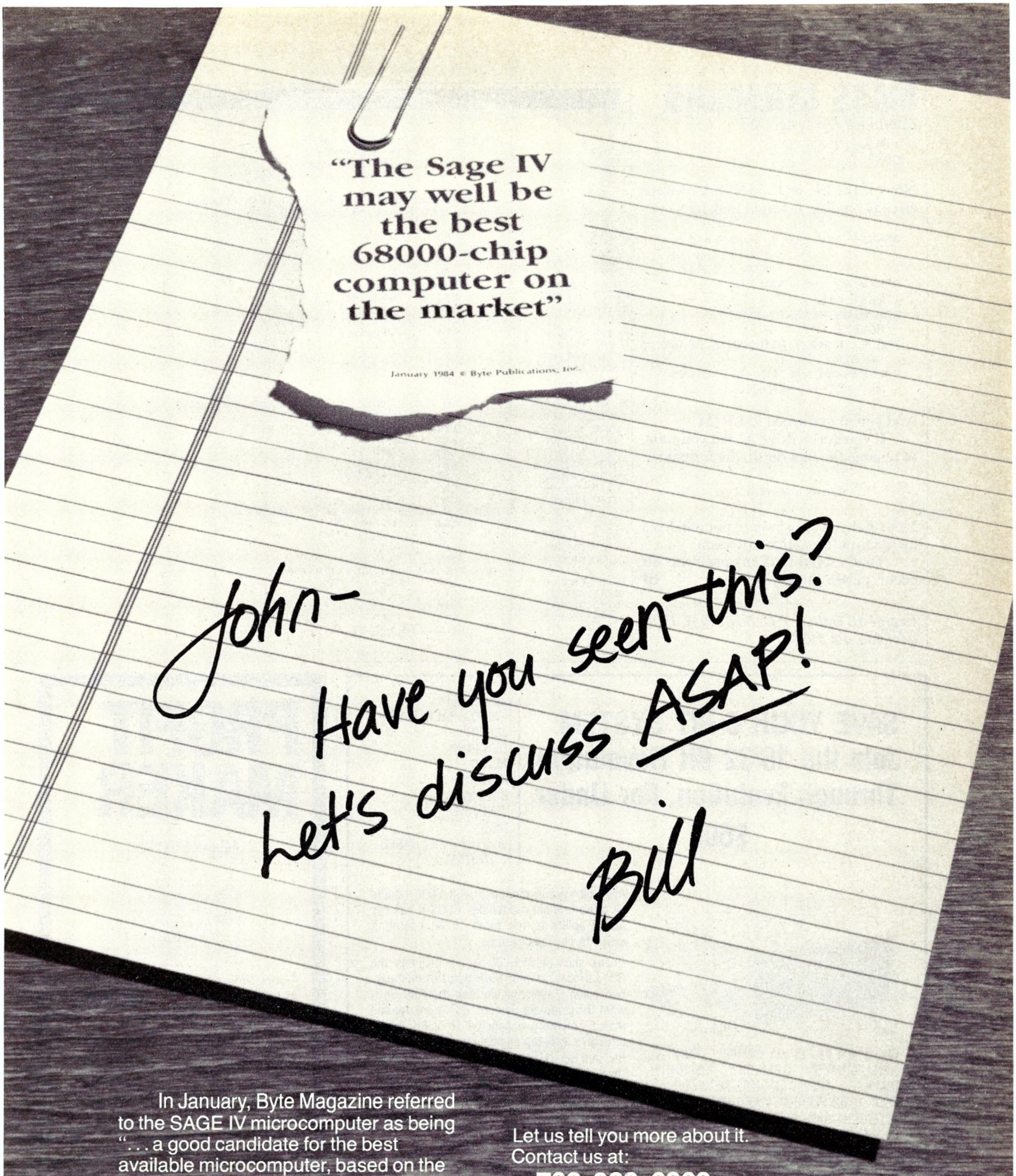
In essence this is what RENEXT (Rename extension) does: it will rename all files having a given filetype to files having the same names but a different filetype. Since RENEXT is not a CP/M function built into the operating system, the program must exist in every user area by which it is to be used. Once the program has been assembled (using ASM.COM) and turned into a .COM file (using LOAD.COM or DDT.COM), it can then be executed. To use ED.COM, ASM.COM, DDT.COM, and LOAD.COM, see your CP/M manual. The RENEXT format is:

```
RENEXT [D:] Sext Dext
```

where [D:] is the drive you want to rename on (A: to P:). If you don't put one in, then the default drive is assumed. Sext is the SOURCE extension or the extension to search for when renaming. Dext is the DESTINATION extension or the extension to rename to once the source extension is found.

The program will display the filename as it was, and then its new name. If it comes across a filename that already exists, it *will not* rename that file. What it will do is stop execution and beep to let you know that this file already exists under the new extension. A sample session would look like this:

```
RENEXT ASM LIB (rename all
               .ASM files to
               .LIB files)
RENAMING TEST.ASM TO
  TEST.LIB
RENAMING RENEXT.ASM TO
  RENEXT.LIB
RENAMING ROUTINES.ASM TO
```



"The Sage IV may well be the best 68000-chip computer on the market"
January 1984 © Byte Publications, Inc.
John-
Have you seen this?
Let's discuss ASAP!
Bill

MASS RENAMING

Continued from page 120

```
  ROUTINES.LIB
DONE...
```

Now, if the file ROUTINES.LIB were already on disk, it would look like:

```
RENEXT ASM LIB (rename all
               .ASM files to
               .LIB files)
RENAMING TEST.ASM TO
  TEST.LIB
RENAMING RENEXT.ASM TO
  RENEXT.LIB
THE FILE ROUTINES.ASM ALREADY
  EXISTS.
PROCESSING HALTED. ALL FILES
  MAY NOT BE RENAMED
```

And you are returned to CP/M.

If you are on drive A: and you want to rename the files on B:, the format is:

```
RENEXT B:ASM LIB
```

Note that you only have to put the drive number on the source extension.

That's about it. If any problems do occur, please contact me.

George M. Gallen, P.O. Box 17622, Philadelphia, PA 19135

```
                 ;
                 ; PROGRAM BY     GEORGE M GALLEN
                 ;                NE SOFTWARE DEV
                 ;                P.O. BOX 17622
                 ;                PHILADELPHIA, PA. 19135
                 ;
                 ; ANY PROBLEMS OR QUESTIONS ABOUT THIS PROGRAM
                 ; CAN BE SENT TO THE ABOVE ADDRESS.
                 ;
0100                     ORG      0100H
                 START:
0100 210000              LXI      H,0
0103 39                  DAD      SP
0104 224803              SHLD     STACK
0107 216A03              LXI      H,STKTOP
010A F9                  SPHL
                 ; SAVE OLD STACK AND SET A NEW STACK
010B 3A5D00              LDA      FCB0+1
010E FE20                CPI      ' '
0110 CA1902              JZ       BF
0113 3A6D00              LDA      FCB0+17
0116 FE20                CPI      ' '
0118 CA1902              JZ       BF
                 ; CHECK TO SEE IF BOTH EXTENSIONS WERE SUPPLIED
011B 215D00              LXI      H,FCB0+1
011E 112802              LXI      D,FCB1+9
0121 010300              LXI      B,3
0124 CD3D03              CALL     LDIR
0127 215D00              LXI      H,FCB0+1
012A 114802              LXI      D,FCB3+9
012D 010300              LXI      B,3
0130 CD3D03              CALL     LDIR
                 ; MOVE THE SOURCE EXTENSION TO TWO SPOTS
0133 216D00              LXI      H,FCB0+17
0136 113802              LXI      D,FCB2+9
0139 010300              LXI      B,3
013C CD3D03              CALL     LDIR
                 ; MOVE THE DESTINATION EXTENSION.
013F 3A5C00              LDA      FCB0
0142 321F02              STA      FCB1
0145 323F02              STA      FCB3
0148 326202              STA      FCB5
                 ; GET THE DRIVE# AND STORE IT.
                 RNLOOP:
```

the best
VALUE anywhere

COMMUNICATIONS SOFTWARE PACKAGE
COMMX-PAC™

- COMMX Smart Terminal and File Transfer Auto Dial and Logon Files for Easy Linkup Protocols for Timeshare, MODEM 7, CRC16 FORTRAN version available for Mainframes
- Unattended "Midnite" Electronic Mail Subsystem
- CONSOLX Remote Operation of Your Computer
- Bulletin Board System with Database Access
- Database File Management System Included
- Utilities for Data Encription, Compression, and More
- CPU License $150   Detailed Manual $25

CIRCLE 119 ON READER SERVICE CARD

Contact Your Local Dealer or
Call or Write For Free Brochure

HAWKEYE GRAFIX

23914 Mobile, Canoga Park, Ca. 91307 • 213/348-7909 U.S.A

WORKMAN & ASSOCIATES
USER-FRIENDLY SOFTWARE

VISIT OUR BOOTH AT THE WEST COAST COMPUTER FAIRE

WRITE

The Writer's Really Incredible Text Editor lives up to its name! It's designed for creative and report writing and carefully protects your text. Includes many features missing from WordStar, such as sorted directory listings, fast scrolling, and trial printing to the screen. All editing commands are single-letter and easily changed. Detailed manual included. WRITE is $239.00.

WORKMAN & ASSOCIATES

112 Marion Avenue
Pasadena, CA 91106
(213) 796-4401

All US orders are postpaid. We ship from stock on many formats, including: 8", Apple, NorthStar, Osborne, KayPro, Monroe, Otrona, Epson QX-10, NEC/PC, DEC VT-180, TI Professional, Access, Morrow. Please request our new catalog. We welcome COD orders.

CIRCLE 278 ON READER SERVICE CARD

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

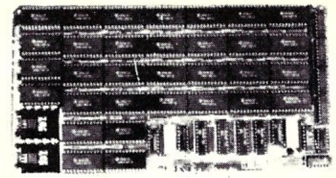
64K S100 STATIC RAM

$199.00 KIT

NEW!

LOW POWER!

RAM OR EPROM!



BLANK PC BOARD WITH DOCUMENTATION $55

SUPPORT ICs + CAPS $17.50

FULL SOCKET SET $14.50

FULLY SUPPORTS THE NEW IEEE 696 S100 STANDARD (AS PROPOSED)

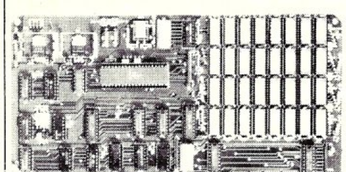
FOR 56K KIT $185

ASSEMBLED AND TESTED ADD $50

FEATURES:
- Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- Fully supports IEEE 696 24 BIT Extended Addressing.
- 64K draws only approximately 500 MA.
- 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- SUPPORTS PHANTOM (BOTH LOWER 32K AND ENTIRE BOARD).
- 2716 EPROMs may be installed in any of top 48K.
- Any of the top 8K (E000 H AND ABOVE) may be disabled to provide windows to eliminate any possible conflicts with your system monitor, disk controller, etc.
- Perfect for small systems since BOTH RAM and EPROM may co-exist on the same board.
- BOARD may be partially populated as 56K.

256K S-100 SOLID STATE DISK SIMULATOR!

WE CALL THIS BOARD THE "LIGHT-SPEED-100" BECAUSE IT OFFERS AN ASTOUNDING INCREASE IN YOUR COMPUTER'S PERFORMANCE WHEN COMPARED TO A MECHANICAL FLOPPY DISK DRIVE.



