THE JOURNAL FOR ADVANCED MICROCOMPUTING

$3.95

Microsystems

Volume 5/Number 1

January 1984

MP/M II is a versatile multiuser multitasking system. And there are ways to make it even better!

MP/M II in the user's hands

Part 1 of Tom Clodfelter's series on enhancing MP/M surveys the basic features, and shows how to fix a few weaknesses. Later parts will discuss powerful new features that can be added.

David Hardy and Ken Jackson show you how to get started with MP/M II.

Bill Wong describes a means of adding concurrency to the MP/M II master console.

Software Reviews

Bruce Hunter reviews MP/M 8-16, the fast MP/M that runs on CompuPro's 8/16 system and gives multiple users the best of the 8- and 16-bit worlds.

Leonard Schwab reviews the latest version of CIS Cobol, a British implementation of Cobol for 8-bit micros.

Hardware Reviews

David Hardy and Ken Jackson review the ProComp 8, a compact S-100 machine running CP/M 2.2 on a Teletek CPU board. The special BIOS makes this one of the fastest Z80 machines to be found.

Eric Beser reviews a useful S-100 graphics board from paraGraphics that has sprite capability.

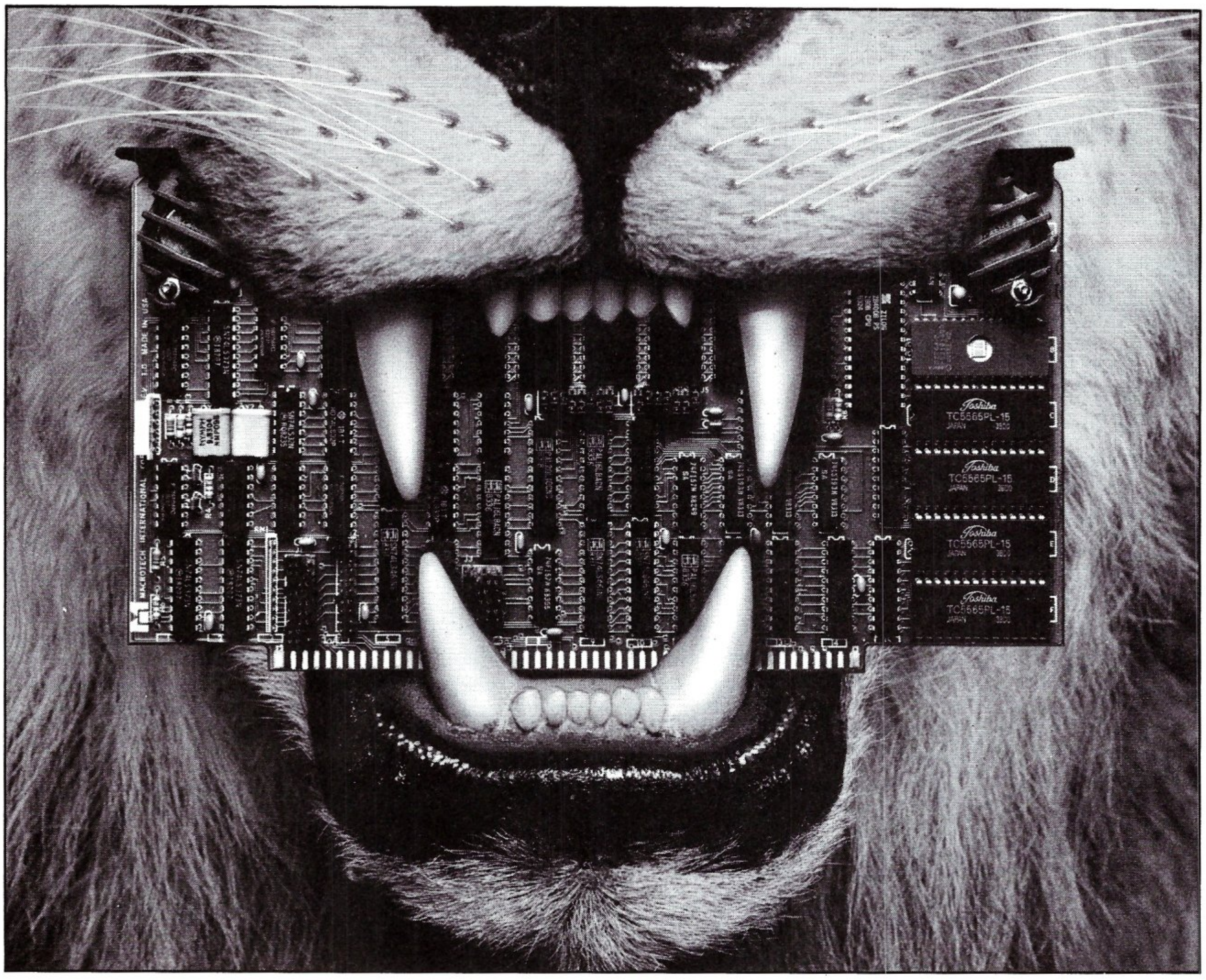
Tutorials

Part 3 of Andrew Bender's series on relocating assemblers discusses macro libraries and libraries for the linker.

Bill Wong provides a brief tutorial on Prolog, combining this with a review of micro-Prolog, a British implementation of the language for CP/M.

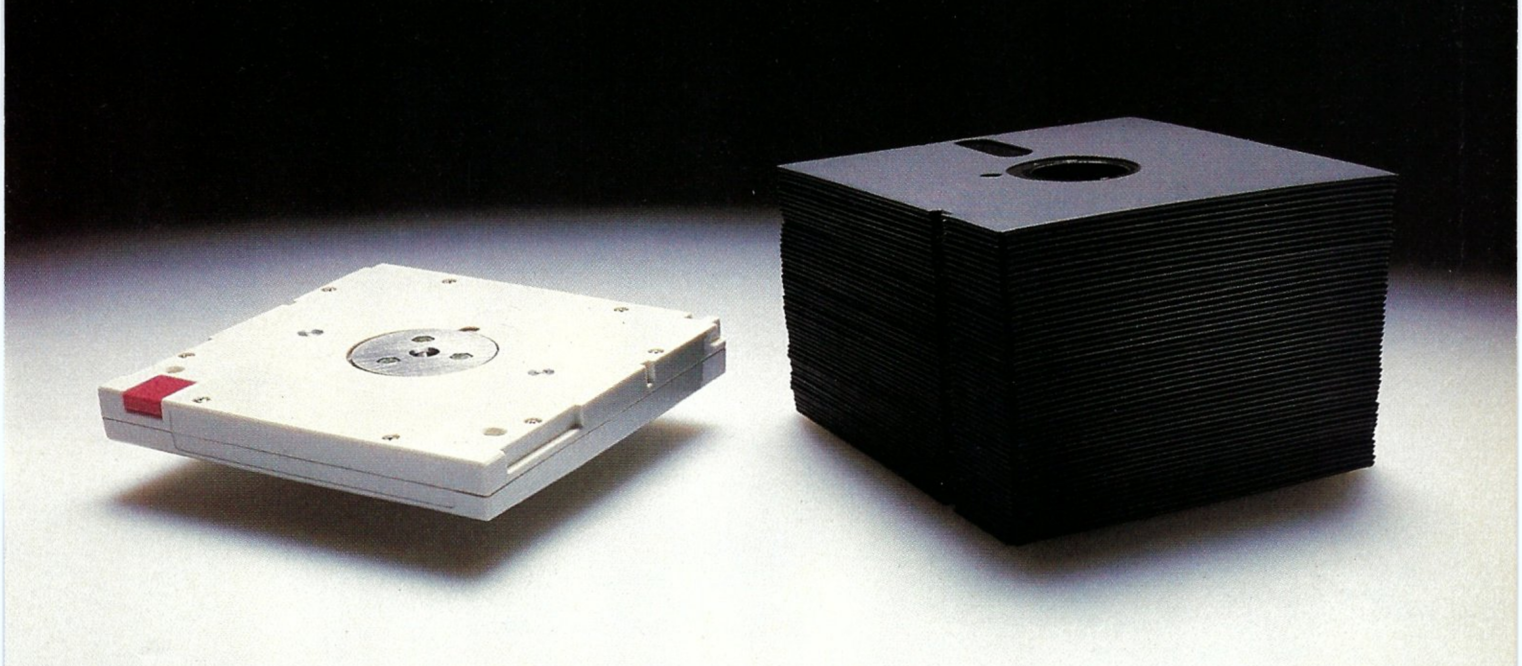
01

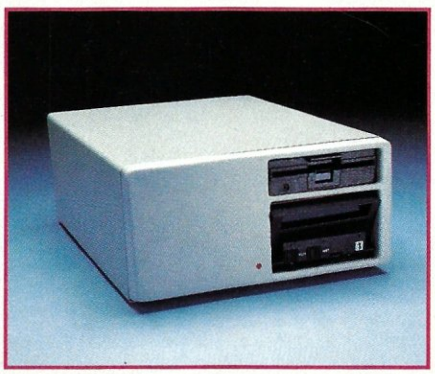
14024 14352



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

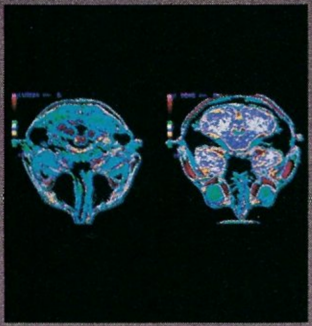
*CP/M is a trademark of Digital Research Corp.
TurboDOS is a trademark of Software 2000 Inc.

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

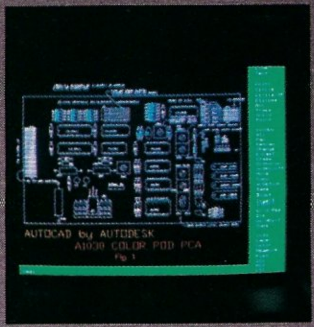
For European Distributor, contact: (714) 891-4004

CIRCLE 148 ON READER SERVICE CARD

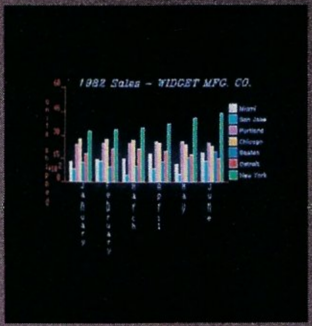
GDL's A-1000™
Why Pay More For Less?



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



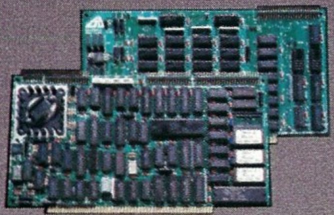
Autocad by Autodesk
R. H. Hymes, GDL



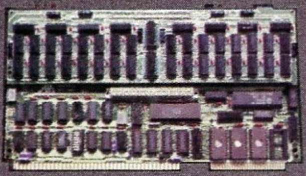
Supercalc Data Graph
Pacific Basin Graphics

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. Also **Hershey** character sets are available.
PBG100 Library	A **Calcomp** compatible subroutine library.
The Analyst	Stock market analysis.

OEM's, Dealers and Systems Integrators

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

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

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

REPRESENTATIVES

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

MOMS COMPUTING
Bldg. 105-5, Fort Cronkite
Sausalito, Ca. 94965
(415) 331-2043

DEALERS

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

MARTIAN TECHNOLOGIES

MOMS COMPUTING

TRACK COMPUTER CENTER
2100 Broadway
Oakland, Ca. 94612
(415) 444-8725

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

MENDOCINO MICROCOMPUTERS INC.
32191 Albion Ridge Rd.
Albion, Ca. 95410
(707) 937-5001

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

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

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

CIRCLE 24 ON READER SERVICE CARD

Contents

Microsystems

Volume 5/Number 1
January 1984

Microsystems

STAFF

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

ADVERTISING SALES OFFICES

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

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

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

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

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

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

Canada
Frank Lederer, *The Pattis Group*
501 Eglinton Ave., E. #202
Toronto, Ontario
M4P 1N4

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

ZIFF-DAVIS PUBLISHING

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

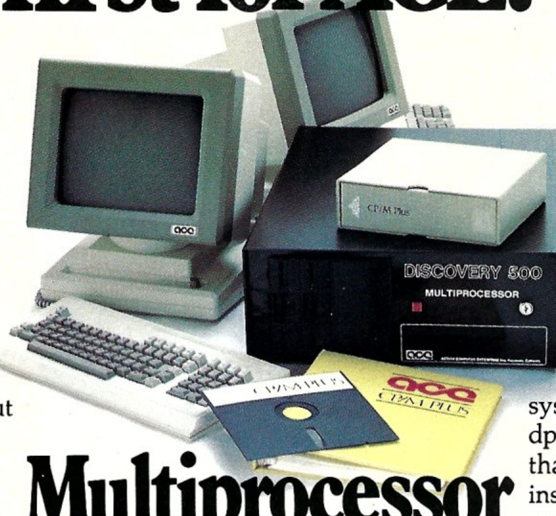
MICROSYSTEMS (ISSN #0199-7955) is published monthly by Ziff-Davis Publishing Company, One Park Avenue, New York, N.Y. 10016.

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

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

For information on commercial advertising, write to: MICROSYSTEMS Advertising Dept., One Park Avenue, New York, NY 10016 or call Jim Beloff at (212) 725-3452.

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.)**

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

CIRCLE 197 ON READER SERVICE CARD

ACTION COMPUTER ENTERPRISE

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

Editor's Page

by Sol Libes

Until now, the IBM formatted floppy 8″ disk has been the only standard disk format in existence. In the 5.25″ floppy disk world an incredible number of different disk formats were used. This created serious problems for software vendors and for public domain software groups seeking to distribute software on disk.

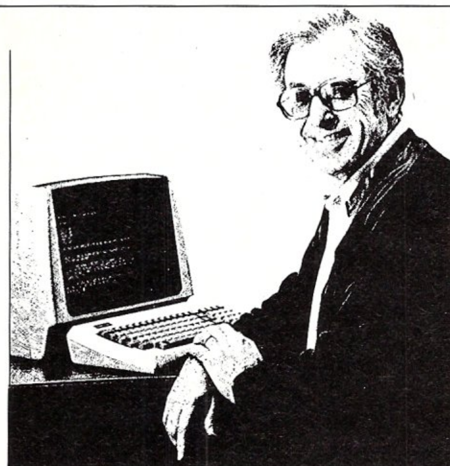
Now a standard appears to be developing in the 5.25″ arena: the format used by IBM for their personal computer. This format allows the storing of 160K on a single-sided disk. The IBM PC, using the new version 2 of the PC-DOS operating system, can in fact store either 320K or 380K on a double-sided floppy disk.

This disk format is now being used by virtually all the manufacturers of IBM PC look-alikes, and several of the non-IBM formatted disks. Thus manufacturers such as Morrow and Osborne now provide utility programs that allow a file to be transferred from an IBM PC formatted disk to the formats used on their machines. And there are similar programs, from other vendors, for machines such as the Kaypro.

Also, there is a trend among software vendors to stop distribution of software on 8″ disks. For example, Digital Research has already ceased distributing CP/M-86 software on 8″ disks and now distributes it only on 5.25″ IBM PC formatted disks. There is no doubt that this trend in increasing and that within another year or two few, if any, software vendors will be distributing software on 8″ disks.

Apple Computer has already announced that they will provide a plug-in board and software to permit the loading and running of MS-DOS software on their Lisa computer. This is a new departure for Apple. In the past, they always tried to make their systems distinctive from everything else in the marketplace, and Apple formatted disks were different from everyone else's. But now even they are responding to the demands of the marketplace.

The likelihood is that, when IBM introduces a new personal computer with a drive smaller than 5.25″, this



will set a new standard in the microdisk drive area. Although Sony has achieved some success in the microdrive market, there is still a wide variety of sizes. Most manufacturers of notebook-size computers are waiting to see what IBM does before deciding on what size drive they are going to use in their machines. I know of one notebook-size computer so designed that it can accept any size of drive from 3″ to 5.25″ without retooling.

There is no doubt that, in the 16-bit world, the MS-DOS operating system is achieving much wider acceptance than CP/M-86. If I were to hazard a guess, it would be that about 90% of the 8086/8088-based 16-bit microcomputer users are running MS-DOS or PC-DOS. Many system manufacturers, particularly the Japanese, are hedging their bets by providing both MS-DOS and CP/M-86 on their systems.

There is already more software available to run under MS-DOS than under CP/M-86. This is true both in the commercial and public domain software areas. Incidentally, *Microsystems* will start reporting on MS-DOS public domain software, just as we have been reporting on CP/M public domain software.

However, a problem exists here to which I must call your attention. In most instances, system manufacturers who provide MS-DOS with their systems are claiming that their systems are "IBM PC compatible." In all too many cases this compatibility is severely limited. For example, Digital Research advertises that all their language products are now available in "IBM PC compatible" form. However, on certain IBM PC look-alikes, they will not run properly. *The point here is that running MS-DOS on a computer does not necessarily make that computer PC-DOS compatible.*

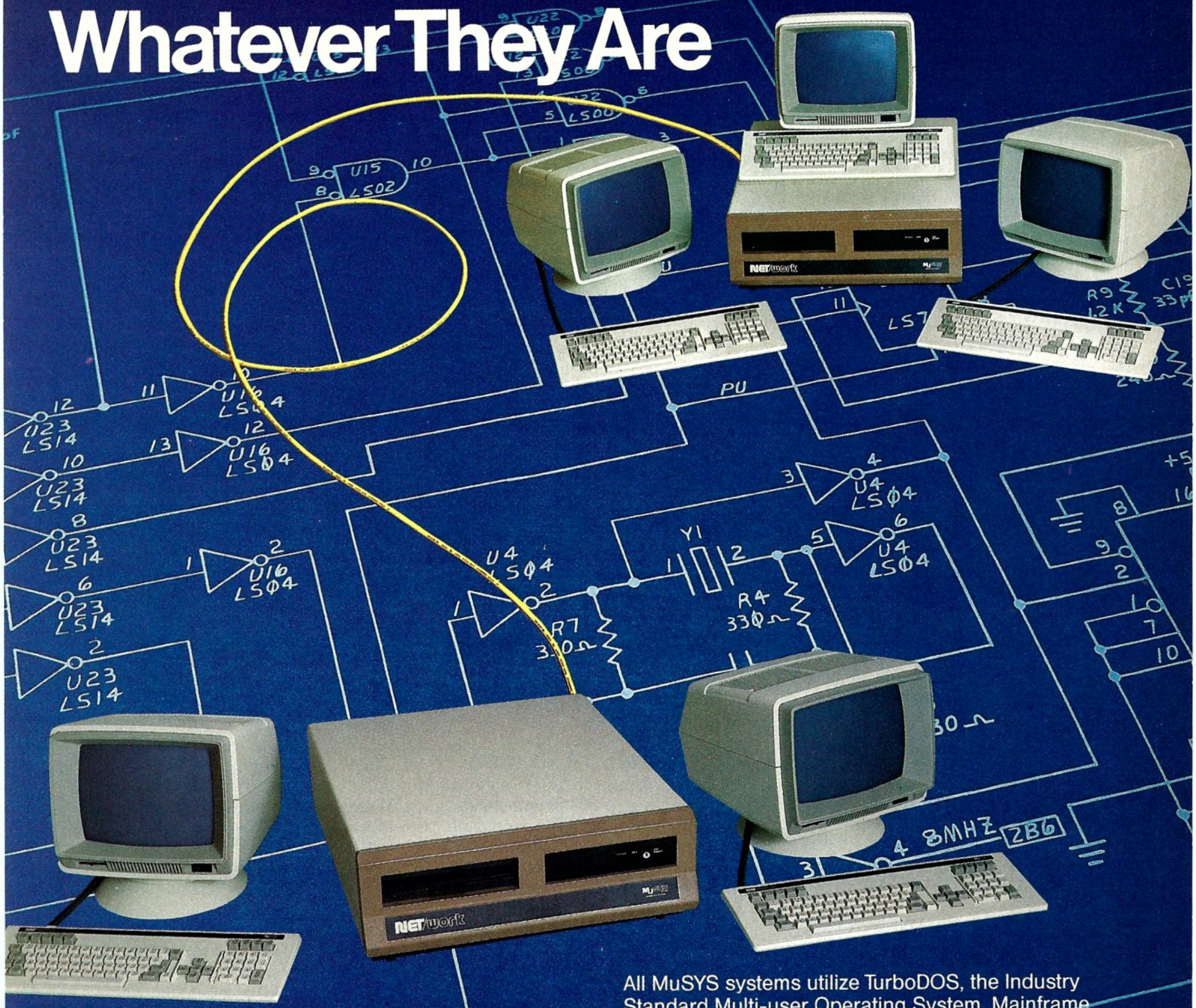
The problem of MS-DOS and PC-DOS compatibility appears due to the fact that IBM uses a bit-mapped display with ROM routines that they have copyrighted. A manufacturer cannot simply furnish a copy of the IBM PC ROM program in his machine. Many software vendors, in their software, use the distinctive features of the IBM PC screen and, further, often make direct calls to routines in the PC's ROM.

Equipment manufacturers, however, are now coming to realize that they have to take these things into account in designing their equipment to have better IBM PC compatibility, so this situation is expected to improve. Also, software suppliers are going to have to learn to use the proper entry points in the DOS, instead of direct calls to ROM routines, if they wish to have their software run on a wide variety of systems. Unfortunately, the software suppliers seem much less likely to change their methods than the hardware manufacturers.

A new subtitle

You will have noticed that our subtitle has changed from "The CP/M User's Journal" to "The Journal for Advanced Microcomputing." Rest assured that this does *not* mean we are abandoning CP/M users or, for that matter, S-100 users. Until recently, serious users with moderate budgets turned almost automatically to S-100 hardware (because of its flexibility) and to CP/M-80 (because of the huge base of available software). But as manufacturers transfer the techniques learned from the S-100 to more compact machines, and as MS-DOS, UNIX (in various flavors), and other operating systems become more prominent, we want to give these users, too, practical help at the systems level—and to let them know it is available in *Microsystems*. We are enlarging our scope, not abandoning old friends!

It Meets Your Needs...
Whatever They Are



Large Multi-user Capability

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

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

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

Dealer and OEM inquiries welcome.

MuSYS CORP

We design the future.

CIRCLE 83 ON READER SERVICE CARD

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

News and Views

by Sol Libes

Random rumors & gossip

Sources report that IBM has delayed introduction of its local area networking system because of problems with the interface chips being furnished by Texas Instruments... there is speculation that Wang Labs may be readying a dictionary on optical disk since it has purchased exclusive electronic publishing rights to the *Random House Dictionary, Concise Oxford Dictionary of English, Roget's International Thesaurus, Black's Law Dictionary* and the *Chicago Manual of Style*.... IBM is also expected to introduce a new multifunction workstation using the Motorola 68000 with an Intel microprocessor working as a coprocessor...

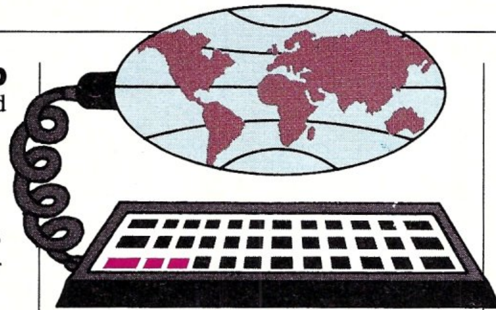
370 on your desktop?

Last year IBM disclosed that a group of their engineers had developed an experimental microcomputer system using the Motorola 68000 microprocessor that executed the IBM 370 mainframe instruction set. Now there are rumors that IBM plans to introduce a desktop computer based on this work, possibly this summer. The rumors are that the machine is all ready for production but that the introduction is being held a back to a point in time when it will not adversely effect the sales either of IBM's 4300-type mainframes or of the PC and XT.

The top 10 micro makers

Future Computing Inc., a marketing research firm in Richardson Texas, has released its chart of the top 10 personal computer manufacturers in the U.S. and their anticipated personal computer sales (in millions) for 1983 (including hardware and software). It is as follows:

IBM	$1,400
Apple Computer	1,100
Radio Shack	1,100
Commodore	800
Hewlett-Packard	550
Texas Instruments	450
Atari	350
Digital Equipment Corp.	200
Victor	150
TeleVideo	100



1983 represents the first time that any company has done more than $1 billion in personal computer sales...and last year three companies passed the mark. Total retail sales for personal computer hardware and software for 1983 is expected to exceed $10 billion. It is also interesting to note that three of the top 10 are newcomers to the microcomputer field, having introduced their first machines last year. They are DEC, Victor, and TeleVideo. This year is expected to bring even greater success for these companies, as it has been estimated that fewer than 7% of U.S. office workers currently have personal computers and less than 10% of U.S. homes have home computers.

Intel delays 186 & 286

Intel Corp. is reportedly having trouble delivering the 80186 and 80286 top-of-the-line microprocessors, and this is expected to delay the introduction of the more powerful versions of the PC/XT expected from IBM. Intel has acknowledged that early shipments of these units contained a microcode bug that rendered the units using these chips incompatible with the PC/XT, which uses the 8088 chip.

Intel reports that they have encountered a much higher demand for these parts than they had anticipated. This is expected to cause shortages in the marketplace for some time to come. Advanced Micro Devices, which is expected to second source the chips, is still some time away from production.

Intel is reportedly also encountering problems in making 286 chips that run above 4 MHz. Thus, 6 MHz units are in very short supply, and the 8 MHz are scarce. Intel is attempting to ramp up production to 100,000 286s a month.

IBM reports record profit increase

IBM reported that third quarter '83 net income rose 25% to $1.3 billion. According to John Opel, IBM chairman, IBM sold 850,000 PC/XT units in '83 and expects to sell about 2 million this year. Personal computer sales accounted for 2-4% of IBM's sales this year and are expected to rise to 6%.

Incidentally, IBM's income from interest on bank deposits and securities tripled to $232 from $80 a year earlier. In other words, IBM has a lot of cash sitting in the bank, and is therefore looking to spend it on increasing its outside holdings in suppliers (e.g., Intel).

The battle of the operating systems

According to John Rowley, president of Digital Research, Digital Research currently has about 900 contracts with OEMs for CP/M (including both 8-bit and 16-bit systems). Current industry estimates are that Microsoft has about 200 OEM contracts for MS-DOS. Further, Digital Research boasts that CP/M is now running on over 1.5 million computers world-wide. World-wide estimates of systems running MS-DOS are less than half this number.

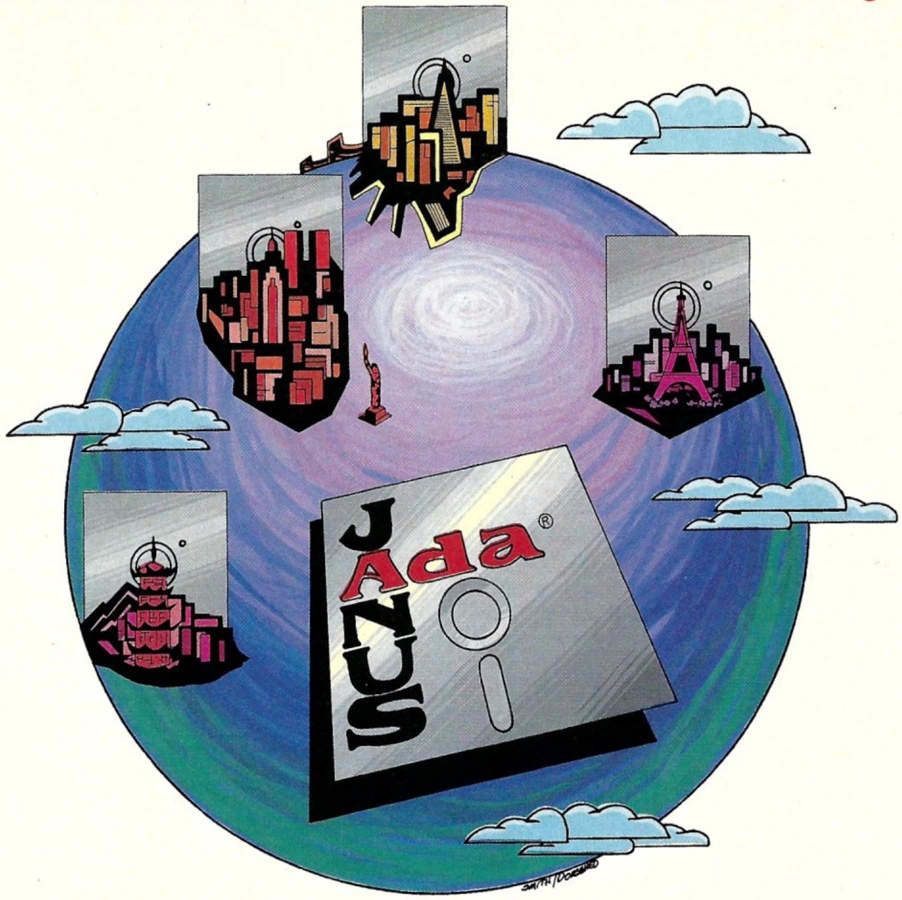
Digital Research is also known to be developing a new version of Concurrent CP/M-86, upgrading it to have many of the features of CP/M-80+, windowing, and the ability to run MS-DOS software. In the meantime, version 3.0 of MS-DOS, which Microsoft had promised to start shipping last fall, has been delayed. The new version of MS-DOS is expected to have compatibility with Microsoft's Xenix multiuser system, and concurrency (à la Concurrent CP/M-86).

In the meantime Digital Research has introduced CP/M-80 on a chip, integrating the operating system and processor onto one chip. This is expected to be used by manufacturers of low-cost home computers.

User group news

Some new user groups have come into existence. They are:

Our Products Get Used...Everywhere



In Business . . .

Toshiba
Sony
General Electric
Union Carbide
ARCO
Hazeltine
Westinghouse
Harris
Grumman
McDonnell Douglas
Chase Manhattan
Honeywell
Poloroid
Magnavox
ITT
Boeing
Lockheed
Xerox
TRW
Raytheon
Sorcim
Quantum
Intel
Sperry Univac
SofTech
E.I. DuPont
IBM
Hughes Aircraft

In Government . .

NASA
Argonne Labs
Jet Propulsion Labs
U.S. Army
U.S. Air Force
U.S. Navy
Naval Postgraduate School
U.S. Department of Commerce
U.S. Department of Treasury
Social Security Administration
FAA
Depart. of Energy, Canada
Depart. of Transportation, Canada
Depart. of Publications, Australia

In Education . . .

Harvard University
Georgetown University
University of Chicago
California Institute of Technology
John Hopkins University
University of Southern California
University of Massachusetts
University of Wisconsin
U.S. Naval Academy
Rochester Institute of Technology
University of Iceland
University of Leicester
University of Trondhiem
University of Zimbabwe

THE MOST COST EFFECTIVE ADA* DEVELOPMENT TOOLS AVAILABLE ON MICROCOMPUTERS!

Our Ada product line is available on the following Operating Systems: CP/M, CP/M-86, PC-DOS and MS-DOS.

Available from the following distributors:

Nationally

Suntex Data
10175 Harwin Suite 100
Houston, TX 77036
(713) 271-9191

CompuView Products
1955 Pauline Blvd., Suite 200
Ann Arbor, MI 48103
(313) 996-1299

Westico, Inc.
25 Van Zant St.
Norwalk, CT 06855
(203) 853-6880

Marfam Corporation
5340 Thornwood Dr. Suite 102
San Jose, CA 95123
(408) 226-0170

Internationally

Nord-Micro Informatique
155, rue du Fauburg
Denis
75010 Paris
(1)205-39-47

Micronix
11 Blackmore St.
Windsor 4030
QLD. Australia
(07)57 9152

S-100 Systems
3687-4 Shobumachi
Obayashi
Saitama, Japan 346-01
04808 (5)0416:
04808 (5)6565

CP/M, CP/M-86, CCP/M-86 are trademarks of Digital Research, Inc.
*ADA is a trademark of the U.S. Department of Defense
MS-DOS is a trademark of Microsoft

RR SOFTWARE, INC. *specialists in state of the art programming*

P.O. Box 1512 Madison, Wisconsin 53701
(608) 244-6436

CIRCLE 190 ON READER SERVICE CARD

- **Unigroup of New York Inc.,** *Box 1931, New York NY 10016.* Membership is $25/yr and entitles you to a quarterly newsletter, local UUCP directory, and a directory of local companies offering UNIX products.
- **CP/M-SIG of Cincinnati Computer Club,** *c/o Ric Allan, 799 Converse Drive, Cincinnati OH 45240.*

New public domain software

SIG/M (Special Interest Group for Microcomputers, Amateur Computer Group of New Jersey, Inc.) has issued nine new volumes of public domain software. They are:

Vol.	Description
137	Building Energy Design Analysis—California Energy
138	Commission (2 vols.)
139	Modem 712 update and misc. utilities
140	Cross Assemblers (6800/68000) and misc. utilities
141	Utilities for Kaypro, Osborne & Big Board
142	Games (incl. Adventure & games for Kaypro & Osborne)
143	Packet Radio program and printer utilities
144	Miscellaneous utilities
145	VFILER (Screen oriented File Manipulator) and misc. utils.

For complete information about SIGM software, send $2 ($2.50 foreign) for printed catalog to: **SIG/M,** *Box 97, Iselin NJ 08830.*

P/D Software, *4691 Dundas Street West, Islington Ontario, Canada M9A 1A7* (tel: (416) 239-2835) is distributing the CPMUG and SIG/M software libraries' 25 different disk formats including 8I, Apple, Kaypro, Osborne, IBM and others. Prices range from $10-$20 depending on disks required.

The New York Amateur Computer Club, Box 106, Church Street Station, NY NY 10008, has released 38 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 volumes are:

Vol.	Description
1	Misc. utilities
2-5	Microsoft Basic programs
6	General Ledger & Startrek programs
7*	RatFor
8*	Misc. Fortran programs and portraits
9*	EBasic Compiler/ Interpreter & programs
10-12	EBasic programs
13*	Original adventure game
14*	Pascal Pascal Compiler
15*	Algol-M
16-17*	Misc. Utilities from SIG/M
18	PC-Talk (modem communications), Pacman & notes
19	PC-File database program
20	RBBS system & utilities
21	Cross Reference and Monitor Utilities
22	Expanded Lister Utility for Basic
23	RATBAS and misc. utilities
24-25	Basic Games
26	8087 sampler programs
27-28	Kermit (PC/mainframe

If you're still looking for a heavyweight performer at a low price, Now, IBC has no competition

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 and it is, 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 20 ON READER SERVICE CARD

News and Views
continued. . .

	communications (Columbia Univ.)
29-30	Misc. Basic Programs
31-32	PC-TALK III (supercedes Vol 18)
33	Misc. application programs
34	DBase & Supercalc templates
35	Misc. Basic utilities
36	IBM keyboard drill system
37	Modem7, program control system, WordStar mods, misc. utilities
38	Disk Directory util., Squish and Lotus 123 mods

* = requires Z80 card
volumes 2-17 taken from CPMUG and SIG/M libraries.

Publications of note

dNOTES, the International dBASE II User's Journal, a 28-page bimonthly newsletter is $44 for 12 issues (includes indexed binder) and is now being published. For information call or write: J.D. Graham, c/o I:D:E:A Industries, Inc., Box 86, Deerfield IL 60015; (312) 940-1010 or (800) DBS-IDEA.

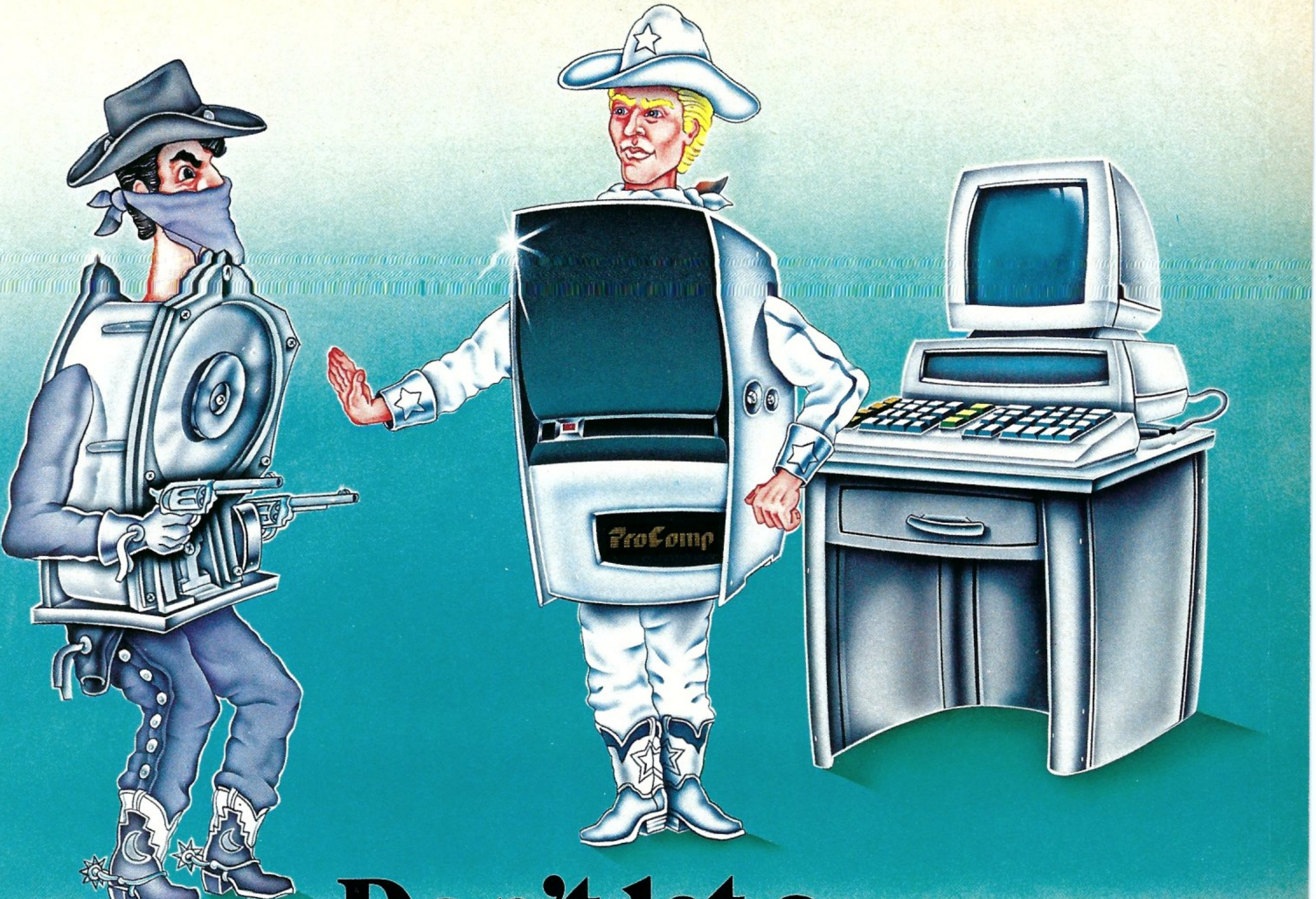
Random news bits

The Department of Defense has given its first official sanction for an ADA compiler for microcomputers. It is the Ada compiler developed by Western Digital Corporation, Irvine CA which runs on the Western Digital 1600 series of systems.... According to Professor Seymour Papert, developer of Logo, "Digital Research is writing CP/M-based software in Logo"....

Advanced Matrix Technology, Newbury Park CA, has announced a letter-quality printer which prints in four colors and can also print on acetate film for overhead transparency projection.

Quotation of the month

"We introduced the 8088 as an 8-bit microprocessor because although the internal architecture is 16-bits the I/O bus interface is that of the 8085. It was not until IBM introduced the IBM PC that we discovered it really was a 16-bit microprocessor. We could tell it because we read it in all the ads in the airline magazines."—David House, Vice President & General Manager, Microcomputer Group, Intel Corporation.



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

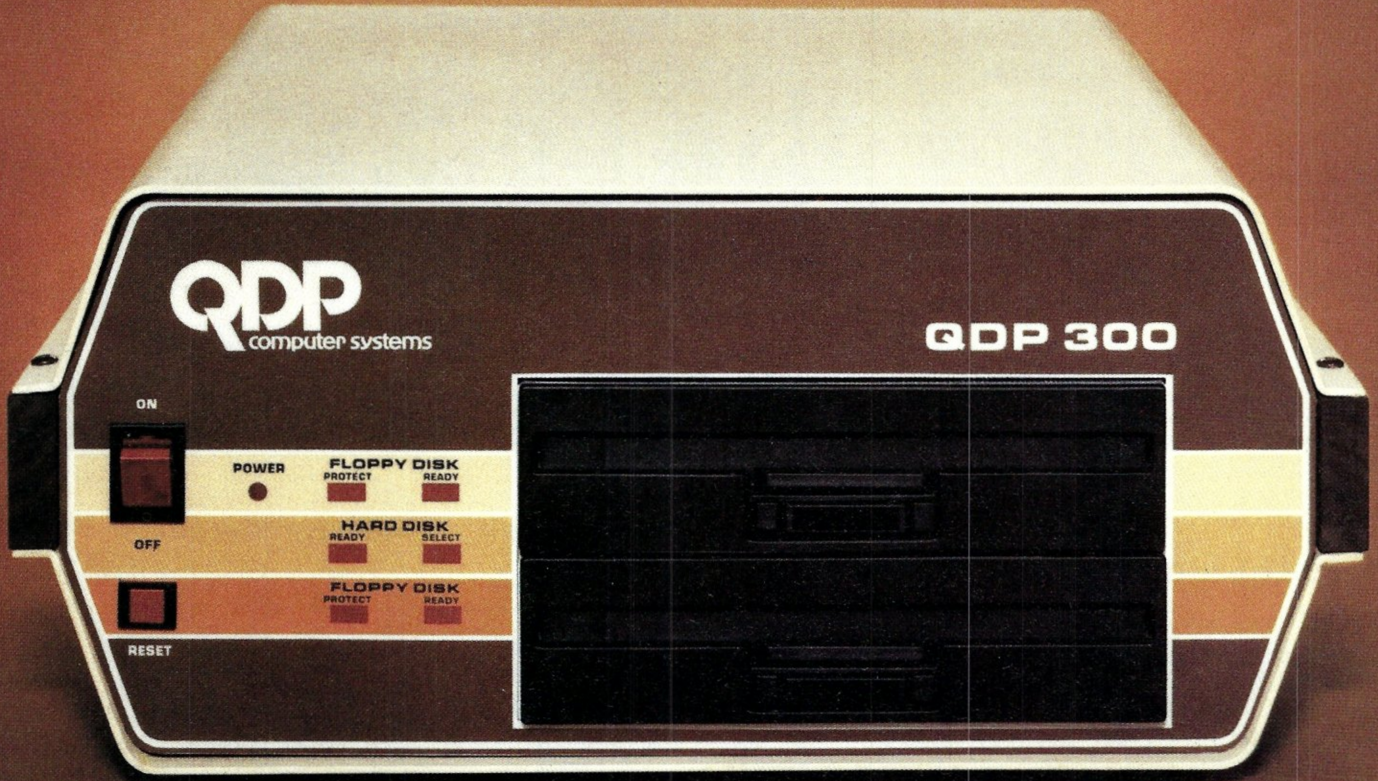
QDP-300

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.

CIRCLE 160 ON READER SERVICE CARD

The S-100 Bus

by Dave Hardy

Readers are encouraged to send in questions about the S-100 bus. Write to Dave Hardy, 736 Notre Dame, Grosse Pointe, MI 48203.

If you've ever tried to run an old 8K RAM board in a 4 MHz or 6 MHz system, then you are probably familiar with "wait states." As promised last time, this month's S-100 Bus will explain what wait states are, why they are needed, and how they can be generated. Some elementary timing diagrams and some simple circuits illustrate the discussion. We'll also have some news about the Dysan DDD alignment procedures mentioned several months ago.

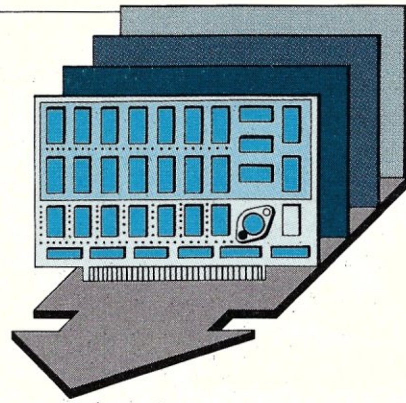
Wait states

In the ideal world, all peripheral (slave) S-100 boards do what they are told, when they are told, and there are never any problems with bus signals not being in the right place at the right time. Unfortunately, many of the boards available for the S-100 bus can't operate as fast as their host (master) CPU, and require the CPU to wait for them to finish their tasks before proceeding with S-100 bus operations.

Good examples of these kinds of boards are older static RAM boards (like those used in old Altair or IMSAI machines) designed to run at the (pre-1975) blazing speed of 500 ns for a 2 MHz clocked system. Also common in the "slow" world are many graphics and I/O boards that use EPROM's such as the 2708 or 2716.

Interestingly, there are many S-100 boards still available today that use slow EPROMs and that have made no provision to work with a CPU clock speed greater than 2 MHz. If this is the case with your board, then you might want to try some of the circuits mentioned here.

CPU manufacturers took this potential problem of slow peripheral bus devices into account when they designed their products (even in the days of the good old 8080) and included special control lines that could cause the CPUs to add extra null cycles to their normal bus operations, allowing a CPU to "wait" for



slow peripheral devices to respond. Hence the name "wait state." In general, any peripheral device can tell the CPU to wait by just asserting a "wait" line at the right time.

Some specifics

Wait states (the extra bus cycles that can be inserted into a CPU's bus operations) come in three basic flavors. There may be more, depending on the exact type of CPU, but these three are the most common ones seen on the S-100 bus:

- "M1" wait states
- "Memory read and write" wait states
- "I/O" wait states

Because the Z80 is probably the most common CPU for the S-100 bus, all of the examples used here will be based on this chip, as it appears to the S-100 bus.

Figure 1 shows the simplest and broadest type of wait state. Notice that each time the WAIT* line of the Z80 CPU is asserted, the following bus cycle is a "wait" type cycle (called T_W in the figure.) Multiple wait states can be inserted by just reasserting the WAIT* line each bus cycle. The M1 cycle is also referred to as the OPCODE FETCH cycle, by the way, and is typically three clock cycles long.

Figure 2 shows the timing for a typical memory read or write bus cycle. This is the most common form of wait state used with slow memories to allow them enough time to stabilize their data outputs or inputs.

Figure 3 represents "I/O" wait state timing. Although used occasionally for special I/O processes, this form of wait state is infrequently seen on the S-100 bus, except on special controllers and interfaces.

S-100 wait states

Generating wait states on the S-100 bus is a bit more complicated than the general theory above would indicate. Obviously, the wait state generator needs access to the CPU clock line (System Clock, pin 24). The generator also needs to see the Z80's M1 signal, which is provided as the pSYNC signal of the S-100 bus (pin 76). Additionally, the generator needs the ability to tell when its own board (or slave) is enabled, so that it will generate wait states only for its own board and not for other slaves on the S-100 bus. It would be a terrible waste to have to generate wait states for an entire system that had only one slow board.

Figure 4 shows a simple circuit that can be used to generate a single wait state on every M1 bus cycle whenever its device is selected by the SELECT* signal. Basically, System Clock is used to clock logic "1s" into a D-type flip-flop that is cleared at the beginning of each M1 cycle, whenever the device is selected. The result is that the output of the flip-flop will be made high (then inverted and sent to the S-100 RDY line) each time an M1 cycle occurs, but only when the device is selected. This way, the RDY line will say "not ready" at the beginning of each M1 bus cycle when the device is selected, and will cause the generation of a wait state. Since the next clock cycle will toggle a logic "1" back into the flip-flop, this generator will cause only a single wait state to occur each time the device is selected.

A more complex multiple wait-state generator is shown in Figure 5. This circuit uses a parallel loading shift to register to allow up to 8 wait states to be generated. Operation is the same as in the circuit in Figure 4, except that the switches must be closed to select the desired number of wait states. For example, closing switches 1, 2, and 3 will cause three wait states to be generated, because three logic "1s" will be clocked out of the shift register by the system clock into the RDY line each time the generator is activated. Similarly, closing all eight switches would cause wait states to be generated, and closing none of the switches would cause no wait states to be generated.

S-100 Bus
continued. . .

Because they only require two connections into an existing circuit (one to the board select logic, and one to the RDY line), these circuits should be easy to add to any existing slow boards that you may have. Many readers have already found this necessary for using 2716 and 2732 EPROMs in a 6 MHz or 8 MHz S-100 system. I have used these circuits myself to "recover" slow EPROM boards for my own S-100 4 MHz systems.

Dysan software

Since the last time I wrote about Dysan Corporation's Diagnostic Disk, they have decided to make public a CP/M version of a program that allows the alignment of floppy disk drives with an inexpensive digital alignment disk (about $40 for the 8" disk). Although this may seem a bit out of place in a column about the S-100 bus, it is actually quite appropriate. Floppy disk drives are a constant source of trouble in most S-100 systems, and anything that makes them less troublesome is well worth

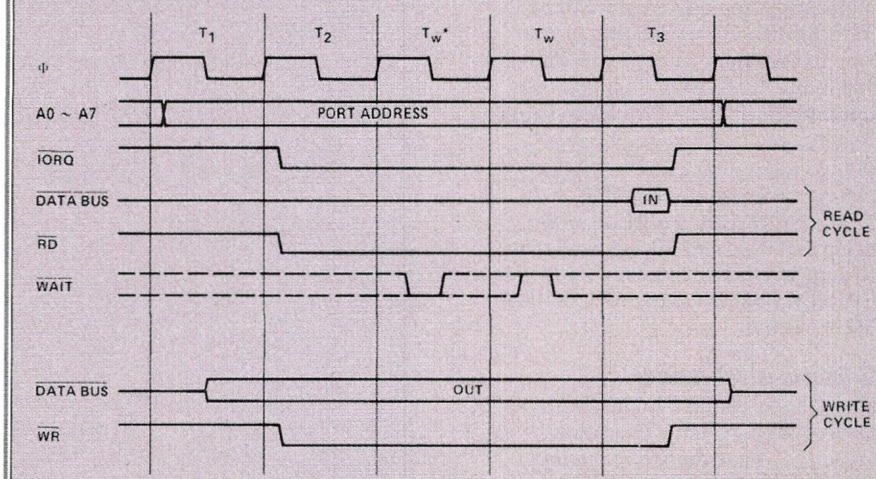


FIGURE 1. M1 CYCLE WITH WAIT STATES

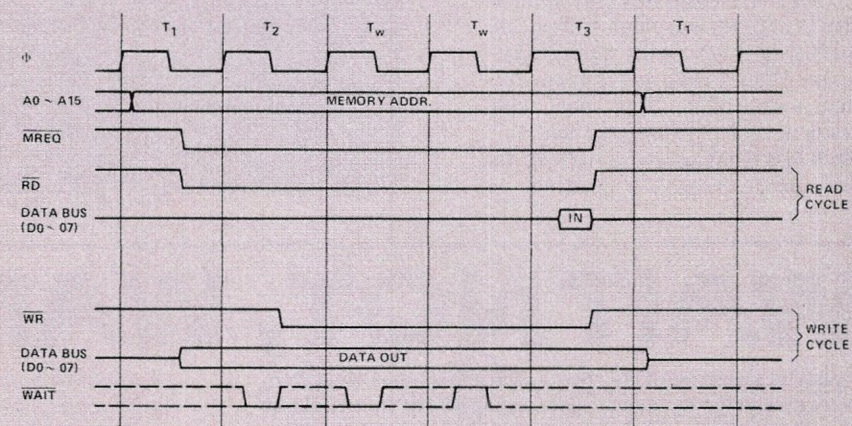


FIGURE 2. MEMORY READ AND WRITE CYCLE WITH WAIT STATES

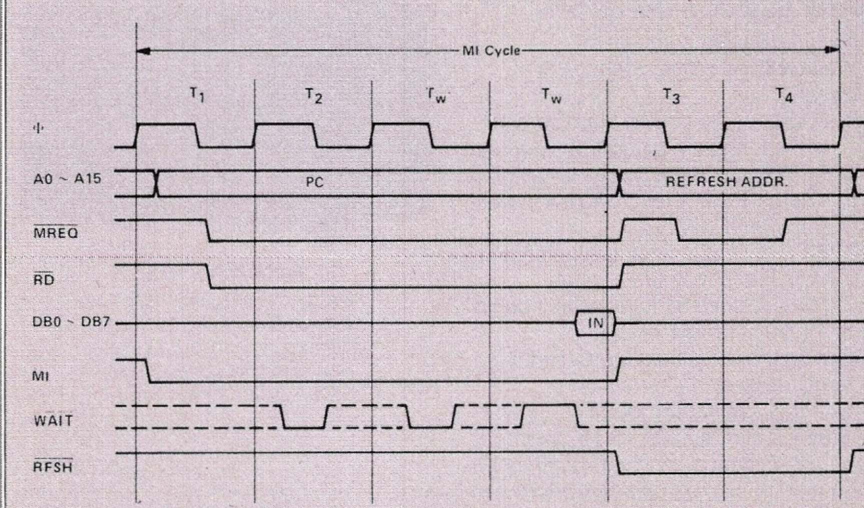


FIGURE 3. I/O CYCLES WITH WAIT STATES

mentioning here.

Incidentally, one of the alignment brochures distributed by Dysan shows an engineer checking the alignment of a floppy disk drive with a special Dysan test stand and rack that contains an S-100 machine (actually a modified IMSAI, I would guess).

The generic alignment program, called DDD.ASM, is already available on many RCP/M systems, and special versions for specific machines are becoming available. The alignment disk and generic program disk are available directly from the CE division of Dysan Corporation on 8″ or 5.25″ CP/M disks.

With just this program and a Digital Diagnostic Disk, you should be able to pinpoint most drive problems in just a few minutes. No additional test equipment is required.

Even if you don't want to align your own drives (few people do), being able to confirm that a system problem is caused by a defective drive, and not system RAM, controller board, etc., makes it well worth the cost. I have spent long hours troubleshooting S-100 systems with analyzers, memory test programs, and the like, only to find that the problems are in the floppy drives. I have also heard from more than a few readers with strange S-100 "horror stories" about random system crashings, etc., whose problems disappeared after a drive alignment or installation of a new drive.

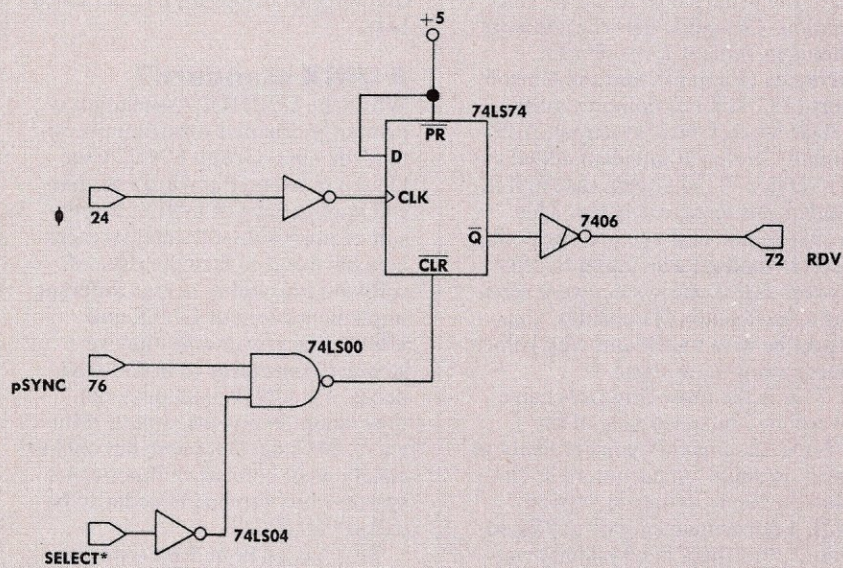


FIGURE 4. A SIMPLE WAIT STATE GENERATOR.

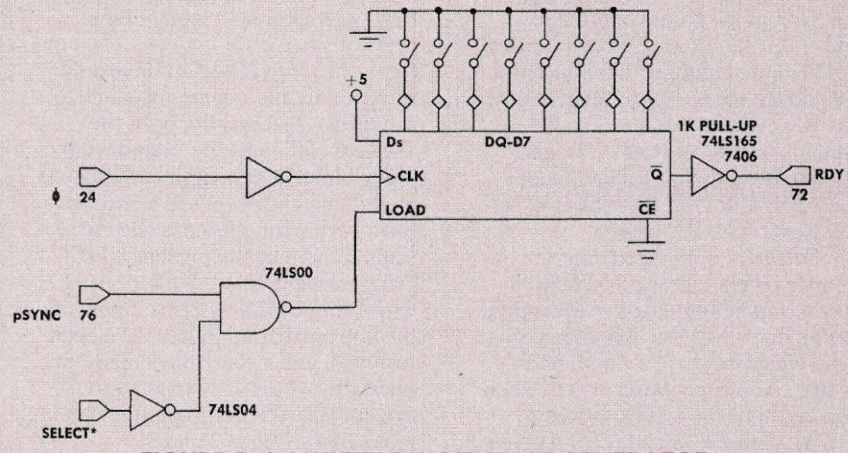
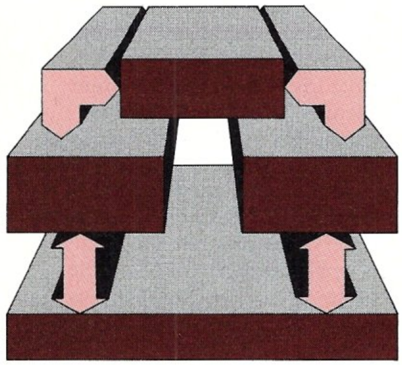


FIGURE 5. A MULTIPLE WAIT STATE GENERATOR

The UNIX File

by Ian F. Darwin

The UNIX File examines many aspects of the UNIX operating system. If you have any comments or questions about UNIX or this column, feel free to write me at the University of Toronto Computing Services (UTCS), 10 King's College Road, Toronto, Ontario, M5S 1A1 Canada. The opinions presented here are those of the author, and are not necessarily those of the University of Toronto or of UTCS.



With this issue the UNIX File celebrates a year of regular publication. I made no predictions last January, so I can't review them. Nor will I step out on a limb to make any predictions for 1984 other than the obvious claim that, if the world doesn't get blown up, more and more people will be using UNIX, UNIX-like systems, and systems layered on top of UNIX. Meanwhile, this month's column reviews a non-textbook on UNIX, looks at the UNIX standard proposed by /usr/Group, reports on UNIX at ACM '83, and touches a few other topics. And speaking of this being January, 1984, if you read this before mid-month you still have time to get to the UNIX conference in Washington, D.C. from the 17th through 20th of January.

The UNIX Guide

Are you still trying to decide if UNIX is right for you? Here at last is a book which does not try to make you an overnight UNIX wizard. The *UNIX System III Guide* from Pacific Micro Tech (5819 Poinsett Avenue, El Cerrito, CA 94530) is intended not as a self-teaching guide to the system, but rather to help you decide whether UNIX is suitable for your needs by describing all the standard UNIX software. The author mentioned to me that the book would also be useful for those trying to get a UNIX system into their organization, since it's more likely that management would read and understand the *Guide* than the entire three-volume UNIX programmer's manual.

The software is categorized by subject, as are most of the "self-teaching" books. But this book offers descriptions of what each program does and what it is useful for, rather than the usual "how to use it" information. The guide lists the software that is in version 7, System III, Berkeley (4.1BSD), and the Unisoft port of UNIX (common to most 68000-based UNIX systems). Additionally, for each program added after version 7, the system that first included this feature is listed. This would be useful if you are evaluating several vendors who claim to offer System III; if one system were missing a few System III utilities, you might be wise to find out why before purchasing the system!

This is the most comprehensive overview I have yet seen of the UNIX versions that you are likely to meet on small "super-micro's" (including Xenix, which is version 7 with a few enhancements not listed here). And there is a System V version of the guide that should be in print by the time you read this. If, however, you have already bought the System III guide, don't feel bad—you can upgrade for half price!

The book assumes that you know something about computer software, but you certainly don't have to be a systems wizard to read it. To give you an idea of what I mean, here's the information on the reminder services which UNIX offers:

"*Calendar* is on electronic reminder service. You can schedule events and be reminded of upcoming events upon request. Alternatively, *Calendar* can send you mail and notify you of upcoming events when you log in to the UNIX systems.

CAL prints a calendar for a given month or all months of a given year."

After reading that, almost anybody who has used a computer system would know what the two programs do, and which to use for a particular purpose. Most of the book is in this vein. If you're looking to see what UNIX can do for you, or if you want a good summary of the range of UNIX software including System III and Berkeley, then I recommend the *UNIX System III Guide* by Bill Freiboth of Pacific Micro Tech.

I've now reviewed most of the current books on UNIX: *The UNIX System*, *A UNIX Primer*, *UNIX Primer Plus*, and others. If there's any interest, I'll put together a summary chart for a future issue, summarizing all the books I've reviewed here.

A UNIX standard?

While the USENIX Association is primarily oriented toward universities, the /usr/Group UNIX User Group is oriented toward commercial applications of UNIX. In pursuit of profitable software, its members have had to face the issue of software portability across different implementations of UNIX and UNIX-like systems. So they've struck a committee to draft "standards" for UNIX systems—not, they claim, in any attempt to influence UNIX development, but only to clarify what core set of functions a system must provide in order to be called "UNIX."

This has all been done with the knowledge and cooperation of Bell Labs, who hold the trademark on the name of the system. Bell even gave them permission to reprint large parts of the UNIX manual. The currently pending UNIX standard describes only the system interface to programs (technically, both the "system call" and the "standard library" interfaces). Both in form and in content, the proposed standard has a very strong resemblance to sections 2 and 3 of the present UNIX programmer's manual distributed with most UNIX systems. Some of the apparent ambiguities have been resolved, and a few things made explicit that you were expected to figure out yourself before, but it still looks like sections 2 and 3.

They attempt to maintain com-

patibility with version 7 (the standard UNIX for many places) as well as the newer 4BSD systems from Berkeley and System 3/System 5 from Bell. This may be hard to do in the future, since the new 4.2BSD (just being received by the first few sites as I write this in October) has a large number of new system calls and changes to existing ones. But that's the committee's problem, not mine.

If all you want is to see what's in the UNIX system interface, then just get the Bell UNIX manuals or one of the technical books which I've reviewed recently. It makes no major changes (other than the addition of record locking, which people want for database applications). There is room in the standard for some features to be marked as optional, but this has not been implemented yet.

Members of the System Interface Standards Committee represent most of the companies doing serious work in UNIX. If you want to see the standard (which may be voted on at the UNIX conference in Washington in January), you can order it from /usr/Group, Box 8750, Stanford, CA 94305-0221. Write for current pricing.

Thompson and Ritchie honored at ACM '83

In recognition of their work in developing the UNIX timesharing system, Ken Thompson and Dennis Ritchie were awarded the ACM's prestigious Turing Award. One of the Association's highest awards, it was presented to the pair at the opening session of its Annual Conference, ACM '83, in New York City, October 24-26, 1983.

In presenting the Turing Lecture, Ritchie gave a good talk discussing the history of UNIX within the Bell Labs research environment. He asked the hypothetical question: Could UNIX have succeeded in a company which regarded computers and software as its main business? His answer was: Quite possibly not. Thompson related some of his cleverest programs, one of which was a simple mod to the "C" compiler to make it compile an altered version of the UNIX **login** command, allowing him to sign on without a legitimate password. His conclusion—a technical argument too long to include here—was that you can't trust software distributed by a vendor to be secure. From this subject, Thompson moved on to a discussion of system crackers such as those in the *War Games* movie and the "414" group. He blamed popular media (rightfully so, in my opinion) for glamorizing these activities rather than condemning them as vandalism. Thompson concluded by calling for a program to re-educate people on the morality of computer penetration.

Thompson and Ritchie also received the Association's Software System Award for the "creation and promulgation of the software components of UNIX comprising a complete system." The recognition arrives at a time when UNIX has at last achieved the recognition it deserves. The awards come more than a decade after the original design of UNIX, which was done in 1969-71.

The chess engine "Belle," which Thompson co-created with Joe Condon of Bell Labs, became the first computer ever to receive U.S. Chess Federation "Master" rating, with a USCF rating of 2203. Belle won several games in the Fourth World

Computer Chess Championship, held in conjunction with ACM '83, but was unable to retain its World Championship. Thompson was also given the 1983 Common Wealth Award for Distinguished Service in Invention for his work on UNIX.

There were other UNIX-related talks among the many technical papers. Steve Johnson and L. Rosler of Bell Labs spoke at length on the evolution of the C language and its variants. Pamela Zave spoke on "operational systems" design for DP projects. P.G. Mathews of Bell spoke on International Work Bench,™ the new CAI for UNIX. Last (and least), your columnist made a few remarks about UNIX on micros at the Microcomputer Operating Systems Panel. Most of the technical talks (or their abstracts) appear in the conference proceedings (available from the ACM order department, Box 64145, Baltimore, MD 21264, phone 301/528-4261). Audio Visual Transcripts (250 West 49th Street, #400, New York, NY 10019, phone 212/586-1972) recorded many sessions; call or write for ordering information.

Another software source

Here's another company set up to package and market UNIX-based software. UniVentures (27 Buckthorn Way, Suite One, Menlo Park, CA 94025, phone 415/325-3283) is primarily oriented to bringing small developers and OEMs together, rather than at selling software directly to end users. They aim to be helpful to software developers and software companies, and can help in deciding which software should be written to achieve good selling potential. They aim to provide OEMs with a source of evaluated, working software. Future plans may include support for UNIX software.

If you have a UNIX software package to sell, contact these people and UniPress (see the last column). And if you, as an OEM, want to buy software for your small UNIX system, then give UniVentures a try. Call or write for a catalog.

Hard floppies

Why build a UNIX system with a fast processor and a slow disk? Because it's cheap, that's why. And many vendors think (with some justification) that the buying public will put up with half the disk throughput if the system costs a bit less. (That's my friend Geoff's favorite sarcasm: "$10 less and half the functionality? I'll take it!") So we continue to see "UNIboxes" (small UNIX systems) coming on the market with slow $5^1/_4$" Winchesters as system disks. Folks, you can't get fast response from a slow disk. When buying a UNIbox, make sure you read the specifications sheet carefully.

One of the critical factors is the disk "access" speed. Transfer times are pretty standard, but make sure the controller uses real DMA—a technique for transferring data to/from main storage or RAM without tying up the main processor (CPU) to read each byte as it comes in. Make sure, also, that you get a disk with an "access time" of 30 msec (milliseconds) or less. The cheap winnies have access times of 90 msec or more. Brent Byer of Textware calls these things "hard floppies," and with good reason. They are almost as slow as floppies. The access time is an average, and there are several variations, so you

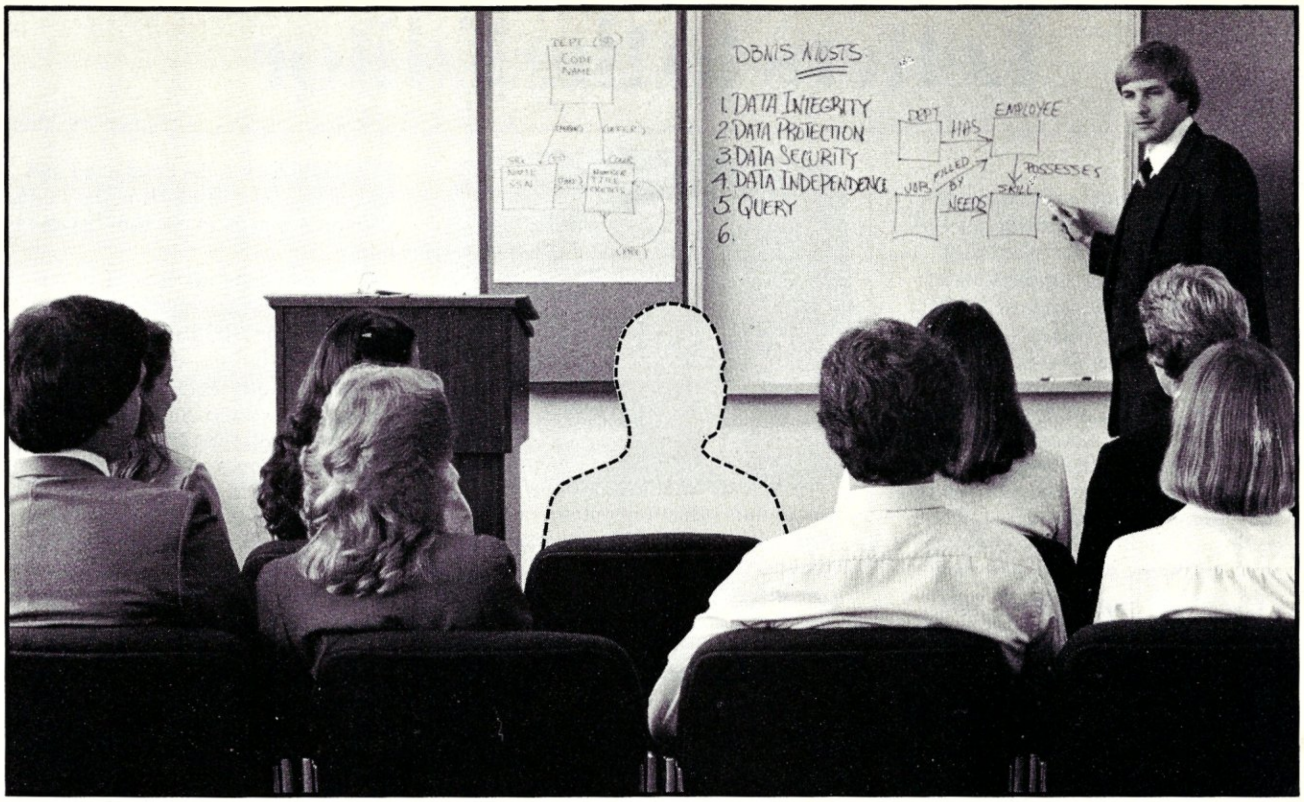
The UNIX File
continued. . .

can't get perfect comparisons. An OEM who evaluated a large number of these drives tells me that one disk with a 40 msec rating is actually faster than several rated at 30 msec. But the ratings are approximatedly right, and a 90 msec disk just doesn't cut it. Don't pay for a hard disk and get stuck with a hard floppy.

One less UNIX-like system

The Marc operating system, a UNIX-like system for the 8080/Z80, has been withdrawn. In a world where computer products are often "released" before they are developed, it is refreshing to see Lauren Weinstein's attitude of not releasing the product until he's sure it's ready. Now he has decided never to release it due to a changing market.

So ends our entry into the brave new world of 1984. Watch for the next session of the UNIX File, which will have a few more shell tips, and probably some information on typesetting and/or networking. ■


DBMS MUSTS
1. DATA INTEGRITY
2. DATA PROTECTION
3. DATA SECURITY
4. DATA INDEPENDENCE
5. QUERY
6.
DEPT
HAS
EMPLOYEE
JOB
FILLED BY
NEEDS
SKILL
POSSESSES

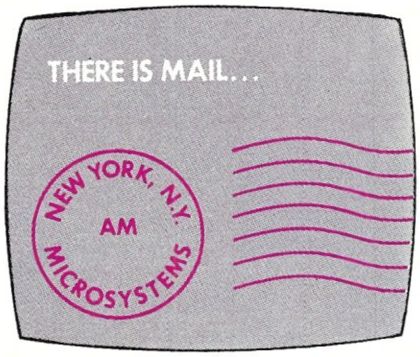
Letters to the Editor

Dear Sir,

In the September issue was a review of the Ithaca Intersystems Encore in which it was implied that switch mode power supplies could not be used with a standard S-100 bus system, hence Ithaca's departure from the standard. This is totally untrue; I have a home-built S-100 system with a switch mode power supply delivering regulated +7 V and ±14.5 V, leaving the very minimum to be dropped across the on-board regulators. This gives the best of both worlds: the light weight and efficiency of an SMPS, with the on-board regulators giving interboard isolation and removing any residual 100 kHz ripple from the SMPS. The on-board regulators are one of the strongest points of the S-100 bus and totally eliminate the board-to-board crosstalk through the power rails that can be so troublesome with other bus structures not so equipped.

Now that at long last the IEEE standard has been agreed and issued, there is no excuse whatsoever for departing from it. Anyone who does depart from it creates a new bus and should not refer to his products as S-100. There are many noncomplying boards on the market, some specifically labelled "IEEE standard," but which lack vital signals or have other nonstandard features. I recently examined a newly introduced 6809 CPU board that had been advertised as IEEE standard, but lacked an sM1 signal. Such a board would be useless in my system, which makes heavy use of this signal. The excuse given was that the 6809 outputs no such signal. Maybe, but it can be easily derived by a little external logic, and its ommission can only be regarded as sheer laziness. The best advice is "caveat emptor," or build it yourself.

I notice that a few months ago you dropped the mention of "S-100" on the cover. I hope that does not mean the gradual phase-out of S-100-oriented articles, which are what I buy *Microsystems* for. CP/M I have little time for, and by itself, it would not make me buy your magazine. Finally, being strongly anticopyright, I find the regular feature "in the Public Domain" of interest, but almost everything mentioned is for running under CP/M (itself copyright!). How about a mention of public domain software that does not presuppose a



particular operating system, or for that matter, public domain operating systems (do any exist, apart from my own effort?)?

Greg Trice
1131 Sandhurst Circle, #111
Scarborough, Ontario M1V 1V5
Canada

Chris Terry replies:

Nobody at Ithaca Intersystems said or implied that it was impossible to use a switching power supply with standard S-100 boards. Rather, it was felt that to do so would unnecessarily add to the cost of the regulators and associated components, as well as requiring larger fans to remove the heat generated by the regulators. In a word, it would not be cost-effective and would partially negate the benefits of using a switching power supply.

Some of the Pascal software already mentioned in the Public Domain column can run under more than one operating system. We shall be covering items issued by the C User's Group in future issues. C is probably the most portable language for micros. We are not aware of any public domain operating system—nor would such a system be of much value unless fully compatible with CP/M or MS-DOS.

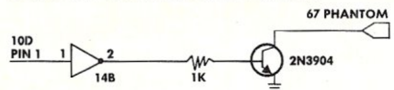
Dear Mr. Libes,

I just finished reading John Gillespie's article (October 1983) and have decided to write you a note. Many people with older systems can use the information it gave; others with new machines will wonder what all the fuss is about. Let me describe some new things North Star has done:

1. The new 32/48/64K RAM boards have switches on the top of the board which allow deselecting the RAM in 8K blocks except for the area from E000H on up. The E000H-FOOOH block can be deselected in 1K sections. This allows the E000H-ECOOH block to be deselected and leaves the rest of the 64K active. You then do not need to use the Phantom line.

2. They have rewritten their versions of CP/M so that is automatically puts the BIOS above the controller. This is done in the CPMGEN program. It allows fast and slow seeking drives and choices of hard disk. When all the choices are made, it lists the memory used by each section of the CPM system and asks if this is what you want. You can then move the parts so as not to conflict with extra hardware or special memory. When you are finished, the program is saved as usual.

I have used the phantom line on my North Star for several years now and have enjoyed using cheap 64K RAM boards which had no address disables. There is one possible problem with Mr. Gillespie's method of creating a phantom. This line can be used only by his disk controller the way it is wired. There may be other boards such as a ROM board that might want to use a phantom. The solution is to drive the bus with an open collector buffer or a transistor. I use the circuit shown below.



I have used both LifeBoat's version of CP/M and North Star's, and have found that the North Star version is much easier to modify. It would still help to have a copy of the BIOS.

Mr. Gillespie's comments about ZCPR are very correct. I have created a program that uses North Star CP/M and ZCPR with an extra BIOS above the disk controller to run 8″ drives. This system runs with a 5MB, 15MB, or HD18. The hard drive adds more to the speed and usability of the North Star than does ZCPR, but all together it makes about the best system I have seen.

Wilton Hart
1545 S.W. Dellwood
Portland, OR 97225

Dear Messrs. Libes and Terry,

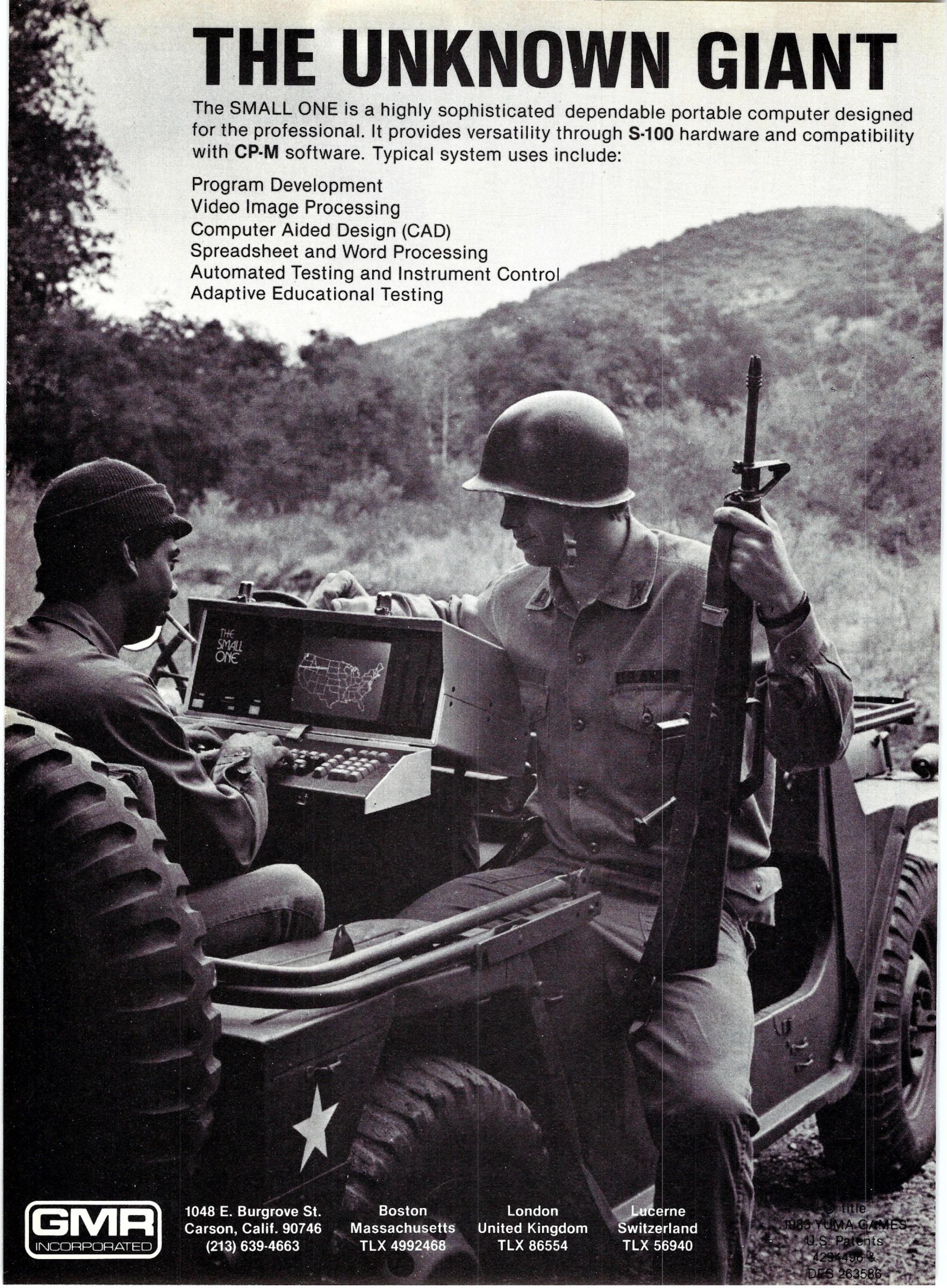
This is a response to Mr. Roger Friedmans's letter that appeared in the October issue of *Microsystems.* First of all, to set the record straight, GSR Computers supplied the boards

Letters continued. . .

for the review in *Microsystems*. Now, let's take a look at Mr. Friedman's comments:

Complaint #1: *SBC is actually a three-board set.* GSR has never advertised its product otherwise. However, the CPU board does contain 1K RAM, ROM, 1 serial and 1 parallel port. If we are permitted to use the term "Single Board Computer" loosely, I believe that it could fall into that category.

Complaint #2: *CPU speed.* While it is a fact that this board was designed a few years ago, it continues to be a popular product. A reason for its popularity is its competitive price, on-board memory and I/O, and its high quality. A similar product with the exact same capabilities running at 6 or 8 MHz would most likely be double in price due to the higher cost that must be paid for all the support chips for Z80s running at that speed. That is not to say that we are not designing a higher speed CPU, but that it will cost more. Also, I believe that it would be safe to say that a majority of the S-100 systems still being sold,



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
YUMA GAMES
U.S. Patents
DES 263586

as well as user-owned systems, are still running at 4 MHz.

Complaint #3: The CPU board uses a standard RS-232 interface signals for an ASYNC connection to a CRT terminal. The RTS/CTS signals mentioned in the article in no way affect a connection to a terminal.

Complaint #4: Correct, our CPU does have only one serial port for a connection to a terminal. However, it also contains a parallel port that is configured for the CP/M list device, such as a printer.

Complaint #5: *Disk parameter block.* All that we can say is that we presently support 27 different CP/M soft-sectored formats—from the IBM PC using CP/M 86, to Superbrain, to the Morrow Micro Decision. Plus, we are adding more! How many other systems can offer this at the same price? Also, Mr. Hardy's comment regarding the reading of the IBM standard disks without placing a table on the disk has been implemented.

Complaint #6: *IEEE-696 compatible.* GSR Computers advises their customers that the boards are in compliance with the IEEE-696, but does not support the full signalization as specified. Also, if Mr. Friedman would look around, he would find that there are very few boards that actually "support" the full signalization as specified by the IEEE-696. We are running an 8-bit CPU, so we don't use a 16-bit data path. True, no extended addressing. Pseudo DMA refers to our method of disk accessing and control. Our CPU does not support Temporary Master/Slave boards, and we never advertised it otherwise. Still, it is a board that is in design compliance with the IEEE 696.

Complaint #7: *Poor choice?* I guess you could say that would be news to the hundreds and thousands of users of our boards. For a single-user CP/M 2.2 system, we offer users high quality, flexibility, and support at a reasonable price. This is not to say that we are not working on a multiuser system, but only that it is not available at this time.

Complaint #8: With our UFDC-I we supply our users with the full Source Code of our BIOS and Disk Formatter Programs so that they can be integrated into existing systems. Possibly a complaint here is that our prices are too inexpensive ($275 A&T for CPU or UFDC-I).

Finally, GSR believes that the article by Messrs. Hardy and Jackson was fair to the products even though it did not fully explain the versatility of the GSR System, with its ability to read almost any CP/M soft-sectored format.

GSR Computers
60-10 69th St.
Maspeth, NY 11378

Dear Mr. Libes,

Bill Kibler's article on the CCS 2422 Disk Controller was very good but did not cover all the problems with bringing up this card. I have been using this controller for over $1\frac{1}{2}$ years on single and double density at 2 MHz. It took several months and several conversations with the CCS technical personnel to sort things out, but it works fine now. The first thing a user should be aware of is that this card will not work properly if the SINP or SOUT signals are latches. This is a problem on older CPU cards.

The monitor EPROM has some

Letters continued. . .

features that can cause problems with some systems. If you do not have the CCS CPU card, the monitor will hang while trying to initialize the UART that isn't there. The next problem is the location of the data and status ports which are at 20H and 25H. Most I/O boards prefer them next to one another. The monitor disk read and disk write code is too slow at double density to work with 8″ drives on a system with a 2 MHz CPU.

The disk read and disk write routines in the CCBIOS work fine for double density at 2 MHz. When trying to read a single-density disk on a track other than 0, the monitor code will fail if the first read is bad. It will try once at single density, then nine more times at double density, and will fail each time.

The controller uses some storage from 0040H to 0053H. These areas must be initialized prior to accessing the disk. When writing a bootstrap loader, the programmer must initialize the density and type, sectors per track, and bytes per sector. These go into CUNIT at 004AH, SPT at 0044H, and IDSV+3 at 0051H. TARGET Planner Calc also uses these areas, so a special BIOS must be written with these areas moved to the BIOS.

CCSINIT has a bug around 0112H. The program saves the drive status and restores it after the program has finished. Unfortunately one RAL was left out of the code. It saves the two-sided flag in the double-density position of the controller command. When the program terminates, it's likely that the system will be confused.

The verify code in VCOPY is not fast enough to work at 2 MHz in double density on 8″ drives. CCSYSGEN has this same problem with its verify code.

Mr. Kibler's comments about needing code to solve timing problems are certainly true. I had "DRIVE NOT READY" messages flash on the screen, until I put a few NOPs here and there. My drives also need a 100 μs delay when changing sides.

In spite of the work necessary to make this board work on a non-CCS system, I think this controller is a very good buy.

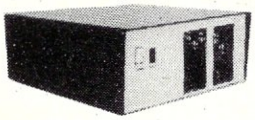
Dennis B. Anderson

Retail Sales
123 East 200 South
Salt Lake City, Utah 84102

The Great Salt Lake Computer Company, Inc.