FEATURES:
- 256K on board, using + 5V 64K DRAMS.
- Uses new Intel 8203-1 LSI Memory Controller.
- Requires only 4 Dip Switch Selectable I/O Ports.
- Runs on 8080 or Z80 S100 machines.
- Up to 8 LS-100 boards can be run together for 2 Meg. of On Line Solid State Disk Storage.
- Provisions for Battery back-up.
- Software to mate the LS-100 to your CP/M* 2.2 DOS is supplied.
- The LS-100 provides an increase in speed of up to 7 to 10 times on Disk Intensive Software.
- Compare our price! You could pay up to 3 times as much for similar boards.

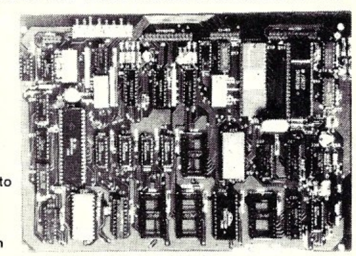
BLANK PCB (WITH CP/M* 2.2 PATCHES AND INSTALL PROGRAM ON DISKETTE) $69.95

$399.00 #LS-100 (FULL 256K KIT)

THE NEW ZRT-80

CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.



FEATURES:
- Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- RS232 at 16 BAUD Rates from 75 to 19,200.
- 24 x 80 standard format (60 Hz).
- Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
- Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- Composite or Split Video.
- Any polarity of video or sync.
- Inverse Video Capability.
- Small Size: 6.5 x 9 inches.
- Upper & lower case with descenders.
- 7 x 9 Character Matrix.
- Requires Par. ASCII keyboard.

BLANK PCB WITH 2716 CHAR. ROM, 2732 MON. ROM $59.95

SOURCE DISKETTE - ADD $10

SET OF 2 CRYSTALS - ADD $7.50

WITH 8 IN. SOURCE DISK! (CP/M COMPATIBLE)

$129.95 # ZRT-80 (COMPLETE KIT, 2K VIDEO RAM)

Digital Research Computers

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

TERMS: Add $3.00 postage. We pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Texas Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50 add 85¢ for insurance.

CIRCLE 64 ON READER SERVICE CARD

MASS RENAMING

Continued from page 122

```
014B 213F02          LXI     H,FCB3
014E 115C00          LXI     D,FCB0
0151 012300          LXI     B,35
0154 CD3D03          CALL    LDIR
0157 0E0F            MVI     C,OPEN
0159 115C00          LXI     D,FCB0
015C CD0500          CALL    BDOS
                ; FIND THE FIRST FILE OF THE TYPE "????????.[SEXT]"
015F FEFF            CPI     0FFH
0161 CA0C02          JZ      ENDIT
                ; IF ERROR=255 THEN THERE ARE NO MORE FILES OF THAT
                ; EXTENSION. PROGRAM IS FINISHED.
0164 215D00          LXI     H,FCB0+1
0167 112002          LXI     D,FCB1+1
016A 010800          LXI     B,8
016D CD3D03          CALL    LDIR
0170 215D00          LXI     H,FCB0+1
0173 113002          LXI     D,FCB2+1
0176 010800          LXI     B,8
0179 CD3D03          CALL    LDIR
                ; MOVE THE FILENAME (NOW WITHOUT "??") WITHOUT EXTENSIONS
                ; TO THE APPROPRIATE SPOT. THE EXTENSION WERE PUT IN
                ; ALREADY IN THE BEGINNING OF THE PROGRAM.
017C 0E10            MVI     C,CLOSE
017E 115C00          LXI     D,FCB0
0181 CD0500          CALL    BDOS
                ; NOW CLOSE ORIGINAL FILE.
0184 213002          LXI     H,FCB2+1
0187 116302          LXI     D,FCB5+1
018A 010B00          LXI     B,11
018D CD3D03          CALL    LDIR
0190 214B02          LXI     H,FCB3+12
0193 011700          LXI     B,23
0196 CD3D03          CALL    LDIR
                ; SET UP A FCB FOR NEW FILE WITH NEW EXTENSION
0199 216202          LXI     H,FCB5
019C 115C00          LXI     D,FCB0
019F 012300          LXI     B,35
01A2 CD3D03          CALL    LDIR
01A5 115C00          LXI     D,FCB0
01A8 0E0F            MVI     C,OPEN
01AA CD0500          CALL    BDOS
01AD FEFF            CPI     0FFH
01AF CAD301          JZ      CONT
                ; IF THE FILE DOES NOT EXIST, GOOD. THEN IT CAN
                ; BE RENAMED TO THAT NAME. IF IT DOES EXIST
                ; THEN WE CAN'T PROCEED ANY FURTHER.
01B2 0E10            MVI     C,CLOSE
01B4 115C00          LXI     D,FCB0
01B7 CD0500          CALL    BDOS
                ; CLOSE UP THE FILE, THEN DISPLAY MESSAGE, THEN LEAVE
01BA 213002          LXI     H,FCB2+1
01BD 11EA02          LXI     D,MSG7
01C0 010800          LXI     B,8
01C3 CD3D03          CALL    LDIR
01C6 13              INX     D
01C7 010300          LXI     B,3
01CA CD3D03          CALL    LDIR
                ; PREPARE MESSAGE WITH THE CORRECT FILE THAT
                ; GAVE US THE PROBLEM
01CD 11E102          LXI     D,MSG8
01D0 C30F02          JMP     HOME
                ; DISPLAY THIS MESSAGE.
                CONT:
01D3 0E17            MVI     C,RNM
01D5 111F02          LXI     D,FCB1
01D8 CD0500          CALL    BDOS
                ; TELL BDOS TO RENAME THE OLD FILE TO THE SAME NAME
                ; BUT WITH A DIFFERENT EXTENSION.
01DB 212002          LXI     H,FCB1+1
01DE 118F02          LXI     D,MSG3
01E1 010800          LXI     B,8
01E4 CD3D03          CALL    LDIR
01E7 13              INX     D
01E8 010300          LXI     B,3
01EB CD3D03          CALL    LDIR
01EE 213002          LXI     H,FCB2+1
01F1 119F02          LXI     D,MSG4
01F4 010800          LXI     B,8
01F7 CD3D03          CALL    LDIR
01FA 13              INX     D
01FB 010300          LXI     B,3
01FE CD3D03          CALL    LDIR
                ; PREPARE MESSAGE WITH APPROPRIATE FILE
                ; NAME AND EXTENSIONS TO LET US KNOW
                ; WHAT IT IS DOING.
0201 0E09            MVI     C,PNTSTR
0203 118502          LXI     D,MSG1
0206 CD0500          CALL    BDOS
0209 C34B01          JMP     RNLOOP
                ; DISPLAY THE MESSAGE, THEN PROCESS THE NEXT
                ; FILE.
                ENDIT:
```

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

MASS RENAMING

Continued from page 124

```
020C 11AE02             LXI     D,MSG5
                ; THAT'S IT. WERE DONE. PREPARE DONE MESSAGE
                ; THEN DISPLAY IT.
                HOME:
020F 0E09               MVI     C,PNTSTR
0211 CD0500             CALL    BDOS
0214 2A4803             LHLD    STACK
0217 F9                 SPHL
0218 C9                 RET
                ; DISPLAY APPROPRIATE MESSAGE (ERROR,SUCCESSFULL,BAD FORMAT)
                ; THEN RESET THE STACK TO WHAT IT WAS THEN 'RET' TO CP/M
                ; THIS RETURN DOES NOT DO A WARM OR COLD BOOT. IT JUST
                ; GOES BACK TO THE 'A>' OR WHATEVER DRIVE IT WAS ON.
                BF:
0219 11BA02             LXI     D,MSG6
021C C30F02             JMP     HOME
                ; GET READY TO DISPLAY 'BAD FORMAT' MESSAGE THEN EXIT.
0017 =          RNM     EQU     23
0010 =          CLOSE   EQU     16
000F =          OPEN    EQU     15
0009 =          PNTSTR  EQU     9
005C =          FCB0    EQU     5CH
0005 =          BDOS    EQU     5
                ; SYSTEM CALL EQUATES AND OTHER MISC. EQUATES.
                FCB1:
021F 00                 DB      0
0220 3F3F3F3F3F         DB      '????????'
0228 3F3F3F             DB      '???'
022B 00000000           DB      0,0,0,0
                FCB2:
022F 00                 DB      0
0230 3F3F3F3F3F         DB      '????????'
0238 3F3F3F             DB      '???'
023B 00000000           DB      0,0,0,0
                FCB3:
023F 00                 DB      0
0240 3F3F3F3F3F         DB      '????????'
0248 3F3F3F             DB      '???'
024B 0000000000         DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                FCB5:
0262 00                 DB      0
0263 3F3F3F3F3F         DB      '????????'
026B 3F3F3F             DB      '???'
026E                    DS      23
                ; MISC. FCB'S USED BY PROGRAM FOR SCRATCH AREA (USED TO
                ; OPEN THE '????' FILE , DETERMINE IF DESTINATION FILE
                ; IS ALREADY EXISTENT, AND IF NOT RENAME IT.
                MSG1:
0285 52454E414D         DB      'RENAMEING '
                MSG3:
028F 2020202020         DB      '        '        ; 8 spaces
0297 2E                 DB      '.'
0298 202020             DB      '   '             ; 3 spaces
029B 20544F20           DB      ' TO '
                MSG4:
029F 2020202020         DB      '        '        ; 8 spaces
02A7 2E                 DB      '.'
02A8 202020             DB      '   '             ; 3 spaces
02AB 0D0A24             DB      13,10,'$'
                MSG5:
02AE 444F4E452E         DB      'DONE.....',13,10,'$'
                MSG6:
02BA 2A20424144         DB      '* BAD FORMAT *',13,10
02CA 52454E4558         DB      'RENEXT [D:]EXT EXT',13,10,10,7,'$'
                MSG8:
02E1 5448452046         DB      'THE FILE '
                MSG7:
02EA 2020202020         DB      '        '        , 8 spaces
02F2 2E                 DB      '.'
02F3 202020             DB      '   '             ; 3 spaces
02F6 20414C5245         DB      ' ALREADY EXISTS. ',13,10
0309 50524F4345         DB      'PROCESSING HALTED. ALL FILES MAY NOT BE RENAMED'
0338 0D0A0A0724         DB      13,10,10,7,'$'
                ; MISC. SYSTEM MESSAGES TO LET YOU KNOW WHAT IS GOING ON.
                LDIR:
033D 7E                 MOV     A,M
033E 12                 STAX    D
033F 23                 INX     H
0340 13                 INX     D
0341 0B                 DCX     B
0342 78                 MOV     A,B
0343 B1                 ORA     C
0344 C8                 RZ
                ; THIS ROUTINE 'SIMULATES' A Z-80 'LDIR' FUNCTION
                ; NOTE: THIS ROUTINE 'DOES NOT' REQUIRE YOU HAVE
                ; A Z-80 MICROPROCESSOR. AN 8080 WILL HANDLE IT FINE
                ; DUE TO THE FACT THAT IT IS NOT AN 'LDIR' IT JUST
                ; DOES THE SAME THING.
0345 C33D03             JMP     LDIR
0348 0000       STACK   DB      0,0
034A                    DS      32
036A =          STKTOP  EQU     $
                ; THE STACK GETS RESET TO HERE (16 LEVELS, 32 BYTES)
036A                    END
```

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.