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

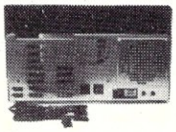
QT 5¼" MAINFRAME

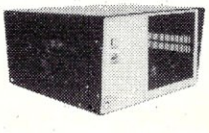


•Provisions for any 2-5¼" drives•15 ea DB 25 cutout •2 ea 50 pin•2 ea 34 pin•1 ea Centronic•EMI filter (fused) •2 AC outlets•Avbl. with 6-8 or 12 slot motherbd• Power supply (+8V16A/+12V5A)

7013/QTC-MF + MD12 (12 slot MB) $560.00

QT 8" MAINFRAME

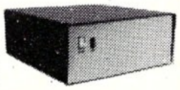




•Provision for any 2-8" drives (hard or floppy)•15 ea DB 25 cutout•2 ea 50 pin•2 ea 34 pin•1 ea Centronic•EMI filter (fused)•2 AC outlets•Avbl with 6-8 or 12 slot motherbd •Power supply (+8V16A/±16V3A/+5V6A/−5V1A/ +24V6A)

Desk Top Version
7009/QTC-MF + DD6 (6 Slot MB) $575.00
7010/QTC-MF + DD8 (8 Slot MB) $625.00
7011/QTC-MF + DD12 (12 Slot MB) $675.00

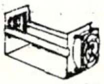
QT STANDARD MAINFRAME



•Provisions for any 2-5¼" drives
•15 ea DB 25 cutout•2 ea 50 pin
•2 ea 34 pin•1 ea Centronic
•EMI filter (fused)•2 AC outlets
•Avbl with 6-8-12-18 or 22 slot MB
•Power supply (+8V16A ±16V3A)

Desk Top Version
7014/QTC-MF + 12 .... $499.00
7016/QTC-MF + 18 .... $525.00
7015/QTC-MF + 22 .... $550.00

QT S-100 CARD CAGES



•Made of anodized steel•Card guides for ea. MB•1-Indicates w/MB
•2-Indicates w/MB + 1 ea fan
•3-Indicates w/MB + 2 ea fans

2064/QTC-CC4	$20.00	2072/CC12	$41.00
2065/CC4-1	$60.00	2073/CC12-1	$140.00
2066/CC6	$22.00	2074/CC12-2	$160.00
2067/CC6-1	$70.00	2075/CC12-3	$180.00
2068/CC6-2	$90.00	2076/CC18	$50.00
2069/CC8	$31.00	2077/CC18-1	$200.00
2070/CC8-1	$100.00	2078/CC18-2	$220.00
2071/CC8-2	$120.00	2079/CC18-3	$240.00
		2080/CC22	$75.00

QT S-100 MOTHERBOARDS

•Silence Plus•Built in Termination•IEE696•Terminal strip for easy power connection

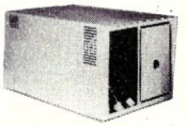
4 Slot Motherboards		**6 Slot Motherboards**	
2051/QTC-MB4BB	$15.00	2057/QTC-MB6BB	$20.00
2052/QTC-MB4A	$40.00	2058/QTC-MB6A	$48.00
8 Slot Motherboards		**12 Slot Motherboards**	
2053/QTC-MB8BB	$25.00	2059/QTC-MB12BB	$30.00
2054/QTC-MB8A	$69.00	2060/QTC-MB12A	$99.00
18 Slot Motherboards		**22 Slot Motherboards**	
2055/QTC-MB18BB	$45.00	2061/QTC-MB22BB	$60.00
2056/QTC-MB18A	$150.00	2062/QTC-MB22A	$185.00

TERMINALS / MONITORS

WITH/2ND PAGE MEMORY

8026/Televideo 925C $715.00
8028/Televideo 970 .... $1095.00
8013/Sanyo 2112A (15MHz) Green $80.00
8008/Princeton HX-12RGB (For IBM) $489.00

QT 8" THINLINE MAINFRAME



•Provisions for 2 ea 8" thinline drives•15 ea DB 25 cutout •2 ea 50 pin•2 ea 34 pin•1 ea Centronic •EMI filter (fused)•2 AC outlets •Power supply (+8V16A/−5V/+24V6A+5V6A)

7012/QTC-IMF + DD6F (6 slot MB) $350.00

DISK DRIVES

8" Disk Drives
5006/801R Shugart SS/DD $359.00
5007/851R Shugart DS/DD $479.00
5011/DT8 (842) Oume DS/DD $490.00
BEST 5014/M-2894-63 MIT DS/DD $389.00
BUY *(Call for qty price)*
5016/M-2896-63 MIT 8" thinline DS/DD $399.00

5¼" Disk Drives
5018/TM100-1 or B-51 for IBM $150.00
5019/TM100-2 or B-52 for IBM (In stock) $225.00
5021/TM848-2 (DS/DD) $400.00

QT DISK DRIVE CABINETS

"All in One" Vertical Disk Drive Cabinet
For 1) 2 ea or 4 ea 8" thinline drive or 2 ea. std. 8"
2) 1 ea hard disk + 1 ea standard 8"
• Power supply (+5V6A/−5V1A/+24V6A)
• Positive pressure fan w/filter•EMI filter
• Power interface cable for any 8" drive

7001/QTC-DDC88V *For 2 standard size 8" drives* .. $269.00
7003/QTC-DDC88T *For 2 thinline 8" drives* $269.00
7005/QTC-DD8V *For 1 ea 8" drive* **$249.00**

For 2 ea 8" standard size drives (horizontal)
• Power supply (+5V6A/−5V1A/+24V6A)
• Interface cable for any 8" drive

7006/QTC-DDC88H $269.00

5¼" cabinets
7017/5¼" DDC5H w/PS horizontal-for 1-5¼" drive 55.00
7007/5¼" DDC5V w/PS vertical-for 1 ea. 5¼" drive 65.00
7008/5¼" DDC55V w/PS vertical-for 2-5¼ drives .. 85.00

MODEMS

0001/Hayes Smart 300 .... 199.00
0002/Hayes Smart 1200 .... 499.00
0003/Multi-Tech MT 212 AD (1200/300) 499.00
0004/Novation J-Cat 300 .... 119.00
0005/U.S. Robotics 212A Auto Dial 469.00
0006/U.S. Robotics Password 399.00

PRINTERS

9152/Brother HR-1A (Par) 17CPS Daisy Wheel 595.00
9017/Daisywriter 2000 (48K Buffer) 25 CPS Daisywheel 1,050.00
9048/Dynax-15 (Par) 13 CPS Daisywheel 475.00
Epson FX100 .... Call
9051/C, I TOH Pro-writer 1 .... 350.00
9062/Juki (Par) 18CPS Daisywheel 569.00
9066/Mannesman Tally (160CPS-40CPS) 589.00
9084/Oki-Data 82A (120CPS) 379.00
9086/Oki-Data 92 (160 CPS) 489.00
9088/Oki-Data 84P (200 CPS) 969.00
Star Gemini 10X and 15X .... Call
9107/Toshiba P-1350 (192/120 CPS) 1,499.00

AC SURGE ELIMINATORS

0011/Grizzly (200W) uninterruptible power system + surge protection 750.00
0012/Grizzly (500W) uninterruptible power system + surge protection 1,802.00
0013/Hawk AC power monitor w/surge protection 160.41
0014/Lemon (6AC outlets-3 prong) 44.00
0015/Lime (5'-3 prong pwr cord w/on-off switch ... 69.00
0016/Orange-AC surge + EMI filter (6 outlets) ... 95.00
0017/Peach (3 outlets) 69.00

COMPUPRO

2001/20 Slot MB	225.00	2014/Interfacer 4	314.00
2002/CPU 8085/88	349.00	2015/System Support I	308.00
2003/CPU 8086/8087	550.00	2016/Active Terminator	65.00
2004/Disk I	399.00	2017/Enclosure 2 (Desk)	725.00
2005/CPM 2.2 for Disk I	150.00		
2006/Disk II	610.00	2018/Enclosure 2 (Rack MT)	775.00
2007/Ram 16-64K	325.00		
2008/Ram 17-64K	299.00	2019/CPU-Z	228.00
2009/Ram 21-128K	650.00	2020/M-Drive H	1,229.00
2011/Interfacer 1	198.00	2021/6 Slot MB	125.00
2012/Interfacer 2	249.00	2022/12 Slot MB	155.00
2013/Interfacer 3	515.00	2023/CPU 68K	510.00

SIERRA DATA S-100 BOARDS

2030/SDS-SBC-100-Z80 (4mhz) master 2 serial 2 par/floppy controller/64k ram $655.00
2031/SDS-SBC-100S4mhs slave/2 serial 2 par/64k ram $565.00
2032/SDS-ZSIO/4-4 serial port I/O bd $250.00
2033/SDS-MUX-RS232 multiplexer bd $235.00
2034/SDS-HDI-M-Hard disk bd for micropolis $129.00
2035/SDS-CPM/B105-cp/m for SBC100 w/BIOS $150.00
2036/SDS-Turbodos-Multi-user for master & slaves $645.00

QT/COMPUTIME BOARD SET

PN-2048 Best Bare Board Set Available
QTC-SBC 2/4 CPU (SBC 880)
QTC-EXP + III 256K (CT256) Memory bd./Expandable to 1MG
QTC-FDC 5/8 Floppy disk controller
Bare Board Set **$150.00**
1) Includes manuals & assembly instructions
2) Parts available 3) Monitor & B10S available

QT COMPUTIME CLOCK/CALENDAR

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

2049/QTC-CCS-BB (S-100) $45.00
2050/QTC-CCS-A (A + T) for S-100 .... $95.00

CPU BOARDS / MEMORY BOARDS / I/O

2038/QTC-SBC 2/4 BB (SBC880) $50.00
2039/QTC-SBC 2/4 A A + T (SBC880) $265.00
2040/QTC-Z + 80 BB (BIG Z) $28.00

DYNAMIC (64K/256K or 1 MEG)
2041/QTC-EXP + III Bare Bd. (CT256) $75.00
2042/QTC-EXP + III 64K A + T (CT256) $375.00

2043/QTC-I/O + BB *2 SER 2 PAR* A + T $75.00
2044/QTC-I/O + A + T $300.00
2045/QTC-ADA *ADA Converter* A + T $400.00
2046/QTC-Dual GPIB-488 *IEEE 488 Interface bd.* .. $795.00

DISKETTES

Diskettes are 3m media packaged and certified by CENTECH. Lifetime warranty —5 colors in each pkg. (Red, Yellow, Blue, Green, Brown)
4001/5¼" Sgl side/dbl den 22.00/10
4002/5¼" Dbl side/dbl den 29.00/10
4003/5¼" 10 sector 24.00/10
4004/5¼" 16 sector 24.00/10
4015/8" Sgl side/dbl den 30.00/10
4016/8" Dbl side/dbl den 40.00/10

DISKETTE STORAGE

ADVANCE ACCESS
4019/AA-5¼ (Holds 82 Disks) 17.00
4020/AA-8 (Holds 82 Disks) 26.00
Smoked Plexiglass Disk Tubs

LIBRARY CASES
4021/CAS-5¼" 2.50
4022/CAS-8" 3.00
4028/Color Burst (Pack of 5) 12.00
Colors Available; color burst asst., beige, black, blue, red, gray

4029/Flip file storage case *(stores 50 diskettes)* ... $27.95
4030/Flip File "Original 5" (holds 50) $21.00
4031/Flip File "Original 8" (holds 50) $29.95

In Continental U.S.
TOLL FREE (800) 545-2633

Inside Utah
(801) 363-3314

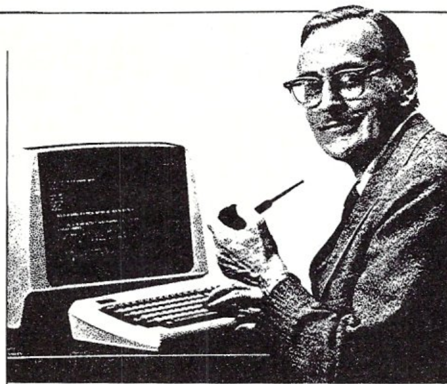
We accept cash, checks, credit cards, or Purchase Orders from qualified firms and institutions. **Minimum prepaid order $15.00** Export cusotmers outside the U.S. or Canada please add 10% to all prices. **Prices and availability subject to change without notice.** Shipping and handling charges via UPS Ground 60¢/lb. UPS Air $1.00/lb. **minimum charge $3.00**

CIRCLE 37 ON READER SERVICE CARD

In the Public Domain

by Chris Terry

The CPMUG and SIG/M public domain libraries have gradually acquired a respectable number of editors, text formatters, and word processing utilities. The usefulness and quality varies considerably; as with most other categories of public domain software, the earlier acquisitions were experimental, hardware-dependent to some degree, and served some specific need of their authors. The later packages are, in general, better designed and use BDOS service requests rather than direct calls to the BIOS. Documentation, often sparse (to say the least) in early volumes of CPMUG, has been much better in packages released during the last two years, and sometimes can be called "excellent."



Editors

EDIT is described as "an Intel-like editor"; it also bears some resemblance to ED, the editor supplied with CP/M. EDIT was originally issued in CPMUG Vol. 16, in .COM form only. A disassembly of this editor was issued as EDIT.ASM in CPMUG Vol. 29, and an updated version (EDITM) with new features appeared in CPMUG Vol. 81. It was originally designed for use with a printing terminal, so it has only a command mode, not a screen edit mode.

TED is another editor for printing terminals, somewhat simpler than CP/M's ED. It appeared in CPMUG Vol. 36 and is also available in SIG/M Vol. 80. I tried TED a year or two ago and found it easy and friendly. **ICE** (In-Context Editor) was issued in SIG/M Vol. 83. It appears to be similar to EDIT and TED, though more powerful, but I have not tried it. In any case, these older line-oriented editors become intolerable after using a good screen editor such as WordMaster, WordStar, or Perfect Writer.

ED and **RED** are much more useful, since they are screen editors written in C. ED, available in SIG/M Vol. 76, was contributed by the Software Tools of Australia group and is based on ED2, a C editor described by Edward K. Ream in the January 1982 issue of *DDJ.* This was designed for compilation with Ron Cain's Small C. RED is a more powerful update, described by Ream in the July and August 1983 issues of *DDJ,* with buffer routines based on "Just Like Mom's Editor," available from the BDS C User's Group. RED is copyrighted and is not in the public domain libraries, but source code (for the BDS C compiler) is available from Mr. Ream, who hopes "you will do anything with this editor except distribute it for profit," and will support it.

Formatters

Formatters vary in complexity from Ian Darwin's "50-line Text Formatter" (*Microsystems,* August 1983) to the super formatter ROFF4 in Vol. 126 of the SIG/M library. Darwin's 50-line Text Formatter provides for line breaks, ragged right edge alignment, and page breaks, and is suitable for letters and short papers. **ZPTEX** (SIG/M, Vol. 22) is a very simple text formatter written in Pascal-Z. **RUN80-V2** (SIG/M Vol. 40) is another text formatter, and **SECRETARY** (SIG/M Vol. 109) is a simple word processor for secretaries. I have not tried any of these yet.

POW (Processor of Words) is a text formatter that uses embedded dot commands and provides headers and footers, right justification, indent and outdent, centering, page numbering, and most of the features needed for straightforward text documents. It was first described in *DDJ* #29, and the assembly language source code appeared in CPMUG Vol. 36. That version did not have CP/M linkages and was found to have several annoying bugs. An updated version (POW2) appeared in CPMUG Vol. 81; in this version all known bugs were fixed and proper linkages to CP/M were added. It is a friendly and useful formatter as long as you don't want to handle footnotes or do anything really fancy.

ROFF4 (SIG/M Vol. 126) was contributed by Professor E. Bergmann of Lehigh University. It is a Cadillac of formatters, patterned after the UNIX formatter **nroff.** ROFF4 not only handles footnotes extremely well, but has algorithms that allow chemical and mathematical equations to be printed with correct partial-line feeds so that they look really good.

You can define your own symbol/character set and invoke any image in this set by means of a translation character visible to your editor (e.g., ˜a might print as a Greek alpha). It also has some built-in macros for simple invocation of complex processes, and facilities for building your own macros. Like ATMS (which runs on IBM mainframes), it allows you to establish a file containing the "style" macros for a particular document, this file can also be invoked in the command line that specifies the text file. And, as Madison Avenue might put it, ". . . much, much, more." We hope to publish an article by Professor Bergmann about this formatter.

Utilities

SPELL is a spelling checker program in CPMUG Vol. 80. It requires Cromemco Structured Basic to run, and is said to be slow because each word is checked in an ISAM file. **INDEXER** (SIG/M Vol. 94) and **GENINDEX** (SIG/M Vol. 143) are both written in Pascal and generate an index from a WordStar file. **WMNOTES** (SIG/M Vol. 42) contains notes for customizing WordMaster and making its control commands closer to those of WordStar. And finally, two programs which I found on RCPM systems but are not in SIG/M or CPMUG: **TEXCLEAN,** which processes a WordStar file and forces bit 7 to 0 in all characters (rather more conveniently than PIP can do it); and **MAGE,** a lifesaver when you get a BDOS Error or Disk Full message while running WordStar. This program allows you to recover the changes that are still in memory but can't be written to the logged-in disk because of the error.

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

Enhancing MP/M II
Part I: Features and Faults

by Tom Clodfelter

MP/M II is an excellent but somewhat unfinished multiuser system for microcomputer systems. It has some very good things going for it, especially the upward compatiblity with CP/M. Though MP/M is no UNIX, it is a vast improvement over CP/M. Current CP/M users are well advised to consider MP/M, even if they have no immediate need for the multiuser features. In this series on enhancing MP/M II, we will show you how to remove some of the remaining drawbacks, and add several new features.

Having worked most of my time with multiuser, multitasking systems, I was always disappointed with the level of system service that CP/M had to offer. CP/M seemed fine to me during my RT-11 days, but once I was spoiled by RSTS, RSX-11M, VMS, and UNIX, I no longer found it exciting. I came to the conclusion that what makes any operating system good is not only its ease of use, but how much it will do for you *at the same time.*

I looked over the alternatives to CP/M very carefully. Only a small number of UNIX-like systems were available at the time and they all required that I buy new hardware. As this was not in my budget, I kept looking. I even considered writing my own operating system, but drew the line at having to write my own assemblers and compilers. I enjoy developing operating systems, but those other programs seem too much like work. I also had a large investment in software that I did not want to throw away.

My next step was to purchase the manual set for MP/M 1.0. I read it through many times. There was really not much information there, but there was enough to get some ideas. Sometimes I think the mark of a good systems programmer is the ability to correctly read between the lines in a system manual. There was one other Z80 operating system that looked good, but it was not CP/M compatible, so I decided to go with MP/M. I felt that MP/M would be a good start and that I would be able to change some of the things I did not like about it. I heard through friends in the CP/M users group that Digital Research was just about to release MP/M II version 2.0, so I requested a spec sheet on MP/M II and fell in love with it (relatively speaking). MP/M II had many more of the features that I required of my operating system.

MP/M II arrived a month and several hundred dollars later. Three hours after coming home with package in hand, I had MP/M II running. I cheated, though, as I had already done my loader BIOS and XIOS from information on the MP/M 1.0 manual. The changes required for MP/M II were few. The MP/M II manuals are much better than the ones for 1.0, and they are seldom wrong. I was able to learn a great deal about the system internals while I waited for more memory boards to arrive. Unfortunately that was about all I could do with it anyway in a 64K system. You will need at least 112K for MP/M II.

Tom Clodfelter, The Arecibo Observatory, P.O. Box 995, Arecibo, PR 00612

MP/M II has many of the features that are needed in a multiuser system. It has record and file locking for shared file access. It provides date- and time-stamping of files. It also has a well-thought-out set of internal data structures through which most of the system services are provided. It provides for event flags, message queues, and real-time processes. But any operating system is a collection of features and faults; MP/M II is no exception.

Digital Research says that MP/M is for a nonhostile environment, and they are right. The operating system is not protected from a user program. This is the most serious shortcoming of MP/M as a multiuser system. I feel, however, that there is a real difference between a nonhostile environment and one that is asking for trouble. Any multiuser system should have a means to keep out unwanted users. Then and only then can you assume that you have a nonhostile environment. In these days of the computer connection, if you are running an MP/M system you may want to have at least one dial-up line. Then you'll want to make sure that only the people you want in the system can get in. You should also be able to find out who is on the system, and send messages to them. Although MP/M II does have a password protection feature for files and disk volumes, it is hard to use in a true multiuser implementation. This will be discussed in more detail in a later article.

The next serious drawback to MP/M was that it had a very poor SUBMIT facility. If you are familiar with CP/M's SUBMIT and XSUB you may be surprised to find out that MP/M's SUBMIT is nowhere near as elaborate. There are several SUBMIT replacements available now, but most of them will not work correctly under MP/M. Even the ones that did work under MP/M fell far short of what I wanted, as MP/M has no XSUB facility at all. With XSUB you can feed input to a program from a file. It only works, however, if the program uses the buffered input system call.

I also wanted a good background batch processor that would run a job or series of jobs in the background *without* tying up a console, feed input to this job from a file, then save the output into another file in a way that showed me when it ran, how long it took, and what it did. Any multiuser and/or multitasking system should provide this function. It turned out to be easy under MP/M II.

A real printer spooler is something nice to have, too. MP/M comes with a spooling package, but it is really just a despooling package. The MP/M SPOOL package allows you to print an existing file, but a good spooler should be able to manage multiple inputs for single or multiple printers. If the printer is busy, the spooler should take any printer output sent to it and store it in a temporary file until the printer is free without any action on your part. As far as the user is concerned, the file goes straight to the printer, and the fact that the spooler is acting like a traffic cop with the printer(s) should be obvious.

I want my operating system to be able to tell me as much as possible about a task running on the system. MP/M provides much and maybe even most of the information that I want, but Digital Research does not provide a utility

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 to look at much of this information. The MPMSTAT utility that they do provide tells me everything I don't want to know about the system.

I felt that any real-time multitasking system should have languages that support its advanced functions. I wanted to be able to write a real-time program with a Basic interpreter, and I wanted a library of functions for the C and Fortran compilers that I use.

Computers are cheap these days. If all you want is several people running programs under CP/M, then you should buy each person a computer. But a good multiuser, multitasking system should provide the many extra facilities that permit several programs to run concurrently, interacting with each other while sharing files and other system resources. MP/M really has a lot to offer in this area. It goes far, but with a little work it can go even further.

These, then, were some of my requirements: protection from unwanted users, an advanced SUBMIT with background batch processing, a sophisticated and transparent spooler, a system status utility, and enhancements for Basic, C and Fortran. If you are an MP/M user, you may have noticed by now that MP/M does not have these features—you must think I'm describing some other operating system. Not so. Even though MP/M does not have them, they can be added. The system itself has most of what it needs to support all of the features that I have mentioned. All it takes is some knowledge of the system and a little programming effort. This series of articles will show you how to turn your present MP/M II system into a real MP/M II system. I have solved all these problems, and am finding more and more exciting things to do with MP/M each day.

It will not be necessary for you to understand all of the MP/M II data structures and system services in order to implement some of the enhancements that will be described in later articles, but it will help. With this in mind, let me give you a short introduction to the system.

Under MP/M II the system is divided into system and user memory segments. If you are running a banked system, the top portion of memory must be common to all banks. In most systems this is a 16K block; since the required resident portion of the system is about 13K, this is just about right. It leaves you with a maximum user memory segment size of 48K, which is enough for most applications. As most of bank 0 is taken up with the resident and banked portion of the operating system, this requires that you have at least one 64K bank and one 48K bank to do much with MP/M. It is possible to add what is known as a Resident System Process, or RSP, to the system in such a way that it resides in bank 0 as part of the operating system image. RSP programs make very nice additions to the operating system, but they also make the system large. Keep in mind that the price of memory is going down. I tell myself this several times a day.

Each process (program) in the system has to be associated with a system data structure known as the Process Descriptor Block, or PDB. The PDB must be unique for each process, and it must reside in the common portion of system memory. For a normal user program the system will create the required PDB on execution of the program. If you write an RSP you must create the PDB for it yourself. Most RSP programs consist of a resident module (RSP) and a Banked Resident System module (BRS). The resident portion usually consists of only what must reside in common memory, such as the PDB and queue buffers. The program code and stack space are in the system bank. The Process Descriptor Block is a gold mine of information about the executing program; it is how the system knows what to do with the program. It also contains the register save area for the program while it is not running. In multitasking systems the processor really only runs one program at a time, but it switches (context switching) between the programs, giving each a little time on the processor (time sharing) so rapidly that it gives the illusion of simultaneous execution. See Figure 1 for a description of the Process Descriptor Block.

Another important system data structure for our purpose is the system queue data structure. Under MP/M a queue is a named buffer that programs may read from and write to. The size and number of messages that a queue will accept is determined by the Linked Queue Control Block (LQCB). In order to access a queue, the user program must open it, somewhat like opening a disk file. The user program opens the queue by passing the address of another data structure to an appropriate system call. This data structure is known as the User Queue Control Block (UQCB). This control block contains the name of the queue to be opened and the address of the LQCB, if known. If the address is blank, the system will fill it in.

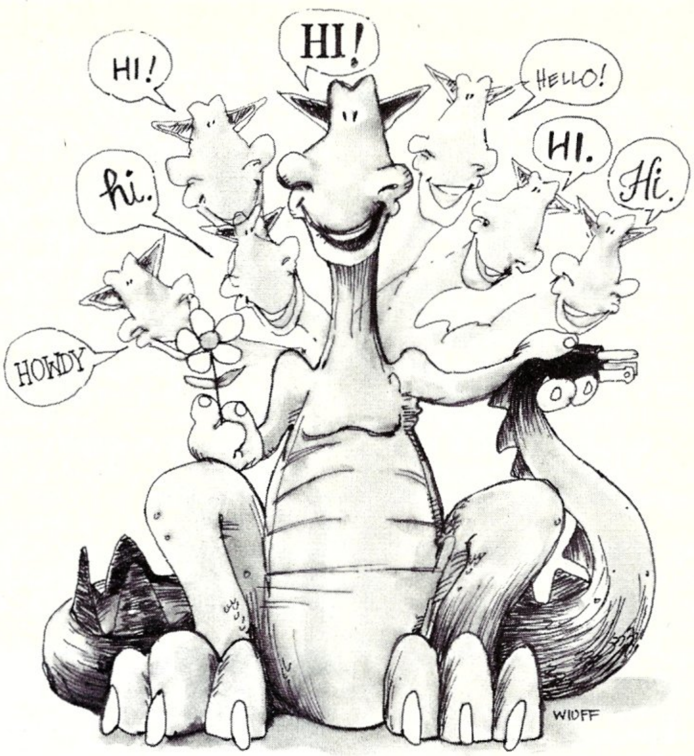
MP/M supports several different types of queues. Linked queues are ones where the messages are longer than two bytes and are pointed to by a linked list of pointers in the QCB. Circular queues are ones in which the messages are less than three bytes long and stored in a circular buffer. Mutual exclusion queues have a special significance to the system. They make up a special type of queue that assists in resourcing nonsharable programs or devices. See Figure 2 for an example of LQCB and Figure 3 for UQCB.

When you write to the queue that you have opened, the system copies the message from the buffer addressed by your UQCB into the next free location in the buffer addressed by the LQCB. If the queue is full, then the system blocks further execution by the enqueuing process until a free location in the queue is available. If your program is trying to read from a queue, it is also blocked if the queue has no message. There are also system calls that return an error code if the queue if full or empty instead of allowing the system to block execution.

Two more important system data structures are the System Data Page (Figure 4) and the Internal Data Segment (Figure 5). These data areas are the most important part of the operating system. They provide the critical information and work areas for the system. The Internal Data Segment (IDS) contains the root address for all of the processes running in the system. It also contains the PDB area for all user memory segments. The System Data Page (SDP)

With a little work, MP/M can go far, providing password protection, an advanced SUBMIT with background batch processing, and enhancements for Basic, C, and Fortran.

SAY HELLO TO HYDRA.



SAY GOODBYE TO DATA NETWORK AND COMPUTER PROBLEMS.

CRC's new HYDRA Computer System has more of everything:

More processors—Up to seven SBC's

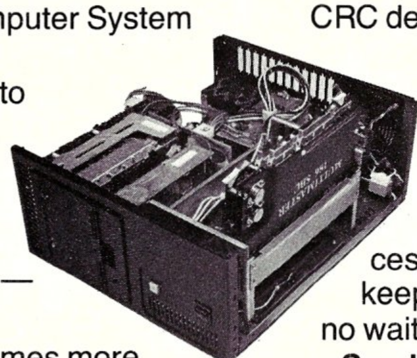
More I/O ports

More memory—256 RAM

More users availability—up to 28 activities

More power—5 to ten times more

More disk memory—Winchester and 5¼ inch floppy



CRC designed the HYDRA around Intel's 80186 8 MHz microprocessor chip to make it the best customized S100 bus computer system available today.

Stand alone or multi processor, multi user, the HYDRA keeps you up and running, with no waiting for more than 20 users.

So, don't buy another piece of computer equipment until you try the HYDRA. And then you won't need any more.

CRC COMMUNICATIONS RESEARCH CORPORATION

A SUBSIDIARY OF ENERGY SCIENCES CORPORATION
1720 130TH AVE. N.E. BELLEVUE, WA 98005 (206) 881-9550
OUTSIDE WASHINGTON, HAWAII & ALASKA — 1-800-426-8075

CIRCLE 10 ON READER SERVICE CARD

contains the system generation values and the pointer to the IDS.

A system like MP/M is made up of many different linked lists. You can find out almost anything you want to know about the system by following the links. For example, if you want your program to be able to find out at what priority it is running, do a system call to get your PDB address. The system will return the address of the currently executing process PDB. This, by definition, is your program (think about it). You can see from the PDB layout (Figure 1) that the priority is in the 4th byte of the PDB. Peek at PDB+3 and the magic is done. You now have the priority. Any of you C freaks out there will appreciate that argv[0] is PDB+6 through PDB+14. This is something you can not get at all from CP/M. A program can even change its name on execution. A little longer look at the PDB will show that you can write a program that can look at what is (was) in the CPU registers of another program while it is (was) executing.

By now you may see that MP/M has very good potential. It is not a *big* system, but then I do not have a big computer. I may go to UNIX for my 68000, but it is overkill for my Z80. The main weakness left in MP/M is that the operating system is not protected from the user, but some of my friends and I are working on that one.

The next installment in this series will cover the implementation of a background batch processor for MP/M II. The source for a simple version of my batch processor will be included with the article. Future installments will cover a replacement for the TMP, a terminal message broadcast system, user accounting, a true printer spooling system, named directories and devices, replacements for many MP/M utility programs, a replacement for the XDOS, and an MP/M-oriented Basic. The TMP and XDOS will handle multiple commands on a line, pipes, and I/O redirection. These functions will be supported by the operating system and not the programs themselves.

The new spooling package, the Terminal Message Process (TMP) replacement, utilities, and RealTime Basic should soon be available as products, but the articles in this series will give the methods for all of my MP/M products for those who would rather roll their own. I hope to show those of you who have MP/M II that it is a very exciting operating system for a microcomputer and to convince those of you considering it to take the plunge.

Before I close, I must mention that I have now seen and used CP/M Plus, and it has many of the nice features which CP/M 2.2 lacked. It is still, however, only a single-user (task) system, and I want more than that. Most of you will understand what I mean when you see several terminals hooked to your computer, all doing different things.

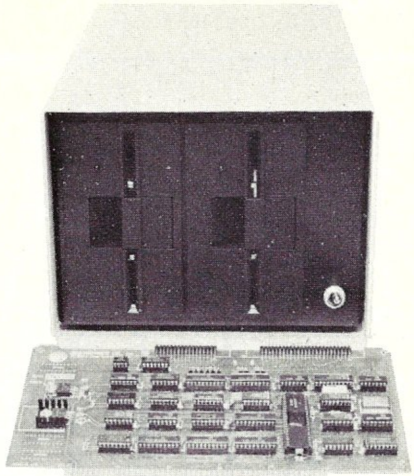
Tom Clodfelter is a senior software engineer at the Arecibo Observatory in Puerto Rico. He is one of the developers of the SemiDisk product and has over seven year's experience in computer programming.

Figure 1.

```
#define BYTE char        /*  8 bit value, 1 byte */
#define WORD unsigned    /* 16 bit value, 2 bytes */

/***********************************************
 *                                             *
 *  PDB Process Descriptor Block Data Structure *
 *                                             *
 ***********************************************/

struct _pdb $
      struct _pdb *nextproc;  /* link to next process */
      BYTE  status;           /* status byte */
      BYTE  priority;         /* process run priority */
      WORD  stack             /* pointer to stack top */
      BYTE  name[8];          /* process name */
      BYTE  cons_lst;         /* console and list numbers */
      BYTE  memseg;           /* memory segment */
      WORD  dparam;           /* dparam */
      WORD  thread;           /* process thread */
      WORD  dma;              /* dma address */
      BYTE  dsk_usr;          /* disk and user number */
      WORD  dcnt;             /* dcnt */
      BYTE  searchl;          /* searchl */
      WORD  searcha;          /* searcha */
      WORD  pdextent;         /* extent */
                              /* the register storage area follows */
                              /* for Z80 and 8080 registers */
      WORD  _HL;              /* HL' */
      WORD  _DE;              /* DE' */
      WORD  _BC;              /* BC' */
      WORD  _AF;              /* AF' */
      WORD  IY;               /* IY */
      WORD  IX;               /* IX */
      WORD  HL;               /* HL */
      WORD  DE;               /* DE */
      WORD  BC;               /* BC */
      WORD  AF;               /* AF */
      WORD  scratch;          /* scratch space reserved for MP/M II */
};
```

Figure 2.

```
#define BYTE char        /*  8 bit value, 1 byte */
#define WORD unsigned    /* 16 bit value, 2 byte */

/***********************************************
 *                                             *
 *  LQCB Linked Queue Control Block            *
 *                                             *
 ***********************************************/
```



DISK MAKER™ MAKES *FIFTY* DIFFERENT DISKS

Solving the 5¼" Disk Hassle

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

Solution: **Disk Maker I**

YOUR BENEFITS

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

DISK MAKER II™

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

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

What is included?

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

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

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

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

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

NEW GENERATION SYSTEMS, inc.

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

(703) 471-5598 • **(800) 368-3359**

CIRCLE 123 ON READER SERVICE CARD

```
struct  _lqcb §
        struct _lqcb *nextque;   /* pointer to next linked que */
        BYTE   name[8];          /* 8 byte queue name */
        WORD   msglen;           /* message length */
        WORD   nmbmsgs;          /* number of messages */
        WORD   dqph;             /* DQPH  dequeue process head */
        WORD   nqph;             /* NQPH  enqueue process head */
        WORD   mh;               /* MH */
        WORD   mt;               /* MT */
        WORD   bh;               /* BH */
†;
```

Figure 3.

```
#define BYTE char        /*  8 bit value, 1 byte */
#define WORD unsigned    /* 16 bit value, 2 byte */

 /*******************************************************
  *                                                     *
  *  UQCB User Queue Control Block                      *
  *                                                     *
  *******************************************************/

struct _uqcb §
        struct  _lqcb *lnkque;  /* WORD pointer to actual queue */
        struct  _qmsg *msg;     /* WORD pointer to message area */
        BYTE    name[8];        /* 8 byte queue name */
†;
```

Figure 4.

```
#define BYTE char        /*  8 bit value, 1 byte */
#define FLAG char        /*  00h = false  0ffh  = true */
#define WORD unsigned    /* 16 bit value, 2 byte */

 /*******************************************************
  *                                                     *
  *  System Data Page structure                         *
  *                                                     *
  *******************************************************/

struct _sysdata §
        BYTE    memtop;            /* top page of memory */
        BYTE    nmbcns;            /* number of consoles */
        BYTE    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 */
        BYTE    xiosjmp;           /* xios jump table page */
        BYTE    resbdos;           /* resbdos page */
        WORD    cpnetadr;          /* cp/net config table address */
        BYTE    xdospage;          /* xdos page address */
        BYTE    rsppage;           /* RSP's (bnkxios top+1) base page */
        BYTE    bnkxiospg;         /* banked xios page address */
        BYTE    bnkbdospg;         /* banked bdos page address */
        BYTE    maxmemseg;         /* maximum memory segment number */
        struct  _memseg mem[8];    /* initial memory segment table  */
        BYTE    bkpointvec[16];    /* breakpoint vector table */
        BYTE    reserved[16];      /* reserved for MP/M II */
        BYTE    stktbl[16];        /* stack pointer table */
        BYTE    reserved2[24];     /* reserved for MP/M II */
        WORD    numbrecs;          /* number of records in MPM.SYS file */
        BYTE    nmbticks;          /* number of ticks per second */
        BYTE    systemdrive;       /* system default drive */
        BYTE    commonpage;        /* common memory base page */
        BYTE    numbrsps;          /* number of resident system processes */
        WORD    listcp;            /* listcp array address */
        BYTE    submitflg[16];     /* submit flag array */
        BYTE    reserved3[43];     /* reserved for MP/M II */
        BYTE    maxlocked;         /* max locked records/process */
        BYTE    maxopen;           /* max open files/process */
        WORD    nmblocked;         /* number of locked list items */
        WORD    *locktbl;          /* pointer to lock table free space */
        BYTE    totallocked;       /* total system locked records */
        BYTE    totalopen;         /* total system open files */
        FLAG    dayfile;           /* dayfile logging boolean */
        BYTE    tempdrive;         /* temporary file drive */
        BYTE    numblpt;           /* number of printers */
        BYTE    reserved4[44];     /* reserved for MP/M II */
        BYTE    bnkxdospg;         /* banked xdos page address */
        BYTE    tmpbase;           /* tmp pdb base */
        BYTE    cnsdatbase;        /* console.dat base */
        WORD    bdosentry;         /* bdos/xdos entry point */
        BYTE    tmpsprbase;        /* tmp.spr base */
        BYTE    nmbbnkrsp;         /* number of banked RSP's */
        BYTE    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 */
†;
```

Figure 5.

```
#define BYTE char        /*  8 bit value, 1 byte */
#define FLAG char        /* 00 = false, 0ffh = true */
#define WORD unsigned    /* 16 bit value, 2 byte */

 /*******************************************************
  *                                                     *
  *  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 */
        WORD    thread;            /* thread root */
        BYTE    nmbcons;           /* number of consoles */
        WORD    attachtbl[16];     /* console attach table */
        WORD    consque[16];       /* console que */
```

Enhancing MP/M II continued . . .

```
        BYTE      nmbflags;           /* number of flags */
        WORD      sysflags[32];       /* system flags list */
        BYTE      nmbseg;             /* number of memory segments */
        struct    _memseg mem[8];     /* memory segment table */
        struct    _pdb pdtable[8];    /* pdb table for memory segments */
        BYTE      nmblpt;             /* number of list devices */
        WORD      lstattach[2];       /* list attach table */
        WORD      lstque[2];          /* list que */
};

/* tod data block */

struct _tod $
        WORD      date;               /* 16 bit days since Jan. 1, 1978 */
        BYTE      hour;               /* BCD hours */
        BYTE      minute;             /* BCD minutes */
        BYTE      second;             /* BCD seconds */
};

/* memseg memory segment table structure */

struct _memseg $
        BYTE      base;               /* memory segment base page */
        BYTE      size;               /* memory segment size in pages */
        BYTE      attributes;         /* memory segment attributes */
        BYTE      bank;               /* memory segment bank number */
};
```

Getting Started with MP/M II: Installing a Basic MP/M System

by Dave Hardy and Ken Jackson

Although MP/M II has been around for quite a while, it has not shared the popularity of its single-user predecessor, CP/M. One reason for this lack of acceptance is undoubtedly the fact that a more sophisticated machine is required to operate under MP/M. Another reason is that MP/M is a much more complicated operating system; one that makes use of multiple banks of memory, interrupts, clocks, timers, and sophisticated I/O. At least, that's what you'd think after a brief look at the MP/M manuals, or a look at a complete banked XIOS source listing.

Actually, neither of these reasons is completely true. In fact, MP/M can be installed, in a minimal system, in virtually any CP/M machine. The only restriction is that the machine be able to run a CP/M system big enough to load MP/M without overwriting itself, which usually means something greater than about 56K of RAM is required. Even a smaller (that is, less RAM) machine could be used, but some fancy programming might be required to load MP/M.

Unless ycu have a 64K CP/M machine, chances are that a minimal MP/M system will be almost completely useless to you. The TPA is so small that only simple programs will be able to run in it. (The XIOS provided here will give about a 32K TPA with two consoles, for example.) Running WordStar is definitely out. So is doing most anything else with any redeeming social value (unless all you want to do is list directories or erase files).

The only real functions of a minimal MP/M system are to get you familiar with MP/M, and provide you with a stepping stone up to the next level, which in this case would be adding interrupts and then banked memory. And adding interrupts and banked memory is deceptively simple, once you understand the basic MP/M implementation.

Installing a minimal MP/M system

Bringing up MP/M on a CP/M-based machine is probably easier than most people think. It is even easier than bringing up a simple, nonbanked CP/M Plus system.

Starting with the source for a CP/M 2.x BIOS, a single evening should be all that is needed to install a minimal (i.e., working, single-user, nonbanked, non-interrupt-driven) MP/M system. Once the basic system is working, the remaining features of MP/M (like interrupts and banked memory) can be added with just a small amount of hair-pulling, and, of course, a few bucks for all that extra RAM.

As usual, the first thing to do is to read the MP/M manuals. Five manuals are provided with MP/M II: the MP/M System Guide, the MP/M Programmer's Guide, the MP/M User's Guide, the MAC Language Manual, and the Link-80 Operator's Guide. All together, these provide about 800 pages of useful information.

Fortunately, the MAC and LINK manuals can be read quickly and set aside. Unfortunately, they are the smallest of the five manuals. The three MP/M manuals should be read completely, of course, with particular attention given to the System Guide, since it contains most of the information needed to install MP/M.

The System Guide provides two sample XIOSes: a simple, nonbanked XIOS for the "micro-2" computer, and a fairly complex banked XIOS for an Altos 8000. The former is straightforward and should provide a good idea of what needs to be done for a simple installation. The latter is extremely machine specific, and can be more confusing than useful to the first time implementor. The System Guide also provides a sample Loader BIOS, which is basically just a stripped-down CP/M 2.2 BIOS that is overlayed onto the MPMLDR.COM file provided with MP/M II.

The following is a five-part step-by-step procedure for bringing up a simple (nonbanked) version of MP/M on an existing CP/M 2.2 system, using the version 2.2 BIOS.

Part 1: Make the MP/M XIOS from the CP/M BIOS

Step 1: Make your BIOS relocatable. The first thing that has to be done to any BIOS before it can be made into an XIOS is that it must be made compatible with a RELOCATING assembler. Any relocating assembler will do, as long as it can make a Digital Resarch LINK-80 compatible relocatable (.REL) file. Because RMAC is supplied with MP/M II, it is the obvious choice if you don't already have a relocating assembler, but many others (e.g., MACRO-80) will work.

Usually, nothing else has to be done to make a BIOS relocatable, but in some cases, certain code may have to be modified to prevent assembler errors. Of course, most absolute origin declarations (like ORG or .LOC or whatever) must be changed. Usually, this means replacing the ORG statement with a CSEG statement, as shown in Listing 1. Be careful not to change any origin declarations that reference code that *is* absolute. For example, many disk controllers us an EPROM that is (obviously) not relocatable, and BIOSes furnished with them may contain code that is acutally located in the EPROM.

The easiest way to see if your BIOS is RMAC compatible is to assemble it with the relocating assembler, and watch for errors. Then look over the .PRN file to make sure that nothing is missing or in the wrong place.

Step 2: Remove your BIOS jump table and boot routines. The jump table will be replaced in the next step. The warm boot and cold boot routines are replaced with simpler code at the same time. Note that the new warm and cold boot codes are contained in the code in Listing 1.

Step 3: Add the code shown in Listing 1 and modify for your I/O. This is actually easier than it looks, because Dig-

Dave Hardy, 736 Notre Dame, Grosse Pointe, MI 48230

ital Research provides simple XIOS source on the MP/M distribution disks (in the file RESXIOS.ASM) so that you can painlessly extract the code shown in Listing 1 with your favorite disk editor and just plug it into your BIOS. Some changes will have to be made to make the code extracted match the code shown in Listing 1, but not much.

Step 4: Modify for your consoles and list device. Modify the "Input/Output Port Address Equates" section to match your console port addresses and change the console and listing device I/O routines to meet your needs (check the status bits and polarity for your I/O ports, and change the file to match them, etc.). Note that in Listing 1, the list device is for a Morrow Designs DJ2D controller board's memory-mapped I/O port. Unless you have a DJ2D, insert your list device drivers instead. None of the other code in Listing 1 should need to be modified to match your system. Be sure to change the status bits used in the con0out:, polco0:, conlout:, polcol:, etc., routines to match your system's I/O status bit assignments.

Step 5: Add the new storage and stack space. Add Listing 2 to your BIOS to provide an interrupt routine stack (even though you won't need interrupts for this simple implementation), register storage, and some other miscellaneous parameters.

Step 6: Make some miscellaneous modifications, as needed. None of the standard BIOS routines should require modification to work with MP/M II. You should, of course, change the names in the new jump table to match the names of the original routines in your BIOS. For example, your Select Disk routine may be called SELDSK instead of SETDRV, as it is called in Listing 1.

Those names listed in the jump table (see Listing 1) that require modifications or additions to your BIOS are shown in upper case. Note that the first jump in the table, which used to be the cold boot jump, is now a jump to COMMONBASE (located just below the jump table), and the second jump, which used to be the warm boot jump, is now a jump to the WARMSTART routine, which is included in Listing 1.

The CP/M PUNCH and READER are not used in MP/M, so they can be eliminated from your BIOS, if you wish. List Device Status is not really needed, so it can be replaced with a RET instruction if you like.

The remaining seven jumps shown in upper case are all included in the code in Listing 1. Any system initialization you wish to perform should be included into the SYSTEMINIT routine. The three bytes following the SYSTEMINIT jump are used to tell MP/M that there is no user-provided Idle procedure. The Idle procedure is a routine executed by an interrupt-driven MP/M system when it literally has nothing to do. Since this is *not* an interrupt-driven implementation, the system is always busy doing something. Therefore, these three bytes must be zero.

Step 7: Assemble the XIOS, fix any errors, and link it. You will probably see some errors the first time you assemble your XIOS. Most will be simple syntax errors, or unknown labels (usually from the location where the warm or cold boot routines were removed, or from unrecognized labels used by your old jump table).

Be sure to set the DEBUG equate to TRUE before you assemble the XIOS, so that your XIOS will work without interupts. After you have brought up MP/M and debugged your XIOS thoroughly, you can install the interrupt code and set the DEBUG equate to FALSE to use the interupt features of MP/M. Note that until you add interrupts, only console 0 will work. Other consoles will appear "dead" because there will be no interrupt services to notify MP/M that they are sending input characters.

Finally, LINK your XIOS (using the "[OS]" option to make a system page relocatable file) with the following command:

```
LINK XIOS[OS]
```

At system generation time, MP/M's system generation program GENSYS will expect to see your XIOS as a file called RESXIOS.SPR, so be sure to rename your XIOS before using GENESYS.

Part 2: Make the loader BIOS and MPMLDR

This is the simplest part of the whole MP/M installation. All you have to do is change the origin of your BIOS to 1700H, assemble it, and overlay it onto the MPMLDR.COM file supplied with MP/M II.

The only things required of a LOADER BIOS are that it be able to perform console output, and that it be able to perform disk reads. So, as shown in Listing 3, you can remove the cold boot, warm boot, list, punch, reader, disk write, and list status routines, if you wish, to make the file smaller. There is really no restriction on the size of the LOADER BIOS, however; so you can just leave all that stuff in if you like. The only time you may wish to remove it would be if it took up too much space in the TPA, or if you wanted to put MPMLDR on the system tracks to allow start-up from reset (in which case, you would have to make MPMLDR smaller than 1A00H bytes to fit on a single-density floppy.

Assemble the LOADER BIOS the same way you would normally assemble your CP/M 2.2 BIOS. It does not need to be made relocatable. The end result should be a .HEX file which can be read in on top of the standard MPMLDR.COM file. The procedure to do this is shown in Listing 4.

The number of pages saved in the above command (26 above) can be calculated by converting the first two digits of the "next" value (1AA6H above) to decimal.

Be careful to save your customized loader as something other than MPMLDR.COM (XMPMLDR is recommended), unless you happen to be a very good programmer.

Part 3: Generate the MP/M system

The next step after assembling and linking the RESXIOS is to generate an MP/M system. MPM.SYS is the file that actually contains the MP/M operating system and is loaded into memory by the MPMLDR program.

Generating the MPM.SYS file is the function of MP/M's GENSYS program. A typical system generation dialog for a system using a simple, nonbanked, noninterrupt scheme like the one described here is shown in Listing 5.

All of the critical (i.e., not already set as a default) entries in the above system generation are marked with "<—Note" flags. If you have any reserved (that is, unuseable) memory at the top of your system, such as a memory-mapped disk controller or a boot PROM, you can

Bringing up MP/M on a CP/M-based machine is easier than most people think. It is even easier than bringing up a simple, nonbanked CP/M Plus system.

tell GENSYS to exclude that area by setting the top page of the operating system.

The number of TMPs, which is actually the number of consoles that you wish to be used in the system being generated, should be set at one or two. With the minimum system used here, it doesn't really matter how many consoles are selected, except that each additional console will use additional memory space in the MPM.SYS file. The minimal system generated here can have only one console when run without interrupts, and only one or two when run with interupts. Of course, under interrupts, additional consoles can be easily added.

It is probably a good idea to enable compatibility attributes in your first MP/M system, so that you will be less likely to have any trouble with some programs that weren't written with MP/M's file security system in mind.

No banked-switched memory is used in this simple implementation, so you should answer "N" to the "Bank switched memory?" question. The number of user memory segments selected should be one the first time you generate a system, so that you will have the maximum size TPA available to work with.

Dayfile logging is a simple time-stamp printed on the console each time a program is loaded into memory. Because there are no interrupts in this simple system, MP/M's internal clock is not valid, so this option is of no use until interrupts are implemented.

None of the RSP files (ABORT, MPMSTAT, SCHED, SPOOL) should be included the first time, since each takes up space in memory, and therefore would make the TPA smaller. These can all be linked in later with GENSYS, if desired, after interrupts are implemented.

The memory segment table contains the list of memory blocks that are available to the system, including the start, end and type of each block. The first entry in the memory segment table is always the block of memory that MP/M itself resides in. MP/M will automatically enter the base (start), size, and attributes for this first bank. It should not be changed. Just hit a RETURN, so that GENSYS will go on to the next question, which asks for the next memory segment. Since only one user memory segment was selected previously in GENSYS, this last memory allocation should use the rest of the available memory, which means that its base would be 0, and its length would be 8FH pages (because that's where the MP/M system begins). The attribute byte should be 0 for all user memory segments. In a banked system, a fourth byte, called the BANK Number, is also requested, which is simply the number of the bank in which the current memory segment resides.

As GENSYS links together each of the modules used to make MPM.SYS, it prints the module's name, followed by its starting address and length. It may be a good idea to write each of the modules's addresses down as they are linked, for use in later debugging or system "exploration" that you may wish to do.

After GENSYS is completed (it will announce when it has finished), you should be ready to run the XMPMLDR.COM file to load and automatically execute MP/M II. When you type "XMPMLDR", you should see the MP/M loader list the modules of the MP/M system as it installs each in its proper place in memory, followed by a copyright message from Digital Research, and finally, the familiar command prompt, (A0>).

Part 4: Debugging

Because of the many modifications and operations performed in the generator of your first MP/M system, don't be surprised if it doesn't work the first time. In fact, if it does, count yourself among the world's elite system implementors. Digital Research, keenly aware of the fact that only its programmers never make mistakes, has cleverly included a few good things to help you at least get a general idea of where your system is messing up.