Systems serviced nationwide by M/A/I/Sorbus Service Division.

Special OEM incentives. Call our OEM Coordinator at (415) 357-8500.

CIRCLE 19 ON READER SERVICE CARD

Software Directory

Software packages designed to enhance the capabilities of your computer system

Program name: DataCURE: Error-correction disk backup/restore utility.
Hardware system: 8080/Z80; CP/M 2.2; 8″ drive
Minimum memory: 48K
Language: 8080 Assembler with some Fortran.
Description: dataCURE is a utility program to ensure the integrity of data or programs on archive or interchange disks. It uses very powerful proprietary error-correction algorithms.

DataCURE rebuilds files affected by bad sectors, restoring the original data to replacement sectors. It can regenerate a complete track, including the directory track.

Information for detecting bad sectors and corrrecting the data in them is kept in duplicate "protection" files occupying about 5% of disk space.

DataCURE supports all CP/M compatible formats and sector sizes. It is self configuring, with both menu and command modes, and gives English-language error messages.
When released: May 1983
Price: $99, including 8″ disk and bound user's manual. Demo disk priced at $19; includes .DOC file. Add $5 shipping.
Available from:
Colorado Online Systems, Inc.
40 Balfour Lane
Ramsey, NJ 07446
(201) 327-5155
CIRCLE 310 ON READER SERVICE CARD

Program name: TXL: Telex Link
Hardware system: A modem; runs on any microcomputer with CP/M, CP/M-86, MP/M, MP/M II, MP/M-86, MS-DOS
Minimum memory: 48K (8-bit); 64K (16-bit)
Language: C
Description: TXL is a telecommunications package to interface most microcomputers to the Western Union Telex II (TWX) network. TXL allows your computer to function as an intelligent telex station, replacing the conventional paper tape telex machine. Edit your telex messages to perfection using your text editor. TXL automatically formats and sends them. Incoming telex messages are received, displayed, stamped with date and time and then written to disk storage.
When released: 1979
Price: $350
Available from:
Cawthon Scientific Group
24224 Michigan Ave.
Dearborn, MI 48124
(313) 565-4000
CIRCLE 311 ON READER SERVICE CARD

Program name: CTL: Computer Telex Link
Hardware system: A modem; any microcomputer with CP/M, CP/M-86, MP/M, MP/M-II, MP/M-86, MS-DOS
Minimum memory: 48K (8-bit); 64K (16-bit)
Language: C
Description: Computer Telex Link is a communications program designed specifically for use with Western Union's EasyLink service. With your personal computer, CTL, and an EasyLink account you can conveniently send and receive messages to and from any telex station anywhere in the world. In addition, You can originate mailgrams, telexgrams, and international cables. CTL is delivered complete and ready to run. No user software patching is required.
When released: 1982
Price: $165
Available from:
Cawthon Scientific Group
24224 Dearborn, MI 48124
(313) 565-4000
CIRCLE 312 ON READER SERVICE CARD

Program name: GINA+
Memory and hardware requirements: 64K with CP/M-80 or 128K with CP/M-86 or PC-DOS, an 80-column CRT; two high-capacity double-sided double-density disk drives.
Language: Pascal MT+
Description: GINA+ is a software program that functions as a point-of-purchase microcomputer retail sales assistant. It is designed to help the salesperson educate, acclimate, qualify, and instill confidence in apprehensive and/or curious prospects. GINA+ helps the prospect decide which of the retailer's systems could meet his needs, but leaves the "fine tuning" and personalized selling to the salesperson.

The GINA+ software system consists of three major components: 1) software and hardware tutors, 2) the consultant, and 3) the proposal. The tutors provide the information required to understand the purpose and use of the system components that need to be purchased. The consultant conducts an interview to obtain information on what is to be done by the system, as well as how much and how often. Follow-up information is also obtained. Based upon the answers supplied, a proposal is generated. The proposal then serves as a model upon which the prospect and salesperson build.
Price: $495
Included with price: a program to configure GINA+ for your terminal, a customizer that allows the dealer to input specific items from his inventory by

brand name, the GINA+ system, and documentation.

Available from:

System Vision Corporation
199 California Drive, #207
Millbrae, CA 94030
(800) 352-9999 or
(415) 697-3861

CIRCLE 313 ON READER SERVICE CARD

Program name: Informail
Hardware system: IBM VMCMS
Minimum memory: 256K
Description: Informail is a proprietary electronic mail system offered to timesharing users of the Informatics data center. It provides each user with an electronic mailbox which acts as a storage device, backs up all files automatically, and only permits users to read mail sent to their own mailbox ID. The directory function allows mail to be addressed to a user's name if the Informail ID is unknown. Even if the user does not supply exact spelling, the directory recognizes names of recipients by abbreviations, partial names and phonetic spelling. When a perfect match is not found, the system presents the user with a list of options. The directory also contains information about organizational structure and allows the user to request the names of all employees reporting to a particular manager. The directory can also create distribution lists for future use. The system's create function

Informail's directory recognizes abbreviations, partial names, and phonetic spellings.

prompts the user for standard memo fields and text entry. A send feature allows the user to specify each receiver's mailbox ID. A read capability allows the user to receive mail from other system users. Mail may be directed to the terminal, the printer or disk. The modify function permits the user to cancel any mail received and add or delete names and IDs from a distribution list. A list of outgoing mail can be created including subject, date and time sent, intended receiver and information on

SOFTWARE

Continued from page 129

whether mail has been read or cancelled.
Price: $25 per hour
Available from:
Informatics General Corporation
6 Kingsbridge Road
Fairfield, NJ 07006
(800) 631-1156
CIRCLE 314 ON READER SERVICE CARD

Program name: SmartKey II
Hardware system: any microcomputer running CP/M; the IBM PC
Minimum memory: SmartKey II: 4.25; SmartPrint: 4K (2.2K when run with SmartKey II
Language: Some Assembler, some C
Description: SmartKey II is a background utility program that allows the user to customize a microcomputer keyboard to reflect software needs. All keys may be redefined at will to become other characters, software commands, or whole words, phrases or boilerplates. Definitions may be changed at any time and saved to disk. Features of SmartKey II include instantaneous installation of special function keys, an expanded buffer capable of handling 3,750 definitions at a time, a text compiler that can reduce entire text files to a single keystroke, and a built-in editor for complex reorganization of key definitions. SmartPrint, an upgrade to SmartKey II, enables users to implement all of a printer's built-in functions and character sets.
Price: SmartKey II: $89.95
SmartPrint: $49.95
Both programs together: $129.95
Available from:
Heritage Software
3757 Wilshire Blvd., #211
Los Angeles, CA 90010
(213) 384-4120
CIRCLE 315 ON READER SERVICE CARD

Program name: MICA General Ledger Package
Hardware system: MS-DOS or PC-DOS
Language: MBasic
Description: The MICA General Ledger Package was designed for first-time users and offers many easy-to-use features, such as on-line help screens, formatted data entry screens, full screen editing during input, extensive error trapping, and batch entry with proofing prior to updating permanent records. The package comes with a comprehensive user's manual that is a self-teaching guide for all aspects of program setup and operation. The chart of accounts is totally user-definable, with account numbers ranging from three to six digits. Departmentalization of revenue and expense accounts is permitted for up to

Departmentalization of accounts is allowed for 99 departments.

99 departments. MICA General Ledger also has master/sub account relationships to provide flexibility in formatting the balance sheet and the P&L statements. Reporting flexibility is also provided through the grouping of accounts in up to 20 user-definable account categories including a special category for nonoperating format, and is balance-verified by the computer to eliminate an unbalanced ledger situation. Transaction entry is also performed in batch mode which enables the user to print a data entry proof and make corrections prior to updating permanent account

balances.
Available from:
Micro Associates, Inc.
2300 Highway 365, #510, LB 131
(409) 724-6583
CIRCLE 316 ON READER SERVICE CARD

Program name: MITE/MS
Hardware system: any microcomputer running IBM PC-DOS or MS-DOS
Minimum memory: 64K
Language: Assembler
Description: MITE/MS is a menu-driven data communications package that converts any computer into an intelligent terminal for use with online services such as Dow Jones, The Source, CompuServe, TWX and for transferring both text and binary files between 8 and/or 16-bit microcomputers. MITE/MS can also be used to access corporate data centers. It has user-protection features that prevent operator errors. All communications parameters (parity, baud rate, etc.) are under full user control on *all* versions. Various user-selectable options allow MITE/MS to capture text from or send text to virtually any asynchronous/ASCII online system. The user may also define up to 10 programmable macro strings which can be invoked via function keys from within the link. This mechanism also sports fully automatic logon on most systems. A full set of system interface functions (DIR, TYPE,etc.) are available from within MITE/MS. It also supports three binary file protocols, (XMODEM, CLINK, CROSSTALK, and HAYES) for compatibility with the largest possible number of other programs. These protocols allow transfer of any file with error checking and automatic retransmission. MITE/MS is now available preinstalled on the IBM PC-XT, Columbia, Compaq, Corona, VICTOR and on other MS-DOS systems. MITE, the CP/M-80 version, is available on Xerox, Exxon, Sanyo, TeleVideo, Heath/Zenith, KayPro, North Star, Radio Shack, and others. MITE/86, the CP/M-86 version, is available on many CP/M-86 machines.
Price: $195
Included with price: 5′ disk and reference manual
Available from:
Mycroft Labs, Inc.
P.O. Box 6045
Tallahassee, FL 32314
(904) 385-1141
CIRCLE 317 ON READER SERVICE CARD

New Products

What's new: a quick roundup of recent innovations and improvements

Hard disk systems for CP/M

CFX Disk Systems, Inc., formerly a part of CF CompuTrax, has announced the release of its first two hard disk systems for CP/M machines: the CFX-10 and the CFX-16. Both these systems incorporate a 5¼″ Miniscribe hard disk.

These systems are currently available for most Z80- or 8080-based machines running CP/M and having the Digital Research utility MOVCPM.COM. Additionally, special models are available for the NEC 8000, Apple II and III, and the IBM PC. Setup and use is fast and simple. Both systems included a manual that gives step-by-step instructions for adding the adapter to the computer and a complete explanation of the software sold with the disk. The software is used not only to format, verify, and run the system, but also to check out the disks, controller card and buffer in the event a problem should arise.

The CFX-10 disk system is actually a 12 MB disk which, when formatted, yields 10 MB of usable space. The CFX-16 disk system is based on a 20 MB disk before formatting.

Prices: Current mail order pricing is $1,995 for the CFX-10 and $2,395 for the CFX-16. Future products will include a tape backup system for the hard disk, 40 MB formatted to 32 MB—to be called the CFX-32—and chaining up to four disks for even greater storage.

CFX Disk Systems, Inc., *P.O. Box 920152, Norcross, GA 30092; (404) 255-3030.*

CIRCLE 320 ON READER SERVICE CARD

High-performance hard disk drives

Percom Data Corporation has announced the release of a new generation of high-performance PHD™ hard disk drives for a variety of personal computers. The new product line will provide speed enhancements and one of the most efficient implementations of memory caching available today.

Percom Data claims to have improved on Winchester technology by implementing a very efficient form of caching driver. Their design permits speed performance increases four times greater than that of key competitors. Percom's benchmark tests show that the PHD hard disk product line is more than five times faster in data transfer than the IBM XT hard disk. The new hard disks are fast, responsive and flexible in performance.

Caching design permits the Percom Data hard disk drive to work with the host computer so that most used files are actually stored and processed in memory—to increase speed—then transparently returned to hard disk memory for data security.