The first thing to determine, if your MP/M system won't boot, is whether or not the loader is working. If MP/M isn't loaded into memory in the right place, it won't work. DR has included a special breakpoint option in MPMLDR that will allow you to load MP/M with it under a debugger, then branch back to the debugger after MP/M is loaded, before MP/M is actually executed. Using this feature, you can at least determine if your XMPMLDR.COM program is working properly. If not, then you should check both your Loader BIOS and the XMPMLDR.COM.

If nothing is printed on the screen after you execute XMPMLDR, then the problem MUST be in the loader. If you see the MP/M load map, as described above, followed by a copyright notice, then it's a good bet that the loader is working OK. Once the copyright notice is printed, your machine is actually running MP/M, so if you get that far, then any problems are most likely in the MPM.SYS file. If you see the system prompt (A0>), but can't input from the keyboard, then the problem is most likely in your console input routines. Also be sure to check that the DEBUG equate in your XIOS is set TRUE. If you don't run in the DEBUG mode, then the system will only read console input after an interrupt, and of course, there are no interrupts in this simple system, so there will be no console input.

If the system dies when control is passed to MPM.SYS, then the problem is most likely in your XIOS. Here are some to check for that might go unnoticed:

- Check that any system initialization that you are doing in the SYSTEMINIT: routine of your XIOS can be done while running CP/M. Sometimes, you can't re-initialize a machine while it's running a program under CP/M (which is what is happening when you execute XMPMLDR.COM).
- Check the values entered into the GENSYS program. Make sure that you have specified the proper top of memory if you have any EPROMs or "holes" in your system memory.
- Make sure that no interrupts are being generated by any system components. Although this simple implementation of MP/M doesn't use interupts, MP/M itself does enable interrupts upon returning from certain system functions. You may have to turn off an interrupt generator, or add some code to the interrupt handler portion of your XIOS to keep the system from blowing up. (It is possible, however, to run an "un-interrupt-driven" version of

MP/M will add multiuser abilities to almost any 8080, 8085, or Z80 system, but at a considerable system overhead.

MP/M with an interrupt-driven XIOS. Many CP/M systems use interrupt-driven BIOSes in the same way.)

- Make sure that the RESXIOS.SPR file linked in by GENSYS is the one you made, and not one supplied by Digital Research as an example. This sort of problem can be more fun than debugging MicroSoft Basic compiler outputs.
- More information on debugging an XIOS is included in the MP/M II System Guide in section 1.4. Bear in mind that it is often the really dumb mistakes that cause the most trouble.
- Be sure to *thoroughly* test your version of MP/M to make sure that there are no hidden troubles. You should, at least, make sure that the system can read from and write to the disk, and that all of the system utilities (STAT, PIP, etc.) work OK. Any problems that you miss now will come back and haunt you later when you add interrupts and multiple banks of memory, because the system will be a great deal more complex.

Part 5: What to do after it works

After everything is working properly, the first thing that you should do is add an interrupt generator, write a simple interrupt handler for your XIOS (it goes into the INTHD: routine, remember?), and get an interrupt-driven version of MP/M running. If your system doesn't have any interrupt ability, an excellent source of information on how to build and program a simple interrupt generator is *Interfacing to S-100/IEEE 696 Microcomputers,* by Sol Libes and Mark Garetz (Osborne/McGraw-Hill, 1981). Its circuits and programs are applicable in this case, even if you don't have an S-100 machine: build the circuits on perfboard and mount this on spacers glued to the motherboard.

You will probably have to add some interrupt control code to the disk routines in your XIOS, and possibly to some other parts, too. If your disk controller generates interrupts and doesn't mess with the bus while it's performing disk I/O, then the Simple XIOS Source provided in the MP/M System Guide will probably provide you with a good model for installing interrupts. If, like many S-100 users, your disk controller "takes over" the bus during disk I/O and is sensitive to external interrupts, you will probably find the DJ2D source mentioned below to be more helpful.

Either way, adding interrupts to the simple MP/M system is by far the most difficult task you will have to face to complete your MP/M system. After interrupts are added, multiple banks (or at least one full 48K bank of memory) should be added, so that more than one task can be performed at the same time.

To make life easier for you once you have installed the basic non-interrupt-driven, nonbanked system, a copiously commented version of an interrupt-driven, multiple-banked MP/M system for the Morrow Designs DJ2D floppy controller is available. The DJ2D is probably one of the most difficult controllers on which MP/M can be implemented, and it provides many good examples of how to work with a memory-mapped device, how to work around an EPROM, and how to deal with a board that can't tolerate interrupts.

Closing notes

The above techniques have been used to bring up simple versions of MP/M on several machines—including several mixed S-100 machines, a Sanyo MBC-1000, and Advanced Digital SuperQuad and a Digital Group system—so they should be of at least some help to the first-time MP/M installer. All of the examples used here are from the DJ2D XIOS, because it is most representative of a typical CP/M machine.

Space does not allow the entire DJ2D nonbanked and banked XIOSes to be listed here, but they are available (along with the DJ2D Loader BIOS) on several RCP/M systems around the country, and also on CompuServe's CPMIG CP/M bulletin board.

One of the most difficult decisions to make regarding MP/M is whether or not you really need it. MP/M will add multitasking, multiuser abilities to most any 8080, 8085, or Z80 system, but at the cost of increased system overhead and additional expense. MP/M systems, by nature, run effectively slower than single-user, single-tasking systems that require no sharing of CPU time or resources. In addition, many programs (especially "universal" utility programs) that work fine under CP/M can't be used with MP/M. Ultimately, it is really up to the user to decide whether MP/M's advantages outweigh its disadvantages.

Either way, if you decide to install MP/M, this article should at least help you get started.

Listing 1

```
        CSEG                        ;Mark as Code Segment
        org     0000H               ;Origin at 0 for relocation

        jmp     COMMONBASE
        jmp     WARMSTART           ;Warm start entry point
        jmp     const               ;Console status routine
        jmp     conin               ;Console input
        jmp     conout              ;Console output
        jmp     list                ;List device output
        jmp     RTNEMPTY            ;PUNCH NOT IMPLEMENTED
        jmp     RTNEMPTY            ;READER NOT IMPLEMENTED
        jmp     home                ;Home drive
        jmp     setdrv              ;Select disk
        jmp     settrk              ;Set track
        jmp     setsec              ;Set sector
        jmp     setdma              ;Set DMA address
        jmp     read                ;Read the disk
        jmp     write               ;Write the disk
        jmp     POLLPT              ;List device status
        jmp     sectran             ;Sector translation
        JMP     SELMEMORY           ;SELECT MEMORY
        JMP     POLLDEVICE          ;POLL DEVICE
        JMP     STARTCLOCK          ;START CLOCK
        JMP     STOPCLOCK           ;STOP CLOCK
        JMP     EXITREGION          ;EXIT REGION
        JMP     MAXCONSOLE          ;MAXIMUM CONSOLE NUMBER
        JMP     SYSTEMINIT          ;SYSTEM INITIALIZATION
        DB      0,0,0               ;FORCE USE OF INTERNAL IDLE PROCEDURE
;
; COMMON BASE for bank-switching (no banks here, but needed anyway)
;
commonbase:
        jmp     coldstart
swtuser: jmp    $-$
swtsys:  jmp    $-$
pdisp:   jmp    $-$
xdos:    jmp    $-$
sysdat:  dw     $-$

coldstart:
warmstart:
        mvi     c,0
        jmp     xdos                ; system reset, terminate process
;
;
;I/O handlers
;
;
;  MP/M II V2.0   Console Bios
;
;
nmbcns  equ     3       ; number of consoles

poll    equ     131     ; XDOS poll function
makeque equ     134     ; XDOS make queue function
readque equ     137     ; XDOS read queue function
writeque equ    139     ; XDOS write queue function
xdelay  equ     141     ; XDOS delay function
create  equ     144     ; XDOS create process function

;
;  Device numbers assigned for use by XDOS POLL function in
;  POLLDEVICE: routine (below).  Note that device number assignment
;  is completely arbitrary, i.e. device order below makes no difference,
;  as long as device number matches device position in device table
;  (called DEVTBL: and located just past POLLDEVICE: routine).
;  In other words, device 0's position in the table must be 0,
;  device 1's position must be 1, etc.
;
pllpt   equ     0       ; poll printer
plco0   equ     1       ; poll console out #0
plco1   equ     2       ; poll console out #1
plco2   equ     3       ; poll console out #2
plci2   equ     4       ; poll console in #2
        if      debug
plci0   equ     5       ; poll console in #0
        endif

;
const:                          ; Console Status
        call    ctbljmp ; compute and jump to hndlr
        dw      con0st  ; console #0 status routine
        dw      con1st  ; console #1 status rt
        dw      con2st  ; console #2 status rt

conin:                          ; Console Input
        call    ctbljmp ; compute and jump to hndlr
        dw      con0in  ; console #0 input
        dw      con1in  ; console #1 input
        dw      con2in  ; console #2 input

conout:                         ; Console Output
        call    ctbljmp ; compute and jump to hndlr
        dw      con0out ; console #0 output
        dw      con1out ; console #1 output
        dw      con2out ; console #2 output

;
ctbljmp:                        ; compute and jump to handler
                                ; d = console #
                                ; do not destroy d !
        mov     a,d
        cpi     nmbcns
        jc      tbljmp
        pop     psw     ; throw away table address
rtnempty:
        xra     a
```

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.

LEARN TO WRITE A SPREADSHEET

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

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

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

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

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

For Visa and MasterCard orders call toll free

1-800-227-2400 X 968

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

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

Dealer & Distributor Inquiries welcome.

Turbo Pascal $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.

CIRCLE 19 ON READER SERVICE CARD

```
        ret
tbljmp:                         ; compute and jump to handler
                                ; a = table index
        add     a               ; double table index for adr offst
        pop     h               ; return adr points to jump tbl
        mov     e,a
        mvi     d,Ø
        dad     d               ; add table index * 2 to tbl base
        mov     e,m             ; get handler address
        inx     h
        mov     d,m
        xchg
        pchl                    ; jump to computed cns handler

;
;
;  Following are the console I/O routines.  These will have to be
;  customized for your machine.  Currently, the LIST device is set up
;  as the DJ2D's memory-mapped I/O port.
;
;
; Input / Output Port Address Equates
;
dataØ   equ     81h
stsØ    equ     8ØH
data1   equ     91H
sts1    equ     9ØH
data2   equ     5Øh
sts2    equ     data2+1
data3   equ     58h
sts3    equ     data3+1
;
;********************** CONSOLE #Ø ROUTINES *************************
;
; Poll Console #Ø Input
;
        if      debug
polciØ:
conØst:
        if      ldcmd
        lda     conØcntr
        ora     a
        mvi     a,Ø
        rnz
        endif

        in      stsØ
        ani     4ØH
        rz
        mvi     a,Øffh
        ret
;
conØin:
        if      ldcmd
        lxi     h,conØcntr
        mov     a,m
        ora     a
        jz      ldcmdØempty
        dcr     m
        lhld    conØptr
        mov     a,m
        inx     h
        shld    conØptr
        ret
conØcntr:
        db      ldcmdØempty-conØldcmd
conØptr:
        dw      conØldcmd
conØldcmd:
        db      'tod '
ldcmdØempty:
        endif
        mvi     c,poll
        mvi     e,plciØ
        call    xdos
        in      dataØ
        ani     7fh
        ret
;
        else
conØst:
                                ; return Øffh if ready,
                                ;        ØØØh if not
        lda     cØinmsgcnt
        ora     a
        rz
        mvi     a,Øffh
        ret
;
; Console #Ø Input
;  DOCUMENTED IN MP/M PROGRAMMER'S GUIDE SEC 3.2, PAGE 113
cØinpd:                                 ;PROCESS DESCRIPTOR
        dw      c1inpd  ; pl            ;ADDRESS OF NEXT PROCESS DESCRIPTOR
        db      Ø       ; status        ;PROCESS STATUS
        db      34      ; priority      ;PROCESS PRIORITY
        dw      cØinstk+18 ; stkptr     ;PROCESS LOCAL STACK
        db      'cØin    ' ; name       ;PROCESS NAME
        db      Ø       ; console       ;CONSOLE & LIST THAT HAVE PROCESS
        db      Øffh    ; memseg        ;MEMORY SEGMENT TABLE LIST
        ds      36                      ;MISCELLANEOUS

cØinstk:
        dw      Øc7c7h,Øc7c7h,Øc7c7h
        dw      Øc7c7h,Øc7c7h,Øc7c7h
        dw      Øc7c7h,Øc7c7h,Øc7c7h
        dw      cØinp   ; starting address

;  DOCUMENTED AS ABOVE, PAGE 116
cØinq:                                  ;QUEUE CONTROL BLOCK FOR ABOVE
        dw      Ø       ; ql            ;QUEUE LINK
        db      'cØinque ' ; name       ;QUEUE NAME
        dw      1       ; msglen        ;MESSAGE LENGTH
        dw      4       ; nmbmsgs       ;NUMBER OF MESSAGES
        ds      8                       ;MISCELLANEOUS
cØinmsgcnt:
        ds      2       ; msgcnt        ;NUMBER OF MSGS IN THE QUEUE
        ds      4       ; buffer        ;=MSGLEN*NMBMSGS

cØinqcb:
        dw      cØinq   ; pointer
        dw      chØin ; msgadr
chØin:
        db      Ø

cØinuqcb:
        dw      cØinq   ; pointer
        dw      charØin ; msgadr
charØin:
        db      Ø

cØinp:
        mvi     c,makeque
        lxi     d,cØinq
        call    xdos    ; make the cØinq

cØinloop:
        mvi     c,flagwait
        mvi     e,6
        call    xdos    ; wait for cØ in intr flag
        mvi     c,writeque
        lxi     d,cØinqcb
        call    xdos    ; write cØin queue
        jmp     cØinloop
```

SAGE™ TECHNICAL BRIEFING

4 It's Fast In Any Language.

When it comes to software development, the difference between a Sage IV computer and other micros is like day and night.

With the Sage Computer it'll take you fewer days and nights to finish your program.

Speed is the reason. Speed resulting from the fast MC68000 microprocessor, fast architecture, fast operating systems, and blinding transfer rates.

The fact is, even we are surprised by the amount of software that's developed on Sage Computers.

So if you're a programmer, maybe you should spend some time learning how much time you could save using a Sage IV.

The Hottest Languages And Operating Systems.

Programs in nearly all of the important languages can be written on the Sage computer.

Included in the price is the **p-SYSTEM** operating system which supports **Pascal, BASIC,** and **FORTRAN.**

Other operating systems are optional. For **UNIX** fans there's **IDRIS,** which runs up to twice as fast (even without a Sage computer). **IDRIS** conforms to /USR/Group Standards Committee standards and programs written under it are highly portable to other micros. **CP/M** advocates please note that Digital Research has developed **CP/M-68K** for Sage hardware, providing a truly complete software development environment. Versions of **Pascal, BASIC, C-BASIC, C,** and **FORTRAN 77,** as well as a very fast **APL,** may be used under this operating system.

The Sage IV is also blindingly fast when running **hyperFORTH** with its extended programmer and user interfaces.

Besides **IDRIS,** other **Multi-User** operating systems that run on the Sage Computer are **PDOS, MBOS,** and **MIRAGE.**

A log of excitement has been brewing in the Pascal World over Niklaus Wirth's new **MODULA 2**—and it's available for Sage computers. So is **ADA,** for the record.

If you found your language here, you just found your computer. The 16-bit, 2-million operation/sec Sage IV micro with up to one MByte RAM and 18-MByte internal hard disk.

For more information and the name of your nearest SAGE dealer, call us today. And be sure to ask about our new soft ware catalog describing over 200 application programs for Sage computers.

Sage Computer Corporate Office, 4905 Energy Way, Reno, Nevada 89502. Phone (702) 322-6868 TWX: 910-395-6073/SAGE RNO

Eastern United States
Sage Computer
15 New England
Executive Park
Suite 120, Burlington, MA
01803 (617) 229-6868

SAGE™ COMPUTER

The computer you don't wait for!

CIRCLE 96 ON READER SERVICE CARD

Announcing A Major New Innovation From The Publishers Of PC

Now, Get Up To 72 Programs For As Little As $2.00 Each.

Introducing PC DISK MAGAZINE. The Only Magazine On Disk That Gives You A Library Of Software Programs For Your IBM Personal Computer ... At A Fraction Of The Price!

Imagine building a library of software programs with amazing ease...adding utilities, subroutines, games and data files to your system...for as little as $2 per program! Introducing PC DISK MAGAZINE—a totally new concept for your microcomputer that gives you all this and more! You get thoroughly tested, ready-to-run programs on a floppy disk—complete with a comprehensive, illustrated User Manual—for the ultimate in ease, efficiency and economy. With your subscription to PC DISK MAGAZINE you'll receive a floppy disk containing up to 12 varied ready-to-use programs and files. Developed by experts and tested by the editors, you'll be able to expand the use of your IBM personal computer.

Business and Finance Aids: Create advanced pie, bar and line chart graphics. Compute loan payment tables. Calculate the Present Value and Internal Rate of Return for your investments. VisiCalc® templates for Real Estate, Lease/Purchase and Tax Shelter analysis.

Programming Tools: Create cross-reference listings for your BASIC programs. Convert MBASIC™ program files to CBASIC™ format or reverse. Use your text editor on your BASIC programs. Convert data between BASIC and VISICALC® or Supercalc® formats.

Home/Personal Applications: Whether you need your own income tax return helper...proven ways to maintain a mailing list ...a personal cash flow analyzer...a speed reading trainer...or SAT test preparation aids, you'll get it from PC DISK MAGAZINE.

Utilities and Diagnostics: Print graphics, screen images on your printer...backup, copy, delete, un-delete or type files with simple menu commands. Use your computer to simulate a conversational terminal with function key control of disk or printer logging ...and more.

Data Files: Tax tables, population statistics, dictionaries and economic times series.

Games: Adventures! Strategies! Test your skills, intelligence and your luck!

Everything Is Ready For You

PC DISK MAGAZINE has everything you need. The accompanying 60-page User Manual is written clearly, concisely, in easy-to-understand terms. You'll be able to try the programs immediately...there's no need to type listings into your computer. Just insert the disk and go! That simply, all the latest software developments are at your fingertips!

```
con0in:
                        ; return character in reg A
        mvi     c,readque
        lxi     d,c0inuqcb
        call    xdos            ; read from c0 in queue
        lda     char0in         ; get character
        ani     7fh             ; strip parity bit
        ret
;
        endif
;
; Console #0 Output
;
con0out:
                        ; Reg C = character to output
        in      sts0
        ani     80H
        jnz     tx0rdy
        push    b
        mvi     c,poll
        mvi     e,plco0
        call    xdos    ; poll console #0 output
        pop     b
tx0rdy:
        mov     a,c
        out     data0
        ret
;
; poll console #0 output
;
polco0:
        in      sts0
        ani     80H
        rz
        mvi     a,0ffh
        ret
;
;********************** END CONSOLE #0 ROUTINES *********************
;
;********************** CONSOLE #1 ROUTINES ************************
;
; Poll Console #1 Input
;
con1st:                 ; return 0ffh if ready,
                        ;        000h if not
        lda     clinmsgcnt
        ora     a
        rz
        mvi     a,0ffh
        ret
;
; Console #1 Input
;
clinpd:
        dw      0       ; pl
        db      0       ; status
        db      34      ; priority
        dw      clinstk+18 ; stkptr
        db      'clin    ' ; name
        db      2       ; console
        db      0ffh    ; memseg
        ds      36

clinstk:
        dw      0c7c7h,0c7c7h,0c7c7h
        dw      0c7c7h,0c7c7h,0c7c7h
        dw      0c7c7h,0c7c7h,0c7c7h
        dw      clinp   ; starting address

clinq:
        dw      0       ; ql
        db      'clinque ' ; name
        dw      1       ; msglen
        dw      4       ; nmbmsgs
        ds      8
clinmsgcnt:
        ds      2       ; msgcnt
        ds      4       ; buffer

clinqcb:
        dw      clinq   ; pointer
        dw      chlin   ; msgadr
chlin:
        db      0

clinuqcb:
        dw      clinq   ; pointer
        dw      charlin ; msgadr
charlin:
        db      0

clinp:
        mvi     c,makeque
        lxi     d,clinq
        call    xdos            ; make the clinq

clinloop:
        mvi     c,flagwait
        mvi     e,8
        call    xdos            ; wait for cl in intr flag
        mvi     c,writeque
        lxi     d,clinqcb
        call    xdos            ; write clin queue
        jmp     clinloop


conlin: mvi     c,readque       ; return character in reg A
        lxi     d,clinuqcb
        call    xdos            ; read from cl in queue
        lda     charlin         ; get character
        ani     7fh             ; strip parity bit
        ret
;
; Console #1 Output
;
con1out:
        in      sts1    ; Reg C = character to output
        ani     80H
        jnz     tx1rdy
        push    b
        mvi     c,poll
        mvi     e,plco1
        call    xdos    ; poll console #1 output
        pop     b
tx1rdy: mov     a,c
        out     data1
        ret
;
; poll console #1 output
;
polco1: in      sts1
        ani     80H
        rz
        mvi     a,0ffh
        ret
;
;********************** END CONSOLE #1 ROUTINES *********************
;
;********************** CONSOLE #2 ROUTINES ************************
;
; Poll Console #2 Input
;
polci2:
con2st: in      sts3    ; return 0ffh if ready,
        ani     2       ;        000h if not
```

GRAPHICS $975

VectorScan 512

Color and Monochrome Graphic Controller Provides High Resolution Graphics on any computer with a RS-232 serial port.

Features include:

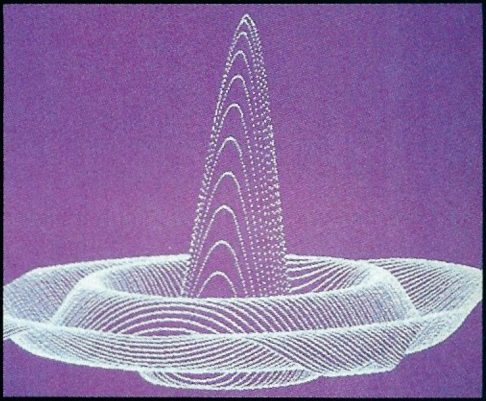
- High Resolution 512 × 480 Pixels × 16 Colors/Intensities.
- Color Lookup Table for each of the possible 16 pixel levels.
- Internal Character Generator for Horizontal and Vertical axes.
- Simple ASCII command structure allows system independence.
- Internal Line, Arc, Circle, Point, Programmable Shape and more.
- Extra RS-232 connector allows "Loopthrough" operation.
- Internal microprocessor with 8K firmware and 128K video RAM.
- Internal printer port for hardcopy on low cost matrix printers.
- Firmware expandable to 16K, Programmable Shape Table to 3K.
- RGB output is IBM-PC compatible allowing low cost monitors.
- Monochrome output is RS-170 composite video and sync.
- Includes firmware for IDS Color PRISM printers.
- Allows 128K video RAM to be used as a 128K printer spooler.

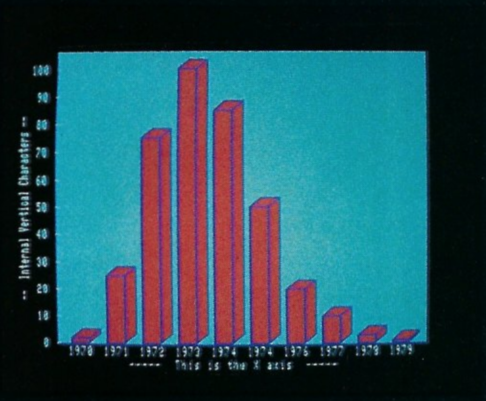
Applications:
CAD/CAM, Medical, Business, Slide Generation, Industrial Process Control, Laboratory Data Display, Mapping, Meteorological Studies, Graphic Artwork, Typesetting, CRT Strip Chart Recorders, Printed Circuit Artwork, Educational Graphics, Mathematical Modeling, Engineering, Bulletin Boards.

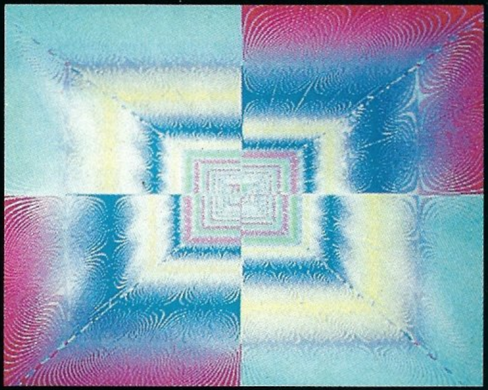
Computers:
IBM-PC, Apple, Epson, Morrow, North Star, Cromemco S-100, Radio Shack, Health, DEC, and most others.

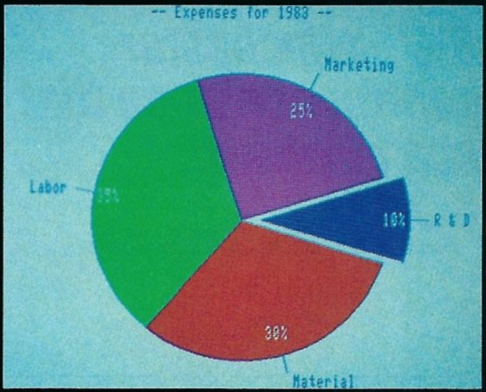
Operating Systems:
CPM-80, CPM-86, PC-DOS, MS-DOS, UNIX, OASIS, RT-11

Languages:
BASIC, PASCAL, FORTRAN, PL/1, FORTH, ASSEMBLY









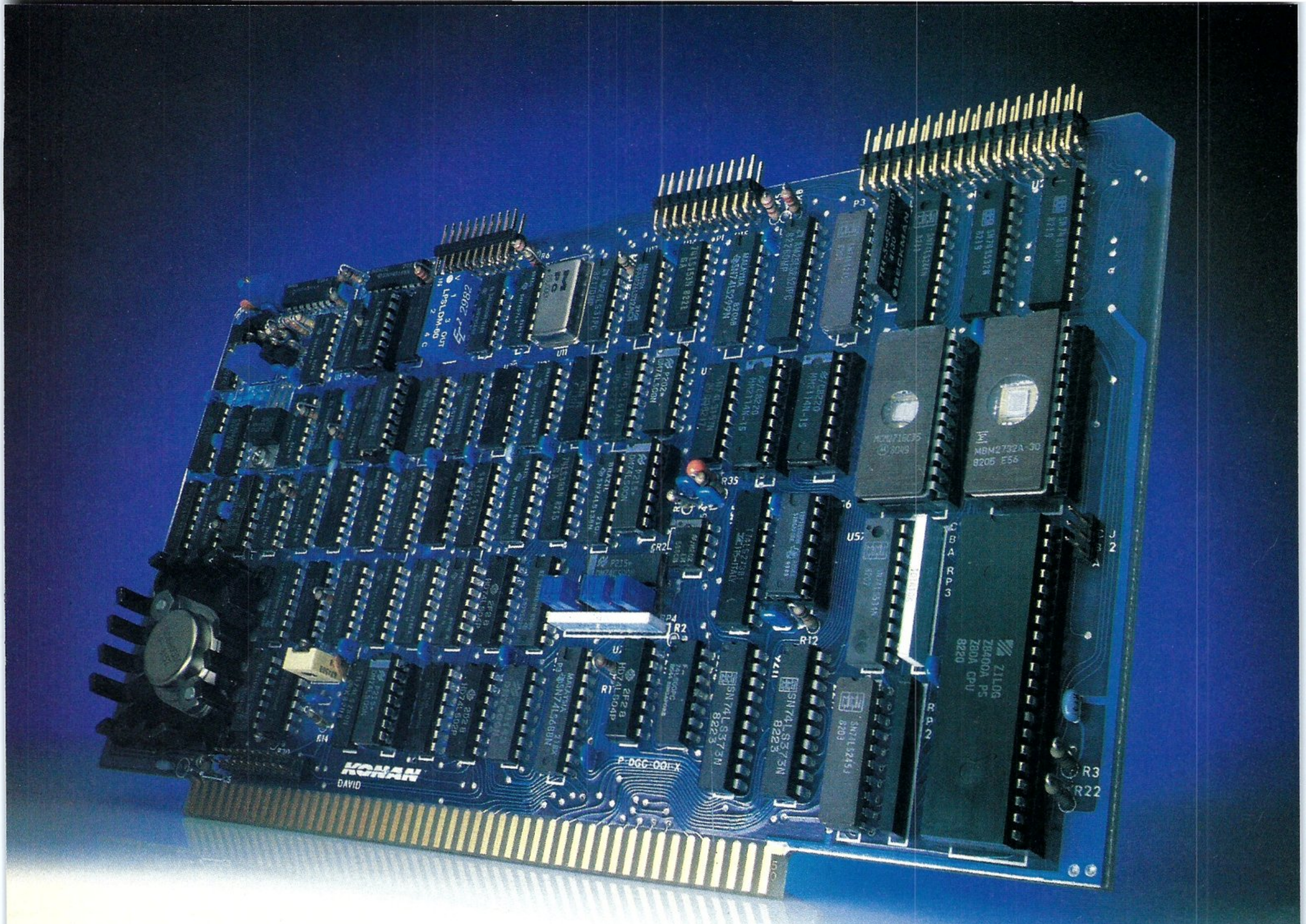
PRICES

VectorScan 512 with 128K RAM	$975.00
12 inch color monitor	CALL
Monochrome Monitor with P-39 tube	$215.00
VectorScan 512 Programmers Manual	$ 3.00

Dealer and OEM discounts available.

Applied Data Systems • 9811 Mallard Drive, Suite 213 • Laurel, MD 20708 • 301-953-9326

CIRCLE 17 ON READER SERVICE CARD



ANOTHER S-100 HARD DISK SOLUTION THAT PUTS YOU IN CONTROL.

5¼'' Winchester control at a price that can't be beat!

$259 *large OEM quantities*

From the spectrum of Konan S-100 solutions comes the DGC-100. Offering full sector buffering, error correction, automatic flaw mapping...all standard. The DGC-100 supports the known spectrum of ST-506 compatible drives. And features what is possibly the best data clock separator on the market today.

Konan's DGC-100. Reliability. High performance. On-time deliveries. And...a low, low price.

For more information or to place an order, write or call:

KONAN T.M.

in control

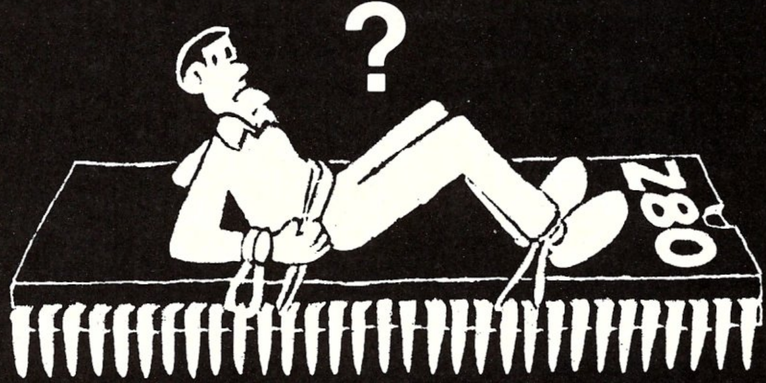
1448 North 27th Avenue, Phoenix, Arizona 85009
(602) 257-1355, TWX/TELEX 9109511552

CIRCLE 242 ON READER SERVICE CARD

```
        rz
        mvi     a,Øffh
        ret
;
; Console #2 Input
;
con2in:                 ; return character in reg A
        mvi     c,poll
        mvi     e,plci2
        call    xdos            ; poll console #Ø input
        in      data3           ; read character
        ani     7fh             ; strip parity bit
        ret
;
; Console #2 Output
;
con2out:                        ; Reg C = character to output
        in      sts3
        ani     Ø1h
        jnz     tx2rdy
        push    b
        mvi     c,poll
        mvi     e,plco2
        call    xdos            ; poll console #2 output
        pop     b
tx2rdy:
        mov     a,c
        out     data3           ; transmit character
        ret
;
; Poll Console #2 Output
;
polco2:
                        ; return Øffh if ready,
                        ;        ØØØh if not
        in      sts3
        ani     Ø1h
        rz
        mvi     a,Øffh
        ret
;
;********************** END CONSOLE #2 ROUTINES ********************
;
;
;********************** LIST DEVICE ROUTINES ***********************
; Line Printer Driver:
;
list:                   ; List Output
lstlout:
        LDA     ORIGIN+3F9H     ;memory-mapped list device
        CMA
        ANI     Ø8H
        JNZ     TXLRDY
        push    b
        mvi     c,poll
        mvi     e,pllpt
        call    xdos            ; poll printer output
        pop     b
txlrdy:
        MOV     A,C
        CMA
        STA     ORIGIN+3F8H
        RET
;
; Poll Printer Output
;
pollpt:                 ; return Øffh if ready,
                        ;        ØØØh if not
        LDA     ORIGIN+3F9H
        CMA
        ANI     Ø8H
        rz
        mvi     a,Øffh
        ret
;********************** END LIST DEVICE ROUTINES *******************
;
;
;   MP/M II V2.Ø    Xios
;
;
polldevice:
                        ; Reg C = device # to be polled
                        ; return Øffh if ready,
                        ;        ØØØh if not
        mov     a,c
        cpi     nmbdev
        jc      devok
        mvi     a,nmbdev; if dev # >= nmbdev,
                        ; set to nmbdev
devok:
        call    tbljmp  ; jump to dev poll code

devtbl:
        dw      pollpt  ; poll printer output
        dw      polcoØ  ; poll console #Ø output
        dw      polco1  ; poll console #1 output
        dw      polco2  ; poll console #2 output
        dw      polci2  ; poll console #2 input
        if      debug
        dw      polciØ  ; poll console #Ø input
        endif
nmbdev  equ     ($-devtbl)/2    ; number of devices to poll
        dw      rtnempty; bad device handler
;

; Select / Protect Memory
;
selmemory:
                        ; Reg BC = adr of mem descriptor
                        ; BC ->  base   1 byte,
                        ;        size   1 byte,
                        ;        attrib 1 byte,
                        ;        bank   1 byte.
; this hardware does not have memory protection or
;  bank switching
        ret
;
; Start Clock
;
startclock:
                        ; will cause flag #1 to be set
                        ;  at each system time unit tick
        mvi     a,Øffh
        sta     tickn
        ret
;
; Stop Clock
;
stopclock:
                        ; will stop flag #1 setting at
                        ;  system time unit tick
        xra     a
        sta     tickn
        ret
;
; Exit Region
;
exitregion:
                        ; EI if not preempted or in dispatcher
        lda     preemp
        ora     a
        rnz
        ei
        ret
;
; Maximum Console Number
```

```
;
maxconsole:
        mvi     a,nmbcns
        ret
;
; System Initialization
;
systeminit:
;
;  This is the place tc insert code to initialize
;  the time of day clock, if it is desired on each
;  booting of the system.
;
;       IF      NOT DEBUG       ;Then set up interrupt jump vector
        mvi     a,Øc3h          ;Initialize the jump at ØØ38H
        sta     ØØ38h           ;  If DEBUG mode, then we can't use this
;       ENDIF                   ;  since we have to use DCON or DDT RST 7
        lxi     h,inthnd

;       IF      NOT DEBUG       ;Then set up rest of interrupt jump vector
        shld    ØØ39h           ;If DEBUG, then don't mess with it
;       ENDIF

        mvi     c,create        ;Create a queue for console #1 if DEBUG mode
        if      debug           ; else create a queue for console #Ø
        lxi     d,clinpd
        else
        lxi     d,cØinpd
        endif
        call    xdos

        call    home            ;Home the booting drive
        ei                      ;Enable interrupts

        RET

;
;
;  MP/M II V2.Ø   Interrupt Handlers
;

flagwait equ    132
flagset equ     133
dsptch  equu    142

inthnd:
                        ; Interrupt handler entry point
                        ;  All interrupts gen a RST 7
                        ;  Location ØØ38H contains a jmp
                        ;  to INTHND.
        shld    svdhl   ;Save all registers
        pop     h
        shld    svdret  ;Get return address, too
        push    psw
        lxi     h,Ø
        dad     sp
        shld    svdsp           ; save users stk ptr
        lxi     sp,lstintstk    ; set up local stack for interrupt handler
        push    d
        push    b

        mvi     a,Øffh          ;set preempted flag so we can make flag set
        sta     preemp          ; calls without enabling interrupts
                                ; (see MP/M II System Guide sec 3.2 p41)
;
; Since this system uses only a simple 6ØHz interrupt (RST7) generator,
; we pass through this routine once every 6Øth of a second (more or less).
; This method is not incredibly accurate here, because interrupts must be
; disabled while doing disk I/O.

        lda     tickn           ;See if any delayed processes
        ora     a               ; test tickn: non-zero indicates
                                ;  delayed process(es)
        jz      notickn         ;If not, then don't flag
        mvi     c,flagset
        mvi     e,1
        call    xdos            ; else set flag #1 each tick
notickn:
        lxi     h,cnt6Ø         ;Now see if time to set 1 second flag
        dcr     m               ; decrement 6Ø tick counter
        jnz     notlsec         ;(best to read a real clock here)
        mvi     m,6Ø
        mvi     c,flagset
        mvi     e,2
        call    xdos            ; set flag #2 every 1 sec
notlsec:                ;Now that timekeeping is done, see if any I/O
                        ; is waiting
        if      not debug       ;then do this, else it's done in conØst
                                ; routine, since no interrupts in DEBUG mode
        in      stsØ            ;See if input waiting from console #Ø
        ani     4Øh
        jnz     cnsØin
        endif
;
        in      stsl            ;See if input is waiting from console #1
        ani     4Øh
        jnz     cnslin
;
        jmp     intdone         ;End of interrupt Handler poll routine
;
;
;  Individual interrupt service routines
;
        if      not debug       ;get the character from console #Ø
cnsØin:
        in      dataØ
        sta     chØin
        mvi     e,6
        jmp     concmn          ; set flag #6
        endif

cnslin:                         ;get the character from console #1
        in      datal
        sta     chlin
        mvi     e,8
;       jmp     concmn          ; set flag #8

concmn:                         ;set the flag contained in register C
        mvi     c,flagset       ;  See Programmer's guide Appendix A
        call    xdos            ;  for flag assignments
        jmp     intdone         ;MP/M 2.1 reserves the following flags:
                                ;Ø=reserved; 1=tick; 2=clock; 3=minute
;
;
intdone:                ; Routine to exit from interrupt handling routines
        xra     a
        sta     preemp  ; clear preempted flag
        pop     b
        pop     d
        lhld    svdsp
        sphl                    ; restore stk ptr
        pop     psw
        lhld    svdret
        push    h
        lhld    svdhl
; The following dispatch call will force round robin
;  scheduling of processes executing at the same priority
;  each 1/6Øth of a second.
; Note: Interrupts are not enabled until the dispatcher
;  resumes the next process.  This prevents interrupt
;  over-run of the stacks when stuck or high frequency
;  interrupts are encountered.
        jmp     pdisp           ; MP/M dispatch
;
```

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 either 128k or 256k RAM.

MSDOS comes with CO-POWER-88, along with your choice of CP/M-86 or MSBASIC. 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:

128k CO-POWER-88 w/MSDOS
and CP/M-86 or MSBASIC.... $550.00
256k CO-POWER-88 w/MSDOS
and CP/M-86 or MSBASIC.... $699.95

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

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.

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

CIRCLE 75 ON READER SERVICE CARD

Listing 2

```
;
cnt60:  db      60      ; 60 tick cntr = 1 sec
intstk:                 ; local stack used by interrupt handler
        dw      0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h
        dw      0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h
        dw      0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h
        dw      0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h
lstintstk:
svdhl:  dw      0       ; saved registers HL during interrupt handling
svdsp:  dw      0       ; saved SP during interrupt handling
svdret: dw      0       ; saved return during interrupt handling
tickn:  db      0       ; ticking flag, TRUE = delayed process(es)
preemp: db      0       ; preempted flag (means don't enable interrupts)
;
```

Listing 3

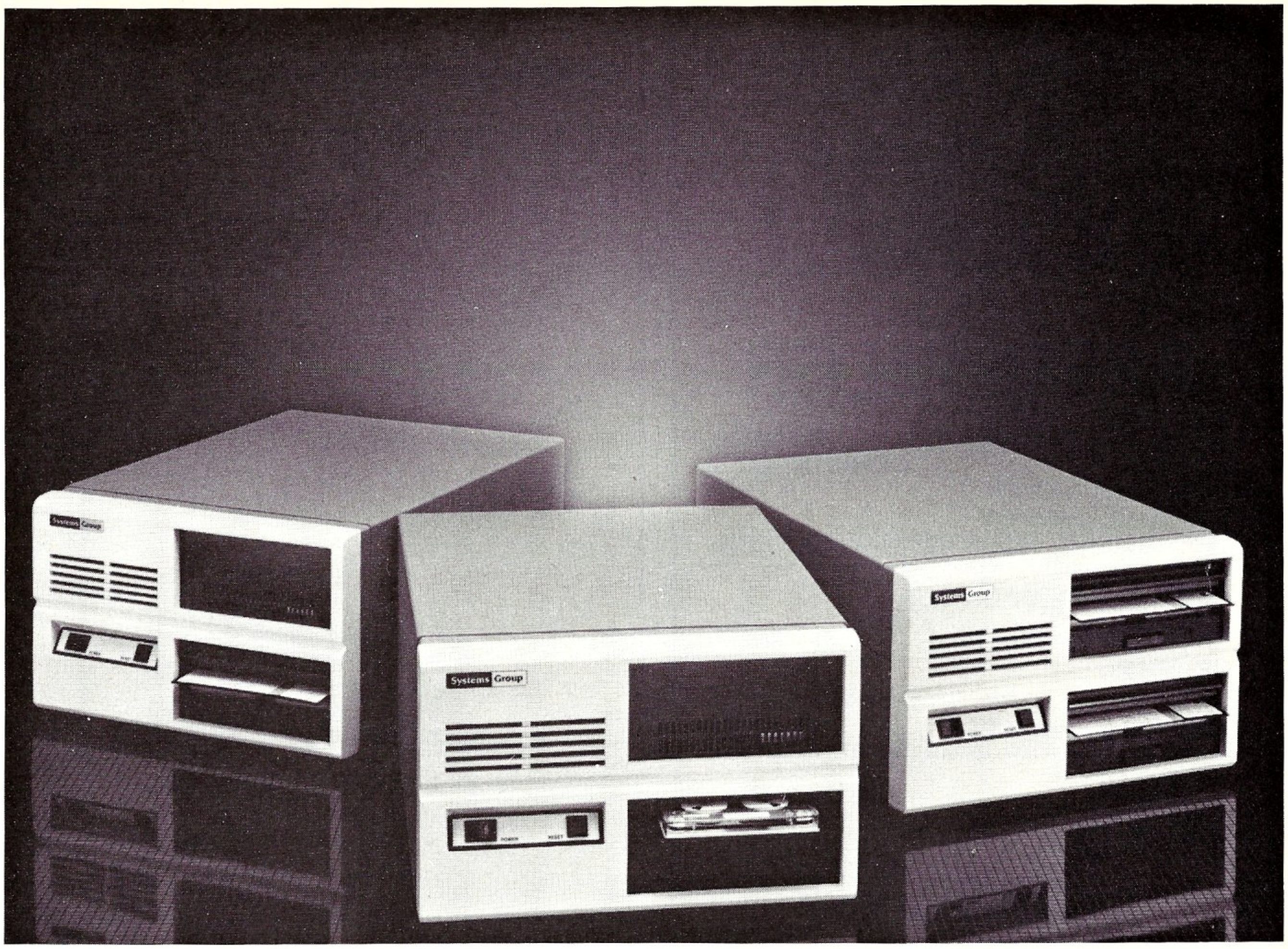
```
        org     1700H           ;LOADER BIOS starting address

        jmp     cboot           ;null entry point
        jmp     wboot           ;null entry point
        jmp     const           ;Console status routine
        jmp     conin           ;Console input
cout    jmp     conout          ;Console output
        jmp     list            ;null device
        jmp     punch           ;null device
        jmp     reader          ;null device
        jmp     home            ;Home drive
        jmp     setdrv          ;Select disk
        jmp     settrk          ;Set track
        jmp     setsec          ;Set sector
        jmp     setdma          ;Set DMA address
        jmp     read            ;Read the disk
        jmp     write           ;null function
        jmp     listst          ;null device
        jmp     sectran         ;Sector translation

cboot:                          ;These are all null devices, i.e.
wboot:                          ;  these routines are not needed by the
list:                           ;  Loader BIOS
punch:
reader:
write:
listst:
rtnempty:                       ;null device return
        RET
```

Listing 4

```
A>MAC LDRBIOS $PZ -S            <---- Assemble your LOADER BIOS


CP/M MACRO ASSEM 2.0

1FB1

007H USE FACTOR

END OF ASSEMBLY

A>DDT MPMLDR.COM                <---- Run debugger and load MPMLDR

DDT VERS 2.0

NEXT  PC

1780 0100

-ILDRBIOS.HEX                   <---- Overlay your LOADER BIOS

-R

NEXT  PC

1AA6 0100

-^C

A>SAVE 26 XMPMLDR.COM           <---- Save your customized MPMLDR
```

Listing 5

```
A>GENSYS


MP/M II V2.1 System Generation

Copyright (C) 1981, Digital Research

Default entries are shown in (parens).

Default base is Hex, precede entry with # for decimal

Top page of operating system (FF) ? F7                    <---- Note

Number of TMPs (system consoles) (#4) ? 2                 <---- Note

Number of Printers (#1) ?

Breakpoint RST (06) ?

Enable Compatibility Attributes (N) ? Y                   <---- Note

Add system call user stacks (Y) ?

Z80 CPU (Y) ?

Number of ticks/second (#60) ?

System Drive (A:) ?

Temporary file drive (A:) ?

Maximum locked records/process (#16) ?

Total locked records/system (#32) ?

Maximum open files/process (#16) ?

Total open files/system (#32) ?

Bank switched memory (Y) ? N                              <---- Note

Number of user memory segments (#3) ? 1                   <---- Note

Dayfile logging at console (Y) ? N                        <---- Note

 RESERVED      F800H  0800H

 SYSTEM   DAT  F700H  0100H

 TMPD     DAT  F600H  0100H

 USERSYS  STK  F500H  0100H

 XIOSJMP  TBL  F400H  0100H

Accept new system data page entries (Y) ?

 RESBDOS  SPR  E800H  0C00H

 XDOS     SPR  C600H  2200H
```



THE MULTI-USER MICRO THAT REPLACES A MINI

Why buy a mini when you can have the big system features of the System 2900 microcomputer series? This versatile, multi-user IEEE 696/S-100 system can support up to eight users with existing popular operating systems like "Super" CP/M*, MP/M* and "Fast" OASIS**. And as your requirements grow, so can the System 2900. When you consider the bottom line—COST PER USER—the system 2900 gives you mini performance at micro prices.

EXPANDABILITY

The System 2900 has built-in expandability. Its modular design allows configuration with combinations of floppy disks, hard disks and tape drives. For applications with big storage requirements, you can use multiple 12MB, 19MB or 40MB Winchester hard disks. You can upgrade the system easily by adding memory modules, I/O interfaces and mass storage subsystems. And with the System 2900, you have industry-standard peripheral and software compatibility.

VERSATILITY

The System 2900 gives you big-system choices in application software.

- Use System Group's high-performance versions of CP/M, MP/M and OASIS to develop your own application software
- Take advantage of the variety of application packages for these operating systems
- Try the BUSINESS EXPRESS™ application development system, and use or customize its word processing and business application packages

Add to that a full one-year warranty on parts and labor for the entire system, **including drives,** and you have the Systems 2900 package. Designed to give you big performance at a micro price.

Join the Group. Systems Group.

* CP/M and MP/M are registered trademarks of Digital Research.
** OASIS is a trademark of Phase One Systems, Inc.
TM Business Express is a trademark of Measurement Systems & Controls, Inc.

Systems Group
A Division of MEASUREMENT systems & controls incorporated

800-448-4888

1601 W. Orangewood Ave., Orange, CA 92668
Telephone (714) 633-4460
TWX/TLX 910-593-1350 SYSTEMGRP ORGE

CIRCLE 105 ON READER SERVICE CARD

Getting Started with MP/M II continued...

```
Select Resident and Banked System Processes:
ABORT    RSP (N) ?
MPMSTAT  RSP (N) ?
SCHED    RSP (N) ?
SPOOL    RSP (N) ?
RESXIOS  SPR  BE00H  0800H
BNKBDOS  SPR  9B00H  2300H
BNKXDOS  SPR  9900H  0200H
TMP      SPR  9400H  0500H
LCKLSTS  DAT  9100H  0300H
CONSOLE  DAT  8F00H  0200H
Base,size,attrib (8F,71,80) ?
Base,size,attrib (00,C0,00) ? 0,8F,0    <---- Note
MP/M II Sys  8F00H  7100H
Memseg  Usr  0000H  8F00H
Accept new memory segment table entries (Y) ?
** GENSYS DONE **
A>
```

MICROSYSTEMS REVIEWS

The MP/M 8-16 from Gifford Computer Systems

Getting more power from MP/M

by Bruce H. Hunter

Most of the software available for micros today is for 8-bit CP/M machines. This software is time-tried, having had ample time to work out the bugs, and as a consequence most of it runs very smoothly and is miserly in the use of memory. But in this fast-paced industry what was once innovative quickly becomes yesterday's news, and today 16-bit micros are where most of the computer industry is concentrating its efforts. 16-bit machines have the advantages of much larger available memory, so they are unfettered by the constraints of 64K addressing. There have been many technological advances for the 16-bus in the last few years, including the Intel 8087 chip, which gives the micro an 80-bit math word. The tremendous impact of the IBM PC on the micro market virtually guarantees an ever-expanding 16-bit software base. 16-bit operating systems are becoming more sophisticated as well, with less of an emphasis on single-user, single-tasking operations and more on concurrent single-user and multiuser, multitasking operations.

The industry shift from 8-bit single-user micros to the more sophisticated 16-bit machines leaves a lot of people who already own perfectly good 8-bit computers with some fast decisions to make. If they jump right into buying a new 16-bit machine, what are they going to do with all that 8-bit software they've been buying for the last few years? 16-bit machines have so much to offer, but 8-bit users are understandably reluctant to abandon their existing software base. The ideal would be to find a transition machine and operating system that works in 8- and 16-bit.

In 1980 Bill Godbout of CompuPro was farsighted enough to see the problem and seek an answer to guide software users through the transitional period. His 8085/8088 CPU board fit the bill very nicely, but like any hardware it needed software, namely, an operating system. CP/M and MP/M were offered in 8- or 16-bit, but not both. What was truly needed was a CP/M compatible operating system to run 8- and 16-bit simultaneously. Concurrent CP/M begins to address this need for single users, but Gifford Computer Systems, a firm that works very closely with CompuPro, created CP/M 8-16 (an enhanced version of Concurrent CP/M) and MP/M 8-16 (an enhanced version of MP/M-86) to take full advantage of the 8085/8088 capabilities.

In a review of MP/M software, it is necessary to consider precisely what is involved in an MP/M environment. Having purchased a multiuser system six months ago, our system requirements and experiences represent one "real-world" example. We needed a minimum of a three-user system, expandable to five or six, enough disk storage to house the text of two books (in editing) simultaneously, a minimum of six programming language versions, a truckload of tools and utilities such as text processors, program development tools, and so forth. In addition to storage for all this, we still needed to have room to thrash. The system also needed the ability to do rapid program development work in 8- and 16-bit, in the neighborhood of 25 programs a month. At least two printers and a sophisticated modem were a necessity.

Having heard of the merits of Gifford Computer System's MP/M 8-16, I consulted with them on the purchase of a computer to handle our needs. They recommended a System 421, a 384K system with 21 MB of formatted hard disk storage. They customized this system with an additional half meg of memory dedicated to being a disk emulator, M/Drive-H, which substantially speeds up not only compiling and linking but other disk-intensive tasks such as dictionary look-ups (The Word) and editing (WordStar and Vedit).

With these hardware specifications in mind, you can appreciate how sophisticated the operating system must be to handle these requirements. Considering the needs of a multiuser environment alone, unenhanced MP/M performs well, and for those coming from a CP/M environment, the transition to MP/M is an easy one. MP/M is a natural extension of CP/M. Working in MP/M is easy for anyone who is already familiar with CP/M. Most of the familiar CP/M commands are still present and work identically, with some extensions. Additional commands have been added as well. ERA is still there to erase files but ERAQ (erase with query) does wildcard expansions with the user option of erasing or bypassing files as they are presented. PIP, the workhorse of file transfer, is present in MP/M, but enhanced to move files across user area boundaries. REN to rename is present and unchanged. DIR is still there, but SDIR has been added to give alphabetically sorted directories with a full display of the directory status, number of bytes, and the number of records. It also shows disk space used and disk space remaining, not unlike a utility called D that has been finding its way into everyone's computer lately. (D was written by Rick Rump, writer of the MicroShell, a UNIX emulator.) STAT is still with us, but SET and SHOW do STAT's tasks better. SET sets disk names, passwords, write protection, time and date stamping of files, and protection levels, and it turns on and off the SYS or DIR attributes, thereby making a file visible or invisible to the DIR command. SUBMIT is basically unchanged, except that it is far more reliable, and TYPE to display ASCII files is unchanged.

There are other additions to MP/M, not present in CP/M. ABORT kills a process in use. Control C will no longer warm boot because it would interrupt all user processes as well. ATTACH restarts a process that has been deliberately detached. CONSOLE tells the user his console number, and DSKRESET, never needed in CP/M,

Bruce H. Hunter, 1020 S. Jenifer, Glendora, CA 91740

The world's largest selection of cross assemblers!

All 2500AD assemblers and cross assemblers support these features:

Relocatable Code—the packages include a versatile Linker that will link up to 400 files together, or just be used for external reference resolution. The Linker allows Submit Mode or Command invocation.
Large File Handling Capacity—the Assembler will process files as large as the disk storage device. All buffers including the symbol table buffer overflow to disk.
Powerful Macro Section—handles string comparisons during parameter substitutions. Recursion and nesting limited only by the amount of disk storage available.
Conditional Assembly—allows up to 248 levels of nesting.
Assembly Time Calculator—will perform calculations with up to 16 pending operands, using 16 or 32 Bit arithmetic (32 Bit only for 16 Bit products). The algebraic hierarchy may be changed through the use of parenthesis.
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—

8086 and Z-8000 XASM includes Source Code Translators

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

Name ______________________
Company ______________________
Address ______________________
City ________ State ________ Zip ________
Phone ______________ Ext. ________
Make and model of computer system ______________________

☐ C.O.D. *(2500AD pays C.O.D. charges)*
☐ VISA or MasterCard #, Exp. Date (mo./yr.)
____

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

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

Total $ ________
shipping/handling ($6.50 per unit) $ ________
Total Order $ ________

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

Signature ______________________

2500AD SOFTWARE INC.

P.O. Box 441410, Aurora, CO 80014, 303-752-4382 tells the system that disks have been changed (a necessity because there is no warm boot). MPMSTAT tells the user the status of MP/M—meaning who is using what process and what processes are standing in line (queuing), waiting for a chance to run. When it is necessary to abort a process, the user does not always know which one to abort. MPMSTAT will tell him. For example, if DRC were being used to compile a C program, it would have been invoked as **drc,** but the process in use will be **r,** the program loader. PRINTER lets the user know which printer is presently attached to his terminal, and is also used to assign the printer to his terminal. SPOOL is a welcome addition; it allows a print file to be attached to the spooler process and then sent off to the printer as a background task. SPOOL detaches itself from the terminal, thus allowing concurrency. TOD gives the time and date. USER sets the user area. In CP/M, user areas are little more than a curiosity; in MP/M they are a necessity, partitioning each disk into 16 distinct areas. Our system has five physical drives which are divided into 7 logical drives (the hard disk comprises drive A-C). There are 16 user areas per disk, giving 112 distinct user areas. Trying to keep track of all the available files and processes in the system without user areas would create pandemonium. A UNIX-like hierarchical file structure would even make it easier, and it certainly would be a welcome addition to MP/M.

Gifford's MP/M 8-16 is an enhancement of MP/M-86, and it more than meets the challenge of Godbout's 8085/8088 coprocessor. An operating system is ported by the OEM (original equipment manufacturer) to the hardware. The input/output system is not furnished by Digital Research, so the OEM makes the system operate with his hardware. While writing the I/O system, the OEM has the opportunity to enhance the basic operating system, and he is also at liberty to add additional processes, such as disk-to-disk file copy utilities, for example. This is where a version of an operating system can be greatly improved, and Gifford's MP/M 8-16 contains substantial enhancements.

Of the many improvements Gifford has made to MP/M, the major difference in MP/M 8-16 is what its name implies, the ability to run 8- and 16-bit processes. It not only runs 8- and 16-bit software, it automatically recognizes which is which (the user doesn't even have to know whether he's running on 8- or 16-bit process). In addition, both 8-bit and 16-bit processes can be mixed or matched within the system and run by a common SUBMIT file.

Gifford has incorporated many UNIX-like features into MP/M 8-16 as well. When a process is invoked and read by the user's shell (similar to the CCP in CP/M), the shell directs a search path through the default user area, finishing up in A0. (A0 is the default area for all processes that are public to the entire system.) It will be looking for a 16-bit process, one with a .CMD extension. If it fails to find the process, it calls SW (the process that runs 8-bit software), which duplicates the original search path, only this time looking for an 8-bit process—one with a .COM extension. When a file with a .COM extension is found, MP/M 8-16 has it read by SW, which causes the process to be read and executed by the 8085 processor. The 8088 still runs the system as a whole, but the 8085 handles the 8-bit processes.

Other UNIX-like enhancements to MP/M 8-16 include MAIL, which allows "mail" or messages to be sent to selected users. Mail can be created, stored, read, passed over, or passed on. As each user logs in, if there is any mail in that user area's "mail box," a message will appear on the screen. Another similar UNIX-like feature is the message of the day. It is present at each terminal on login and is set by the system "supervisor." We find it a handy substitute for a "things to do" pad. Both work well and are bug free. HELP brings up general and specific help menus and almost eliminates the necessity of using the manual once the system is initially understood. There is a process called SCHED that will invoke processes at predetermined times, thereby allowing disk- and/or CPU-intensive processes to be scheduled for off hours. For example, a payroll could be run at 2:00 a.m., long after everyone has gone home. While it is not advisable to schedule unattended runs of processes that involve the printer, because of the possibility of a jam and subsequent file hazard, there is nothing to stop these processes from outputting a PRN file to be SPOOLed when personnel are in attendance. Another process called TIME times all processes in use at any time. If users or departments or clients are to be billed on a time-sharing basis, this utility is invaluable. A utility called SWAP rotates processes to RAM-based disks for faster execution. Warp Drive is an automatic part of the operating system (as long as you have sufficient memory to support it). Another UNIX-like feature is WHO, a process to tell who is currently logged in.

I have never warmed up to assembly language programming, which explains why I have not mentioned DDT86 and ASM86, also furnished with MP/M 8-16. Having to change a BIOS or XIOS in MP/M is a chore I would dread. However, I love to program in C, and MP/M 8-16 is written in C. Device assignments are done by way of simple ASCII files. Passwords, printer assignments, and terminal assignments as well as other devices such as modems are easily added to, taken away from or modified by the TTYS, PASSWD, and LPRS files. The files allow baud rates, the number of stop bits, handshaking, protocols, and teletype names to be set. The PASSWD file allows the user and his password (if any) to be set, his terminal and printer to be selected, and the default programs that may automatically be brought up. For example, you can limit a specific user to word processing only by having WordStar come up on login; you also arrange that when the word processing clerk attempts to exit from WordStar, he automatically exits from the system also. On the other hand, other users (such as the "system manager") can log on and have access to the entire system. Once I got past the initial familiarization with the system, I found I could reconfigure, add or

MP/M 8/16 is a pleasure to use. Not just for what it does, but for what it doesn't do. It doesn't keep us waiting, and it doesn't lose data formerly lost by passing files from one CP/M system to another.

delete devices and users in a minute or two. Try to modify a BIOS and assemble it that rapidly. This MP/M 8-16 feature alone rates seven stars!

Except for observing user areas and avoiding warm boots, working in MP/M 8-16 is not very different from working in CP/M, especially Concurrent CP/M. However, bringing up MP/M 8-16 (starting up the system) is very different. Hard disk or not, the system comes up on a floppy disk (which is not too surprising—even mainframes do so). If any automatic processes are to be run, MPMINIT runs them before allowing any hands-on operations. Next, the system automatically enters the multiuser mode if a file called "AUTOST" is found. If not, the system waits on the main console. I avoid AUTOST so that I can do any system housekeeping before turning the system over to the other users. It is a good time to kill backup files and roam around everyone else's areas, looking for unbacked-up files and files to be deleted from the system—there is an archive bit that can (and should) be set in the directory to indicate whether a file has been backed up or not. Once these housekeeping chores are done, the system terminal is logged out and all terminals in the system display the banner message of the day and the logon prompt. As each user logs on, he will get notice of mail, if any exists, and then it is business as usual.

There are other differences in MP/M 8-16 relating to operating philosophy. In typical MP/M systems the available user memory partition is substantially reduced by the RSPs (resident system processes). 8-bit memory partitions are 64K, but are usually reduced to 48K or so by heavy overhead. MP/M 8-16 has an entirely different approach. RSPs are stored in a high-speed memory area available to all users. The individual user's memory partition is penalized by only 1K or so for jump points to the systems process area. It costs the system a full 92K to do this, but not at the expense of individual users. Most of the system processes are in high-speed memory and do not have to be brought up from disk. This saves substantial time. Another time-saver is the use of a cache memory buffer for the hard disk. Rather than wait for the hard disk to be accessed physically, disk output goes to the cache buffer and then is read to disk every 30 seconds by an interrupt.

MP/M 8-16 is a pleasure to use. Not just for what it does, but also for what it doesn't do. It doesn't keep us waiting. It doesn't give any problems moving across drives and user areas, and it doesn't lose data formerly lost by passing files from one CP/M system to another. I haven't had a major loss of a file since the system arrived. On our single-user systems I had lost entire disks form "disk full" errors, one of many risks associated with floppy disk CP/M systems. Not part of MP/M 8-16, but available with it, is Modem 8-16. This is a very versatile modem program that actually operates under MP/M, provided that memory is available, without disrupting the system. It has the ability to use files for automatic operation.

Documentation for 8/16 includes the usual three-inch

TURN-KEY, TURBO-DOS BUSINESS SYSTEM

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

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

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

SAVE OVER $3000.00

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

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

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

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

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

STD HEIGHT 5.25" FLOPPY DISK DRIVES

CCD08207-02 MPI B51 40 trk, 1 side $199.95
CCD08201-02 MPI B52 40 trk, 2 side $264.95
CCD08206-02 MPI B91 80 trk, 1 side $269.95
CCD08202-02 MPI B92 80 trk, 2 side $369.95

HALF HEIGHT 5.25" FLOPPY DISK DRIVES

CCD08204-05 SA-465, 80 trk, 2 side $379.95
CCD08203-02 MPI 902, 80 trk, 2 side $369.95

STD HEIGHT 8" FLOPPY DISK DRIVES

CCD08205-02 Qume 842, 77 trk, 2 side $579.95

HARD DISK DRIVES, 5.25"

CCD08501-02 ST-419, 19 meg $999.95

HARD DISK SYBSYSTEMS, 5.25"

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

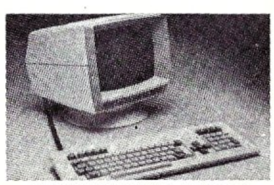
PRINTERS, RS-232 SERIAL

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

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

CCD08204-05 SA-465

HARD DISK SUBSYSTEMS



CCD08601-02 LSI, ADM-11

CRT TERMINALS, RS-232

CCD08601-02 LSI, ADM-11 .... $559.95
CCD08602-09 QUME, QVT 102 .... $559.95

TELETEK IEEE-696 S-100 CARDS

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

NOVATION MODEMS

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

8" CP/M SOFTWARE

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

(713) 488-9754

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

COMPUTER COMPONENT DISTRIBUTION

2600 Skywalker, #2101
Houston, Texas 77058

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

stack of Digital Research documentation plus Gifford's manual. The Gifford manual is reasonably well written. It is divided into a Reference Manual and a User's Guide. The Reference Manual is well indexed, so look-up time is reduced to a minimum. The User's Guide is a bit sparse but it covers the ground well, and a thorough reading should put a new user on his feet in a minimum of time. It even contains helpful hints on installing WordStar. Regrettably the documentation for Modem 8-16 is not well written, but I have been told that it is going to be redone.

Error handling is superb. No more "BDOS error on A:" or equally nebulous CP/M error messages. MP/M is tenacious. It doesn't quit and "go away" when a process fails to work. The system will output an appropriate error message and then go back about its business. It has only "panic stopped" once or twice in six months of our 16-hour-a-day, 7-days-a-week operation, and then only with good reason. The only thing that I have found in the way of software that will bring MP/M 8-16 down is new programs that we are developing in pointer languages like C. We developed a habit of screaming "Archive!" before trying a new program! New compilers (i.e., version 1s) also bring MP/M 8-16 down. Other than that, it is as stable as a rock.

Existing software interacts well with MP/M 8-16. The only programs that failed to make the journey from my old 8-bit library were the Z80-only packages like UNICA and a Z80 version of Vedit. Most new software that I acquire now is 16-bit. Even MicroShell runs well, although I was at first worried about its reactions to extended addressing. WordStar gave us a problem; once it had accessed the printer, it would not release the printer to the other users until the user exited from WordStar. Fortunately there is a patch for that. I did acquire a new MP/M version of Vedit which, like all software, is better in the newest version. So far, the only compiler that takes full advantage of the entire available memory area is Digital Research C, which has the ability to specify memory models. Lowell Wolf of DRI has told me that the next release of PL/I will also have the ability to specify memory models.

The only problem I have found with the system as a whole is its need for large amounts of memory. Our system has 384K, which seems like all the memory in the world; in practice it allows the use of only three 64K partitions and a little room to spare. When we have two users in 8-bit processes and I start doing some heavy 16-bit compilation, we inevitaly get an "out of memory" error. Another 64K would alleviate the problem, but 128K would be better. Also, in order to use Modem 8-16 effectively, another 128K is just about a necessity. However, adding more memory is no problem for MP/M 8-16. Simply insert one more board in the card cage, and the system will automatically become aware of the difference. Besides knowing its memory limit at all times, MP/M 8-16 takes a running memory check constantly, testing from bottom to top and starting all over again whenever the CPU is not

MP/M 8-16 continued. . .

scheduled to do anything else. MP/M 8-16 is never idle.

Service and support from Gifford Computer Systems is very good, perhaps the best in the industry. The hotline is always attended, usually by Pat Miller. Gifford has maintained a policy of never allowing a customer to be down for more than a full business day. There has been a minor deterioration of service lately, primarily due to Gifford's rapid growth, but you will have to look long and hard to find better customer support.

To the best of my knowledge, MP/M 8-16 is available only on CompuPro systems, either from CompuPro distributors or Gifford Computer systems. It is available in hard disk and floppy versions, but a floppy-only MP/M system is almost an impossible dream because of the limited storage and the slow disk access with multiple users. Gifford systems shipped with MP/M 8-16 are shipped with CP/M-80 and CP/M-86 as well.

For more information, contact:

CompuPro
3506 Breakwater Court
Hayward, CA 94545
(415) 786-0909
CIRCLE 304 ON READER SERVICE CARD

Gifford Computer Systems
2446 Verna Court
San Leandro, CA 94577
(415) 895-0798
CIRCLE 303 ON READER SERVICE CARD

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.

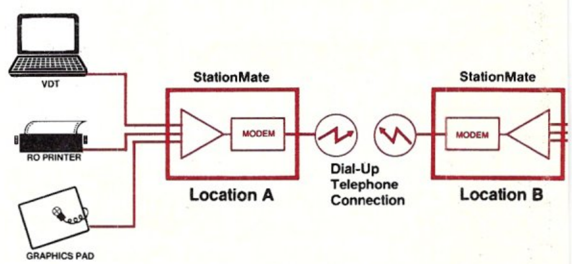
Besides local area networking, StationMate permits access to remote workstations via an internal dial-up modem. And it serves as a gateway for teleprocessing access to all LAN resources. You couldn't get all of this in one unit. Until now. And the suggested retail price is only $1450.00. StationMate makes communications simple. *Incredibly.*

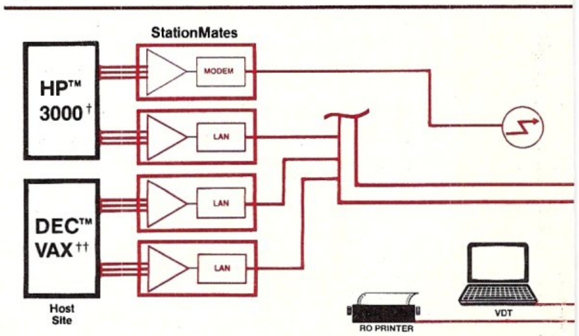
StationMate. The link you've been missing.

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™

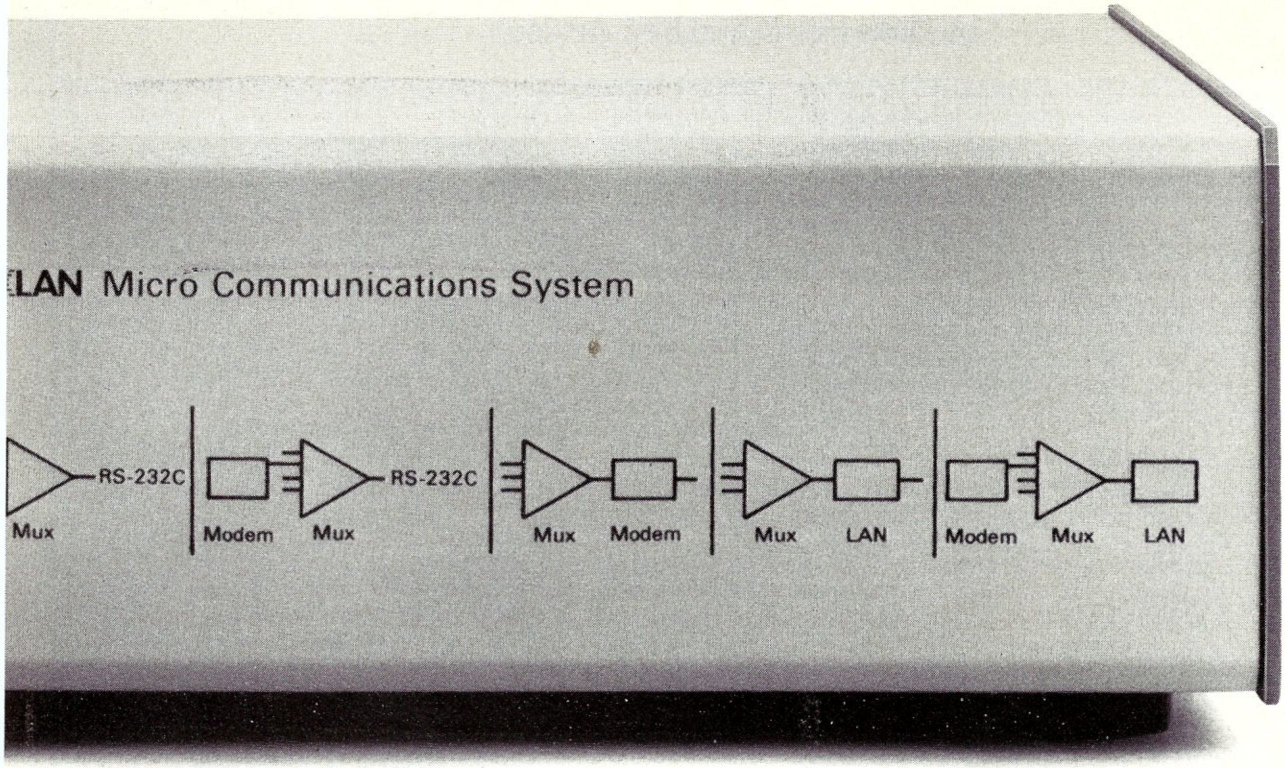
Teleprocessing



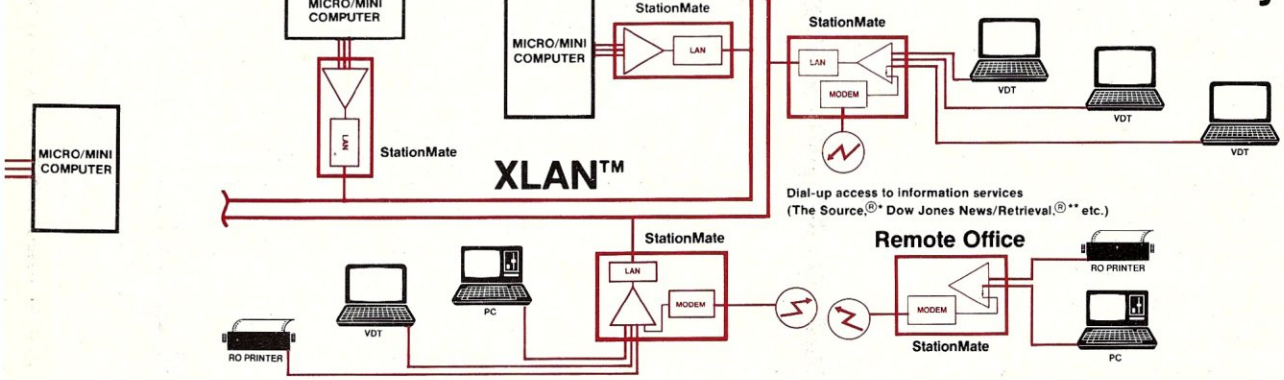


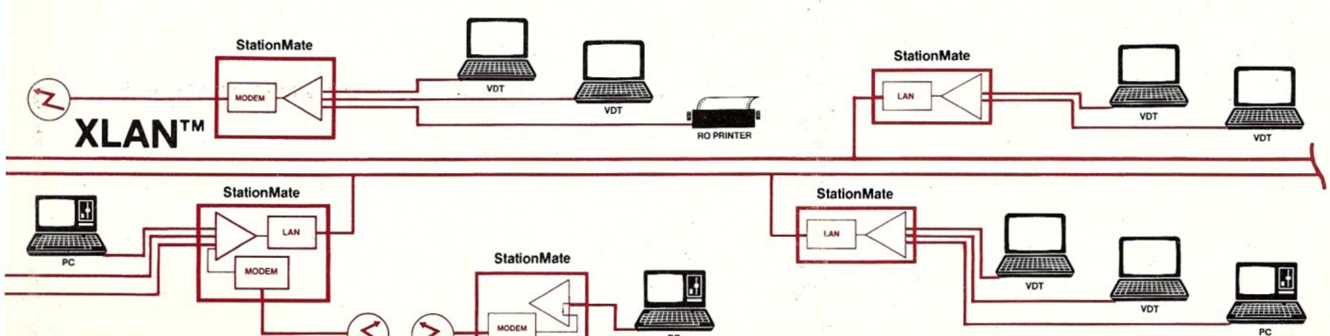
Local Networking

*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.



LAN/TP Gateway





CIRCLE 51 ON READER SERVICE CARD

The Pro-Comp 8 Computer

by Dave Hardy and Ken Jackson

The Pro-Comp 8 computer system is actually a Teletek SysteMaster board with a custom EPROM, and a very customized operating system. Although the board has been reviewed here previously, the software provided with the Pro-Comp 8 makes it worth a second look. Operating systems available for the Pro-Comp 8 include CP/M, MP/M, CP/NOS, and TurboDOS. The system reviewed here uses the CP/M 2.2 operating system. In addition to the improved software, Pro-Comp provides its own hardware manual for the SysteMaster, along with monitor source and several utilities.

Hardware features

As mentioned previously, the Pro-Comp 8 uses the Teletek SysteMaster single-board computer. Briefly, the SysteMaster offers the following features:

- Z80A @ 4 MHz
- IEEE-696 compatible
- 2 serial ports (Z80A SIO)
- 2 parallel ports (Z80A PIO)
- Counter/timer (Z80A CTC)
- Z80 mode 2 interrupt driven, also supports S-100 vectored interrupts
- 765 floppy disk controller (up to four drives)
- 64K dynamic RAM (4564)
- Bank-switching supported via I/O ports
- Z80 DMA controller for floppy I/O; also user-programmable
- Type-ahead console buffer

Pro-Comp 8 configurations are available with 5¼″ or 8″ floppy disk drives, and various Winchester configurations. High-speed tape backup is also available.

Software features

The software is what sets the Pro-Comp 8 computer apart from most other systems. More than a dozen utilities are provided to make configuration and use of the system a great deal easier and faster than normal. Some of the utility programs provided are shown in Table 1.

The following is a specific discussion of some of the more interesting utilities mentioned above.

FKSET

FKSET is supplied in three forms: machine language (.COM), source, and as a Basic program.

The machine language version can display the function key settings, or set them individually. Up to 16 function keys may be defined, and they can each contain an entire string, not just a single character. When the program is exited, it automatically writes the new function key definitions onto the system tracks of the A: drive, so that they will be permanently available at each cold boot.

The Basic program is similar to the machine language program, except that it only defines the keys temporarily, so that new key definitions can be temporarily assigned while running a Basic program.

Dave Hardy, 736 Notre Dame, Grosse Pointe, Michigan 48230

Table 1. Pro-Comp 8 software utility programs

COMM	A basic communications terminal program.
COPY	A fast disk copying program that will clone an entire disk in about two minutes. Also allows copied data to be verified, and can be used to check disks to see if they are readable.
FKSET	A function key definition utility that allows individual (unprogrammable) keys on a terminal to be used as multiple-keystroke function keys.
FORMAT	A disk format utility that can format 5¼″ and 8″ disks. 8″ disks can be formatted in single, double, and "Extended" densities.
MENU	A "menu" program that can set the system up to list .COM files in a menu, to simplify system operation.
SETFMT	A utility to set the system up for single or double-sided 5″ drives.
SETTOD	A utility used to set the system time from an optional battery backup clock.
SETUP	A sophisticated program that allows the user to define several default system parameters, including auto-command selections, directory size, drive search path, device assignments, and much more.
SINGLEA	A handy program that temporarily stops the system from reading the A: system tracks during warm boots. This allows unSYSGENed and single-density disks to be used in the A: drive.
SPOOL	A printer despooling program that allows wildcard file commands, and several options.
STAT	A modified version of the standard STAT program that uses less confusing device names, such as SPA: for serial port A, PAR: for the parallel port, etc.
SYSGEN	A modified version of the standard SYSGEN program that automatically cold boots if used to put a system on the A: drive, to prevent a possible crash.
TOD	A utility that can set and read the time and date.
USRINT	A do-nothing example program supplied to demonstrate how to include user-defined interrupt handlers in the CP/M sytem. Supplied in SOURCE form.

FORMAT

This program is used to format 5¼″ and 8″ disks. Its three 8″ formats include the standard single-density IBM format, the Teletek double-density format, and Pro-Comp's "extended" density format, which is the same as the Morrow Designs and Godbout double-density formats. These formats were chosen to maintain compatibility with other 8″ disk systems (single-density format), to maintain compatibility with Teletek SysteMaster software (512 byte-sectors), and to provide the maximum possible storage on an 8″ floppy disk (1024-byte sectors in the "Extended" format).

In addition, when formatting in "Extended" density, the user is given the option of formatting the whole disk or just the system tracks. The manual states that this feature may be used to convert disks with a single-density track zero (such as the TRS-80 Model II) to double-density track zero so that they can be SYSGENed and used as a Pro-Comp system (A:) disks.

Double-sided disks are automatically recognized by FORMAT when using the double and "extended" density options. However, only single-sided disks may be formatted in single-density mode.

The 5¼″ format chosen is compatible with the Eagle series of computers. The system must be told via the SETUP or SETFMT programs whether the 5¼″ disk being formatted is to be single- or double-sided.

SETUP

The SETUP program is used to set up virtually all of the user-definable features of the Pro-Comp system. SETUP modifies a section of the BIOS called the "mode area" that contains all of the "environment-dependent" options, as mentioned previously. The program allows the user to specify more than 30 options. The questions asked by SETUP are:

- Autoload command on warm boot, cold boot, both or none?
- 256 or 128 files on "extended" density disks?
- Automatically search drive A: for .COM files?
- Display user number as part of system prompt? (e.g. 0A:)?
- Make user 0 a public file area for all other user numbers?
- Console (CON:) on serial port A or B?
- Make LST: serial A, serial B, serial C, or parallel?
- Make RDR: serial A, serial B, serial C, or serial D?
- Make PUN: serial A, serial B, serial C, or serial D?
- Make serial A printer protocol ETX/ACK, XON/XOFF, or none?
- Want hardware handshaking on serial A port?
- Serial A baud rate 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, or 19200?
- Serial B baud rate (same as above)?
- Serial C baud rate?
- Serial D baud rate?
- Type of serial expansion board?
- Disable BREAK key option?
- Enter type-ahead buffer flush character:
- Have battery backup clock option?
- Enter parallel printer interface type:
- Floppy disk step rate (1-15 ms):
- Number of floppy drives?
- Set extended terminal parameters? (next four questions)
- Function key delay constant?
- Function key lead-in character?
- Use 2 or 3 character function codes?
- Enter 16 character string of function designator codes
- Save changes, re-enter parameters, or quit?
- Incorporate these changes into the existing system?

This is not a complete list of all of the questions asked by SETUP, but by now you probably get the idea. SETUP is so complete that it virtually eliminates the need to generate a new system the "old way," (by reassembling the BIOS, overlaying it onto a memory image of CP/M, and then SYSGENing it onto a disk).

As shown above, SETUP will automatically put the new system onto the A: disk if desired, or it can create a file of the system image for later use. In addition, all changes can be made just temporarily, that is, until the next cold boot. Although we were provided with complete BIOS source for our tests, because of SETUP, we never had to use it to generate any other systems.

SPOOL

Externally, SPOOL is very much like most other printer despooling programs. Basically, it works the same way, sending characters to the printer only while the system is waiting for console input. Unlike most of the others, however, SPOOL will accept wildcard filenames to print out. SPOOL also has several options that make it even more useful:

/B:n	Use n 256 byte pages of memory as file buffers
/C:n	Print n copies of the file(s) specified
/D	Delete the file(s) after printing
/F	Issue a formfeed character after each copy or file
/U:n	Look in user n for the files to print
/*	Do not check printer status before printing

SPOOL also accepts three keywords while executing:

!PAUSE	to suspend printer output until !RESUME is issued
!RESUME	to resume a print job after being suspended
!STOP	to immediately cancel any printing and terminate SPOOL

After reading the previous descriptions, it should be obvious that the quality (and quantity) of the software provided is much higher than is usually found in this type of computer system. We found the software to be generally bug-free and easy to use (if a bit overwhelming at times). The simplified nature of the set-up programs makes customizing this system easy enough for a beginner, while still

Knowing that most real computer users don't read the manual until later, the folks at Pro-Comp included a simple procedure that can be followed like a road map, right to the system prompt.

allowing advanced system features.

Source code for the BIOS, MENU and SPOOL programs is available from Pro-Comp at additional cost. The source code they provided was well commented and easy to understand.

Documentation

The documentation provided consists of several files included on the system disks, and a three-part ring binder consisting of a software user's manual, a hardware user's manual, and a ROM monitor user's manual. The software and hardware manuals are indexed and well written. The ROM monitor manual is only three pages long, so an index for it is not really needed.

The largest manual (44 pages) is the software manual, which covers in depth all of the software provided with the system, and is broken into three parts: CP/M User's Guide, CP/M Implementation Notes, and Appendices.

The User's Guide contains start-up instructions, several system notes, and information about the operation of all of the utility programs.

The CP/M Implementation Notes contain information about the operation of the system's interrupts, and various special programs like SPOOL and SETUP. They also provide guidelines for user-written interrupt handlers.

The Appendices contain various tables and lists, including a page zero map, an error message summary, a listing of the mode area, some useful addresses inside the BIOS, and default interrupt table values. Appendix C is a detailed explanation of the disk layout, including the DPB's and skew tables.

The hardware manual (14 pages) contains a concise board overview, board configuration information, drive setup information, peripheral interfacing information, and port maps for on-board and external ports used by the board.

The board overview is extremely brief and includes no theory of operation, diagrams, or schematics. If you want a theory of operation, etc., you can always get it from Teletek.

Board configuration involves setting several optional jumpers, most of which are already set at the factory. Information is included only for setting the disk write precompensation values to match various brands and sizes of floppy disk drives. Basically, precomp values of 0, 125 ns, 250 ns, and 500 ns are available, and separate values can be set for 5¼″ drives and 8″ drives that exist on the same system. Common memory size selection is also provided with jumper options (4K, 8K, 16K, or 32K); this permits off-board memory to be employed as multiple bank systems.

Drive strapping information is provided for all Shugart 5¼″ and 8″ drives, and also for the Qume Datatrak 8 drive. To use 5¼″ drives, a minidisk adaptor board is required.

Peripheral interfacing information includes serial port layouts and pin definitions, baud rate selection values for the on-board CTC, parallel port layouts and pin definitions, and information on fabricating a Centronics-compatible cable and a Data Products compatible parallel cable.

Bringing it up

Bringing up the Pro-Comp 8 system took about five minutes. Knowing that most *real* computer users don't read the manual until later, the folks at Pro-Comp cleverly included a simple procedure (called "How to get the system running without reading the manual") that can be followed like a roadmap, right to the system prompt. The hardest part is setting up the terminal and drives, which usually requires finding your configuration manuals and setting a few switches and jumpers. The "How to" procedure even tells you what to check if it doesn't work.

Conclusion

Our benchmarks indicate that, although processing speed remains the same, the improved disk I/O makes the Pro-Comp 8 considerably faster overall than the stock Teletek system that we tested several months ago.

The biggest advantage of the Pro-Comp 8 is its software. It is much better than the software provided by the board's manufacturer, and is well worth the extra cost. The extra 400H bytes used by the Pro-Comp system (compared to the Teletek CP/M) are a small price to pay for the added speed and versatility. The "extended" format increases disk I/O speed considerably, while simultaneously increasing the disk capacity.

All of our tests were performed in a standard IEEE-696 compatible frame (ParaDynamics PRONTO) using Shugart 860 8″ thinline floppy disk drives. An adaptor board was not provided, so no tests were performed with 5¼″ drives.

If you already have a SysteMaster, you can upgrade to the Pro-Comp version of CP/M for $185, which includes CP/M and a modification kit for your board. If you send Pro-Comp your original Teletek CP/M disk (as proof of ownership of CP/M), the price is only $95. For an additional $50, Pro-Comp will do the modifications for you.

If you don't have a SysteMaster, you can buy it from Pro-Comp, along with the Pro-Comp software (but without CP/M) for $895.

BIOS & Utility source are available for $500, as are SPOOL source, and MENU source.

The Pro-Comp 8 is also available as a system with two double-sided 5¼″ floppy drives for $3,000, with single-sided 8″ drives for $3,500, or with double-sided 8″ drives for $3,750. Several other options are available, and Pro-Comp software will soon be available for other boards and processors. For more information, contact:

Pro-Comp Systems, Inc.
333 West 52nd St.
New York, NY 10019
(212) 246-0074

CIRCLE 302 ON READER SERVICE CARD

The software is what sets the Pro-Comp 8 computer apart from most other systems. More than a dozen utilities are provided to make configuration and use easier and faster than normal.

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.

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

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

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

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

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

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

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

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

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

HARDWARE

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

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

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

APPLE Z80 DEVELOPER
Includes: 6502X ($150), MACRO II ($100), MACRO III ($150), QSAL ($200), QED ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), ZDDT ($40) and BUSINESS BASIC ($200)
VALUE: $1250 **NOW $325**
$175 when purchased with AMT "Chairman" Board

DISK FORMATS

When ordering software specify which disk format you would like.

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

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

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

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

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

MICROSYSTEMS REVIEWS

The paraGraphics "Game Board"

An economical solution to high quality S-100 graphics

by Eric L. Beser

The "Game Board" from paraGraphics is more than a S-100 video graphics interface. It's an I/O-mapped terminal emulating the Zenith H19 in cursor commands and graphics characters, and it's capable of emulating the DEC VT-52. The baud rate is slow, relatively speaking, with character speed about 9600 baud. WordStar, dBASE II, and other screen-oriented programs work very well using the "Game Board" as a terminal. However, terminal emulation, as I will point out, is only a small part of this product. Actually, this product can be so much more than simply a game board (with or without the quotation marks, which the company uses) that the name seems unnecessarily limiting.

Product description

The "Game Board" is well constructed and well laid out. Since the board is I/O mapped, it doesn't take up memory space in the system. The board may be addressed at any lower 8-bit boundary and is selectable by a DIPswitch. It has a 68B09E CPU and uses the 6845 CRT controller chip. There is 64K of dynamic memory used onboard, as well as a 2764 EPROM, which contains the firmware used to drive the board. The CPU was chosen because of its ability to operate synchronously with the 6845 CRT controller. Both are driven with different phases of the same clock, which means that the CPU has access to memory when the CRT controller doesn't, allowing the CRT controller and CPU to run at full speed without contention. This technique eliminates hash and flicker and other contention problems. As a result, the display is smooth and clean.

The display board has a low-resolution graphics controller of 512 by 286 pixels and a high-resolution capability of 512 by 575 pixels. Also, you can set the horizontal frequency pulse width to permit use with different monitors. Either a positive or negative pulse synchronization is selectable via a DIPswitch. There's a composite video interface as well.

To provide for different monitors and to have maximum resolution, the horizontal rate is driven at 19,040 Hz and the display is interlaced. In the interlaced mode, frame time is divided into even and odd alternating fields, resulting in a displacement of scan lines. As a result, the character density is doubled. On a cheap ($99 variety) monitor, this interlace will show up as a flickering display. A slow phosphor monitor, on the other hand, won't exhibit this effect. On my TECO monitor, which cost $89 and has a slow phosphor CRT, this flicker shows up in high-resolution mode; however the board works very well. I've tried this board also with the Zenith monitor and found that the horizontal scan rate has to be adjusted.

Eric L. Beser, 2509 North Calvert Street, Baltimore, MD 21218

Firmware

The support firmware on the board is just short of excellent. There are some limitations, some shortcomings, but the versatility of the firmware makes up for them.

Upon power-up, the board is initialized as a 24-line H19 terminal with a subset of H19 cursor commands. Resolution is set to maximum (512 × 576), which will cause problems for some monitors. This shortcoming is compensated by a Set Resolution command that allows for a global change of resolution; also a Set Register command that allows for dynamic setting of the CRT controller registers. You may find values that are optimal for the monitor in use, and may write an initialization routine that feeds these values to the board on power-up.

Terminal emulation

Operating the "Game Board" as a terminal, you may run H19-compatible software, as the full graphics set is also implemented. You can set the terminal to the ANSI mode and use DEC VT-52 cursor codes. The terminal commands implemented are XY Cursor Positioning, Cursor Movement (up, down, right, and left), Inverted Video, Line Insert and Delete, Erase to End of Line, Erase to End of Screen, Clear Screen, and Cursor Home. Additionally, you have a choice between 24 lines and 40 lines, but no 25th line (status line) is implemented. The characters are readable in the 24-line mode but somewhat crowded in the 40-line mode. The cursor may be removed and the scrolling disabled, a useful function when using low-resolution or high-resolution graphics.