This will benefit sophisticated users, programmers, office managers, network resource managers, and anyone anticipating future applications with multitasking, multiuser functions.

The new Percom Data PHD product line is immediately available for the IBM-PC, and supports PC DOS 2.0. High-performance PHDs for IBM-compatible microcomputers will soon be available.

Prices: Caching driver "personality kit," $149.95. Percom PHD hard disk subsystems will continue to be priced from $1,895 (for a 5 MB version) and up. A caching driver sofware upgrade kit is also available direct from Percom Data for $50. This will enable present Percom IBM DOS 2.0 PHD owners to upgrade to the high-performance version. Cost of the upgrade package is $50 and is available only from Percom Data. Minimum memory requirement: 256K RAM.

Percom Data Corp., *11220 Pagemill Rd., Dallas, TX 75243; (214) 340-5800.*

CIRCLE 321 ON READER SERVICE CARD

16-bit SBC built around Intel 80126

Advanced Digital has announced the introduction of the SUPER 186™, the first 16-bit S-100 single board computer build around the Intel 80186. The 8 MHz SUPER 186 can be configured as a stand-alone bus master or bus slave to serve both single or multiple users with superior speed, versatility, and reliability.

SUPER 186 features 256 KB of memory, expandable to 1 MB, and a floppy disk drive controller that can simultaneously support both 8″ and 5¼″ disk drives. It also comes with four serial RS-232 and two paralled 1/0 ports, DMA controller, parity and monitor EPROM to aid in initial loading. The board is compatible with CP/M-86, MP/M86, Turbodos and MS-DOS operating systems. Performance of existing CP/M-86 systems can improved by as much as 2½ times with SUPER 186.

In a TurboDOS environment, SUPER 186 can also function as a powerful 8/16-bit bus slave. By using the SUPER 186 for large processing demands, the user frees the remaining, less-powerful boards for other tasks, while the SUPER 186 board rapidly completes the assigned job.

Price: $1,650; quantity discounts available for OEMs and independent sales organization.

SUPER 186 is the latest addition to Advanced Digital Corporation's S-100-based modular microcomputer line that includes SUPER SYSTEM II, one of the most powerful multiuser microcomputer systems available; SUPER STAR, the first S-100 system with both a fixed and removable Winchester drive and expandable to four users; SUPER SIX (16 MHz, Z80-based) and SUPER QUAD (4MHz, Z80-based) 8-bit, single board computers; SUPER SLAVE, a powerful single-board processor; and a compact, microprocessor-based hard disk controller.

Advanced Digital Corp., *5432 Production Drive, Huntington Beach, CA 92649; (714) 891-4004.*

CIRCLE 322 ON READER SERVICE CARD

Disk subsystems

PH Associates, Inc. has announced three new additions to its continuing disk subsystem product line. The DSS-5B, DSS-10B, and DSS-15B are designed to provide enconomical disk storage for those users not requiring the high performance (capacity and speed) of the MARK series. These 85 ms average access drives use industry standard Seagate ST-400 series compatible drives. The DSS series comes as a complete subsystem assembled in its own stand-alone chassis or in OEM versions consisting of drive, cables and controller for use by system integrators.

The subsystems include a smart controller with intelligent formatting and automatic alternate sectoring. Features include • 85 ms average access speed (buffered seek) • 90-day parts and labor warranty on *both* the controller *and* the disk unit, including heads and platter • 5¼" floppy physical size compatibility • 5, 10, and 15 MB disk capacities (formatted) • Heat dissipation less than 40 watts • Interfaces to DEC Q-BUS, S-100, and any Z80 processor such as the Northstar Advantage, Apple IBM PC, TRS-80, Zenith Z-100, Motorola VME BUS, Multibus, and NEC as well as any 8-bit parallel port • Complete software currently available for CP/M and Apple, TRS-0, NEC, IBM-PC • 5 megabit per second disk transfer rate.

Prices: Formatted capacities of 5, 10 or 15 MB are available with list prices of $1,995, $2,295, and $2,695 respectively. Significant quantity and OEM discounts are available.

PH Associates, *8720 Old Courthouse Rd., Vienna, VA 22180; (703) 281-5762.*

CIRCLE 323 ON READER SERVICE CARD

Stand-alone array processor for APL software

Analogic Corporation has announced the APL Machine, the first stand-alone general-purpose computer system using array architecture to deliver mainframe APL performance at a fraction of the cost. Featuring the interaction and response of a personal computer, the APL Machine is a convenient yet powerful development tool for high-level decision making. Designed to replace traditional costly and less flexible methods of processing APL, the APL machine can be used for strategic planning and forecasting, as well as financial, actuarial, graphics animation, and scientific applications.

A typical configuration consists of a 4 MB array processor, an IBM PC as the programmer's workstation, a 124 MB hard disk, a dual-mode tape drive, an I/O processor supporting up to 8 terminals, and the Analogic software system. System architecture supports several levels of overlapped and parallel processing, permitting a very high level of performance and throughput.

While the APL interpreter runs in the 12.5 MHz 16/32-bit super micro control processor, handling all of the aspects of syntax and conformance checking, the array portion of the machine concurrently executes arithmetic and logical expressions. Moreover, it provides several high-speed processing elements executing additions, multiplications, and logical operations at the same time. The APL Machine's array processor can execute up to 10 million full floating-point operations per second. Because of hardware design efficiencies, pipeline throughput is competitive with machines rated at 15 million operations per second.

The Analogic 32-bit floating-point array processor used in the APL Machine is also programmed directly in APL. Since the primitive functions and operators of APL reside in pipeline microcode, high-speed execution in most applications is achieved directly from APL. For applications using different software such as compiled or assembled code, Analogic's APL implementation provides shells to incorporate the non-APL code. These "shells' are similar to UNIX shells and handle non-APL code in the same manner as if it were written in APL.

The Analogic multiuser, virtual memory APL environment is designed to bypass the need for setting up independent workspaces and files. A sophisticated memory manager allows the user to nest applications and to share code among concurrent processes.

NEW PRODUCTS

Continued from page 133

A sophisticated operating environment on the PC workstation, designed by Analogic and called InSight, allows the user to display up to 10 concurrently running tasks on overlapping and sliding windows. The PC environment is itself a mulitasking, virtual memory system that communicates to the array processor through a packet protocol. Its capabilities extend to using the PC for concurrent interaction with muliple processes on multiple systems.

Analogic is engaged in the design and manufacture of high-precision data conversion products, medical imaging systems, control instrumentation, and signal processing computer equipment.

Prices for a basic APL Machine with ½ megabyte of memory start at $44,000. The typical system described above is about $85,000.

Analogic Corp., *Audubon Rd., Wakefield, MA 01880; (617) 246-0300.*

CIRCLE 324 ON READER SERVICE CARD

Sierra Data Sciences 8 MHz micros

Sierra Data Sciences is producing two new S100 bus, Z80H single-board computers (SBCs). The boards, a master (SBC 100/8M) and a slave (SBC 100/8S) are the first production-run SBCs to feature 8 MHz speeds. In addition, this new generation of high-speed, very powerful microcomputers takes full advantage of the design benefits offered by the IEEE-696//S-100 bus standard for exceptional throughput rates. Typically, there is a 100% increase: SBCs operating under the latest versions of Sierra Data-supported CP/M or TurboDOS are significantly faster than most 8-bit micros.

On board each standard SBC, master or slave, is a Z80H central processor with 64 to 512K RAM, bank selectable in 8K segments. Additional support chips include DART for two serial ports, 4K to 32K EPROM, and an optional math coprocessor.

Two parallel ports and clock timing are provided for on the master via Zilog CIO. Similar functions are handled through a PIO and a CTC on the slave. The slave also features a 4K static RAM buffer for high-speed 8- or 16-bit bus transfers and upward compatibility to 16-bit systems.

The master uses a DMA controller that provides for very fast data transfers without CPU intervention and no wait states. Disk reads, for example, are typically 200 to 500% faster. DMA transfers are possible when moving data from master memory to slave memory and from master to or from floppy disk or hard disk. Additionally, a floppy disk controller that can simulaneously run 5¼″ and 8″ media is another onboard feature. A high-speed Winchester DMA port is also standard.

Built for speed and functional reliability, the SBCs are multilayer boards with separate ground and power planes. Yet both boards meet the IEEE-696/S-100 standard and therefore fit any compatible system.

In stand-alone configuration, the master makes an exceptionally powerful system for personal computing or standard data processing tasks. Applications requiring high-speed real-time monitoring, dedicated process control and communications are especilly well suited to the new SBCs.

Prices: Standard boards (128K RAM) are "quantity-one" priced at $995 for the SBC 100/8M and $895 for the SBC 100/8S.

Sierra Data Sciences Inc., *25700 First Street, Westlake, OH 44145; (216) 892-1800.*

CIRCLE 325 ON READER SERVICE CARD

"Watchdog" timer added to multipurpose Q-Bus board

Codar Technology, Inc., has announced a new version of its Q-Timer™ single-board system module that adds a watchdog timer to existing functions such as a calendar clock and nonvolatile CMOS memory.

Designed for integrators of real-time systems based on the DEC family of microprocessors, the dual-wide Q-Timer board is the only product of its kind with complete software support stored in an on-board EPROM. The board provides built-in compatibility with DEC's 16, 18 and 22-bit bus architectures and, in some instances, can replace up to four dual-wide modules.

The watchdog timer makes the Q-timer particularly suitable for unattended or remote operations in that it will reboot a system if program execution stops because of a noncatastrophic problem, such as a bad memory cell, static discharge, or a failure to transfer data from a peripheral device.

If, for some reason, a programmed instruction is not completed, the machine may sit there and wait forever unless a person or an escape technique intervenes. The watchdog device will automatically activate power-fail/restart logic if a user program does not read the watchdog register at least once during a user-defined interval. Thus, when program execution stops prematurely, the timer is not reset and the system is rebooted.

Also integral to the Q-timer is a CMOS calendar clock that provides tenths of seconds, seconds, minutes, hours, days, the day of week, month and year, with automatic leap-day insertion. It uses a 3.768 KHz crystal oscillator for reference.

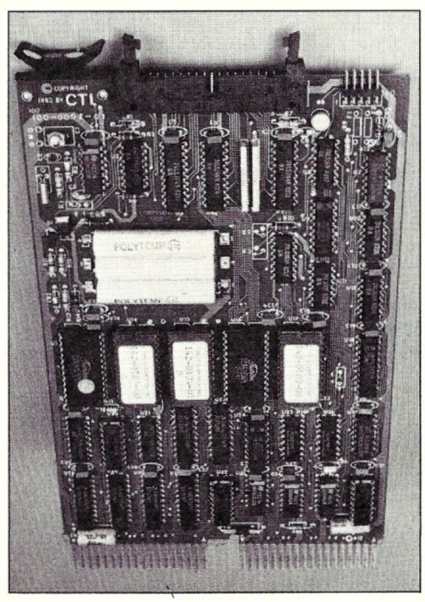
An on-board CMOS RAM memory provided 2K X 16 bits of storage for data and/or programmed parameters. This feature is particularly useful for scientific or processs applications where there is no disk storage.

CMOS devices are backed up by an on-board NiCad battery pack (with recharging circuit) that will maintain operational readiness for a minimum of 60 days.

All of the software necessary to operate the Q-Timer functions is stored in EPROM, eliminating the requirements of a SYSGEN or patching of either RT-11 or RSX operating systems. An EPROM software monitor provides self-diagnostics of on-board devices and has bootstrap routines for RL01/02, RK05, RK06/07, RM02, TU58, and TM11 storage devices. In addition to the self-diagnostic routine, the software will run complete CPU and memory diagnostics.

Expandable 8-bit parallel input and output provide a path by which the Q-Timer can perform a variety of I/O functions in addition to routine calendar clock tasks, often eliminating a DRV11-type interface board.

Codar's Computer Products Dvision was formed in 1982 to market DEC system enhancement products developed by the company for its own use in a ruggedized data acquisition/HF radar system used in remote sensing of ocean surface conditions.



Price: Codar Technology Inc., *437 Main St., Longmont, Co 80501; (303) 772-2782.*

CIRCLE 326 ON READER SERVICE CARD

S-100 modules for Intel 80286

Performics Inc. has recently introduced a pair of S-100 modules aimed at the high-performance end of the computer market. The P-286 CPU module is designed around Intel's 80286 micro, allowing it to execute software developed for the 8086/8088. On-board resorces include a socket for the optional 80287 math coprocessor, three 16-bit programmable and cascadable system timer, programmable interrupt controller, and sockets for 64K bytes of IPROM/ROM. The P-128 is a 128K-byte that features 8K x 8 CMOS RAM chips for low power consumption. The boards interface via an IEEE-696 interface and a private acces bus for no wait states.

Prices: Initially, the P-286 and P-128 boards are being offered as a pair at a price of $3,000.

Performics Inc., *P.O. Box 3207, Nashua, NH 03061; (603) 881-8334.*

CIRCLE 327 ON READER SERVICE CARD

Hard disk backup

Stok Software, Inc. has announced nine major enhancements to BackRest, their hard disk backup program.

BackRest now runs on TurboDOS, MS-DOS (PC-DOS) and local Area Networks such as CP/NET and OMNI/NET. Other enhancements include: • Ability to select predefined backup routines • Password protected file backup and restore • Repeated backup to the same hard or floppy disk • Improved error handling • Optional use of screen color attributes • German language version with European date format.

On Local Area Networks the user can back up files on the server or any node. The destination disk can be at any location. It has successfully been tested on CP/NET and OMNI/NET.

Unique backup routines can be predefined for flexible control. For example, individuals in a large organization can have their own backup procedures. With complete security BackRest can now back up and restore password-protected files on MP/M, MP/M-86 and Concurrent CP/M.

BackRest can now use another ard or floppy disk as the destination th the option of updating or scratch- previously backed up files.

Improved error handling now pro- es errors that occur while other er- are being handled.

hree popular features that never l improvement have been re- 1) BackRest creates manage- ports to show what it has done ides statistics of hard disk us- handles sparse files; and 3) files

NEW PRODUCTS

Continued from page 135

too large for a single disk are split. BackRest is available for CP/M, MP/M, CP/NET, CP/M-86, MP/M-86, Concurrent CP/M, TurboDOS, MMMost, and MS-DOS.

Stok Software, Inc., *17 West 17th St., New York, NY 10011; (212) 243-1444.*

CIRCLE 328 ON READER SERVICE CARD

ESDI/ESTI drives

OMTI has announced the introduction of the Series 6000 family of intelligent data controllers designed to attach disk and tape drives having the ESDI/ESTI (Enhanced Small Disk/Tape Interface) to a variety of host systems. The host computer bus is the industry standard SASI (ANSI-SCSI) bidirectional bus interface, with its associated high-level command set.

The ESDI/ESTI disk and tape interface was developed by a cooperative effort of disk and tape drive manufacturers interested in meeting the demand for increased performance and capacity while providing an industry standard interface. Described as an evolution of the ST506/412 interface, the ESDI/ESTI interface standard is data rate independed (NRZ), encoding method independent, and defines a serial command and response protocol. This makes it possible for controllers to dynamically determine the drive parameters. The ESDI/ESTI interface is unique in the sense that both disk and tape drives can be attached on the same bus without sacrificing the performance characteristics or cost of either.

The Series 6000 controllers are designed around OMTI's third-generation advanced VLSI data controller chip set. This 15-megabit chip set provides performance features such as consecutive sector transfers, 2.0 MB host data transfer rates, and intelligent buffer management.The Series 6000 family will initially consist of the Model 6100 ESDI controller and the Model 6300 ESDI/ESTI controller, with other products to follow.

The Model 6100 disk controller is packaged in the industry standard $5\frac{1}{4}''$ form factor and will support up to four ESDI disk drives. Features include disk data rates up to 10 megabits per second, 2.0 MB host data transfer rates, consecutive sector transfers, overlapped seek operation, automatic configuration of disk parameters (heads, cylinders, sectors/track, data encoding method, etc.) on power-up, format using the drive manufacturers prerecorded defect list, 48-bit ECC error correction code with 20-bit error correction capability, defective sector/track handling and interdevice copies.

The Model 6300 disk/tape controller is also packaged in the industry standard $5\frac{1}{4}''$ form factor and will support up to four ESDI/ESTI and/or ST506 devices. The ESTI devices may be streaming, start/stop, and/or block addressable tape drives. In addition to the Model 6100 features, the Model 6300 disk/tape controller supports tape data transfer rates to 5 megabits per second, automatic configuration of tape parameters (tracks, tape speed, tape density, data encoding method, etc.) and interdevice copies.

The ANSI-proposed Small Computer System Interface (SCSI) (also know as SASI) is implemented in the OMTI Series 6000 intelligent controller family. SCSI defines a host level I/O bus structure that can be operated at data rates in excess of 1.5 MB per second. The primary objective of this interface is to provide host computers with device independence, so that disk drives, tape drives, and future devices can be added to the host computer without requiring moodifications to the generic system software or hardware. The ESDI/ESTI device level interface supports this concept by suppling the important device parameters upon request.

Prices: (in quantities of 1000): Model 6100, $255; Model 6300, $350.

OMTI, *557 Salmar Ave., Campbell, CA 95008; (408) 370-3555.*

CIRCLE 329 ON READER SERVICE CARD

PC bubble memory board

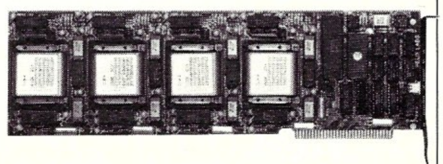
Helix Laboratories, Inc., has introduced the PC Bubble board, a half-megabyte bubble memory expansion board designed to emulate a "mini-Winchester" in the IBM Personal Computer. The board brings many of the convenient features of the fixed disk to the PC without requiring an increased or external power supply.

The PC Bubble board responds to fixed disk commands under most operating systems including MS-DOS 2.0, Softech Pascal IV.13, and CP/M-86 for the PC/XT. Software features such as the RESTORE and BACKUP commands as well as partitioning to hold multiple operating systems are available to the user. If the AUTOSCAN ROM BIOS is installed, the PC Bubble can also cold boot any of the above operating systems.

The all solid-state PC Bubble contains four Intel 7110 one-megabit bubble memories, providing rugged, nonvolatile mass memory with no moving parts, as well as immunity to dusty, hostile environments. Operation is completely quiet. Mean access time (40 ms) and data transfer rate (400 kbit/s) of the board result in file transfers that are several times faster than floppy disk. Bubble memory technology makes the PC Bubble several orders of magnitude more reliable than either the fixed or floppy disk.

In addition to the PC, the PC Bubble works in the IBM XT and most IBM-compatible computers. The lower power demands and compactness of the PC Bubble make it a unique enhancement to compatible portables, such as the Compaq and Corona.

Price: $1,495, with a four-to six-week availability time.



Helix Laboratories, Inc., *16776 Bernardo Center Dr., Ste. 106A, San Diego, CA 92128; (619) 451-0270*

CIRCLE 330 ON READER SERVICE CARD

Monolithic Power FET Arrays

Supertex Inc., active in CMOS and DMOS technology, has introduced monolithic power FET arrays designed to drive high-voltage, nonimpact printers or flat panel (plasma, LCD or electroluminescent) displays. The 8-channel, AN01 (N-channel) and AP01 (P-channel) devices are fabricated using lateral DMOS technology.

The devices, which are available in 18-pin DIPs, reduce the cost per channel by up to 40% compared to similar circuits designed with discrete TO-92 parts. In addition to cost benefits, the arrays reduce insertion costs and improve board space utilization.

The arrays have common-source construction with underdedicated gates and drains. This provides the circuit design flexibility by allowing each FET to be independently driven. Designers using these arrays can directly interface CMOS logic with high voltages.

The AN01 and AP01 can sustain continuous drain currents of 30 mA and 15 mA per channel respectively and are rated from 200 to 400 volt drain to source voltage. Both are available from stock in production quantities.

Prices: Pricing depends on voltage level and whether the devices are plastic or ceramic. AN01 20NA (200 volts): $249 in 1K quantities; AP01 20NA (200 volts): $309 in 1K quantities. Both also available at 300 and 400 volt levels; prices are higher for the ceramic models.

Supertex Inc., *1225 Bordeaux Dr., Sunnyvale, CA 94086; (408) 744-0100.*

CIRCLE 331 ON READER SERVICE CARD

Z80 Software

DEBUG QED LINKER MODEM MACRO TPM BASIC I ZDDT ZTEL QSAL ZEDIT TOP I 6502X TPM II

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....

MasterCard VISA

OEMS:

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

Dealer Inquiries Invited.

For Phone Orders ONLY Call Toll Free...
1-(800) 458-3491 (Except Pa.)
Ask For Extension #15

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

Computer Design Labs

342 Columbus Avenue/Trenton, NJ 08629

CIRCLE 84 ON READER SERVICE CARD

LETTERS

Continued from page 45

bank is currently selected. This memory is known as common, or global, memory. Code which changes banks must execute in common memory so that the instruction executed after the one which switches banks is the following instruction, rather than whatever happens to be at the corresponding address in another bank. This arrangement can be achieved using memory boards smaller than 64K, by having one board for the common memory addresses which ignores the extended address.

In order to try to solve this problem with memory boards of 64K or larger, things begin to get tricky. I would like to discuss both a straightforward solution and a more sophisticated one. Since these involve more complicated circuits, rather than simple changes to Bender's circuits, I will discuss them only in general terms.

The straightforward solution checks the regular address lines to determine whether an address is in common memory or not. (With an 8K common memory, it would check lines A13-A15.) If the address is not in common memory, the action is as before, using the contents of the latch to assert the extended address lines; but if the address is in common memory, a constant value of 0 is used. In either case, the ADSB*-gating is used, of course. In this way, addresses in common memory will always be mapped to bank 0.

The problem with the straightforward solution is that the memory in other banks which is "hidden" by the common memory in bank 0 will never be used, and is thus wasted. If this is unacceptable, then a more sophisticated memory mapping circuit must be used, which maps "logical" addresses (as specified by the address lines of the CPU and an extended address latch port) onto arbitrary 24-bit physical addresses. With such a circuit the physical memory could be arbitrarily divided up into a common memory section and some number of banks of whatever size is desirable, and no memory would be wasted. A memory mapping circuit could be built with a RAM (or a ROM, if your configuration doesn't change much) and some support circuitry, but there are two disadvantages to keep in mind: the necessary circuit is fairly complicated, and it requires interception of the CPU address lines before they reach the bus; therefore modification of, and a jumper cable to, the CPU board is necessary.