Graphics

As a graphics terminal, the "Game Board" is quite powerful, in that you can manipulate each pixel on either a 512 × 288 screen (with two planes) or a 512 × 576 screen (with only one plane). The high-level graphics commands allow you to plot points, draw lines, and manipulate sprites. (Unfortunately, it is a monochromatic display, with no color and no gray levels.) With this resolution and the firmware capabilities, you may emulate many other systems that use HPLOT, HLINE, VLINE, PLOT, SET, RESET, etc. Run this board with an S-100 development system, and you can plan and write graphic software that will run on other computers (adding the color, of course). Other graphic commands include Draw Lines, Plot Points, Draw Relative (meaning from where you left off the last time) Lines and Points, Set, Reset, or Compliment Lines and Points. You can display the ability to alternate planes, as well as to define sprites and models and move them around the screen.

One of the more interesting features of the "Game Board" is the ability to manipulate sprites. It allows you to write arcade games, use large cursor displays with mouse applications, do menu processing, define large-screen characters, and provide animated displays for user-friendly programs.

Sprites are defined by first defining a model of the image you want to display. This model defines a rectangular part of the screen. Sprites are represented as a dynamic construction of the model, and as such can be moved around the screen. Figure 1 illustrates a model for a friendly little yellow guy we all know well, and Figure 2 is the sprite defined by Figure 1, which can be moved around the screen. My first listing, written in C, shows how this little guy may be generated and moved with ease. Notice that the model can be rotated and another sprite can be generated that can be moved in the opposite direction. The function ROT(SPRITE) handles this rotation by 0Ring the bits in the model. The interface functions to the graphics software

	BYTE 1	BYTE 2	BYTE 3
0	0 0 0 0 0 0 0 1	1 1 1 1 1 1 1 1	1 1 1 1 0 0 0 0
1	0 0 0 0 0 0 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 0 0
2	0 0 0 0 0 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
3	0 0 0 0 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 0 0
4	0 0 0 0 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 0 0 0
5	0 0 0 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 0 0 0 0
6	0 0 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 0 0 0 0 0 0
7	0 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0
8	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 0 0 0 0 0 0
9	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 0 0 0 0
10	0 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 0 0 0
11	0 0 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 0 0
12	0 0 0 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
13	0 0 0 0 0 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 0 0
14	0 0 0 0 0 0 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 0 0 0
15	0 0 0 0 0 0 0 1	1 1 1 1 1 1 1 1	1 1 1 1 0 0 0 0
	1 MSB	1 LSB	7 6 5 4 3 2 1 0

Figure 1. Data model bit map.

are commented. This listing also demonstrates the ease with which programs may interface to the firmware. I've provided a second listing written in MBasic that defines a model of a little guy and then moves this guy back and forth on the screen. From this listing, you can see how easy it is to manipulate screen images. Also shown is a rudimentary listing of graphics subroutines within the Basic program, also showing how easy it is to use the firmware with other languages.

Software

paraGraphics provides an 8″ SSSD disk containing text programs; a complete interface package written in assembly language that may be used with Basic, Fortran, and other Microsoft-compatible languages; a graphics interface in C with a linkable library of functions; and a series of Basic, C and assembler test programs that demonstrate clearly how the board is interfaced. The instructions invite you to make as many modifications as you desire. The package is well documented and provides a means of testing all functions of the board once installed.

Documentation

In many graphics displays, documentation is the weak part of the product. Many displays come with about 10 or 20 pages of photocopied information that assume extensive knowledge on your part. It is sad that many otherwise fine displays are lacking in this area. Fortunately, paraGraphics spent a great deal of time planning its documentation, as it did with its interface firmware. My board came with a 1″ loose-leaf notebook that presented assembly information in a clear and concise manner, with proper warnings about eye protection and acid solder, plus installation instructions and one page on each graphics command. The documentation, like the board itself, is a class act.

Accessibility and assembly

My "Game Board" came as a kit. Other choices are to purchase an assembled model or to purchase a bare board with key parts (CPU, CRT Controller, PAL ROM, and firmware ROM). The hours for contacting the paraGraphics technician is noted in the manual. The engineer who is available to help you with interface problems,

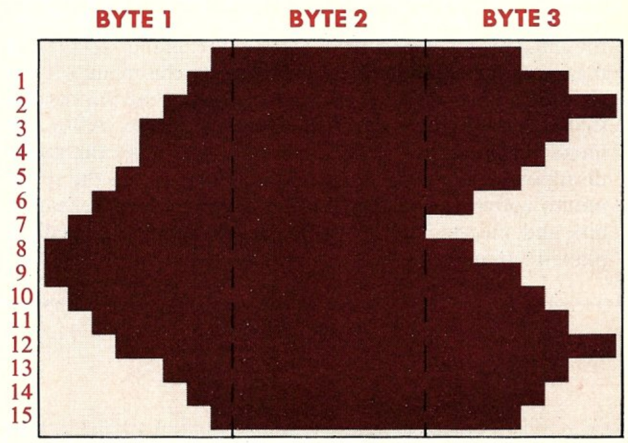


Figure 2. Sprite display on screen—overlay bit map.

assembly problems, and troubleshooting is the same one who designed the board. paraGraphics obviously supports its product.

Assembly of my kit took several hours and was straightforward. Sockets for ICs were included, although I was short several resistors and capacitors. The missing capacitors were common bypass types, which I had on hand, so there was no delay in construction time.

My first try at bringing the board up in my system was a disappointment. What was supposed to be a cursor on a blank screen was simply a blank screen. With schematic in hand and a CRT Controller Handbook to help me out, I started to troubleshoot the board. Several phone calls to paraGraphics later, I localized the problem. A bad 75451 dual AND driver was the culprit. The technician was very helpful and, since he was the guy who designed the board, he helped me localize the problem. But telephone conversations being as they are, and my background in digital electronics being limited, I never did pinpoint the problem. I sent the board back to paraGraphics, and within one day the board was fixed and returned. There was no charge. My impression is that paraGraphics backs its product 100 percent.

If I had to pick a second time, I would buy the "Game Board" again; paraGraphics is young company with a mature product.

Board operation

The board works very well in my system as both a terminal and as a graphics device. There are very few quirks that have to be flexed out, and there are no severe flaws in the firmware. Since purchasing the board, I have received three updates in the firmware ROM. I was impressed enough by this board to want to write about it.

Missing features

Now for my "wish list." The line drawing algorithms are nice, but how about circle plots, shading, gray scales, and user-defined character fonts? Since the character fonts are defined in memory (there is no character ROM) the latter request may be one of the easier ones to implement. Since I have hopes for using this board with a mouse to do window processing, the ability to move data from one plane to another in the form of screen updates would be very useful. While using sprite mode, the display plane is shadowed, meaning that as the sprite moves over the displayed image, the image is recovered from the shadow plane. To place a menu on the screen requires displaying the menu on the shadow plane, switching the planes, giving the impression that the menu overlays the displayed image. After the menu selection, switching planes should make the menu disappear, leaving the image intact. However, as the sprite moves across the image, the menu begins to reappear in bits and pieces. This problem can be eliminated with a plane-to-plane transfer command.

Conclusions

If I had to pick over again, I would buy the same product. It's that simple. paraGraphics is a young company, but it has a very mature product. For anyone who wants to write video games, learn about interactive graphics, emulate software written for other graphics computers, or write very friendly user interfaces, this board is ideal. As responsive as paraGraphics was to my needs, I would even recommend you buy the kit and assemble it yourself.

Specifications

paraGraphics, *58 Needham St., Norfolk, MA 02056; (617) 620-4513*

CIRCLE 305 ON READER SERVICE CARD

CPU: 68B09E

CRT Controller: 6845

Memory: 4164-150 (150 ns access time)

Resolution	Horiz Rate	Max Lines of Text	Max Graphics
MODE 0	19,040 Hz	24 or 40	512×576 or 512×28
MODE 1	17,547 Hz	22 or 40	512×528 or 512×26
MODE 2	15,980 Hz	20 or 40	512×480 or 512×24

Price: $295 for bare board, documentation, CPU, CRT controller, PAL firmware ROM, and disk with interface software. $525 for full kit.

```
LISTING 1: SPRITE MANIPULATION PROGRAM WRITTEN IN C

/*
        This is an example of sprite control for
        use with "The Game Board" by paraGraphics.
*/

#include "bdscio.h"
#include "graphic.h"

/*
        Rot will simply rotate an already define pacman.
        this is just here to make building other pacmen easier.
*/

rot(sprite)
int sprite;
{
        int a;

        a = 0;

        if ((sprite & 1) > 0)
          a = a | 0x80;
        if ((sprite & 2) > 0)
          a = a | 0x40;
        if ((sprite & 4) > 0)
          a = a | 0x20;
        if ((sprite & 8) > 0)
          a = a | 0x10;
        if ((sprite & 0x10) > 0)
          a = a | 8;
        if ((sprite & 0x20) > 0)
          a = a | 4;
        if ((sprite & 0x40) > 0)
          a = a | 2;
        if ((sprite & 0x80) > 0)
          a = a | 1;

        return(a);
}
/* end of rot */

main()
{
        char pac0[64];              /* PACMAN BUFFER */
        char pac1[64];
        char pac2[64];
        char pac3[64];
        char pac4[64];
        char pac5[64];

        int l, i, n, q, m, addr;

        pac0[0] = 0x0;
        pac0[1] = 0xf;
        pac0[2] = 0xf0;
        pac0[3] = 0x0;

        pac0[4] = 0x0;
        pac0[5] = 0xff;
        pac0[6] = 0xff;
        pac0[7] = 0x0;

        pac0[8] = 0x3;
        pac0[9] = 0xff;
        pac0[10] = 0xff;
        pac0[11] = 0xc0;

        pac0[12] = 0xf;
        pac0[13] = 0xff;
        pac0[14] = 0xff;
        pac0[15] = 0x0;

        pac0[16] = 0x3f;
        pac0[17] = 0xff;
        pac0[18] = 0xfc;
        pac0[19] = 0x0;

        pac0[20] = 0x7f;
        pac0[21] = 0xff;
        pac0[22] = 0xf0;
        pac0[23] = 0x0;

        pac0[24] = 0xff;
        pac0[25] = 0xff;
        pac0[26] = 0xc0;
        pac0[27] = 0x0;

        pac0[28] = 0xff;
        pac0[29] = 0xff;
        pac0[30] = 0x0;
        pac0[31] = 0x0;

        pac0[32] = 0xff;
        pac0[33] = 0xff;
        pac0[34] = 0x0;
        pac0[35] = 0x0;

        pac0[36] = 0xff;
        pac0[37] = 0xff;
        pac0[38] = 0xc0;
        pac0[39] = 0x0;

        pac0[40] = 0x7f;
        pac0[41] = 0xff;
        pac0[42] = 0xf0;
        pac0[43] = 0x0;

        pac0[44] = 0x3f;
        pac0[45] = 0xff;
        pac0[46] = 0xfc;
        pac0[47] = 0x0;

        pac0[48] = 0xf;
        pac0[49] = 0xff;
        pac0[50] = 0xff;
        pac0[51] = 0x0;

        pac0[52] = 0x3;
        pac0[53] = 0xff;
        pac0[54] = 0xff;
        pac0[55] = 0xc0;

        pac0[56] = 0x0;
        pac0[57] = 0xff;
        pac0[58] = 0xff;
        pac0[59] = 0x0;

        pac0[60] = 0x0;
        pac0[61] = 0xf;
        pac0[62] = 0xf0;
        pac0[63] = 0x0;

        for (i = 0; i < 64; i++)
          {
          pac1[i] = pac0[i];
          pac2[i] = pac0[i];
          }

        pac1[15] = 0x0f0;
        pac1[18] = 0x0ff;
        pac1[19] = 0x0fc;
        pac1[22] = 0x0ff;
        pac1[23] = 0x0c0;
        pac1[26] = 0x0f0;

        pac1[38] = 0x0fc;
        pac1[43] = 0x0c0;
        pac1[42] = 0x0ff;
        pac1[47] = 0x0fc;
        pac1[46] = 0x0ff;
        pac1[51] = 0x0f0;

        pac2[15] = 0x0f0;
        pac2[18] = 0x0ff;
        pac2[19] = 0x0fc;
        pac2[22] = 0x0ff;
        pac2[23] = 0x0fe;
        pac2[26] = 0x0ff;
```

```
pac2[27] = 0x0ff;
pac2[30] = 0x0ff;
pac2[31] = 0x0ff;
pac2[34] = 0x0ff;
pac2[35] = 0x0ff;
pac2[38] = 0x0ff;
pac2[39] = 0x0ff;
pac2[43] = 0x0fe;
pac2[42] = 0x0ff;
pac2[47] = 0x0fc;
pac2[46] = 0x0ff;
pac2[51] = 0x0f0;

for (n = 0; n < 16; n++)
  for (i = 0; i < 4; i++)
    {
    pac3[((n*4)+(3-i))] = rot(pac0[((n*4)+i)]);
    pac4[((n*4)+(3-i))] = rot(pac1[((n*4)+i)]);
    pac5[((n*4)+(3-i))] = rot(pac2[((n*4)+i)]);
    }

printf("What's your game board base address in hex ? ");
scanf("%x", &addr);
ingbio(addr);

set_mode(SPRITE);               /* enter sprite mode */
in_model();                     /* initialise models */
in_sprite();                    /* initialise sprites */
set_mode(CURSOR);               /* stop cursor display */
t_e_comm(CLEAR);                /* clear screen */

model(0,4,16,pac0);             /* define models */
model(1,4,16,pac1);
model(2,4,16,pac2);
model(3,4,16,pac3);
model(4,4,16,pac4);
model(5,4,16,pac5);

a_pl(0,0,511,287);      /* plot line for backgraound */

c_sprite(0,0,4,16,0,100);       /* create sprite */

q = 0;
while (TRUE)
  {
  l = TRUE;
  while (l && (q < 400))
    {
    for (n = 0; n < 3; n++)
      {
      a_smove (0,n,q,100);
      q++;
      for (i = 0; i < 10; i++)
        {
        a_smove(0,0x80,q,100);
        q++;
        }
      }

    if (kbhit())
      {
      getchar();
      l = FALSE;
      }
    }
  l = TRUE;
  while (l && (q > 100))
    {
    for (n = 3; n < 6; n++)
      {
      a_smove (0,n,q,100);
      q--;
      for (i = 0; i < 10; i++)
        {
        a_smove(0,0x80,q,100);
        q--;
        }
      }
      if (kbhit())
        {
        getchar();
        l = FALSE;
        }
      }
    }
  }
/* end of main */
```

LISTING 2: SPRITE MANIPULATION PROGRAM WRITTEN IN MBASIC

```
10 REM          This is a basic program which uses sprites to create a
20 REM  man who walks across the bottom of the screen.  It makes use of the
30 REM  BPG.BAS file of routines to perform I/O to the paraGRAPHICS board
40 REM
50 REM Place the mman at the lower oeft corner.
60      MAN.XCOORD = 500
70      MAN.YCOORD = 21
80      LEG.HEIGHT = 6
100 GOSUB 40000
110 GOSUB 46000
120 REM This is a program that uses the sprite commands to display a man
130 REM                 walking across the screen.
140 REM Set the mode to sprite mode.
150     MODE$ = "E"
160     GOSUB 47000
170 REM Initialize the models and sprites.
180     DATA8  = &H97
190     GOSUB 49000
200     DATA8  = &H98
210     GOSUB 49000
220 REM Build some arrays containing some body and leg data.
230     GOSUB 3000
240 REM Set up the models for the body and the three leg positions.
250     GOSUB 1000
260 REM Move the body model and the first leg model  into their sprites.
270     GOSUB 2000
300 REM Now walk the man across the screen from right to left.
310     NR.STEPS = 50    : YPOS = MAN.YCOORD
312     XPOS = MAN.XCOORD
315     STRIDE = -3
320     FOR I = 1 TO NR.STEPS
330         GOSUB 800
370         MODEL.NUMBER = LEG3.MODEL
380         GOSUB 700
390         GOSUB 800
400         MODEL.NUMBER = LEG2.MODEL
410         GOSUB 700
420         GOSUB 800
430         MODEL.NUMBER = LEG1.MODEL
440         GOSUB 700
450     NEXT I
500 END
700 REM Move a leg sprite and insert a new model into it.
710     SPRITE.NUMBER = LEG.SPRITE
715     YPOS = MAN.YCOORD - BODY.HEIGHT
717     XPOS = XPOS + STRIDE
720     GOSUB 900
730     RETURN
800 REM Move the body one partial step.
810     SPRITE.NUMBER = BODY.SPRITE
820     MODEL.NUMBER = &H80
825     YPOS = MAN.YCOORD
830     GOSUB 900
840     RETURN
900 REM Do a relative move of a sprite.
910     DATA8 = &H8E
920     GOSUB 49000
930     DATA8 = SPRITE.NUMBER
940     GOSUB 49000
950     DATA8 = MODEL.NUMBER
960     GOSUB 49000
975     DATA16 = XPOS
980     GOSUB 48000
990     DATA16 = YPOS
```

```
995     GOSUB 48000
997     RETURN
1000 REM Set up the body of the man.
1004 PRINT
1005 PRINT "BODY MODEL"
1010    DATA8  = &H8C
1020    GOSUB 49000
1030    DATA8  = BODY.MODEL                        : REM sprite #1.
1040    GOSUB 49000
1050    DATA8  = BODY.WIDTH                        : REM Sprite width in bytes.
1060    GOSUB 49000
1070    DATA8  = BODY.HEIGHT                       : REM Sprint heigth.
1080    GOSUB 49000
1090    NR.BYTES = BODY.WIDTH  * BODY.HEIGHT
1100    FOR I = 0 TO NR.BYTES - 1
1110        DATA8  = BODY (I)
1130        GOSUB 49000
1140    NEXT I
1160 REM Set up the first leg model.
1164 PRINT
1165 PRINT "LEG1 MODEL"
1170    DATA8  = &H8C
1180    GOSUB 49000
1190    DATA8  = LEG1.MODEL
1200    GOSUB 49000
1210    DATA8  = LEG.WIDTH
1220    GOSUB 49000
1230    DATA8  = LEG.HEIGHT
1240    GOSUB 49000
1250    NR.BYTES = LEG.WIDTH  * LEG.HEIGHT
1260    FOR I = 0 TO NR.BYTES - 1
1270        DATA8  = LEG1(I)
1280        GOSUB 49000
1290    NEXT I
1300 REM Set up the second leg model.
1310    DATA8  = &H8C
1315 PRINT : PRINT "LEG2 MODEL"
1320    GOSUB 49000
1330    DATA8  = LEG2.MODEL
1340    GOSUB 49000
1350    DATA8  = LEG.WIDTH
1360    GOSUB 49000
1370    DATA8 = LEG.HEIGHT
1380    GOSUB 49000
1390    FOR I = 0 TO NR.BYTES - 1
1400        DATA8  = LEG2(I)
1410        GOSUB 49000
1420    NEXT I
1430 REM Set up the third leg model.
1435 PRINT : PRINT "LEG3 MODEL"
1440    DATA8  = &H8C
1450    GOSUB 49000
1460    DATA8  = LEG3.MODEL
1470    GOSUB 49000
1480    DATA8  = LEG.WIDTH
1490    GOSUB 49000
1500    DATA8 = LEG.HEIGHT
1510    GOSUB 49000
1520    FOR I = 0 TO NR.BYTES - 1
1530        DATA8  = LEG3(I)
1540        GOSUB 49000
1550    NEXT I
1560    RETURN
2000 REM Create the body sprite from the body model.
2004 PRINT
2005 PRINT "BODY SPRITE"
2010    DATA8  = &H8D
2020    GOSUB 49000
2030    DATA8  = BODY.SPRITE
2040    GOSUB 49000
2050    DATA8  = BODY.MODEL
2060    GOSUB 49000
2070    DATA8  = BODY.WIDTH
2080    GOSUB 49000
2090    DATA8  = BODY.HEIGHT
2100    GOSUB 49000
2110    DATA16 = MAN.XCOORD
2120    GOSUB 48000
2130    DATA16 = MAN.YCOORD
2140    GOSUB 48000
2150 REM Create the first leg sprite from the first leg model.
2155 PRINT
2156 PRINT "LEG1 SPRITE"
2160    DATA8 = &H8D
2170    GOSUB 49000
2180    DATA8 = LEG.SPRITE
2190    GOSUB 49000
2200    DATA8 = LEG1.MODEL
2210    GOSUB 49000
2220    DATA8 = LEG.WIDTH
2230    GOSUB 49000
2240    DATA8 = LEG.HEIGHT
2250    GOSUB 49000
2260    DATA16 = MAN.XCOORD
2270    GOSUB 48000
2280    DATA16 = MAN.YCOORD - BODY.HEIGHT
2290    GOSUB 48000
2300    RETURN
3000 REM Data table for the body of the man.
3010    BODY.MODEL = 1
3020    BODY.SPRITE = BODY.MODEL
3030   BODY.YCOORD = 21
3040    BODY.WIDTH  = 2
3050    BODY.HEIGHT  = 14
3060    BODY.SIZE = BODY.WIDTH  * BODY.HEIGHT
3070    DIM BODY (BODY.SIZE)
3080    FOR I = 0 TO BODY.SIZE - 1
3090        READ BODY (I)
3100    NEXT I
3110    DATA &HC7, &HC0,  &HC5, &HC0,  &HC7, &HC0,  &HC3, &HC0,  &HC7, &HC0
3120    DATA &H01, &H0C,  &H0F, &HE0,  &H0F, &HE0,  &H0F, &HE0,  &H0F, &HE0
3130    DATA &H0F, &HE0,  &H0F, &HE0,  &H0F, &HE0,  &H0F, &HE0
3140 REM Data table for all the three leg models.
3150    LEG.HEIGHT = 6
3160    LEG.WIDTH = BODY.WIDTH
3170    LEG.SPRITE = 2
3180 REM Date table for the first leg model, legs together.
3190    LEG1.MODEL = 2
3200    LEG.SIZE = LEG.WIDTH * LEG.HEIGHT
3205    DIM LEG1(LEG.SIZE)
3210    FOR I = 0 TO LEG.SIZE - 1
3220        READ LEG1(I)
3230    NEXT I
3240    DATA &HC2, &H80,  &HC2, &H80,  &HC2, &HC0,  &HC2, &H80,  &HC2, &H80
3250    DATA &H07, &HC0
3260 REM Date table for the second leg model, legs slightly parted.
3270    LEG2.MODEL = 3
3280    DIM LEG2(LEG.SIZE)
3290    FOR I = 0 TO LEG.SIZE - 1
3300        READ LEG2(I)
3310    NEXT I
3320    DATA &H02, &H80,  &H02, &H80,  &H06, &HC0,  &H06, &HC0,  &H0C, &H40
3330    DATA &H18, &HC0
3340 REM Data table for the third leg model, legs fully parted.
3350    LEG3.MODEL = 4
3360    DIM LEG3(LEG.SIZE)
3370    FOR I = 0 TO LEG.SIZE - 1
3380        READ LEG3(I)
3390    NEXT I
3400    DATA &H02, &H80,  &H06, &HC0,  &H0C, &H70,  &H18, &H30,  &H30, &H10
3410    DATA &HE0, &H30
3420    RETURN
```

INTERFACE SUBROUTINES FOR USE WITH THE PARAGRAPHICS BOARD:

```
40000 REM------------------- SET UP CONSTANTS --------------------------------
40010 REM
40020 REM              This routine sets up some variables that are used
40030 REM        throughout these routines as constants.  The purpose of using
40040 REM        is either to allow the hardware dependant routines to be
40050 REM        changed to match a different port mapping, or to avoid errors
40060 REM        when a constant is used repeatedly.
40070 REM
40080   DATA.PORT = &H80                       : REM paraGRAPHICS data port.
40090   STATUS.PORT = DATA.PORT + 1            : REM   "        " status port.
```

"The Game Board" Now with:

SUPERGRAPH

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

Features:

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

For more information and ordering contact:

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

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

VISA MasterCard

paraGraphics

CIRCLE 21 ON READER SERVICE CARD

EUREKA!™

"Up the creek" Searching for files?

Ever wasted time looking for a file on disk after disk? Next time let **EUREKA !** find it for you.
For CP/M 2.2 systems, **EUREKA !** is the most thorough easy to use disk cataloging system available.

- 'Menu driven' with quick and smooth response
- Put key comments in files as you work on them
- Find files by name or key text
- See disk information, including space available

EUREKA !... search no more ...

Only $50

add $2.00 shipping — Calif. residents add 6%

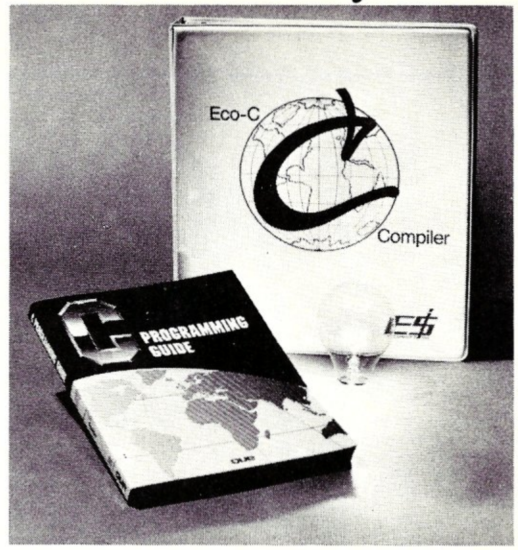
MENDOCINO SOFTWARE COMPANY, INC.
(707) 459-9130

Dept. M-1 — P.O. Box 1564 — Willits, CA 95490

CP/M is a registered trademark of Digital Research Corp.
EUREKA ! is a trademark of Mendocino Software Co., Inc.

CIRCLE 78 ON READER SERVICE CARD

We make C easy...



and work!

Eco-C compiler... we've got it all.

Whether you're a seasoned professional or just getting started in C, the Ecosoft C compiler has everything you'll ever need.

COMPLETENESS:
Our Eco-C compiler is a complete implementation of C and supports all operators and data types (including long, float and double).

EFFICIENCY:
The compiler generates extremely efficient Z80 code using Zilog's mnemonics. On the benchmarks tested, typically we finished either first or second using substantially less generated code.

PORTABILITY:
The ECO-C library contains over 100 functions that are UNIX V7 compatible, and includes a complete transcendental package. Programs developed with the Eco-C compiler can be moved to virtually any system with little or no change.

EASE OF USE:
The Eco-C compiler includes Microsoft's MACRO 80 assembler, linker, library manager and supporting documentation. The assembler (M80) generates industry-standard REL file output. The linker (L80) is fast and uses only the functions you request in the program. Program development is a snap.

The user's manual is clear, concise and full of useful information. For those of you just getting started with C, we also include a copy of the **C Programming Guide** (Que). This B. Dalton Best Seller has been adopted by a number of leading universities around the country and is included with each compiler. The book is designed to help you learn C from the ground up. We ought to know... we wrote the book.

We've made the compiler easy to work with for the professional and beginner alike. Most error messages, for example, tell you in English (not just a number) the line number and character position of the error, what was expected and a page reference to the **Guide** to consult for help if you need it.

PRICE:
We saved the best for last; we've cut the price by $100.00. Now you can buy the Eco-C compiler for only $250.00 (MACRO 80 and the book alone are worth $218.00!). Shop around and we think you agree that the Eco-C compiler is the best value available.

The Eco-C compiler requires a Z80 CPU, CP/M, 54K of free memory and about 240K of disk space (one or two drives). An IBM-PC version will be available in the first quarter of 84. To order your Eco-C compiler, call or write.

ECOSOFT INC.

Ecosoft Inc.
P.O. Box 68602
Indianapolis, IN 46268
(317) 255-6476

VISA MasterCard

TRADEMARKS:
Eco-C (Ecosoft), MACRO 80 (Microsoft), CP/M (Digital Research)

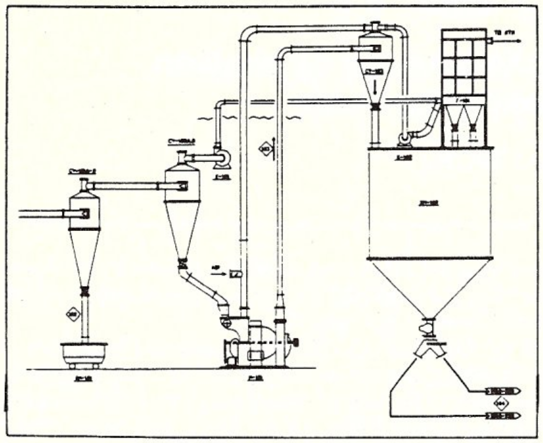
CIRCLE 154 ON READER SERVICE CARD

paraGraphics continued . . .

```
40100   STATUS.BIT% = &H80                    : REM   "          " ready bit.
40110   BLACK = 0                             : REM Color constant.
40120   WHITE = 1                             : REM   "       "
40130   ESCAPE = 27                           : REM <ESC> character.
40140   RETURN
40150 REM
40160 REM
45000 REM-------------------- LINE ABSOLUTE --------------------------------
45010 REM
45020 REM           LINE ABSOLUTE will draw a line from the coordinates
45030 REM       (x1%, y1%) to the coordinates specified by (x2%, y2%).  The
45040 REM       line may be either black or white as specified by the variable
45050 REM       called color%.  Black is specified by color% = 0, white is any
45060 REM       other value.
45070 REM
45080   IF COLOR = 0 THEN COMMAND = &H86 ELSE COMMAND = &H89
45090   DATA8 = COMMAND                       : REM Set up white or black.
45100   GOSUB   49000
45110   DATA16 = X1                           : REM Set up x coordinate.
45120   GOSUB   48000                         : REM Send out as integer.
45130   DATA16 = Y1                           : REM Set up y coordinate.
45140   GOSUB   48000
45150   DATA16 = X2                           : REM Same for end point.
45160   GOSUB   48000
45170   DATA16 = Y2
45180   GOSUB   48000
45190   RETURN
45200 REM
45210 REM
46000 REM-------------------- CLEAR DISPLAY --------------------------------
46010 REM
46020 REM           CLEAR DISPLAY will clear the screen of all graphics,
46030 REM       and text, and it will home the cursor.
46040 REM
46050   DATA8 = ESCAPE                        : REM Set up <esc> lead-in.
46060   GOSUB   49000                         : REM Send it out.
46070   DATA8 = ASC("E")                      : REM Specify clear command.
46080   GOSUB   49000
46090   RETURN
46100 REM
46110 REM
47000 REM--------------------- SET MODE ------------------------------------
47010 REM
47020 REM           SET MODE will set the display mode of the paraGRAPHICS
47030 REM       board.  The mode characters are specified in the user's manual.
47040 REM
47050   DATA8 = ESCAPE                        : REM Set up <esc> lead-in.
47060   GOSUB   49000                         : REM Send it out.
47070   DATA8 = ASC("x")                      : REM Specify mode command.
47080   GOSUB   49000                         : REM Send it out.
47090   DATA8 = ASC(MODE$)                    : REM ASCIIize the mode char.
47100   GOSUB   49000                         : REM Send it out.
47110   RETURN
47120 REM
47130 REM
48000 REM-------------------- SEND AN INTEGER ------------------------------
48010 REM
48020 REM           SEND AN INTEGER will send the integer DATA16% to the
48030 REM       paraGRAPHICS board.  It calls on SEND A BYTE so the ready bit
48040 REM       is checked before the data is sent.
48050 REM
48060   DATA8 = DATA16   / 256                : REM Get upper8 bits ready.
48070   GOSUB   49000                         : REM Send them out.
48080   DATA8 = DATA16 AND &HFF               : REM Get lower 8 bits ready.
48090   GOSUB   49000                         : REM Send them out.
48100   RETURN
48110 REM
48120 REM
49000 REM---------------------- SEND A BYTE --------------------------------
49010 REM
49020 REM           SEND A BYTE will send the lower 8 bits of the variable
49030 REM       data8%% to the paraGRAPHICS board.  The status is first checked
49040 REM       to see if the board is ready, then the data is sent to the
49050 REM       data port of the board.
49060 REM
49070   STATUS% = INP(STATUS.PORT)            : REM Read the status word.
49080   IF (STATUS% AND STATUS.BIT%) = 0 THEN GOTO 49070
49090   OUT DATA.PORT, FIX(DATA8)
49100   RETURN
```

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

LARGE INVENTORY AVAILABLE FOR IMMEDIATE DELIVERY

SMITH SYSTEMS INC.

215 Joliet St., Suite #1
Dyer, IN 46311 ● PH. 219/865-6010

MIDWEST DISTRIBUTOR FOR

HDL RESEARCH S-100 MAIN FRAMES	
MCS-112 DESK 12 SLOT W/CVT PWR	$737.
MCS-122 DESK 22 SLOT W/CVT PWR	$881.
RM-12 RACK 12 SLOT W/CVT PWR	$879.
RM-122 RACK 12 SLOT W/CVT PWR	$1036.
DDE-8 DISK ENCLOSURE W/CVT PWR	$910.
DDE-8R DISK ENCLOSURE W/CVT PWR	$1015.
TF-12 12 SLOT W/SP FOR 2-5" DRVS	$960.

Ready To Run Systems Available Now!!
Send For Our System's Flyer
(Dealer Inquiries Invited)

COMPLETE COMPUTER SYSTEMS

MORROW MD-3 64K W/MDT-20 TERMINAL & PERFECT'S SOFTWARE	$1895.
SANYO MBC-1000 64K 1-DS DD DRIVE W/MICRO-PRO LMTD QTY	$1795.
SANYO MBC-1200 64K 1-DS THIN DRV W/MICRO-PRO SOFTWARE	$1995.
SANYO MBC-550 128K 1-SS THIN DRV 16-BIT PC COMP W/SOFTWARE	$995.
LNW 80 II WITH 2 DRIVES SOFTWARE & AMDEK AMBER MONITOR	$2350.

SYSTEM COMPONENTS

8" QUME DT-8 DS/DD	$475.	8" MITSUBISHI THINLINE DS/DD	$464.
CCS 2810 Z-80 4MHZ CPU	$245.	TARBELL DD Floppy Disk CONT	$395.
MEMORY MERCHANT 64K Static	$419.	PMMI 300 BAUD S-100 MODEM	$339.
SSM I/O-8 8 PORT SERIAL	$449.	GEMINI 10X PRINTER PAR	$320.
ADDS VIEWPOINT ADM-3A +	$545.	AMDEK V300A AMBER	$165.

SPECS ON OUR BARCODE SYSTEM

Our system consists of the HP/wand and our software. Machine language installable file allows access from micro soft basic, Basic/Z, and Dbase II version 2.4. We also include programs to print barcode in the following printers, Epsom mx-80 & mx100 & fx, Citoh 1550 & 8510, Gemini 10X. Sample printout in Next AD. Our system will work with almost any CPM/80 computer that has a spare single bit Parallel input. Write for complete brochure or stop in and see a live DEMO. $250. dealers welcome.

Terms: we accept Visa/MC, prepay, check, or money order. Please allow personal checks two weeks to clear before shipment. $5.00 handling fee on all orders under $50.00. 15% restocking fee. All orders shipped via U.P.S. unless otherwise specified. All U.P.S. C.O.D. orders require cash or cashiers check. Prices subject to change without notice.

CIRCLE 55 ON READER SERVICE CARD

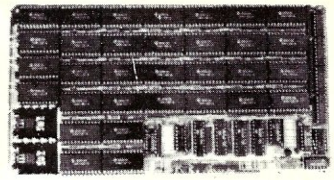
ALL SALES SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. FREE COPY UPON REQUEST.

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)

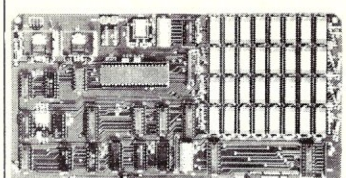
FOR 56K KIT $185

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

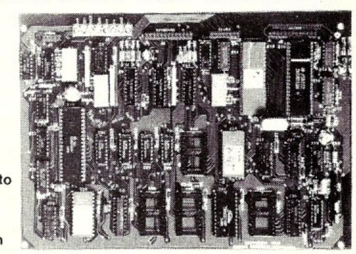
$399.00 #LS-100 (FULL 256K KIT)

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 Master-Card. Texas Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50 add 85¢ for insurance.

CIRCLE 65 ON READER SERVICE CARD

Adding Concurrency to MP/M II

Add multitasking and split-screen operation to your MP/M master console

by William G. Wong

One of the hottest new operating systems is Concurrent CP/M-86 from Digital Research, Inc. (DRI), a single-user, multitasking operating system for the Intel 8088/8086-based machine. DRI also sells MP/M II, an 8-bit multiuser, multitasking system, which, however, lacks many features of the Concurrent CP/M-86. By enhancing the Extended Input/Output System (XIOS) to include support for features found in the 16-bit Concurrent CP/M-86, users of 8-bit systems can have features such as multiple display pages with split-screen operation.

Concurrent MP/M II system structure

The XIOS becomes fairly large, nearing 14K. Of this, however, only 2K need be located in common memory; the remainder is part of the banked XIOS. This leaves a good number of free pages for Resident System Processes (RSPs). There is also some free space left in banked memory for banked RSPs as shown in Figure 1. All of the routines and data for the main console and all input queues should be in banked memory. This allows the number of enhancements to grow without increasing the size of the XIOS in resident memory. In fact, a large part of the console input interrupt routines are also located in banked memory.

Banking the interrupt routines is made possible by enhancing the XIOS SELMEMORY routine so that it remembers what the current memory bank is. Interrupt routines using banked memory consist of both common and banked portions. The common portion simply saves this information and selects the system bank. Control is then passed to the banked portion. When interrupt handling is complete, the original program bank is then restored by the common portion of the interrupt routine. The interrupt procedure can work with single- or multiple-level interrupts.

Only disk and console input is interrupt driven, as recommended by DRI in their documentation. Console input is one of the rarest interrupts on the system timewise because even the fastest typist is no match for the computer. Since the overhead due to console interrupts is low anyway, increasing that overhead by some small percentage does not significantly degrade system performance.

The actual interrupt support operation is the same as with a normal XIOS with the additional overhead of changing banks. The changing between banks can be handled easily through the use of "coroutines" as shown in Figure 2. Coroutines are used instead of RSPs or queues because coroutines have very little overhead and because the coroutine stacks are located in various banks as required by the banked memory scheme. Actually, there are four coroutines per console.

A coroutine acts like an independent process in a multitasking system in that it has its own program and data space, including its own stack. The difference is that a coroutine is called like a subroutine, but never runs concurrently with the task that calls it. However, a coroutine returns its results to the task that calls it, just as a subroutine does. In fact, the calling task does not know the difference between a subroutine and a coroutine.

When a program asks for a character from a console, it does so through MP/M. The XIOS Input Coroutine takes over and first switches to its stack, located in common memory. It then changes the memory bank from the program bank to the system bank and passes control to the Input Coroutine in the banked portion of the system bank. This coroutine obtains a character and returns it to the common memory coroutine, which then switches the memory bank back to the program bank. It is then safe to switch to the program stack located in the program bank just selected. Control is then returned to MP/M and subsequently to the program that requested the character. Output requests are handled in a similar fashion.

Now, using these coroutines may seem like a lot of overhead, but, in fact, the overhead is less than in conventional implementations, especially for the main console. This reduction occurs because much of the state information associated with a coroutine is located on its stack. Saving and restoring this information is simply a matter of PUSHing it onto the stack or POPing it off. These instructions are faster and shorter than the load (LHLD) and store (SHLD) instructions needed to do similar memory operations. The stack can be used with the coroutine approach because the coroutine program is written as an infinite loop. This makes a coroutine simpler and easier to write. The following is a typical example of the coroutine code used in this XIOS.

```
loop: push h             ; save parameters
        call coroutine   ; de: = coroutine stack
                         ; pointer
      pop h              ; restore parameters
      push d             ; save coroutine stack
                         ; pointer
      call support       ; call support routine
      pop d              ; de: = coroutine stack
                         ; pointer
      jmp loop           ; loop forever
```

Here the "support" is the routine used to provide the XIOS operation and "coroutine" is defined as the following code:

```
coroutine                ; preserves bc and psw
      lxi h,0            ; de has new stack
```

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

```
                        ; pointer
dad sp                  ; hl: = current stack
                        ; pointer
xchg                    ; hl: = new stack pointer
sphl                    ; sp: = new stack pointer
ret                     ; de: = old stack pointer
```

Note that the return (ret) instruction returns, not to the program that initially called the coroutine, but to the coroutine in the new stack. The calling coroutine will be continued when the other coroutine calls the coroutine procedure. The procedure is small, simple, and fast.

One additional advantage to using coroutines and banked support routines is that if the video RAM for the main console is located in the system bank, it is directly accessible to the banked support routines. Therefore, switching to the system bank performs two functions: selecting the video RAM and accessing the corresponding support routines.

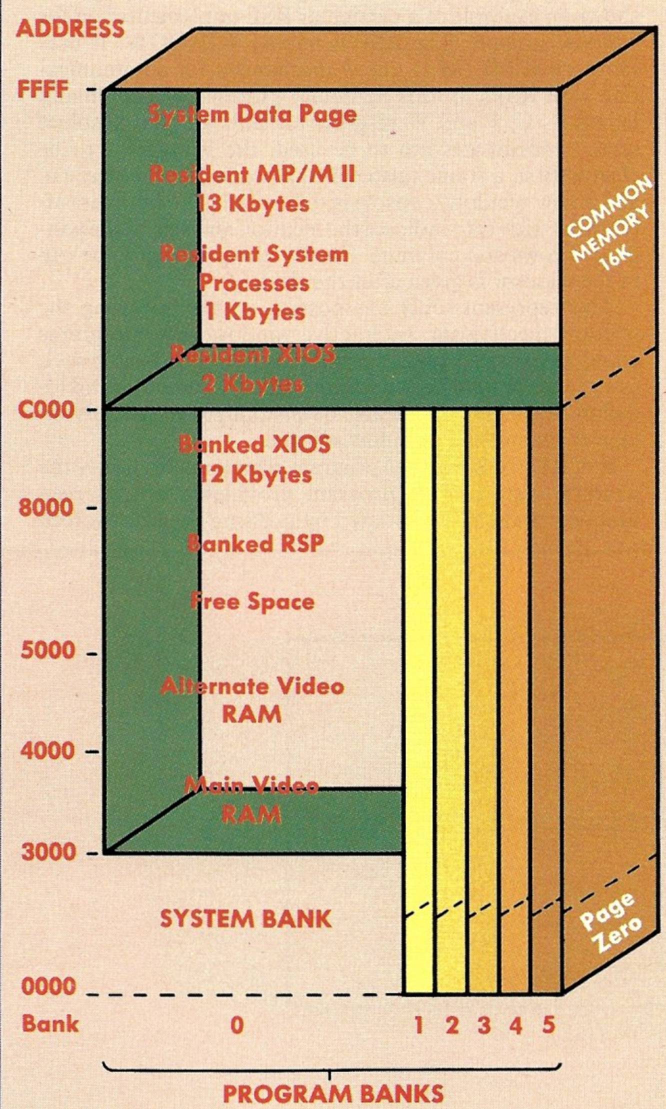


FIGURE 1

Although MP/M II is somewhat restrictive to the systems implementer, other approaches are possible. This particular approach to XIOS implementation lends itself to clean, powerful support routines. It also allows significant enhancements to be made while keeping the size of the common system memory to a minimum. This keeps the size of the program banks within their limit of 48K and still leaves room for user RSPs.

Main console concurrent support

The main console has special features not available to the remote consoles; these special features include independent multiple screens with multiple windowing capability on each screen. Also, each screen supports a subset of the ADM-31 and TeleVideo 925 control sequences plus extensions for reading the screen and programming the 20 function keys. The enhancements described below depend upon having a memory-mapped display for the main console. Concurrent support for other terminals can be done if they also have memory-mapped displays.

When the system is initialized, the screen on the main console displays the normal MP/M II prompt for console 0. Any program could now be run. Note that the display is actually showing one of two independent display pages (A and B). Page B can be shown by pressing a special function key that causes page A to be replaced by page B on the display. Page A is not lost, it is just not displayed. All keyboard input is now directed to page B. Press the same key again and page A reappears, replacing page B. Thus you can run two programs from the main console, each with its own screen to work with.

There are further benefits. A page can have one or more windows mapped onto it. A window corresponds to an MP/M console and maps to a rectangular region on a screen. Any program can be attached to a window/console. The size and location of a window can be changed dynamically, using escape sequences; a utility program can assist in this task. The result is that you can have more than one program running on a screen, and they can be set so as not to interfere with each other. Figure 3 shows an example of what might be seen on a screen when using multiple windows and running different programs. The borders separate the two windows. Each window scrolls independently of any nonoverlapping windows and has its own independent cursor. In this particular example, we see page A with window/console 0 on top where the directory program has just been run. At the same time, in the lower window, we see the last part of the output from the system status program MPMSTAT. This output is actually much larger than one window, so it scrolled out of sight; however, since the two windows are nonoverlapping and independent, this data did not overwrite the information in window 0.

We have two display pages, and there are now two programs on one page. Take a peek at Figure 4 for what might be happening on page B while all this is going on page A. Looking at page B requires just a flick of a button. This next example is intentionally more complex to show how much can be done with just one screen. There are actually three separate windows in this example. The top one is allocated to DRI's Z80 debugging program ZSID. The bottom right is allocated to the copy utility, PIP, and the bot-

A page can have one or more windows mapped onto it. Since each window has its own cursor and scrolls independently, several programs can run concurrently.