A few unrelated complaints: the Bender article mentioned a "Listing 1"; try as hard as I might, I couldn't find one. Also, in general, I have found the placement of listings and figures in your magazine to be distressingly distant from the text which references them.

Kurt Gollhardt
128 E. 7th St., Apt. 308
Plainfield, NJ 07060

Editor's note:

The listing to Dr. Bender's article, which was inadvertantly omitted, was subsequently published on page 140 of the December 1983 issue.

Dear Mr. Libes,

I found both "The New 16-bit Super Microprocomputers" and "An Introduction to Local Area Networks: Part II" unnecessarily hard to read because the figures were so many pages beyond the text. I would prefer smaller figures, all on one or two pages, or the text squeezed among the illustrations so they flanked the text or (perhaps best) a combination of the two, since I felt many of the figures were too large for their content and thus could have been smaller and closer to the text.

I suspect that the figure on page 40 could have been clearer, since it seems the table names (GDT, etc.) are in the "visible" part, while the actual registers are in the "invisible" part. Also, having the top section of a table white (e.g., "BASE ADDRESS") usually indicates that it is a title block, which does not seem to be the case here.

Mike Firth
104 N. St. Mary
Dallas, TX 75214

Dear Mr. Libes,

Andrew Bender's hardware review of the CompuPro 86/87 system (November 1983) touched on many of the problems involved in configuring CompuPro boards with older hardware. Here are some observations resulting from my own recent experiences.

1) Older I/O boards: the problem with many I/O boards is not so much a problem of speed as it is a problem of bus compatibility. Just as the CompuPro CPU and disk controller will not work on a nonterminated motherboard, some of the older I/O boards (at least the TEI 3P + 3S vintage 1977 I have) will not work on terminated motherboards. In my case, the addition of address buffers fixed the I/O board.

2) Stepper motors employing the steel band widget technology such as found in the Shugart 850 will run much cooler than the screw mechanism types.

3) For some reason the CBIOS is written with the ACT86 assembler rather than the assembler supplied with CP/M-86. One is therefore compelled to buy an additional assembler if one wants to make significant changes to the CBIOS. Finally, those with System Support Boards (or other similar boards) can use the EPROM at the top of memory to initialize any nonstandard hardware they might have. That is, put a jump at the reset point (FFFF0) to an area in the EPROM, do whatever you have to do, *then* jump to the disk loading process.

David Langmann
2900 Connecticut Ave. NW
Washington, DC 20008

Dear Mr. Libes,

I have some comments on the S-BASIC Compiler that was reviewed in the October 1983 issue. First, the reviewer, when stating how fast S-BASIC was, did not make clear whether he was referring to execution speed or the amount of time taken to compile the source code. Having written quite a few programs in it over the last year, I must say that it compiles code fairly swiftly and compactly, but the execution times are extremely slow, in some cases slower even than DR's CBASIC2. Intrepreted MBASIC-80 runs circles around it, even on long programs. Floating point calculations are especially tedious, running almost 10 times slower than MBASIC for equivalent precision.

Speaking of floating point, there is a bug in S-BASIC's square root algorithm such that the error of the function increases with increasing value of the operand. Users are advised to use the equivalent (x) ^.5 instead, which gives accurate results. Also, when doing character or string I/O via PRINT # statements, CHR$(0) is not allowed. And, although procedures and functions can have local variables, the names of these variables must be unique, i.e., all of your math functions cannot use 'X' as their dummy parameter. This is an annoyance, especially when creating libraries of functions and procedures, which the compiler handles so nicely.

Excepting these minor (?) quibbles, I must agree with Mr. Parker's assessment of the power and flexibility of the language. If it weren't for the lack of speed in execution, it would be my favorite BASIC for the 8080/Z80. Considering its low cost, it is still quite a deal.

If only it were fast. . . .

W. John Bau
Director
SPICA
1601 Paseo de Peralta
Santa Fe, NM 878501

Dear Mr. Libes,

Re: Loading and calling machine code from Basic:

A simpler way to sort out a subroutine and poke it into memory.

In the October 1983 issue of *Microsystems,* Costa and Leibson describe a method by which a Basic program can load a .HEX file produced by an assembler such as ASM. Here is a simpler way, which depends on the realization that you can read a .HEX file directly from Basic, sort out the subroutine (you know it's got to be in there somewhere), and poke it into memory.

Listing 1 shows the .PRN file of a demonstration subroutine. When this subroutine is called, it prints a message to the console and returns to Basic. The associated .HEX file is shown in Listing 2. Notice the correspondence between the second column of the .PRN listing and the data field of the .HEX listing (F5 at 8000H, C5 at 8001H etc.). The file is structured in what is called Paper Tape Record Format. The colon signals the start of the record. The record type is 00 for data records. The sum of all the bytes in the record, starting with the record length and ending with the checksum, should be evenly divisible by 256. A record length of zero indicates end-of-file.

Listing 3 is a program in Mbasic to load the subroutine and run it. To use this method you will need to splice the loader segment (lines 210 to 350) into your program and provide the appropriate memory limit, subroutine name and call location (lines 130, 140 and 150). A different Basic dialect may require, in addition, some attention to the syntax of the file handling instructions, variable naming restrictions, and hexadecimal number representation. In some of the earlier Mbasic version 5.xx releases, the Clear instruction (line 130) doesn't work. You can accomplish the same thing by using the highest memory initialization option when loading Basic (e.g., A > MBASIC /M:&H7FFF).

Dave Russell
12 Winchester Street
Southboro, MA 01772

LISTING 1

```
                    ;"SUBROUT"
                    ;DEMO MACHINE CODE SUBROUTINE
                    ;D. RUSSELL 10-01-83

0005 =              FDOS     EQU    5              ;CPM SYSTEM ENTRY POINT
0009 =              PCBFN    EQU    9              ;PRINT CONSOLE BUFF FUNCTION NO.
8000                ORG      8000H                 ;STARTING ADDRESS OF ROUTINE

8000 F5             SRENTRY:PUSH    PSW            ;SAVE
8001 C5                     PUSH    B              ; CPU
8002 D5                     PUSH    D              ; REGISTER
8003 E5                     PUSH    H              ; CONTENTS

8004 0E09                   MVI     C, PCBFN       ;SET UP FUNCTION NUMBER AND
8006 111180                 LXI     D, MSG         ; MESSAGE LOCATION PARAMETERS

8009 CD0500                 CALL    FDOS           ;GO DO THE JOB

800C E1                     POP     H              ;RESTORE
800D D1                     POP     D              ; CPU
800E C1                     POP     B              ; REGISTER
800F F1                     POP     PSW            ; CONTENTS

8010 C9                     RET                    ;RETURN TO BASIC

8011 0D0A484552MSG:         DB      0DH, 0AH, 'HERE I AM', 0DH, 0AH, '$'

801F                        END
```

LISTING 2

```
TYPE SUBROUT.HEX

RECORD MARK | RECORD LENGTH | LOAD ADDRESS | RECORD TYPE | DATA | CHECKSUM

:10800000F5C5D5E50E09111180CD0500E1D1C1F10D
:0F801000C90D0A48455245204920414D0D0A240B
:0000000000

END OF FILE
```

LISTING 3

```
100 REM "LOADER"
110 REM DEMO MACHINE CODE LOADER PROGRAM
120 REM D. RUSSELL  10-01-83
130 CLEAR, &H7FFF                                  'FENCE BASIC OUT
140 SRNAME$ = "SUBROUT"                            'NAME OF SUBROUTINE
150 SRENTRY = &H8000                               'SUBROUTINE CALL LOCATION
160 REM
170 REM
180 REM ****************
190 REM SUBROUTINE LOADER
200 REM ****************
210 OPEN "I", #1, SRNAME$ + ".HEX"
220 INPUT #1, R$                                   'READ A RECORD
230 IF EOF(1) THEN GOTO 350                        'END OF FILE?
240 RECLEN = VAL("&H" + MID$(R$, 2, 2))            'NUMBER OF DATA BYTES IN RECORD
250 LAH = VAL("&H" + MID$(R$, 4, 2))               'FIRST
260 LAL = VAL("&H" + MID$(R$, 6, 2))               ' DATA BYTE
270 LOADADDR = 256*LAH + LAL                       ' LOAD ADDRESS
280 FOR BYTENUM = 1 TO RECLEN                      'SORT OUT
290 ADDR = LOADADDR + BYTENUM - 1                  ' EACH
300 POINTER = 2*BYTENUM + 8                        ' DATA BYTE
310 BYTE = VAL("&H" + MID$(R$, POINTER, 2))        ' AND
320 POKE ADDR, BYTE                                ' LOAD IT
330 NEXT BYTENUM                                   ' INTO MEMORY
340 GOTO 220                                       'GO READ ANOTHER RECORD
350 CLOSE #1                                       'ALL DONE LOADING
360 REM END OF LOADER
370 REM
380 REM
390 REM RUN SUBROUTINE
400 INPUT "DO IT NOW"; DUMMY$                      'ANY KEY MAKES IT HAPPEN
410 CALL SRENTRY
420 PRINT
430 PRINT "HOW ABOUT THAT!"                        'IT WORKED
440 END
```

αβγδεζηθικλμνξοπρστυφχψωΓΔΘΛΞΠΣΤΦΨΩ∞+

Turn WORDSTAR™ into a TECHNICAL WORD PROCESSOR by adding CHARTECH.
Use with most popular printers having dot-addressable graphics.

$$\Lambda = \sum_{n=1}^{\infty} \left[\Gamma_{\alpha\beta} + \Omega_{\gamma\nu}\right]$$

94 special characters printed.
New characters easily added.
Special characters displayed on the screen for some systems.
Screen formatting of equations.
All printer functions accessable.

Plain **CHARTECH** is $95, Enhanced version is $130
(Plain version displays characters resident in the terminal, enhanced version forms screen characters in software for certain systems.)
Special screen character ROMs available for some systems.
Presently available for 8-bit systems using CP/M™.
Ask your local dealer or order directly from

TECHWARE, 2510 Cresta de Ruta
Eugene, OR 97403, (503) 343-0566.

State computer, disk density, and release of WordStar.

αβγδεζηθικλμνξοπρστυφχψωΓΔΘΛΞΠΣΤΦΨΩ∞+

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

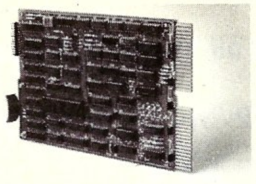
Hextran/Hexrecv™© Transposes binary files
- You can now download or upload executable files on to any communications network (i.e. The Source* Compuserve*

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.

Run Intel software on your LSI-11 or PDP-11 Minicomputer



DECMATION offers you our D100 processor card with extensive software which allows you to run 8080 programs which execute under ISIS or CP/M®.
- Executes Intel language processors 2-3 times faster.
- Runs Intel software from any terminal on multi-user systems.
- Uses all Dec peripherals.
- Interchanges files with your Intel development system using floppy disks or a serial link.
- Runs under RT-11, TSX+, RSX-11M and RSTS/E operating systems on your minicomputer.
- Runs CP/M® with its many software packages. We include an editor/word processor at no extra charge.
- Price: under $2500.

DECMATION
3375 Scott Blvd. #422 Santa Clara, Ca. 95051
(408) 980-1678

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

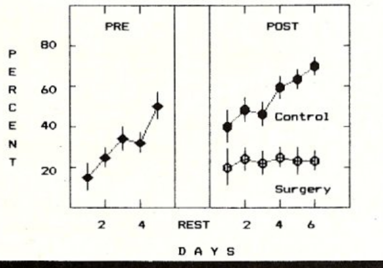
CIRCLE 250 ON READER SERVICE CARD

Publishable Graphs

on MANY DOT MATRIX PRINTERS

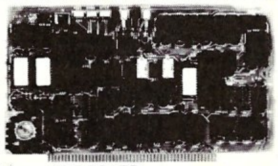
Requires no Graphics Screen
Line Graphs and Scatterplots
Easy to Use, No Programming
CP/M 80, CP/M 86, PC-DOS
Most disk formats.
Only $50+$3 shipping. Manual Alone: $10.

DataPlotter from Lark Software
7 Cedars Rd, Caldwell, N.J. 07006
(201) 226-7552 Visa, M/C



CIRCLE 249 ON READER SERVICE CARD

80 CHARACTER VIDEO BOARD – S-100



All This on ONE BOARD:
- Keyboard port with TYPE-AHEAD buffer
- 8275 CRT controller with light pen port
- Two 2716's – program & character rom's
- Optional 2716 for CHARACTER GRAPHICS
- All screen & keyboard ram
- SIMULTANEOUS I/O or Memory mapped
- Z-80 MPU – 2 or 4 Mhz system clock
- Easy to adapt Software
- Uses only EASY-TO-GET parts
- Use in any S-100 system
- 696 Bus Compliance: D8 M16 I8 T200
- Build for less than $200
- **!Now includes Crystal and Heat Sink, $9.85 value.**

Introducing The VDB-A
Bare board with Documentation **$49.50**
+ $2.50 s&h (Ill.res. add 6% tax)
Add 3% for MC & Visa

Simpliway PRODUCTS CO.
P.O. Box 601, Hoffman Estates, IL 60195
Dealer Inquiries Invited
312/359-7337

CIRCLE 252 ON READER SERVICE CARD

SIMPL-SIMON SOFTWARE **EZ-ASM**

Makes CP/M assembly language programming almost as easy as BASIC. *Automatically* selects from 99 subroutines.

SIMPL-SIMON SOFTWARE **CP/M.D. II**

Recover erased files, cure other CP/M disk problems easily. Dozens of functions.

For CP/M 2.2. Manuals included. Specify 8-inch SSSD or Superbrain SSDD.

Your choice: $39.95

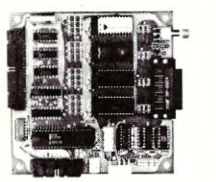
Or Both for $75

Add $3.00 per order for shipping (USA). Send check for prompt shipment. Georgia residents add 4 per cent tax. CP/M is a trademark of Digital Research.

TELEPRINT, INC.
PO Box 10-B, Sylvania, Ga. 30467

CIRCLE 251 ON READER SERVICE CARD

SIBEC 51



8051-Based Single-Board Computer with Monitor/Debugger

- Versatile - 4 28-pin byte-wide sockets; monitor will program EEPROMs.
- Affordable - just $335.
- Perfect for System Development and Educational Applications

Binary Technology
P.O. BOX A-59 • HANOVER, NH 03755 • 603/643-2881

Xerox Warranty
Service & Maintenance

Promotional Prices for COMPUTER PROFESSIONALS

We set up your sale/leasebacks.

CompuPro® SYSTEMS

MORROW DESIGNS *Tarbell*

PRINTERS, TERMINALS, SOFTWARE

Computer professionals, we want you as a customer, a resource and a reference. Because we sell as many computers by referral as directly from our ad, we want to expand our professional customer base. We are offering promotional prices **only** to knowledgeable users, for a short while.

TERMS: Cash with order. You pay all freight. Add 3% for 2% 10/Net 30 AAA1 or Fed./State P.O. 2%/mo. carrying chg.
CALL US 9 AM to 10 PM
205 879-5976
205 879-4735

CostPlus COMPUTERS
P.O. Box 6114 • Birmingham, AL 35259-6114
Our bank is 1st National of Birmingham, (205) 326-5120

CSE

CSE: A full-screen text editor written in C
- full-screen editing with complete cursor control, insert/overtype, single and multiple line insertion and deletion, forward and backward scrolling
- Powerful text buffer commands include find, find/replace, block move and copy, file inclusion, and nested macro commands
- Installation program allows easy customization for most popular terminals and configuration to use function and special editing keys
- Uses intelligent terminal features, such as insert line and delete line, to avoid unnecessary redisplay
- Available for CP/M-86[1], MP/M-86[1], CP/M 2.2[1], MS-DOS[2], IBM PC[3], (PC-DOS and CP/M-86)
- Requires 64K CP/M-86 or equivalent MP/M-86; 64K CP/M 2.2; 64K MS-DOS; 64K IBM PC-DOS; 64K IBM CP/M-86
- Includes object code, C source code, and manual
- Available in 8" SSSD format for CP/M-86, MP/M-86; CP/M 2.2, MS-DOS; 5" SSDD format for IBM PC-DOS and IBM CP/M-86
- **$60.00**, including UPS

SIM 80

SIM80: An 8080 simulator for the 8086/8088
- Run CP/M object code (.COM files) on any CP/M-86 or MP/M-86 system: ASM, DDT, dBase II[4], MBASIC[5], etc.
- Retain applications software when upgrading from CP/M to CP/M-86 or MP/M-86
- 8K overhead, TPA can be 61K
- 1/3 to 1/10 as fast as a 5 Mhz 8085 (**not recommended** for highly interactive programs such as Wordstar)
- Includes object code, ASM-86 source code, and manual
- Available in 8" SSSD format for CP/M-86, MP/M-86; 5" SSDD format for IBM PC (CP/M-86 only)
- **$50.00**, including UPS

Both CSE and SIM80 for $100.00

NMD Northwest Microsystem Design
P.O. Box 10853 • Eugene, OR 97401 • (503) 484-7129

[1]tm, Digital Research; [2]tm, Microsoft; [3]tm, IBM; [4]tm, Ashton-Tate; [5]tm, Micropro

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

WANT TO TRADE new unopened Wordstar, mailmerge, supercalc 2, D Base II, all CP/M 86 8" disks, calcstar CP/M80 8", for Pascal MT + CPM 86 8", Access Manager 86 8", and RAM 21. R. Morrison, P.O. Box 188, Cannon Beach, OR 97110. Phone (503) 436-1122.

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.

FREE CATALOG!

Just let us know and we'll mail you a FREE *Creative Computing* Catalog—16 pages filled with books, buyer's guides, magazines, and more!

To get your FREE catalog, write to: *Creative Computing* Catalog, Dept NA1X 39 East Hanover Ave., Morris Plains, NJ 07950.

LETTERS

Continued from page 139

Gentlemen,

The machine code loader for MBASIC-80 presented by Larry Costa and Steve Leibson (*Microsystems*, October 1983) is relatively fast when compared with the use of DATA and POKE statements, as described by the authors. By the use of a technique from the Microsoft Softcard BASIC Interpreter Reference Manual, an even greater increase in speed and in efficiency can be obtained.

Costa and Leibson make use of the ASC and MID$ function in a FOR-NEXT loop to pick out each individual byte of machine code in a 128-byte sector, and then POKE it into memory. The loop is repeated until every sector is loaded into memory. Instead of this byte-by-byte method, each sector of code can be placed into memory by the use of a single LSET statement. This is done by first setting up a dummy string, C$, of 128 bytes in length, and pointing it into the area reserved for machine code, via the VARPTR and POKE functions. To the MBASIC interpreter C$ now consists of the 128 bytes of memory where the sector of machine code is to reside. This sector of code is

Instead of using the byte-by-byte method, place each sector of code into memory via a single LSET statement.

then read from the disk, and placed into C$ by the LSET statement, which left justifies the string and so avoids dynamic reallocation of string space for C$. Then the pointers are updated to point to the next 128 bytes of memory, ready for the next sector of code.

In order to implement this method, line 30040 of Costa's and Leibson's MBASIC code should be replaced with the lines shown in Listing 1.

LISTING 1

```
30040 C$=T$: P%=VARPTR(C$)+1:             Set up 128-byte dummy string,
                                          and get location of address
                                          pointers.
      L%=&HOO: H%=&HB8                    Get low byte, high byte of
                                          base address of machine code.
30045 FOR I=1 to NS%:                     Loop through each sector of
                                          code.
      POKE P%,L%: POKE P%+1,H%:           Set the pointers of C$ to
                                          point to the machine code
                                          reserved area.
      GET#2,I: LSET C$=T$:                Get the code, and put it
                                          into C$.
      L%=(L% XOR 128): H%=H%-(L%=0):      Update the pointers to point
                                          to the area for the next
                                          sector.
      NEXT I                              Loop around for the next
                                          sector.
```

The speed of this technique really becomes evident when it is used to set up a large machine-language environment in high memory of Basic. I have used it to load an area of 4K in less than 6 seconds. It is a vesatile method, able to be adapted to save and/or move a block of memory quickly. The use to which it is put in the Microsoft manual is the loading and saving of high-resolution graphics screens.

Peter Tyler
Department of Clinical Chemistry
The Queen Elizabeth Hospital
Woodville, South Australia 5011

Dear Mr. Libes,

The article on "Floppy Problems" in the December '83 issue struck home with me, as it probably did with many S-100 readers. In particular, the editorial comments at the end of the article were most interesting.

However, they presented only half the problem. The gist was that an S-100 board manufacturer does not (and should not have to) provide suport for system configurations using its board other than the systems for which the manufacturers made the board.

Consider the poor customer. S-100 board manufacturers rarely put in their ads or catalogs all the 'fine print' which describes the gotchas. My favorite horse which, I think, richly deserves flogging is the Texas company that advertised its disk controller in stand-alone ads. Naive purchasers like me found to their dismay that they faced a considerable dilemma. Either there were many other purchases required from the same company because all the software was tied together (key portions of the BIOS coming with each of several boards), or there was considerable programming to be done to bring the controller up. The company involved was monumentally uncooperative. After a considerable period of time, I disposed of the board on principle, preferring the products and approach of a more honest company.

I think a prospective purchaser seeing an ad unfettered by restricitons is entitled to expect that the board can be used in a variety of systems of the customer's choosing. Details to facilitate the user's reasonably expected needs should be included. This certainly doesn't mean that the manufacturer itself has to go through all the effort of interfacing with every board and every conceivable combinaiton of products; there are many capable customers buying stand-alone S-100 boards who will be glad to accept that challenge, given sufficient information on the product in the manual.

I guess the solution is to ask manufacturers to be more honest and forthcoming in their ads, and to give us customers a fair shake when preparing a manual. Let's be frank before purchase about the limitations and restrictions as well as the benefits. Most of us will be glad to share our unusual applications and solutions with the manufacturer (user-group style), providing the manufacturer doesn't unfairly commercialize our work.

Joseph Reymann
953 Avenida Ladera
San Dimas, CA 91773

Dear Mr. Libes,

Ian Ashdown's XERA (August 1983, page 54) is a lovely program and I am glad to have it—but alas for those "six hours of file reconstuction" which prompted it! And alas too for the hours it cost him to write it! The easy solution for his inadvertent erasures was to load DUU (CP/MUG Vol.71), go to Track 3 and replace all those E5 bytes with 00. Simple.

R. W. Odlin
2125 Hoogdal Rd.
Sedro-Woolley, WA 98284

ADVERTISERS INDEX

Four Answers To Your S-100, Multi-User Problems.