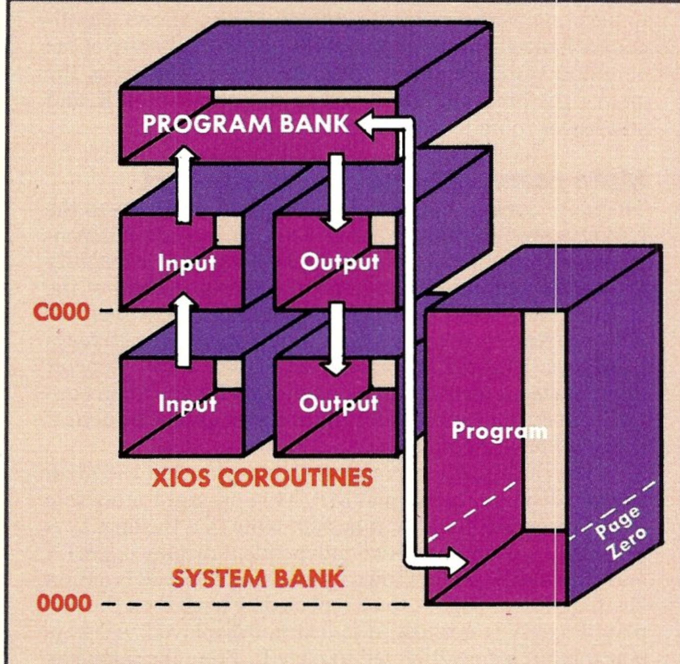


FIGURE 2

tom left shows the MP/M command line interface that was used to start the WINDOW program to set up the other windows.

Obviously, this use of multiple windows could be of great assistance when debugging programs. The possibilities are limited only by your imagination. Figure 5 shows a common use of the multiwindow system. Here, two windows are shown, without borders. Both are running different copies of the popular word processor, WordStar. They have been modified to check the size of the window they are running in, thereby adding split screen operation to WordStar—and WordStar itself is none the wiser.

Since each window is independent, it is possible to run a copy of Wordstar in one window and a copy of dBase II in another and have the function keys set up for both. A function key is used to switch between them. Each time it is pressed, the cursor on the current window is turned off. The cursor in the next window on the page is then turned on. All keyboard input is directed to the window with the active cursor. The windows on a screen are linked together in a logical loop, so any window can be selected by pressing the special function key a sufficient number of times.

Window characteristics can be modified under program control or by using an RSP designed for the job. Figure 4 shows an example of a particular RSP implementation for window support. The RSP is named WINDOW. It uses the normal MP/M II queue mechanism for communication with RSPs. In this particular example, the command letters A, C, F and W respectively indicate that absolute screen coordinates are to be used, the window is to be cleared first, a frame (also called a border) is to be drawn, and a new window/console is to be used. The numbers after the at-sign (@) indicate the location and size of the window in rows and columns. Finally, the program to be run in the window is given after the slash (/).

This represents only one possible method of using the window mechanism. Interactive window setup programs could be written, possibly employing pointing devices such as a mouse or light pen to specify window attributes and location. Applications can also be customized to take advantage of the multiple window scheme.

PSCREEN is another program which makes use of the window system. This program provides a print screen function that can be initiated by pressing the print screen

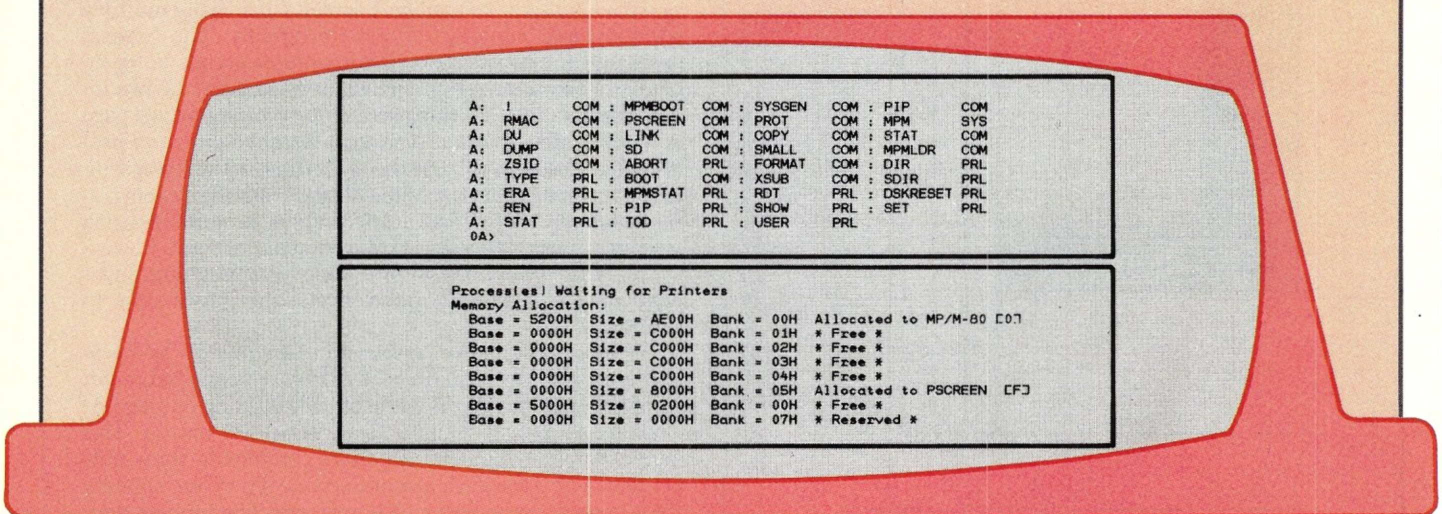


FIGURE 3

All keyboard input is directed to the window with the active cursor. The windows on a screen are linked together in a logical loop, so that any window can be activated by pressing a special function key.

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

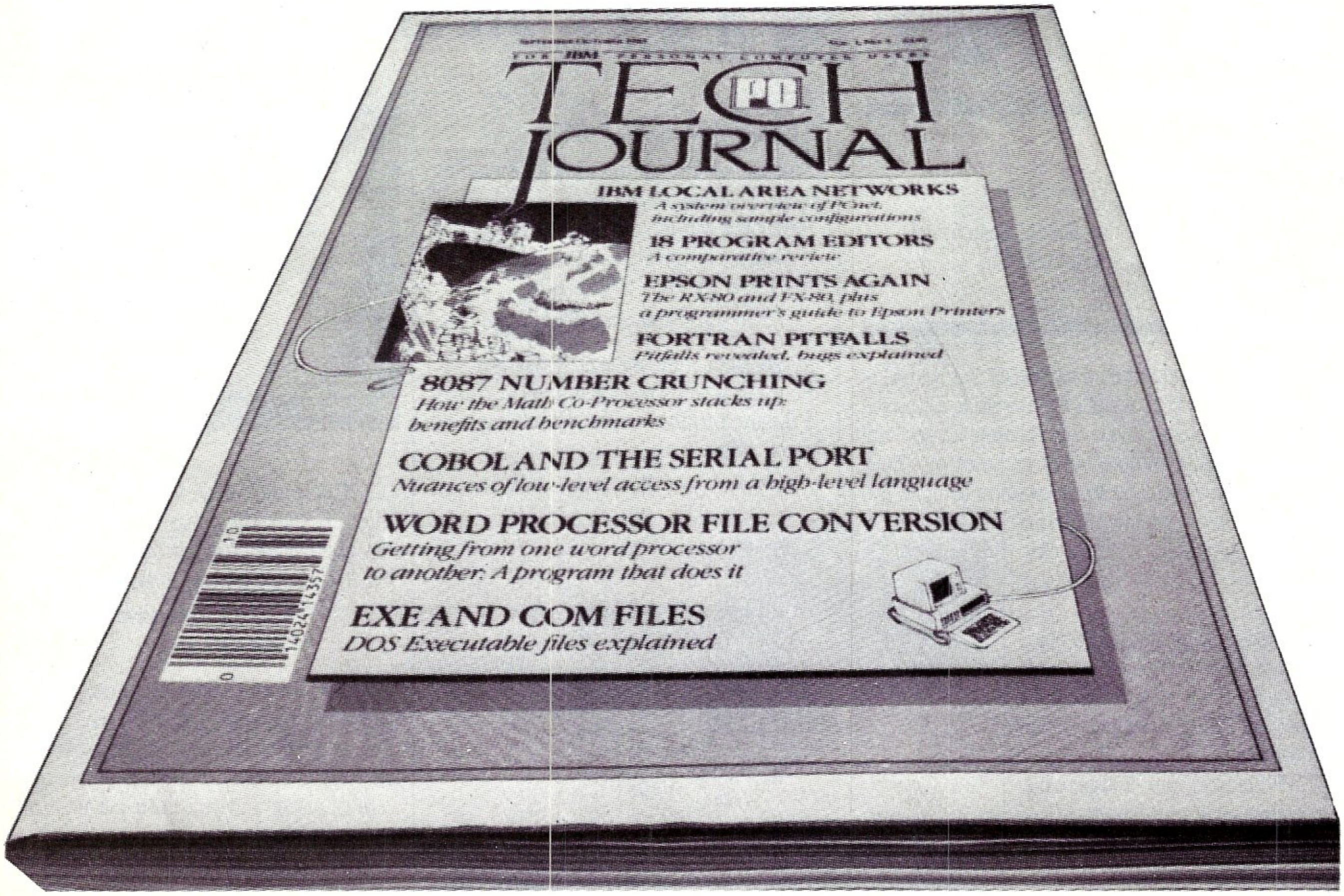
8080, Z80 or IBM PC	$150
CP/M-86 or MSDOS	$195
Manual only	$18
On-Line Tutorial	$15
Mail Merge	$15

CIRCLE 22 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

If You're an Advanced IBM™ PC User, You're Ready for



If you are serious about microcomputers and want to untap the host of possibilities offered by your IBM PC, PC TECH JOURNAL is the magazine for you.

Brought to you from the publishers of *PC*, PC TECH JOURNAL is the definitive authority on how to maximize the potential of your IBM personal computer, as well as develop new applications and face creative new challenges.

WHY A TECHNICAL JOURNAL?

Because:

- PC TECH JOURNAL gives you more than simple problem solving exercises with your PC
- PC TECH JOURNAL tells you how your PC works and how you can get it to do more!
- PC TECH JOURNAL shows you how to tailor your IBM PC to your own needs
- PC TECH JOURNAL gives you information you cannot find anywhere else!

EACH ISSUE IS FULL OF USEFUL ARTICLES ON TOPICS

Such as:

- How to fully exploit BIOS
- Examinations of component hardware available for the IBM PC
- Programming methodologies which exploit the IBM PC's present and potential operating system environments
- Evaluation of the IBM PC's communication protocols and standards: computer to computer connections, terminal emulation, and distributed data processing
- PLUS networking, mass storage and new user development tools

In short, if you have a genuine appetite for state-of-the-art information about your IBM PC, you will want to subscribe to PC TECH JOURNAL today! Just fill in the handy order form!

TECH PC JOURNAL
P.O. Box 2966
Boulder, Colorado 80322

YES, please send my PC TECH JOURNAL for:

☐ One year (12 issues) only $24.97 and SAVE 17%
☐ Two years only $43.97 and SAVE 27%
☐ Three years only $57.97 and SAVE 36%
Savings based on full one-year subscription price of $29.97.

Mr./Mrs./Ms.________________ (please print) 8H310
Company________________
Address________________
City__________State__________Zip__________

CHECK ONE: ☐ Payment enclosed ☐ Bill me later

Charge my:
☐ American Express ☐ Visa ☐ MasterCard

Account No.________________Exp. Date______

Add $12 a year in Canada and all other foreign countries. Please allow 60 to 90 days for delivery of first issue.

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

key or through an escape sequence. PSCREEN starts by selecting an unallocated window. It then sets up the window to be on the physical page, which is the one always being displayed. The window covers the whole screen. The program then waits on an MP/M flag. This flag is set when the print screen key is pressed.

At this time, PSCREEN tries to attach to the list device and then issues a read screen escape sequence to the window. It then reads the contents of the screen using normal console input commands and prints the characters on the printer. When the screen had been printed, the list device is released so that other programs can use it. This processing repeats forever. PSCREEN also turns off the cursor for the window so it cannot be selected by the special function keys. Thus the print screen window and program are essentially invisible to the user.

The implication of this print screen implementation is twofold. First, the print screen option is loaded only if required by the user, and it takes up no space when not in use. Second, the implementation requires neither modification of the operating system nor direct access to the screen memory. This means that the user can customize the print screen program for various printers or screen conditions. It is even possible to have more than one print screen program. In fact, the screen could be copied to a file instead of the printer. Try that with any other system.

A few comments about this particular window system. First, the video RAM location is hard wired. Swapping in the alternate page on the screen is done by copying this area into an alternate memory buffer and replacing it with the contents of the alternate page. This is actually accomplished by a high-priority task that waits on an MP/M flag associated with the CALC MODE key. It is possible to have more than two display pages; however, two seems to be adequate given the multiple window system. The main console can actually be used to run as many programs as there are program banks.

Setting the attributes of a window is accomplished through escape sequences including read screen data and attributes that are initiated with escape sequences. Programming the function keys is also accomplished with escape sequences. This approach provides a flexible yet portable interface to the window system. The following are some examples of the sequences used to control the screen:

Sequence	Description
ESC = (row) (column)	Position cursor at (row) (column)
ESC ESC 0	Set upper left window limit
ESC ESC 1	Set lower right window limit
ESC ESC 2	Reset window to full screen

Note: ESC is the escape character (27 decimal).

The first sequence matches the ADM-31 and TeleVideo 925 cursor positioning sequence. The remaining three sequences are all that are required for setting the size and position of a window. Positioning a window is simply a matter of moving the cursor to the top left, issuing ESC ESC 0, moving to the lower right corner and issuing ESC ESC 1. Absolute window positioning is done by first issuing the "reset window to full screen" sequence.

Although operating system function calls may be faster, the control sequence approach is more portable. It also means that an entire setup can be contained within a text file. In fact some window demonstrations are actually done by simply printing a file on the console. Note that the escape sequences listed here are for a particular implementation. Other concurrent implementations may support different sequences.

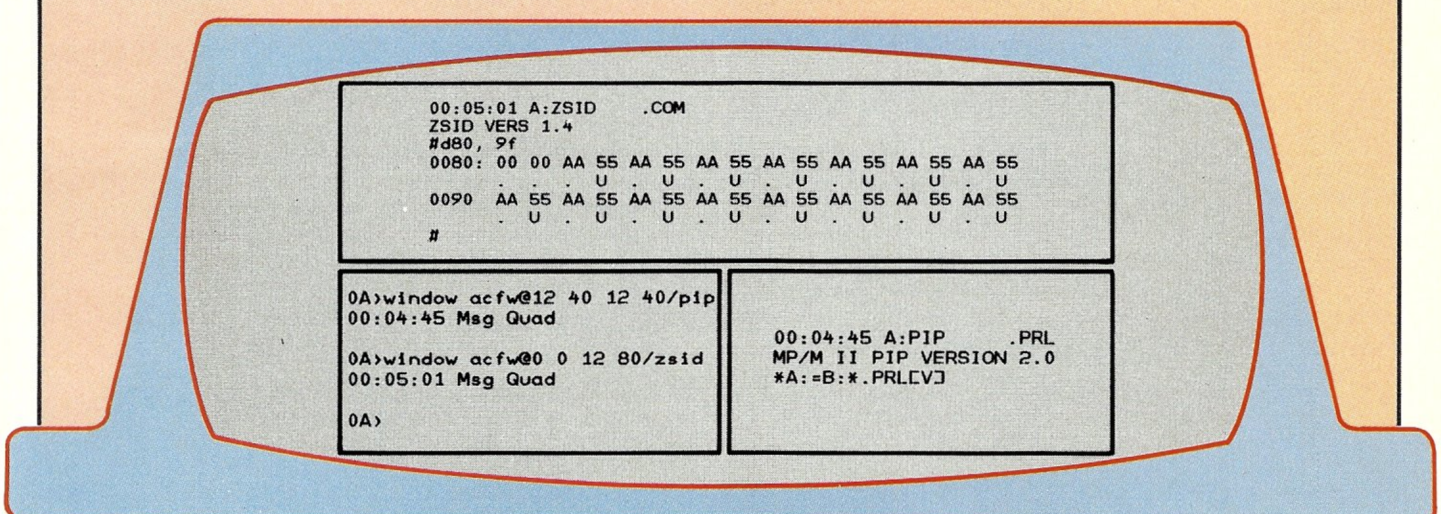


FIGURE 4

Since each window is independent, it is possible to run WordStar in one window and dBase II in another, and have the function keys set up for both. A function key is used to switch between them.

Remote console support

Even though the main console is the flashy part of the system, it still provides access for only one user: it is, however, possible to extend the use of the system by supporting remote consoles in addition to the main console.

The remote consoles' numbers start at two and can be used as communication ports or as MP/M II consoles, depending upon the system generation parameters. The console number order should be such that ports intended primarily as communication ports have higher console numbers than MP/M consoles. Using a remote terminal as an MP/M console is simply a matter of selecting the proper number of system consoles in the MP/M system generation process. Other consoles can be used as communication ports.

The serial ports can also be used to support communication programs if they are not used as MP/M II consoles. The communication programs can also make use of the interrupt-driven support within the XIOS, without having to modify MP/M or the XIOS. The programs can also use the mutual exclusion system built into MP/M to make sure that two users do not try to use the same communications port at the same time. How is all this accomplished?

Well, first the communication ports are still accessible as consoles under MP/M, even if they are not system consoles. However, most communication programs use the console to talk to the user, so using the console interface is not the best way to access the ports. Even so, the exclusive access to a console can be used to prevent two programs from simultaneously using a particular communications port. Therefore, the first thing a communications program does is to use the MP/M II Assign Console function (149) to get exclusive access to the communications console while retaining access to the users console. Attaching the communications console to the program is a form of mutual exclusion, since only one program can be attached to the console. Any attempt to assign a console currently in use will fail. If the attempt fails, the communications program should tell the user that the port is in use.

The second thing the communications program must do is locate the entry points for the four communications routines for the particular port. These routines are:

- Receiver input status
- Transmitter output status
- Get a character from the receiver
- Send a character to the transmitter

They access the interrupt-driven queues in the XIOS with the appropriate handshakes. Each communications port has a block containing these entry points as a jump table, along with additional information on the serial port. The blocks are easily found in this implementation, since they reside just above the Z80 interrupt table, which can be found using the Z80 interrupt register.

Another method for making this information known would be to place it at a fixed location above the MP/M COMMONBASE location. Using these entry points also means that the communications program never needs to modify the interrupt vectors used by the system. Additional information on baud rate settings and such can be placed after the jump vectors.

Releasing the port is simply a matter of detaching from the communications port. In fact, this is automatically done by MP/M II if the program terminates without doing so. Using this approach for communication port access is both easy and secure. It also increases the stability of the overall system, since users do not have to make changes to the operating system to get their favorite communications program to work.

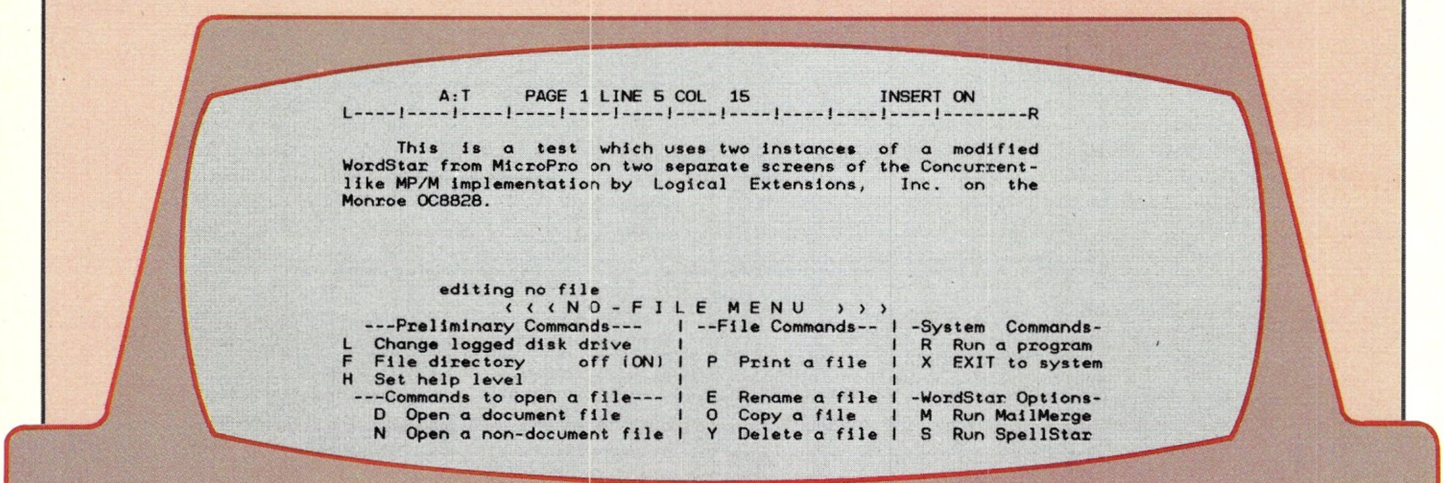


FIGURE 5

Interactive window setup programs could be written, possibly using pointing devices such as a mouse or light pen to specify window attributes and location. Applications can also be customized for a multiple window scheme.

Concurrent MP/M II continued. . .

Summary

MP/M II, when enhanced with concurrency, provides a flexible and powerful working environment. Multiple window and screen support on the main console allow a user to work concurrently with multiple programs. Multiple consoles can also be added to the system, allowing people to share facilities such as the printer and hard disk.

The particular implementation described here was done by Logical Extensions for the Monroe 0C8828. Remote consoles are supported, but only the main console has the support for concurrent operation. Implementing MP/M II with concurrent features on other machines is possible and should not be difficult for anyone who has already developed a XIOS. The results are well worth the additional implementation effort.

Vendor Information

MP/M II with concurrent features for the Monroe 0C8828 is available from:

Logical Extensions, Inc.
26 Broad Street
Cranford, NJ 07016
(201) 272-1793

CIRCLE 329 ON READER SERVICE CARD

Price per Computer: $995 in 1-4 quantity; $750 in 5-9 quantity.

Price includes disks for the 0C8828, documentation, and an MP/M II license from Digital Research, Inc.

PRODUCT REVIEW

CIS Cobol from Micro Focus

by Leonard Schwab

CIS (Compact, Interactive, Standard) Cobol should be of great interest to vendors who want to sell micros to corporate systems managers, to systems managers who want some reassurance that their present programming staff will be able to deal with the "personal" computers proliferating within their companies, and to mainframe computer professionals who want to become involved in the growing microcomputer applications industry. CIS Cobol is also an excellent vehicle for the teaching of Cobal. It is distributed by Micro Focus, a firm specializing in Cobol language processors.

The CIS (pronounced "kiss") Cobol system includes a compiler that produces a file of intermediate code (INT-code) and a runtime system (RTS) that executes the intermediate code. Two important optional products are available: FORMS-2, a source code generator that translates screen formats into Cobol data-division statements, and ANIMATOR, a super debugging package. Together, they make up an environment that supports the efficient development of highly portable Cobol applications.

Cobol in the micro environment

The Operating Guide for CIS Cobol opens with this predictable sentence: "Cobol (Common Business Oriented Language) is the most widely and extensively used language for the programming of commercial and administrative data processing."

Cobol is a very old language by computer industry standards. It was conceived and is still maintained by the Conference On Data Systems Languages (CODASYL), an amalgam of government agencies, business users, and hardware manufacturers. The fundamental specifications were established in 1960 and adopted by the American National Standards Institute (ANSI) in 1968. The last major revision of ANSI Cobol was published in 1974. Despite its popularity on mainframes and minis, Cobol has not been widely used on microcomputers. The 1974 version, upon which CIS Cobol was based, is primarily a batch or file-processing language. It requires substantial amounts of memory and storage and has virtually no capability to support interactive programs. The need for an interactive Cobol has been recognized in the mainframe world, and appropriate revisions are now being considered. However, as often happens, practice moves faster than the standard-setting process, and many Cobol vendors have come forth with their own versions of interactive facilities.

In CIS Cobol, Micro Focus has produced a compiler that operates easily within the memory capacity of 8-bit microcomputers, thus earning its appelation, "compact." The ACCEPT and DISPLAY statements of standard Cobol have been substantially enhanced to take advantage of the capabilities of modern CRT terminals; these justify the "I" in "CIS."

Leonard Schwab, 2728 Garber St., Berkeley, CA 94705

This compact system, with its nonstandard interactive facilities, is still able to produce certifiably standard Cobol programs. The compiler has met the requirements of the U.S. Government's General Services Administration (GSA) at the "low-intermediate" level. Most minicomputer compilers are rated no higher than this. The GSA certification allows CIS Cobol to be used in government projects and also provides nongovernmental users with assurance that applications will conform to standard COBOL, an important consideration when portability is required.

The 1974 ANSI specification is divided into a number of "modules" and provides for two implementation levels: "minimum standard," or level 1, and "full standard," or level 2. CIS Cobol includes a full implementation of the level 1 standard, including tables; sequential, relative and indexed file processing; debugging; ability to copy source code from library files; and ability to segment and overlay program modules. Level 2 features that have been implemented include Nested IF statements and the PERFORM UNTIL statement, among others.

CIS Cobol does not include the following verbs, which are found in some other microcomputer Cobol implementations:

- STRING and UNSTRING
- COMPUTE
- PERFORM VARYING

IF-statement conditions may not be connected with AND or OR, but proper nesting of IF statements will achieve the same results. Missing also are level-88 condition names, which allow a programmer to assign names to predefined logical conditions.

Because CIS Cobol has been implemented as a "semi-compiled" language, program functions that are specific to a particular environment rather than to a particular application program may be placed into the runtime system. The most obvious class of such functions is that which implements the features available in the system console device or terminal, e.g., screen clearing, cursor addressing, etc. Micro Focus has made it easy to modify the runtime system to take care of the requirements of the various types of console devices that might be encountered.

Furthermore, since only the relatively small RTS needs to exist in the native language of the host processor (e.g., an 8080, 8086, Z8000 or 68000), programs in the INT-code "language" may be easily transported to different systems. This feature of CIS Cobol should be very important to software distributors.

Beyond the obvious need to provide for various terminals, Micro Focus has given the programmer a substantial degree of freedom to customize the runtime system. Machine-language subroutines may be installed into the RTS and called from a Cobol program at the sacrifice of a degree of portability. It is also possible to drop certain parts of the RTS, such as the debugging facility or the indexed sequential facility, when they are not needed. Finally, CIS Cobol allows the INT-code for a completed application to be linked with the RTS, forming an execution module that can be loaded and run with a single command.

Of course, there are trade-offs. A program in INT-code, which must be interpreted when executed, is not likely to run as fast as one that exists in the native code of the machine. The CIS compiler and the utilities, ANIMATOR and FORMS-2, are all in INT-code form. After using CIS Cobol for several weeks, however, it was my impression that the system has adequate execution speed. The speed of the display updating routines is especially impressive.

Our review copy of CIS Cobol was version 4.5, revision 1, released by Micro Focus in May, 1982. The system, as tested, runs in any 8-bit CP/M machine with at least 48K of memory. It includes internal checks that allow it to be run under either CP/M 1.4 or 2.2, though the latter gives superior performance. Versions of CIS Cobol are available for other operating systems and for other CP/M configurations, including 16-bit systems.

The designers of this system appear to have been particularly aware of the specific requirements of a professional programmer. The packaging reflects this understanding. Each product comes in a separate three-ring binder. The binder fits on a desk shelf and opens into an easel for desk reference. The pages are heavy, coated paper, and the binder rings are flat on one side so that the pages lie flat. A flap covers the rings, protecting the distribution disk in its pocket in the left cover.

A Software Issue Bulletin provides information specific to the particular microprocessor and operating system. An operating guide contains most of the guidance needed by an experienced Cobol programmer using CIS Cobol for the first time. Finally, there is a Language Reference Manual in the obtuse style of most Cobol manuals. Written in highly technical language and containing no examples, it is definitely not for beginners. Nonstandard extensions to standard Cobol are clearly indicated in the manual.

Most of the information needed to use CIS Cobol is in the Operating Guide. There is no index, but a detailed, well-organized table of contents usually leads one to the desired reference. The language manual has a table of contents, an index, and a helpful glossary of terms.

Micro Focus has also provided a short "Getting Started" manual for first-time users. Created primarily for CP/M-86 systems (e.g., IBM PC) but generally applicable, it guides you through the process of opening the package, configuring an RTS for the user's terminal, and compiling and running the sample programs that are supplied. A substantial part of "getting started" duplicates material in the operating guide, though in somewhat less technical language. One of the sample programs provides a nonsignificant error so that the error-reporting features of the compiler may be demonstrated.

Finally, Micro Focus provides a small (4" × 5") pocket guide containing a complete syntax outline, lists of compiler and RTS directives, and other useful information.

Creating, compiling and running a CIS Cobol program

CIS Cobol source code files may be created with any utility text editor capable of producing a "pure" ASCII file. The compiler cannot deal with embedded control characters and, like other Cobol compilers, expects source code records to conform to a particular structure. I found it inconvenient not to be able to use the tab key when editing source code. (My editor does not expand tabs to spaces.)

When the compiler is invoked, the full filename and extension of the source code file must be explicitly entered. Though the default extensions for source code and intermediate-code files are ".CBL" and ".INT" respectively, neither program will supply an extension to the filename.

Fortunately, the compiler does not require a source file to have a filename extension. I found it most convenient to avoid the use of filename extensions when naming my source files.

The nature and destination of compiler outputs may be modified by including appropriate "directives" in the command line (see Table 1). The normal output of the compiler includes a listing file, an INT-code file, and messages to the user's console. Listing files are rarely needed, except for final documentation and for logic debugging. If ANIMATOR is available, listing files aren't even necessary for debugging. Compilation time is substantially shortened when the listing file is suppressed. I routinely specify the NOLIST option when compiling CIS Cobol programs, but I resent having to do so.

The compiler checks the syntax of the source code and reports any errors. In the absence of a contrary directive, each line found to contain an error is displayed on the terminal, along with a numeric error code and a message describing the error. If an error is generated, it is similarly reported in the program listing output file.

Since the error messages are contained in a disk file, there is some delay in processing whenever an error message must be fetched from the message file. (This system cries out for fast disks or disk-emulating RAM.) The error display visually indicates the specific part of the line in which the error was detected, aiding the programmer in interpreting and correcting the mistake. Certain types of errors reported by the compiler are actually due to programmer mistakes in prior lines (such as the omission of a required period), and the line with the actual error is not shown. Experienced Cobol programmers will be familiar with this effect, since it is common to all Cobol compilers, but beginning Cobol users may be confused.

The compiler is not particularly fast but not annoyingly slow either. It "feels" somewhat slower than CBasic, another semicompiling system, but appreciably faster than Microsoft Cobol, a native-code compiler. Compiled programs appear to run with adequate speed. The display functions are especially fast.

If the compiler is loaded without specifying a source-code input file, the compiler displays the messages "COMPILING CONSOLE INPUT". The user can then type in lines of source code. The lines are checked for syntax as entered. Output is to files named "CONSOL.LST" AND "CONSOL.INT". This mode of program creation is only suggested in a flowchart in the manuals, and does not appear to be otherwise documented (or useful).

If the runtime interpreter is loaded without specifying an input file, the system loads and pauses with no message. By trial and error, I discovered that the user may then type in an "INT" filename and processing will continue. I

The error display visually indicates the specific part of the line in which the error was detected, aiding the programmer in interpreting and correcting the mistake.

found no documentation of this action.

For users who have a "GO" command available in their operating system, the command is not useable for the compiler, which seems to modify its memory image when run, but can be used to reinvoke the runtime interpreter.

Configuring the RTS

Unless your system includes a ADM-3 compatible terminal device, an RTS must be configured prior to running any compiled program. Micro Focus provides a utility program, called CONFIG, which is used to "customize" the RTS for specific terminal characteristics. CONFIG may also be used to reserve an area for custom machine-language routines, if needed.

A CONFIG session involves answering about 35 questions about your console device. The operating manual contains the answers to be given for more than 30 specific terminals. The customization is very thorough. In addition to defining cursor control, screen clearing, highlighting, and other terminal functions, you may also redefine certain keys, e.g., tab, escape, return, which the RTS will recognize during data-entry operations. One may even specify that I/O be done directly through the hardware ports, rather than by use of the operating system. Micro Focus seems to have thought of everything.

Well, not quite everything. My console device requires output to a port in order to activate the console alarm. This is not provided for in CONFIG. Fortunately, however, the distribution disk I received included an undocumented assembly language listing of patches for an unusual console device that could not be accommodated by CONFIG. The listing helped me find the "hooks" in the RTS that I needed to patch the RTS for my console alarm. This information may not be included with the system, as it is currently being delivered.

The dialogue with CONFIG is completely sequential. No provision is made for going back to a question, once completed. The only recourse is to abort CONFIG and start all over. This is a minor inconvenience, since CONFIG will not be run very often.

Users should also be aware that CONFIG doesn't attempt to find and open an existing RTS file for input and modification, until after the long dialogue is finished. If it is unable to open the file named by the user at the start of the CONFIG session, all of the work done during the session will be for naught.

A commercial software house could generate a number of customized versions of the RTS, making it very easy to deliver operating, interactive applications to customers with different terminals without having to change the application programs themselves. The specifications for new terminal types may be saved to disk, precluding the need to rerun CONFIG for the same type of terminal.

Interactive facility

Interactive programs in CIS Cobol are implemented by using the extended DISPLAY and ACCEPT verbs. Blank forms are defined by appropriate data-division records. The areas within the form where data entry is to take place are designated by redefining the blank form record. An entire form is brought to the screen by a single DISPLAY statement, and all the data-entry areas within the form are processed by a single ACCEPT statement. Alternative forms of both verbs allow one to specify the line and column coordinates at which a statement will be executed.

The process of coding the data-division statements for an interactive data-entry program may be automated by use of FORMS-2, an optional source code generator. FORMS-2 can also generate simple, stand-alone programs to maintain indexed sequential files.

FORMS-2 will not eliminate the difficult work involved in designing a set of data-entry screens for a particular application, but it does mechanize the tedious process of writing the data-division statements. The code generated by FORMS-2 may by COPYed directly into applications programs. The time it takes to learn how to use the many powerful features of FORMS-2 will be rewarded if one has to create data-entry programs on a regular basis.

When an ACCEPT statement is processed, the cursor is under absolute control by the system and may not be moved into any area not defined as a data-entry area. Each keystroke is examined and, for example, alphabetic characters will not be accepted into a numeric field. I am very pleased by these aspects of the implementation.

On the other hand, some negative aspects must be noted.

Unless one is very careful in designing and defining the record descriptions, substantial amounts of runtime memory may be wasted in the form of blank spaces generated by FILLER items within a form record. This is seen in the sample programs that illustrate the use of DISPLAY and ACCEPT.

Perhaps more significantly, a multifield record is processed by ACCEPT as a unit. This means that it is not possible for a program to return the cursor to, say, the third field in a form if the program detects invalid data in that field. Actually, it is not impossible to achieve the described action, but doing so will require careful record descriptions and complex procedures. In summary, the ability of CIS Cobol to process a multifield form with a single ACCEPT statement, while interesting, may not be useful in many real-world applications.

During execution of an ACCEPT statement, certain

```
01 SCREEN-HEADINGS.
02 ASK-CODE      PIC X(21)     VALUE "STOCK CODE          <   >".
02 FILLER        PIC X(59)
02 ASK-DESC      PIC X(16)     VALUE "DESCRIPTION         <".
02 SI-DESC       PIC X(21)     VALUE"                         >".
02 FILLER        PIC X(43)
02 ASK-SIZE      PIC X(21)     VALUE "UNIT SIZE           <   >".
```

Figure 1. *Example of a 'form-mask' record, defining a "blank form" to be placed on the screen.*

```
01     ENTER-IT REDEFINES SCREEN HEADINGS.
02 FILLER                    PIC X(16).
02 CRT-STOCK-CODE            PIC X(4).
02 FILLER                    PIC X(76).
02 CRT-PROD-DESC             PIC X(20).
02 FILLER                    PIC X(60).
02 CRT-UNIT-SIZE             PIC 9(4).
02 FILLER                    PIC X.
```

Figure 2. *Example of an 'input-mask' re-defining data entry areas in a previously defined form-mask area.*

cursor-control keys become active. The left-arrow key backs the cursor by one position, nondestructively. If the cursor is already in the first position of the first field, the console alarm beeps. If the cursor is in the first position of any except the first field in an input mask, the cursor moves back to the last position of the prior field. The right-arrow cursor control key operates in a mirror-image fashion. The up-arrow key moves the cursor to the first position of the current field; the down-arrow puts the cursor at the start of the next field, if any. The "home" key moves the cursor to the first position of the first field.

If the character input entered by the operator fills the area allowed for a particular field, the cursor automatically moves to the next field. The "return" key terminates the ACCEPT statement, and the program continues at the next executable statement in the procedure division. The action of the RETURN key is not affected by the position of the cursor within an input mask. In other words, pressing "return" always causes an ACCEPT statement to terminate, whether or not all the input fields have been processed.

It is useful, in many applications, for data entry to terminate automatically without operator intervention when a field is filled. But this is not always the best convention, and auto-termination can, in some cases, be confusing to the operator. Furthermore, because of these conventions, the operator of an CIS Cobol application program needs special training. Generally, one terminates field input by pressing the return key; with CIS Cobol, one ends field input either by filling the field or, if the field is not filled, by pressing the down-arrow key. It may be possible to avoid the training problem by redefining the "return" and down-arrow keys, using the CONFIG program.

Debugging CIS Cobol programs

An impressive array of debugging tools is available. The compiler includes both a version of the standard COBOL debug module and an extended interactive runtime monitor. In addition, ANIMATOR, an extra-cost optional debugging system, might be characterized as the Rolls-Royce of debugging tools.

The standard Cobol debug module provides for tracing the flow of execution through all or a part of a program and for the inclusion of standard Cobol statements in the source code, which are to be compiled and executed only if certain switches have been activated. These statements could, for example, cause information about changing data values to be displayed only during the debugging process.

The extended runtime monitor provides facilities to test the logic of a program while it is being executed. The user may set breakpoints, trace paragraphs being executed, execute one statement at a time, and examine and modify contents of data storage areas. Groups of debugging commands may be combined into named "macros," but the macros may not be saved from session to session. The monitor includes commands that allow the user to include formatting instructions, and comments in macros to improve the readability of displayed information. In order to use this facility, the user must have a printed copy of the compiler's output listing to determine the hexadecimal addresses of program and data areas required by the monitor.

ANIMATOR

ANIMATOR is an debugging facility for programs compiled by the CIS Cobol compiler. It virtually eliminates the need for printed listings of a program under development, and no special statements are required in the source code. When ANIMATOR is to be used, a compiler switch is set which causes three special disk files to be generated in addition to the normal output of the compiler. The INT-code module generated by the compiler is not materially affected, so there is no need to recompile the program for normal execution.

ANIMATOR places a copy of the start of the procedure division of the source code onto the screen and shows the flow of execution by moving the cursor from statement to statement in real time. Additional parts of the source code are paged from disk as they are needed. The content of the "user screen," i.e., of the screen as it would appear if the program were being run normally, is maintained in memory and may be displayed at any time.

ANIMATOR makes clever use of the screen, and several features are worth noting. Whenever the screen is updated, the source code is positioned so that the statement about to be executed is on the third line. This enables the user to view the context of the statement. The bottom three lines of the display are reserved for ANIMATOR prompts and messages. A delimiting line of hyphens sets off this area from the rest of the screen.

The source code display area (top 20 lines) may be subdivided into two independent windows. The relative size of the two windows may be adjusted by the user. Using this feature, it is possible to minimize time-consuming paging operations when the flow of the program involves jumps between two separated sections of the source code.

The user controls ANIMATOR operations by means of commands entered from the keyboard. Most ANIMATOR commands consist of two characters. A prompt line showing the first character for sets of commands is usually displayed near the bottom of the screen. Depressing one of these keys will cause the prompt line to display the subcommands available. After a short period of use, one becomes familiar with the most-used commands and, by rapid entry of both characters, the submenu display may

The compiler "feels" somewhat slower than CBasic, another semicompiling system, but appreciably faster than Microsoft Cobol, a native code compiler.

be circumvented in the style of WordStar.

In general, the operands are indicated by the position of the cursor within the source code display so that the process is to move the cursor to a specific place and then to press the desired command sequence. When the display is split, commands that scroll source code will operate only on the window in which the cursor is displayed.

During a session, the user may scroll through the source code or find a specified string. When a target string is found, the display is updated, if necessary, and the cursor is positioned at the character following the target characters. This is somewhat annoying, usually, since you really need to have the cursor positioned at the start of the word in order to invoke some other ANIMATOR command, and it becomes necessary to back the cursor manually to the desired location.

The user may set breakpoints (places in the program where execution will be stopped) simply by positioning the cursor at the first character of the appropriate Cobol verb and invoking the set-breakpoint command. Up to four breakpoints may be set at any time. Commands are available that will find the next breakpoint in the program, unset a breakpoint pointed by the cursor, or cancel all breakpoints.

Program execution and ANIMATOR is controlled by these single-key commands:

X — Executes one statement
G — Starts continuous execution
K — Skips the statement being pointed to
I — Operates like "G," with an implied breakpoint at the next IF statement
Z — Starts execution without animation, subject to breakpoints, if any, or to depression of the space bar

In the continuous execution mode (G), program flow is graphically demonstrated by the movement of the cursor from verb to verb through the source code display. The speed of execution is controllable over a wide range by depressing numeric keys (again, like WordStar). If desired, ANIMATOR can be instructed to show only major blocks of code being executed (e.g., by pointing to PERFORM statements rather than each verb within the block being PERFORMed.)

If a statement to be executed is not on the screen, the display is updated. Execution may be stopped at any time by depressing the space bar. It will automatically stop if a breakpoint is encountered or a command entered.

The user is able to query ANIMATOR at any time for the current value of any data item. This may be done by pointing to the name of the item in the source code or by typing in the name. After displaying the value, ANIMATOR gives the user an opportunity to change the value if desired. Any single data item may be continuously monitored during the execution of the program. If a data item is being monitored, its value is displayed constantly at the bottom of the screen.

Upon encountering an ACCEPT statement, the display switches from source code to the "user screen," as it would currently exist in normal operation. The user is then able to input data from the keyboard. Control is returned to ANIMATOR when the input operation is finished.

DISPLAY statements that contain cursor positioning are directed to the invisible user screen buffer and cannot be seen unless the user screen is explicitly called up. DISPLAY statements without cursor control will cause data to be displayed on the last line of the ANIMATOR screen.

My overall impression of ANIMATOR is very favorable. It is easy to learn and to use. I wish that I had a debug module like this for every language I use. My major objection to ANIMATOR is that it is somewhat slow. The load time is lengthy, and the paging of source code from disk to screen takes time. Of course, this criticism must be tempered by the fact that the paging system makes possible the animation of a much larger program than would otherwise be possible in a given amount of memory. Furthermore, systems that use

Table 1. CIS Cobol compiler directives

I. Directives affecting listing file output

A. NOLIST—Suppresses output of listing
B. LIST (file name)—Directs listing to named file or output device. Default: source-file-name.LST
C. FORM (integer)—Sets listing page length, in lines. Default: 60 lines. Minimum: 5 lines.
D. NOFORM—Suppresses pagination of listing. Default: paginate and head each page.
E. ERRLIST—Suppresses listing of source code lines which contain no error. Default: list all lines.
F. COPYLIST—Causes listing of all lines from library modules COPYied into a program. Default: no listing of COPYied lines.
G. RESEQ—Causes generation of Cobol line numbers in columns 1 through 6 of listing, in increments of 10. Default: no line numbers or line numbers in sourcefile are treated as comments.
H. NOREFF—Suppresses generation of hexadecimal memory references on the right side of listing. Default: relative offset references into data or code partitions are shown on listing.
I. DATE—Allows programmer to include a date in listing page headings and in DATE-COMPILED paragraph of program.
J. FLAG (level)—Causes generation of flags on lines in program which exceed specified GSA Cobol implementation level.

II. Directive affecting console output

A. NOECHO—Supresses display of error messages on console device. Default: lines containing errors are listed along with error codes and messages

III. Directives affecting P-code file output

A. NOINT—Supresses output of P-code file.
B. INT (filename)—Directs P-code to named file. Default: source-file-name.INT
C. ANIM—Causes generation of data needed to use the optional ANIMATOR debugging program

The normal output of the compiler includes a listing file, an INT-code file, and messages to the user's console device.

CIS Cobol continued . . .

hard disks or disk emulators will handle the paging much more efficiently than a floppy disk system.

In summary, CIS Cobol, FORMS-2, and ANIMATOR should be considered by anyone looking for a well-designed microcomputer Cobol implementation. In the hands of a skilled user, CIS Cobol should be an efficient medium for the development and maintenance of very sophisticated applications programs.

Micro Focus now offers three types of Cobol compilers. CIS Cobol is an ANSI-74 Standard Cobol compiler certified by the GFA at the low/intermediate level (typical level of implementation for most micro and minicomputer Cobol). In addition, Micro Focus offers compact Level II Cobol and high-performance Level II Cobol. Both are ANSI-74 Standard and GFA certified at the high level. This is the highest level of implementation within the ANSI-74 standard and is comparable to most mainframe Cobol compilers. Compact Level II Cobol requires about half the memory of high-performance Level II Cobol, and the high-performance Level II Cobol compiler is about five to 10 times faster. All three compilers are fully compatible with the Micro Focus Visual Programming Tools, ANIMATOR and FORMS-2.