Intercontinental Micro Systems makes everything you need for S-100 bus multi-user systems, networks or single user systems.

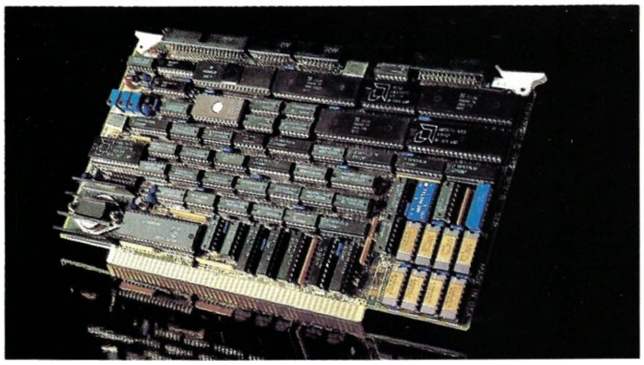
At a price that won't break your budget.

Quite simply, our single board computers, slaves, 256K memories and personality boards let you build a system *now,* not later. The hardware works, the software works, and the prices are what you'd expect from a company that uses the most advanced design, software and production techniques to keep costs down.

What you won't expect is the almost awesome sophistication of Intercontinental Micro System's products.

So stop messing around with multiple sourcing, hardware integration problems and software nightmares. Come to Intercontinental Micro and get it all — price, performance and delivery.

Read the specs, then call, write or circle the bingo number below. We'd be glad to send more information and help solve your S-100, multi-user system problems.

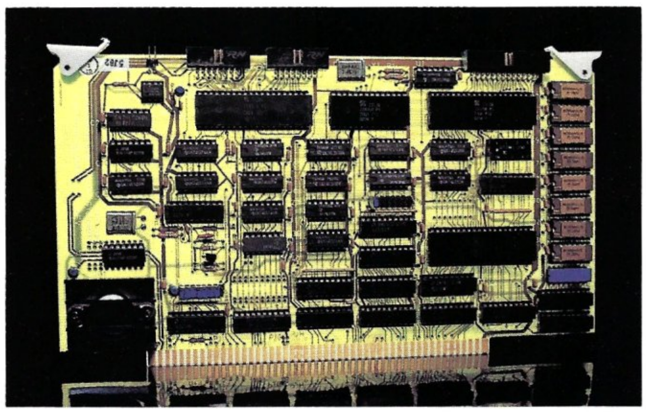


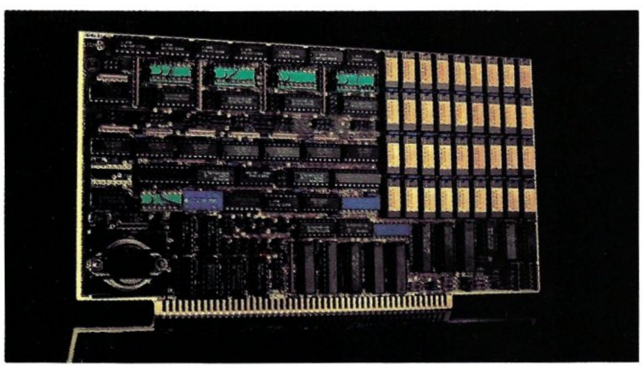
1 CPZ-48000 SINGLE BOARD COMPUTER.

□ IEEE 696.1/D2 S-100 compliance. □ Z80A,™ 4MHz Operation. □ Floppy disk controller (FDC). Single or double sided. Single or double density. 8" or 5¼". □ Two synchronous or asynchronous serial I/O channels (SIO). □ Two parallel I/O channels (PIO). □ Four channel DMA controller. □ 64K on board RAM. □ Memory management unit (MMU). Addresses up to 16 megabytes of system memory. □ Eight Vectored priority interrupts. □ Provisions for 2K or 4K onboard EPROM. □ Software selectable baud rates. □ IBM Bisync, HDLC, SDLC and other protocols. □ CP/M,™ MP/M,™ and TurboDOS™ operating systems available. □ Turbo-Disk® implementation included.

2 CPX-MX SLAVES.

□ IEEE 696.1/D2 S-100 compliance. □ Compatible with CPZ-48000 SBCP, any Z-80A based CPU with extended address capability or 16 bit based CPUs complying with IEEE 696.1/D2 bus specification. □ Z-80B™ 6MHz (CPS-6X) or Z80A 4MHz (CPS-4X) operation. □ Two synchronous (CPS-MS) or asynchronous (CPS-MA) serial I/O ports. □ TurboDOS™ & CP/NET™ compatible. □ Master confiscation of slave memory for diagnostic purposes. □ Two parallel I/O ports; eight data bits + 2 handshake lines per port. □ 64 Kbytes of onboard dynamic RAM. □ Master/slave memory-to-memory transfers under DMA control @ 571 Kbyte/sec transfer rate when used with CPZ-48000 SBCP. □ Software selectable baud rates. □ Usable as an intelligent I/O processor in single user system.



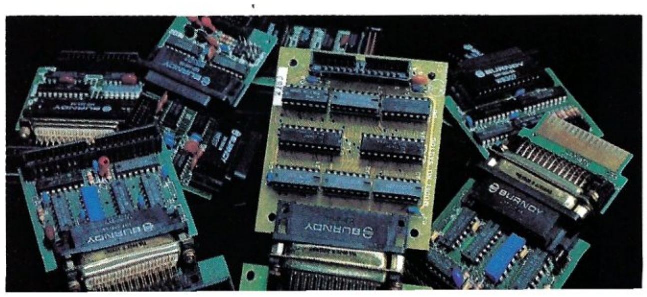


3 256KMB-100 256K MEMORY.

□ IEEE S-100 bus, spec 696.1/D2 compliance. The 256KMB-100 is compatible with most IEEE S-100 board products now on the market. □ Linear addressable to 2 megabytes. □ 225 nanosecond access time, maximum, 160 nano-seconds, typical. □ 295 nano- second read-write time, minimum. □ Bank selectable 16K increments. □ I/O port address bank selection. □ Configures for phantom deselection. □ Parity error detection, visual and/or interrupts. □ Bank selection compatible with CROMIX,™ CP/M2.2,™ MP/M,™ Alpha Micro, and other major systems.

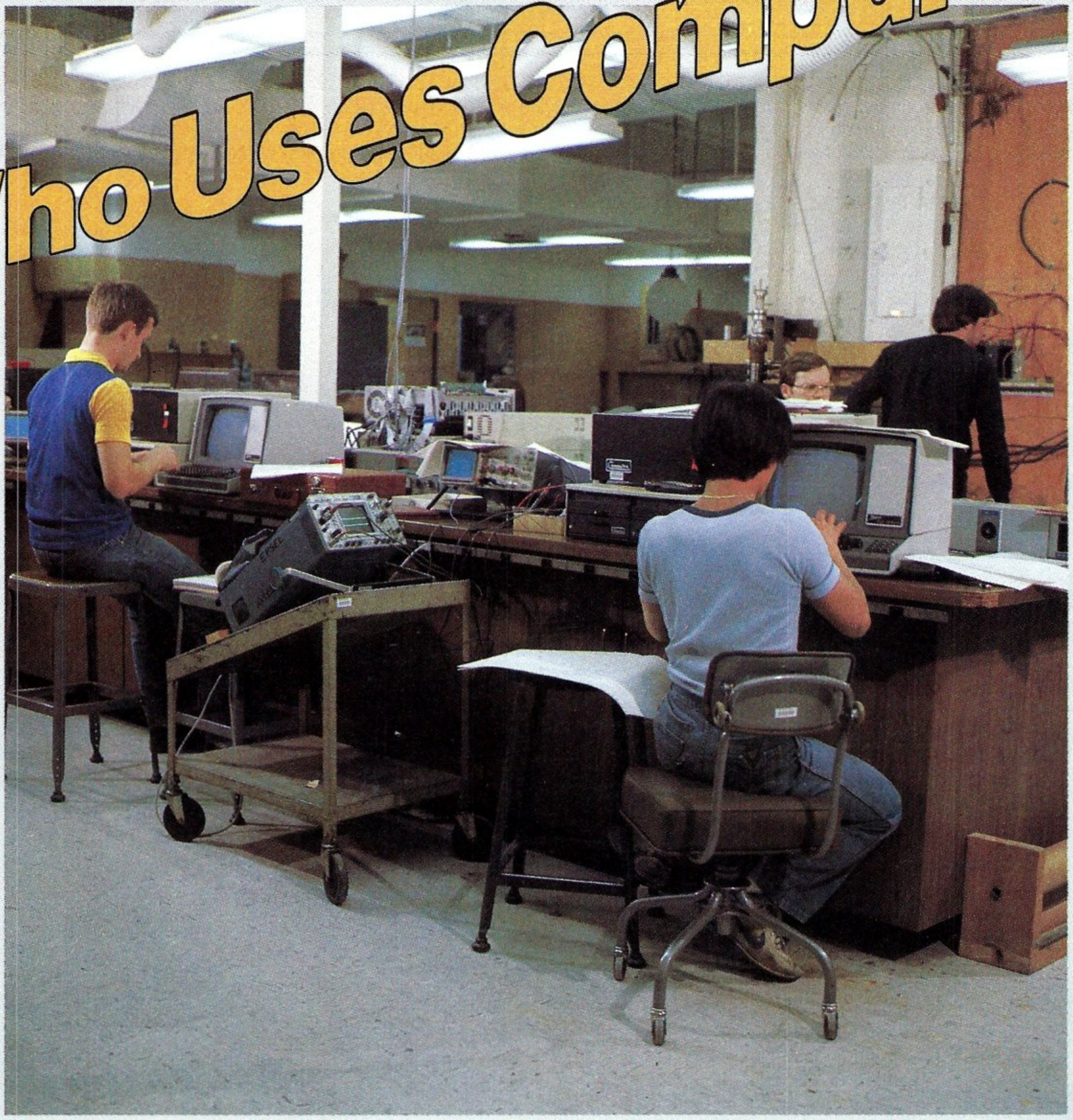
4 PERSONALITY BOARDS.

□ Centronics printer. □ 8 inch floppy disk. □ 5¼ inch floppy disk. □ RS232 serial communications. □ Synchronous/asynchronous modem. □ Priam smart/smart E hard disk. □ Long distance serial communication (2000 ft @ 9600 baud). □ Shugart Associates Systems Interface (SASI). □ Clock/calendar. □ Koṇan David, Jr.™ hard disk. □ Archive tape drive.



4015 Leaverton Court

Who Uses CompuPro?



©1983 CompuPro

In a laboratory at the Massachusetts Institute of Technology, students in computer-assisted engineering courses rely on a network of **CompuPro** systems.

Dr. Richard Thornton and Dr. Jeffrey Lang teach future engineers to use computers for realtime control of "everything in principle from automobiles to a nuclear power plant to a wind turbine to heat pumps," he says.

Several factors persuaded Dr. Thornton to put **CompuPro** in his lab. Associates who use **CompuPro** systems recommended them as very fast and highly reliable, which was an important consideration because the lab's machines rarely rest. Also, other equipment in the lab relies on the same S-100 bus that **CompuPro** uses, which makes interfacing easier.

"And it has the flexibility too," Prof. Thornton says. "If we want to change processors or add more memory, we can do that."

To students or staffers, **CompuPro** delivers performance, quality and reliability. Call **(415) 786-0909** ext. 206 for the location of the **Full Service CompuPro System Center** nearest you.

CompuPro®