For more information on CIS Cobol, contact:

Micro Focus
1860 Embarcadero St.
Palo Alto, CA 94303
(415) 856-4161

CIRCLE 301 ON READER SERVICE CARD

A MICROSYSTEMS TUTORIAL/REVIEW

Prolog:

The new AI language being used on 5th generation Japanese computers can now be run under CP/M

by William G. Wong

Wouldn't it be nice if we could tell a computer some facts, ask it some questions, then get the answers without having to tell it how to do so? Well, Prolog may not let you do all of this in every case, but it is a step in the right direction.

Prolog is a programming language that differs tremendously from conventional languages such as Pascal and Basic because Prolog is a descriptive language. You tell Prolog *what* has to be done, not *how* it is to be done. This article presents some of the features of Prolog, as well as a review of a CP/M-based Prolog—called **micro-Prolog**—available from Logic Programming Associates, Ltd.

Prolog stands for PROgramming in LOGic. It is a logical programming language that has roots in mathematics and logic. Like Lisp, it was first used for artificial intelligence research but has since found wider acceptance. In fact, it is being used as the programming language for the Japanese fifth-generation computer project as well as numerous domestic and European research projects. This article presents some of the ideas behind Prolog, as well as the facilities and performance provided by micro-Prolog.

Introduction

Prolog is based on unification, Horn clauses, backtracking and first-order predicate logic. If you are ready to quit reading, don't. These are some of the terms that researchers in artificial intelligence use to describe Prolog, but it is actually less complicated than it sounds. Prolog is easier to learn and use than more conventional languages. In fact, people who know other programming languages may find it harder to learn Prolog than nonprogrammers do, because Prolog works so differently.

First, the basics. Prolog has a database that is sometimes called a dictionary. You can enter information into this database and then ask questions about it. Prolog will try to answer them using the information in the database. This information can include simple facts such as:

```
Mark-Twain wrote Tom-Sawyer
Ernest-Hemingway wrote For Whom-The-Bell-Tolls
Arthur-Miller wrote Death-of-a-Salesman
Charles-Dickens wrote Oliver-Twist
Charles-Dickens wrote Great-Expectations
William-Shakespeare wrote Macbeth
William-Shakespeare wrote Romeo-and-Juliet

Mark-Twain is American
Ernest-Hemingway is American
Charles-Dickens is English
William-Shakespeare is English

Death-of-a-Salesman is-a novel
For-Whom-The-Bell-Tolls is-a novel
Tom-Sawyer is-a novel
Macbeth is-a play
Romeo-and-Juliet is-a play
Oliver-Twist is-a novel
Oliver-Twist is-a musical
Great-Expectations is-a novel
```

Now we could ask questions about specific items in the database, such as "Did Charles Dickens write *Oliver Twist*?" which would be written in Prolog as:

```
Does (Charles-Dickens wrote Oliver-Twist)
```

In this case Prolog would answer YES, since this information is in the database. It would answer NO to the following question, since this information is not in the database:

```
Does (Charles-Dickens wrote Mother-Goose)
```

Well, so far we can answer simple questions, but what about a question of the type "Who wrote *Oliver Twist*?" Prolog does this using what is called unification in conjunction with *uninstantiated variables*. Unification is a form of pattern matching. Uninstantiated variables are variables that initially have no value but are instantiated (assigned a value) through unification with a constant. For example: (1 x) unifies with (1 2) where x is the variable whose instantiated value is 2, and 1 and 2 are constants. The parentheses delimit a list. But, back to the question. Try the following Prolog statement:

```
Which (x x wrote Oliver-Twist)
```

In this case, Prolog answers "Answer is Charles-Dickens." It then continues searching the database and prints "No (more) answers" because only one person wrote the novel as indicated in the database. The "Which" predicate prints its first parameter, in this case "x," if it can solve the remaining portion, which it did. Instead we can try asking the following question:

```
Which (x x is-a novel)
```

In this case we get the following results:

```
Answer is Death-of-a-Salesman
Answer is For-Whom-The-Bell-Tolls
Answer is Tom-Sawyer
Answer is Oliver-Twist
Answer is Great-Expectations
No (more) answers
```

We can now query the database and get all sorts of answers, but Prolog is actually much more powerful. For example, let's ask Prolog to find the English authors. The following Prolog question would do this:

```
Which (x x wrote y and x is English)
```

Now the answers may seem a bit redundant.

```
Answer is Charles-Dickens
Answer is Charles-Dickens
Answer is William-Shakespeare
Answer is William-Shakespeare
No (more) answers
```

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

This is due to the nature of the question: we did not specify that duplicates be deleted. Since both authors wrote two items they showed up in the answer list twice. It just goes to show that you need to be explicit with this version of Prolog. Obviously, the multiplicity of variables within a question leads to some very powerful queries. Nevertheless, it is quite tiresome entering the entire question many times. It would be nice if this information could be placed into the database, and, of course, it can. Assume that the following statement is in the database:

```
English novelist x if x wrote y and y is a
  novel and x is English
```

Now this seems to be a reasonable statement. It essentially says that "x" is an English novelist if "x" wrote "y" and "y" is a novel (making "x" a novelist), and "x" is also English. Note that the "x" in this statement is not the same "x" that is often used in a query. Like normal procedure in other languages, this statement represents a pattern and the variables represent relations within the statement. Each time the statement is used, a copy of the variables is used where the new copy is unique. We then ask the following question and get the subsequent answers:

```
Which (x English-novelist x)
Answer is Charles-Dickens
Answer is Charles-Dickens
No (more) answers
```

One thing which should be mentioned about Prolog is its versatility. Database queries can be used to verify or retrieve information. In both cases, the same definitions and mechanisms are used. Prolog can also use the same definitions to construct new information as well. The following section on lists, strings and numbers shows how the SUM predicate can be used in all three modes. Any other language would require three definitions.

A quick note about the backtracking operation of Prolog is also in order. Prolog searches the database in a linear fashion. When a match is found, the information is returned; however, a marker is placed at that point. If, for some reason, the first item found fails to satisfy a subsequent condition, then Prolog backtracks to the last marker and tries to find the next matching item. This is how Prolog finds multiple answers to a question. The backtracking is done automatically and is controllable by the user.

There is no restriction on the type of clause that can be entered into the database. A clause may contain any number of variables and conditions. The main thing to note at this point is that so far we have told Prolog only what was a fact and asked questions. We did not tell it *how* to search the database or how the information is to be stored. Any other language would require pages of program code just to tell the computer how to do the search and store functions.

A Prolog program is simply a set of statements that describe the contents of database, as well as relations between elements within that database. Check out the references listed at the end of this article if you want more information on Prolog.

micro-Prolog user interfaces

The user interface to micro-Prolog consists of a prompt, followed by an input which is then evaluated. This loop is repeated until the program is terminated. Micro-Prolog is actually supplied with a number of user interface programs.

The SIMPLE.LOG interface was described in the introduction. It is the most English-like of the three and definitely the one to use when first starting out. Error trapping and tracing is sophisticated and includes helpful information when things go wrong. SIMPLE.LOG is a text file containing the micro-Prolog program for the interface, which can be enhanced by the user. It is also a good example of what micro-Prolog programs look like. This interface requires about 12K.

The other two interfaces are the MICRO.LOG and the built-in interface. The MICRO.LOG interface is similar to SIMPLE.LOG, but it contains fewer features. The gain is in the amount of workspace. MICRO.LOG occupies only about half the space required by SIMPLE.LOG. The built-in interface provides the least amount of support and is intended for application development or experienced users. It is terse and likes error numbers instead of messages, but provides the largest workspace of the three.

The following examples were used earlier and follow the SIMPLE.LOG syntax. Note that this format closely matches the English statement of the same fact.

```
William-Shakespeare wrote Macbeth
English-novelist x if x wrote y and y is a
  novel and x is English
```

Now the corresponding micro-Prolog basic syntax for the same statements is:

```
((wrote William-Shakespeare Macbeth))
((English-novelist | x) (wrote x y) (is a
  y novel) (is x English)
```

Note the similarity between the basic micro-Prolog syntax and Lisp. The translation between this basic syntax and the SIMPLE.LOG syntax is performed by the SIMPLE.LOG program, which is essentially an infix-to-prefix notation conversion. Using the list as the basic data structure greatly simplifies manipulation of definitions by programs. The basic list notation will be used for subsequent examples.

Lists, strings, and numbers

micro-Prolog has one built-in list operator, |. It must appear between parentheses and is the same as the dot in Lisp list dot notation. (1 | 2) represents a list whose first element is 1 with the remaining portion of the list being 2. This one operator can be used as either a constructor or extractor function depending upon the context at evaluation time.

Although | is the only built-in list operator, it is very easy to extend the language by adding new definitions. For example, the following is the definition for a list append routine.

```
((append () x x))
((append (x0 | x1) y (x0 | z)) (append
  x1 y z))
```

The first statement says that appending a list "x" to the empty list results in list "x." The second statement says that a list "y" can be appended to a nonempty list by taking the first element and making it the first element of the result and getting the rest of the result by appending "y" to the rest of the first list.

Strings are micro-Prolog constants which may be up to 60 characters. A string can be converted to a list of charac-

Prolog differs tremendously from conventional languages such as Pascal and Basic because it is descriptive: You tell Prolog what has to be done—not how.

ters and back; however, these rudimentary operations, along with the maximum size of a string, severely limit the text processing capabilities of the system. Even so, this is usually sufficient for many of the intended Prolog applications.

Sixteen-bit integers and eight-digit decimal floating-point numbers are supported. The decimal floating point is nice, since there is no roundoff error when entering a number and printing it. All the clauses used to define arithmetic operations are multifunctional and work with both integers and floating-point numbers. The following examples show off the multifunctional nature of the built-in micro-Prolog SUM clause.

```
(SUM 1 2 3)
(SUM 2 2 8)
(SUM 1 2 x)
(SUM 1 y 3)
```

The first two clauses are used to check values. The first clause would succeed since 1 plus 2 is 3, whereas the second would fail since 2 plus 2 is not 8. The third example is used to generate a result. In this case, x is unified with the constant 3. The third example shows how to subtract two numbers. In this case, y is equal to 3 minus 1.

Only the basic arithmetic functions are provided. There are no predefinedd trigonometric or scientific functions provided, although they can be defined using the basic functions.

Input/Output support

micro-Prolog's I/O support is good and includes random as well as sequential file access. Devices such as the console and printer are also supported as files. File directory manipulation operations are possible and very flexible. For example:

```
(DIR (ambiguous file name) (file name list))
```

is the format of the built-in directory clause. The (ambiguous filename) is a standard CP/M filename and the (filename list) is the list of all filenames that match the pattern. The following definition could be used to print a directory listing:

```
((PRINT-DIR x) (DIR x y) (PP y))
```

where PP prints its parameters and starts a new line. Evaluating (PRINT-DIR *.*) would print all the files on the current disk.

The random file access provided is sufficient but somewhat primitive. Record position is provided using a list whose head is the CP/M record and whose tail is the byte index. For example, (1 | 32) corresponds to record 1, byte 32. Logical records can span CP/M records which are 128 bytes. The major drawback of the system is that a file can be open either for input or for output, but not for both at the same time; also, opening a file for output creates a new file, so random database update is not possible.

Formatted records between 1 and 254 bytes in length are also supported. Formatted file read and write operations are included in the basic system. It does not support sophisticated floating-point formatting.

Editor

A resident list editor is available and is written in micro-Prolog. It can be modified and enhanced, and it includes some pattern matching as well as normal structure manipulation operations. Since micro-Prolog definitions are lists, the editor can be used to create and modify definitions as well.

Support for saving/restoring definitions from text files is also provided, so conventional word processors can be used to edit a micro-Prolog database. Large databases will normally be built using a word processor, whereas the resident editor will be used when developing a system or when experimenting with micro-Prolog.

Error handling

micro-Prolog has excellent error handling capabilities which can be tailored by the user. Errors can occur, however, when you try to evaluate a condition. In that event, the clause evaluation is changed to the following form:

```
('?ERROR?'' (error number) (error condition)
```

Error-handling routines are added by including the appropriate definition in the dictionary. For example, the following definition would print the (error number), the (error clause) and succeed.

```
((''?ERROR?'' x y) (PP Error x in y)
```

The default option is to simply print the (error number) and abort. The various micro-Prolog front end programs usually print a text description of the error and allow the user to select the way to continue execution. This mode of operation allows a beginner to have a friendly interface to the system, while a developer may dispose of this support to gain the additional space taken up by the friendly error interface.

Garbage collection

The micro-Prolog garbage collector runs automatically, as in Lisp, when all known free space is exhausted. It runs very fast and its operation is usually not noticeable. Elapsed time is usually under one-half second. It can be initiated manually by finding the current amount of free space using the (SPACE x) condition.

One feature of this garbage collector is that the dictionary is also checked so that unreferenced names are removed. Some systems do not perform this operation, thereby wasting the space occupied by unused names.

Optimizations

micro-Prolog contains a number of features that provide optimizations in space, speed, and modularity. The module support in micro-Prolog allows a module to contain

It should be mentioned that Prolog is versatile. Database queries can be employed to verify or retrieve information by using the same definitions and mechanisms. Prolog can also use those same definitions to construct new information.

any micro-Prolog definitions. The *exported* and *imported* definitions and names are explicitly listed as part of a module. All other internal definitions are inaccessible to the outside world. Modules also speed up execution, since internal module definitions are not searched when the main dictionary is checked.

In terms of space, micro-Prolog provides a form of virtual memory support. In this case, a file is used to contain Prolog definitions. A memory-resident definition is used to access the file. A special module is provided to create and modify definition files. The files are randomly accessed for speed, and one file may contain any number of different definitions. The file contents are ASCII text so they may be printed; however, the information is position-dependent, so using a text editor on such a file will not work.

Tail recursion and *success popping* are two features which provide space and speed optimizations. In short, the micro-Prolog interpreter is smart enough to know when certain pieces of information kept on the stack will never be used again and throws them away, thus saving space. This information need not be checked later, so programs run faster. An additional benefit is that deterministic definitions can run, even if the data or operation is large or possibly infinite. Take the following micro-Prolog definition, for example:

```
((forever) ((PP 1)) (forever))
```

Calling "forever" would print the number 1 forever. Without tail recursion, this process would repeat until the stack was used up in keeping track of the subsequent calls to forever.

Documentation

micro-Prolog comes with two books designed for two different types of users. The Primer is designed for first-time users. It describes the SIMPLE.LOG interface to micro-Prolog. The book is very complete, and is full of examples and explanations. Each section includes problems and answers. It can be used as a learning tool for Prolog in general, or specifically for micro-Prolog. The table of contents is complete and makes up for the lack of an index. The bibilography directs you to more information on Prolog.

The second book is the micro-Prolog 3.0 Programmer's Reference Manual. This, too, lacks an index, but the table of contents is sufficient. This book completely describes the user interface and includes numerous examples. Every programmer's reference manual ought to be as comprehensive. All supplied Prolog functions and files are described, including the interfacing of assembly language programs. Even the internal operation of the system is presented for those interested in the architecture of a Prolog machine.

Both books are in bound paper-back form, which is adequate for occasional use; however, I recommend cutting the binding and placing the books into a ring binder if the system is to be used to any great extent. If you write a great program and add the same type of documentation you should do the same with the packaging and support.

Performance

The micro-Prolog interpreter is very powerful for its small size (15K). It requires a Z80, 8088, MS-DOS, or CP/M-86 processor, whose more powerful instruction set saves space and adds speed. This leaves quite a bit of space for programs and data, usually 32K on a 64K CP/M system. This is sufficient for some very interesting and sophisticated programs.

Performance of micro-Prolog is quoted to be 240 resolutions per second on a 4 MHz Z80 under CP/M V2.2. This number is quite useless to most people, since there is nothing to compare it against; it is, however, very respectable. micro-Prolog compares favorably with Lisp and Basic interpreter implementations under CP/M for most operations, including numeric processing and file input and output operations. List support is on a par with Lisp, which is far superior to Basic or Pascal.

The area where micro-Prolog shines is in pattern matching and database search. It is much faster and easier to use than any other language implementation I have seen on a Z80. Using the virtual memory support is acceptable on a hard disk or memory disk.

Summary

Prolog is a very flexible and powerful language. This brief presentation can only give a small taste of what Prolog can do. Using a logical programming language like Prolog can help you create more logical programs that are easier to understand and to debug. What may also surprise you is that Prolog compilers on larger machines generate code that is as efficient as C or Lisp, so that programming logically does not necessarily imply inefficiency.

At $275 per copy, micro-Prolog is a bit expensive if you just want to try out Prolog, but it is an excellent product and, if you can afford it, it is well worth the price. The facilities provided by micro-Prolog are very good and the documentation is well written and complete. I would highly recommend its use in all areas except those which require compiled machine code for reasons of speed or security. Hopefully, Prolog compilers will become available on micros in the near future.

Prolog is one of the first logic programming languages. It is widely used in Europe, and now in Japan, especially for advanced computer research. If you do not know Prolog now, micro-Prolog offers a good way to learn it. If you know Prolog, you will find this implementation very complete.

References

Clark, K.L., Ennals, J.R., and McCabe, F.G., *A micro-PROLOG Primer*, Logic Programming Associates, Ltd., London, 1982.

McCabe, F.G., Clark, K.L., *micro-PROLOG 3.0: Programmer's Reference Manual*, Logic Programming Associates, Ltd., London, 1982.

Clocksin, W., Mellish, C., *Programming in Prolog*, Springer-Verlag, New York, 1981.

Kowalski, R.A., "Algorithm = Logic + Control,"

The area where micro-Prolog shines is in pattern matching and database search. It is much faster than any other language implementation I have seen on a Z80.

CACM, August, 1979.

Kowalski, R.A., *Logic for Problem Solving,* North Holland, New York, 1979.

Pereira, L., Pereira, F., Warren, D., *User's Guide to DEC System 10 Prolog,* Dept. of AI, University of Edinburgh, U.K., 1978.

micro-Prolog Listing of Sample Database

```
((English-novelist | X)
    (wrote X Y)
    (is-a Y novel)
    (is-X English))
((is-a Tom-Sawyer novel))
((is-a For Whom-the-Bell-Tolls novel))
((is-a Death-of-a-Salesman novel))
((is-a Macbeth play))
((is-a Romeo-and-Juliet play))
((is-a Oliver-Twist novel)
((is-a Great-Expectations novel))
((is Mark-Twain American))
((is Ernest-Hemmingway American))
((is Arthur-Miller American))
((is Charles-Dickens English))
((is William-Shakespeare English))
((wrote Mark-Twain Tom-Sawyer))
((wrote Ernest-Hemmingway For-Whom-The-Bell-Tolls))
((wrote Arthur-Miller Death-of-a-Salesman))
((wrote Charles-Dickens Oliver-Twist))
((wrote Charles-Dickens Great-Expectations))
((wrote William-Shakespeare Macbeth))
((wrote William-Shakespeare Romeo-and-Juliet))
((dict English-novelist))
((dict dict))
((dict is-a))
((dict is))
((dict wrote))
```

Vendor Information

Logic Programming Associates, Ltd.
Software Consultants
10 Burntwood Close
London, SW18 3JU
England
CIRCLE 306 ON READER SERVICE CARD

Prices:

micro-Prolog 3.0 requires Z80, 8088, MS-DOS, or CP/M-86 includes diskette, primer, and reference manual	$275
micro-Prolog documentation	$35
A micro-Prolog Primer	$15
Software without documentation	$250

All prices include air mail to U.S. Disketted formats include 8 inch single sided, single density, and 5 1/4 inch for IBM PC (for Z80 card), North Star, Zenith Z89, Osborne, and Apple II (for Z80 card). The documentation is now distributed in a ring binder.

Two Users on a CP/M System

by Richard Benser

Wouldn't it be nice to have a "communications package" or a "terminal program" to use with your CP/M system? If you had one, and a modem connected to your computer, you could use your computer in new ways. You could talk to other computer users with compatible hardware, access time-sharing systems, and exchange programs with any of these other sources. The necessary hardware is rapidly becoming very affordable. I saw a 0-300 baud direct connect modem listed for only $79. It was even a brand new device, not one removed from other equipment in "operating condition." If you have a spare RS-232 port on your computer, or even one with a printer attached to it, you can use this modem.

To use one port for two devices is a lot easier with an RS-232 switch, but I have carefully changed connections enough times to know that a switch is really a luxury. However, even this luxury is becoming affordable. I saw a switch advertised for only $39.95 assembled, or $34.95 as a kit. I didn't actually buy one of these switches, but if I had decided to need one I probably would have. They look like complete mechanical switches that would be very functional units in my house. They might not be suited for rough commercial use, but they look fine for my not-too-harsh, not-too-often requirements. They come without a case and would require a cover, which would be a project that I would consider within my capabilities as a hobbyist.

While I was considering these options for my own use, a friend of mine offered me a deal. He was a real estate agent, and wanted me to write some programs that would allow him to maintain a computerized database of lease properties. In return he would buy me an automatic answer modem and an I/O board to attach it to, and pay me a monthly fee to access my computer from his office via a terminal. Sounds great, but much hard and expensive experience has taught me that hardware without software doesn't do very much for me. While I was considering his offer, I decided that I could use a modified BIOS for communicating through the modem. I would simply treat the modem as the console. This scheme was the beginning of COMmunicating BIOS (COMBIOS).

What is COMBIOS?

BIOS is that part of CP/M written for your specific hardware configuration. It accesses and controls your peripherals (disk drives, console, printer. . .). COMBIOS is simply my name for a BIOS that uses two consoles in parallel. Output is allowed to both consoles simultaneously. Input can be accepted from either console. A standard BIOS usually contains three or four sections of code that deal with the console. These are: an input routine (gets a character from the console), an output routine (displays a character on the console) and a status routine (answers the question "has any key been struck on the console keyboard since the last character was read?") and possibly some setup code (set baud rate and the like). The first three routines are part of the requirements of CP/M for any BIOS. The fourth may be required by your particular hardware, or it may not be necessary. The changes described here don't directly affect the fourth area, so it will be ignored. However, if you are using the same port for a printer and a second console, you might have to change the fourth area as well.

COMBIOS didn't start to become a reality until I had done a lot of preliminary work. First, I wrote the data manipulation routines required by the real estate system and a home-grown database system. Then I set up a more or less standard BIOS using my new modem and I/O board (a CompuPro Interfacer 3 with five ports) as the console. This was enough to get my friend started in the wonderful world of automated data processing. It was only then that I had the time to play with my idea.

A COMBIOS cookbook

I am going to describe how my system developed into its current condition and use that as a logical approach. It worked for me. The first routine that I modified on my system was my output routine, CONOT. The modification was easy. While my friend was using the system remotely through the modem, I wanted to be able to see what he was doing, so I could explain any mistakes he made or help with any problems. The modification was just to write any output characters to my console and then to the port with the modem attached. It was a simple change to add the check for a ready status on my console, wait for a ready and then write the character before doing the same thing to the port with the modem. Listing 1 shows the original code of my BIOS and the modified code. I could now play "Big Brother" to my remote user, and that became phase 1 of my COMBIOS.

The second routine that I modified was my console input routine. I really wanted to be able to show my friend how to run some of his reports, so I needed input in my role as "Big Brother." My original input routine, CONIN, checked for a byte being ready from the modem and read it if one was ready, or checked again if no character was ready. My first modification to CONIN simply added a check of my console if there was no character ready from the modem. If my console check failed, the routine looped back to check the modem again, but if there was a character ready, the character was read and CONIN returned with the console input. This modification is demonstrated in Listing 2. I still had a problem, though. Some of my programs used code to check the status of the console before calling CONIN. So I had to modify my console status routine, CONST. For my console to be fully functional as a parallel console, it too had to be checked by CONST. This also was a simple change. It was only necessary to check the status of my console as well as the remote console before returning a no-character-ready status. Listing 3 shows these changes to CONST.

I now was a fully functioning "Big Brother." I could monitor my friend by just turning on my console, and if I

Richard Benser, JC Systems Inc., 93 Glenwood Ave., Leonia, NJ 07605

EE/EPROM PROGRAMMERS & UV ERASERS

AFFORDABLE ★ RELIABLE ★ AVAILABLE

UV ERASERS
QUV-T8/1
$49.95
*HOBBY

QUV-T8/2N
$68.95
*INDUSTRIAL

QUV-T8/2T
$97.50
*WITH TIMER
& SAFETY SWITCH

GANGPRO-8™
$1,295.00
(GANG
PROGRAMMER)

RS-232 serial, STAND ALONE, INTELLIGENT

*EASY DUPLICATION *USER FRIENDLY *128K BUFFER

SUPPORTS: MOST 8K, 16K, 32K, 64K, 128K, 256K EPROMS
PROMPRO-8: KEY PAD OPTION, EPROM SIMULATION MODE
Microcomputer Chips: 8748 (H), 8749H, 8750, 8751, 8741, 8742, 8755A
SOFTWARE DRIVERS: MDS ISIS, TEKTRONICS 8002, IBM PC, ATARI, APPLE II, CPM, FLEX, TRS-80
DIRECT HOOK UP TO ANY DUMB TERMINAL OR COMPUTER.

DISTRIBUTOR INQUIRY WELCOME.

LOGICAL DEVICES INC.
1321E N.W. 65 Place, Ft. Lauderdale, FL 33309
Phone Orders (305) 974-0967 TWX 510-955-9496

CIRCLE 191 ON READER SERVICE CARD

the best
VALUE anywhere

COMMUNICATIONS SOFTWARE PACKAGE
COMMX-PAC™

- COMMX Smart Terminal and File Transfer Auto Dial and Logon Files for Easy Linkup Simple Operation Via Local or Command Menu COMMunications eXchange protocol for Timeshare, RCPM (XMODEM) and other COMMX Systems (Error Free Reliability)
- Over 60 configurations available (8 + 16 bit)
- FORTRAN version available for Mainframes
- "Midnite" Electronic Mail Subsystem
- Infoworld Software Report Card Rated "A plus"
- CPU License $150 - Detailed Manual $25

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

CIRCLE 119 ON READER SERVICE CARD

NOW!
CP/M-80
SOFTWARE USERS...

KNOW YOUR SOFTWARE PROGRAM BEFORE YOU BUY...
CHECK OUT - THE ENCYCLOPEDIA OF CP/M-80 SOFTWARE©...

A comprehensive, indepth, single source of information about the "vast amount of software" for your CP/M-80 8-bit microcomputer. Always current, updated, containing all of the CP/M-80 software available plus the latest price reductions. You can save more than the cost of an annual subscription on your very next software purchase by knowing what is available, what it will do and what it will cost. Answers most questions on every program.

CP/M is a trademark of Digital Research.
© Copyrighted by ENCYCLOPEDIA PUBLISHING CORP.

Rush your order now for the Spring Edition.
One year subscription (four quarterly issues) $129.00*
*Foreign orders $169.00 payable in U.S. Funds.
Visa, Mastercard, American Express, Carte Blanche, Diners Club. 24 hour toll free telephone orders.
1-800-522-1500 Ext. 865
Or mail subscription orders prepaid to:
ENCYCLOPEDIA PUBLISHING CORPORATION
950 Hampshire Road, #108
Westlake Village, CA 91361
(805) 497-8002

CIRCLE 122 ON READER SERVICE CARD

WOW!

P-R-O-P-O-R-T-I-O-N-A-L Spacing on WordStar

You are reading text **printed by WordStar** in proportional spacing, providing a professional, **easy to read,** typeset appearance.

Complete details for printing in proportional spacing **directly from WordStar,** setting **two or more fully justified columns on a page,** and underlining spaces between words, are provided. The techniques will work on **all** versions of WordStar, and will drive Diablo, **Xerox, Qume, NEC,** and other daisywheel printers.

Above text printed on a daisywheel printer direct from WordStar.
Now you can have the professional appearance of typeset text. Using PS is as easy as turning on bold or underline and is done right in your document, then printed by WordStar automatically!

PS ON WordStar-$20 postpaid

Please send me_______ Copies. Enclosed is my check (or Visa/MC# and exp. date) for $__________, made out to:

WRITING CONSULTANTS
Suite 119 / 11 Creek Bend Drive
Fairport, New York 14450
Call Toll Free 1-800-227-3800 Ext. 7018
Dealer inquires invited.

CIRCLE 52 ON READER SERVICE CARD

wanted, I could demonstrate some part of my system by typing on my console while he just watched. I should have been satisfied. However, I wanted still more. I wanted to be able to use the system myself while my friend wasn't using it. When he went out for lunch, I wanted to be able to print his bill, or even modify one of my real estate programs while still leaving the system accessible to him.

I could do that with my current modifications, but I wasn't happy running my console at 300 baud just because when my friend was connected, that was his transmission rate. My solution was a switch in BIOS. The switch would control whether or not a copy of console output would go to my remote user. To implement this function, I set aside a byte (called VIDBIT) and chose the least significant bit for this switch. When this bit is on (a one) BIOS sends console output to both consoles. When this bit is off, BIOS sends output only to my local console. That was easy enough to set up. I simply added the check in CONOT before attempting to display on the remote console. Listing 4 shows this addition. Next, I needed a way to change that bit. I went back to CONIN. When CONIN sees a control R typed on my local console, it reverses that bit. Now I can turn off my friend's terminal output, and turn it back on again from my keyboard. When the remote output is turned off, my console doesn't have to run at the slower speed, and I'm much happier running my system.

Listing 5 shows CONIN modified to check for the control R and flip the proper bit when it detects one. I chose the control R because it would stand for "Remote" and would be easy to remember, but any other character that makes sense to you, will do. Notice, however, that the only console with the power to flip that bit is the local console. My friend cannot turn himself on and off. He wouldn't understand if just the right set of noises on his telephone line turned his output off. After using this code for awhile, I added one more change. Instead of allowing the control R to go on to CP/M, I changed it and all subsequent COMBIOS control characters to a delete (rubout). In that way at the beginning of a line they are not used at all by the system, and now, before I use one elsewhere, I type the last character twice so CP/M will delete one for me. I had to allow some characters to go back, in case CONST had already been used to verify that there was a character ready from a console, and CONIN was called to get it.

At this point COMBIOS was functional. It wasn't all I could ask for, but it worked. The biggest hassle with it was that CP/M saw everything typed either remotely or locally as a command. My friend and I could type messages to each other, but CP/M would always answer first to say that he couldn't run that program, unless a message started with a word that was also a program name. In that case CP/M would start to run the program. It was, to say the least, inconvenient and unprofessional. I clearly needed a second switch.

For this function, I decided to use the bit next to my remote shutoff bit. To change the bit, I decided to use the character control Q, for "Quiet." When this bit was off, CP/M would see both consoles as completely quiet. This was the messiest feature to add. It required modifications to CONIN and CONST. CONIN had to be changed so that when the "Quiet" bit was off, CONIN didn't return a character. What CONIN had to do was to display it on the console(s), and then return to read another character until CONIN received a control Q from the local console to turn on the "Quiet" mode bit. CONIN also had to add a line feed after each carriage return, or else the consoles would continually overprint the same line in "Quiet" mode. Listing 6 shows CONIN with this code added.

The last problem was now with CONST. If any program called CONIN while COMBIOS was in the "Quiet" mode, there was no character returned until the "Quiet" mode bit was turned on. CONST had to show this state to any program that called it. The modification that I settled on goes something like this. If the "Quiet" mode bit is on, CONST functions normally and may either return a ready or a not-ready status depending on the consoles' states. But if the "Quiet" mode bit is off, CONST must return a not-ready status.

The way I achieved this was to add code to CONST to be executed when there is a character ready on either console. If there is none, CONST functions normally. Also, if the "Quiet" mode bit is on, the flow is not changed. But if the "Quiet" mode bit is off and there is a character ready on a console, CONST first turns the "Quiet" bit on, so that CONIN will function normally. Then CONST calls CONIN. Since there is a character ready, CONIN gets the character and returns immediately. After receiving the character, CONST flips the "Quiet" mode bit again. This returns it to its original state of off, unless CONIN has already done that. In that case, CONST turns the bit on, which is exactly what a control Q is supposed to do in these circumstances. After flipping the bit, CONST calls CONOT to display the character. Then CONST loops to its beginning. In that way CONST goes on until there is no character ready, then it returns a not-ready status. This allows the calling program to go on as though there was no console activity. See Listing 7 for all these modifications in CONST.

One more adjustment remained to be made. No timesharing system that I knew about could deal with a terminal that echoed all data sent to it back to the system. There were some systems that would echo to the terminal (full duplex), but none that would accept an echo from the terminal. CP/M, however, echoes anything typed on the console keyboard back to the console display. This made timesharing unavailable to COMBIOS. The fix for this problem involved more changes to CONIN and CONOT. CONIN had to save the last character typed by the remote console in a location I labelled DUPLEX. Then CONOT had to check something else before displaying on the remote console. The first check is for the function controlled by the control R, which can block all remote output. The next check is of the third bit in VIDBIT. This bit, toggled by control D (for duplex), controls the echo to the remote console. If the bit is on, CONOT goes ahead and displays the character. However, if the bit is off, CONOT first checks the location in memory called DUPLEX. If the character about to be displayed is the same as the one in DUPLEX, CONOT does not display the character. Instead it returns to its caller after wiping out the location

I get a big kick out of my single-user OS running a program while I am using it to converse with a remote user.

DUPLEX. This is to prevent other attempts to display the character from being caught in the same trap. For example, imagine my friend trying to type the word "see" and only seeing "se" on his terminal, no matter how many e's he typed. See Listings 8 and 9 for the code in CONIN and CONOT implementing the half-duplex function.

There are two features which I plan to add to COMBIOS. One is a way to suppress the automatic line feed in "Quiet" mode. This will avoid double spacing while using time sharing systems. The other is a way to send a break down the line to interrupt a listing or program. They will be added before this is printed. And should make COMBIOS an easy, effective package to use.

Conclusions

Now, COMBIOS is a fairly powerful tool. I can easily converse with my remote user. All I have to do is put my system into "Quiet" mode and type. If I want to run a quick directory listing without slowing down for my remote user, I can use the control R function to not send output to the remote teminal, and not slow my local console down to his speed. Also I have modified at least one program, a memory test program, to run as a background task during a conversation. The program calls CONST every few instructions. In that way it can run as a background task and still let our conversation go on. I get a big kick out of my single-user operating system running a program while I am using it to carry on a conversation—even though I must admit that my memory test program ran a lot faster without all those calls to CONST in it. That, however, is the nature of a background task: to be run only when the system has nothing better to do, or at a very low priority and very slowly.

I have also used COMBIOS to access time-sharing systems and even to transfer files from one system to another. By setting my system in the "Quiet" mode, I can issue dial commands to my modem (a DC Hayes Smartmodem) by typing them on the local console with the remote switch on. When my modem makes the connection, I simply use my system as a dumb terminal and type whatever I need for a sign-on sequence. Then when I am ready to transfer a file, I can turn off the remote interface, remove myself from "Quiet" mode, and get CP/M ready to transfer or receive a file. Then all I have to do is turn the remote interface back on, and a single carriage return starts the exchange going. When it is complete, I can use the remote switch and the "Quiet" mode to do other work on either system, or terminate either my edit session or the time-sharing session.

There is one limitation to COMBIOS that I should mention. It will not transfer non-ASCII files as it is currently written. If it ever becomes necessary for me to get or give a non-ASCII file, I will have to write a program to do it. The program will have to select some means of exchanging the data, or match a similar program on someone else's system. I don't see that as much of a problem, and I expect to be able to do it fairly easily if the need should arise. Also, after receiving an ASCII file from another system, I may have to do some editing to clean it up before I am satisfied. Usually simple things—such as having two line feeds at the end of each line—aren't quite right.

I had one fear when I started this project, but thankfully it proved unfounded. I was afraid that all this code would slow down my console output so much that I would never use the system. There is almost no difference in printing speed with all these enhancements to my BIOS compared to the original. I assume that the CPU is so much faster than my console that the extra code merely uses up some of the time the CPU would (before the modification) have spent waiting for the console to be ready for another character.

One problem I had to solve was that I needed more than the first two tracks of my single-density floppy disk for CP/M with COMBIOS. My solution was to use one of the last six sectors of the last track of my disk (track 76). The way CP/M allocates disk space, in eight-sector (1024 byte) chunks, precludes the use of the last six sectors by any CP/M file. The actual code for the solution is unimportant in the context of COMBIOS and may not even be necessary with newer double-density controllers. It also was not original work on my part. It has been described in at least one article in a computer magazine. If my filing system (a.k.a. that mess in the basement) were better, I would cite the magazine and date. But under the circumstances I can only say that I read it at some time, and was able to recall and use the technique without being able to credit the author and magazine by name.

Listing 1

```
;WRITE A CHARACTER TO CONSOLE DEV
;       ON INPUT CHARACTER IS IN C REG.

CONOT:  EQU     $
        IN      21H             ;GET STATUS
        ANI     1               ;LOOK AT BIT #0
        JZ      CONOT           ;NOT READY - TRY AGAIN
        MOV     A,C             ;MOVE CHAR. FROM C TO A
        OUT     20H             ;OUTPUT TO CONSOLE
        RET
```

A) original CONOT with only remote console

```
;WRITE A CHARACTER TO CONSOLE DEV
;       ON INPUT CHARACTER IS IN C REG.

CONOT:  EQU     $
        IN      11H             ;GET LOCAL CONSOLE STATUS
        ANI     4               ;CHECK FOR READY
        JZ      CONOT           ;NOT READY - TRY AGAIN
        MOV     A,C             ;MOVE CHAR. FROM C TO A
        OUT     10H             ;OUTPUT TO LOCAL CONSOLE

CONOTR: EQU     $
        IN      21H             ;GET REMOTE CONSOLE STATUS
        ANI     1               ;CHECK BIT #2
        JZ      CONOTR          ;NOT READY - LOOP
        MOV     A,C             ;MOVE CHAR FOR OUTPUT
        OUT     20H             ;OUTPUT IT
        RET                     ;RETURN
```

B) first modification - two consoles get all output

Listing 2

```
;READ A CHARACTER FROM CONSOLE
;       ON RETURN CHARACTER IS IN A REG.

CONIN:  EQU     $
        IN      21H             ;READ CONSOLE STATUS
        ANI     2               ;LOOK AT BIT #1
        JZ      CONIN           ;NOT READY - CHECK AGAIN
        IN      20H             ;READ DATA BYTE
        ANI     7FH             ;KILL POSSIBLE PARITY
        RET                     ;RETURN TO CALLER
```

A) original CONIN routine to accept input from remote console

```
;READ A CHARACTER FROM CONSOLE
;       ON RETURN CHARACTER IS IN A REG.

CONIN:  EQU     $
        IN      21H             ;GET REMOTE CONSOLE STATUS
        ANI     2               ;IS THERE A CHARACTER READY
        JZ      CONINL          ;IF NOT CHECK LOCAL CONSOLE
        IN      20H             ;ELSE GET CHARACTER
        ANI     7FH             ;DESTROY PARITY BIT
        RET                     ;RETURN WITH CHARACTER
CONINL: IN      11H             ;ELSE CHECK LOCAL CONSOLE
        ANI     2               ;IS THERE A CHARACTER READY
        JZ      CONIN           ;IF NONE LOOP AGAIN
        IN      10H             ;ELSE GET CHARACTER
        ANI     7FH             ;THROW AWAY PARITY BIT
        RET                     ;RETURN WITH CHARACTER
```

B) console input routine with provision for two consoles

Listing 3

```
;CHECK CONSOLE INPUT STATUS
;       ON RETURN IF NOT READY A = 0, IF READY A = FF

CONST:  EQU     $
        IN      21H             ;READ CONSOLE STATUS
        ANI     2               ;LOOK AT BIT #1
        RZ                      ;RETURN WITH NOT READY
        MVI     A,0FFH          ;READY SET A=FF
        RET                     ;RETURN TO CALLER
```

A) original single console CONST routine

```
;CHECK CONSOLE INPUT STATUS
;       ON RETURN IF NOT READY A = 0, IF READY A = FF

CONST:  EQU     $
        IN      21H             ;GET REMOTE CONSOLE STATUS
        ANI     2               ;LOOK AT BIT #1
        JZ      CONSTL          ;IF NOT READY CHECK LOCAL
        MVI     A,0FFH          ;ELSE SET READY ( A = FF )
        RET                     ;AND RETURN
CONSTL: IN      11H             ;GET LOCAL STATUS
        ANI     2               ;CHECK BIT #1
        RZ                      ;IF NOT READY RETURN WITH ZERO
        MVI     A,0FFH          ;ELSE SET READY
        RET                     ;AND RETURN
```

B) two console console status routine

Listing 4

```
;VIDBIT - CONSOLE CONTROL BYTE IS A BYTE WITH EIGHT ONE
;       BIT SWITCHES.  AT PRESENT THE BITS USED ARE

;       BIT 0   - REMOTE COPY MODE [ 0 MEANS NO REMOTE COPY ]

VIDBIT: DB      0FFH            ;START OFF WITH ALL ON
```

1984 Business Activity Planner

Imagine! A complete Appointment Book, Financial Record Keeping System, Travel Information Guide, Health and Fitness Aid and Directory... All in one book!

No matter how organized and successful you are today, our Planner will make you even better in the future... or your money back!

IT'S A GREAT GIFT IDEA, TOO!

IT'S AN APPOINTMENT BOOK!

SEE-A-WEEK CALENDAR

■ A full week... at a glance! Time planning becomes easy when your appointments, meetings, luncheon and dinner engagements and special activities are in front of you.

HALF-HOUR TIME SCHEDULING

■ Convenient scheduling... seven days a week.

THINGS TO DO TODAY

■ Keep track of important follow-ups, telephone calls and correspondence.

SEVEN SNAP-LOCK RINGS

■ The seven ring format insures that your pages won't pull out... unless you want them to!

BEAUTIFUL, SOFT, LEATHER-LIKE BINDER

■ Padded front and back, your 9½ x 11½ Planner is comfortable to carry. You can take it with you wherever you go. The sturdy, permanent bookmark always keeps you in the right week and the two inside pockets enable you to keep all your important papers in your **1984 Activity Planner.**

YOUR CHOICE OF IMPRINTS

■ Order the 1984 Business Activity Planner for everyone in your office ... and select the cover imprint for each person! Available with Business Activity Planner, Chairman of the Board Activity Planner, Executive Woman Activity Planner, or Executive Activity Planner.

IT'S MUCH MORE!

■ Each Section is Index-Tabbed for Instant Retrieval of Information.

FINANCIAL RECORD KEEPER

■ 12 Income and Expense Reports
■ 1984 Business Tax Calendar
■ Executive Compensation Checklist
■ Personal Investment Portfolio

HEALTH AND FITNESS AID

■ Fat - Cholesterol - Calorie Charts
■ Value of Various Exercises
■ Substitution Chart for Better Health
■ First Aid Tips

TRAVEL INFORMATION

■ Average Temperatures for Most Major Cities
■ Rank and Population of the Top 100 U.S. Cities
■ Traveling Distances
■ Toll-Free Telephone Numbers for Major Airlines, Auto Rentals, Hotels, Motels and much more!

MISCELLANEOUS

This section is filled with valuable information including Principal Holidays, French and California Wine Vintage Charts and more.

DIRECTORY

Places all emergency and important telephone numbers at your fingertips.

ZIFF-DAVIS ACTIVITY PLANNER P.O. Box 16-2354, Miami, FL 33116

Please send me the following 1984 Activity Planners at $38.95* each plus $5 per copy postage and handling. Outside U.S.A. add $10 per copy postage and handling.

Specify quantity for each cover imprint ordered:

____ Business Activity Planner(s)
____ Chairman of the Board Activity Planner(s)
____ Executive Woman Activity Planner(s)
____ Executive Activity Planner(s)

QUANTITY DISCOUNTS
(on total number of Planners)

6–20	10% off	($4 P&H each)
21–50	15% off	($3 P&H each)
51–or more	20% off	($2 P&H each)

☐ Payment enclosed. ☐ American Express ☐ Visa ☐ MasterCard ☐ Purchase Order enclosed.

Card No. ____________ Exp. Date ______

Signature ____________

Name ____________
(please print)

Company ____________

Address ____________

City/State/Zip ____________

*Florida residents add 5% sales tax.

SATISFACTION GUARANTEED or return within 10 days for a prompt refund!

MS01

FOR FASTER SERVICE CALL TOLL FREE 800-327-1969

(In Fla. and Canada call (305) 595-8744.)

```
;WRITE A CHARACTER TO CONSOLE DEV
;        ON INPUT CHARACTER IS IN C REG.

CONOT:  EQU     $
        IN      11H             ;GET LOCAL CONSOLE STATUS
        ANI     4               ;CHECK FOR READY
        JZ      CONOT           ;NOT READY - TRY AGAIN
        MOV     A,C             ;MOVE CHAR. FROM C TO A
        OUT     10H             ;OUTPUT TO LOCAL CONSOLE

        LDA     VIDBIT          ;GET SWITCH BIT
        ANI     1               ;CHECK REMOTE OUTPUT SWITCH
        RZ                      ;IF OFF DON'T DISPLAY REMOTELY
CONOTR: EQU     $
        IN      21H             ;GET REMOTE CONSOLE STATUS
        ANI     1               ;CHECK BIT #2
        JZ      CONOTR          ;NOT READY - LOOP
        MOV     A,C             ;MOVE CHAR FOR OUTPUT
        OUT     20H             ;OUTPUT IT
        RET                     ;RETURN
```

Listing showing use of remote output switch in console output code

Listing 5

```
;VIDBIT - CONSOLE CONTROL BYTE IS A BYTE WITH EIGHT ONE
;        BIT SWITCHES.  AT PRESENT THE BITS USED ARE

;        BIT 0    - REMOTE COPY MODE [ 0 MEANS NO REMOTE COPY ]

VIDBIT: DB      0FFH            ;START OFF WITH ALL ON

;READ A CHARACTER FROM CONSOLE
;        ON RETURN CHARACTER IS IN A REG.

CONIN:  EQU     $
        IN      21H             ;GET REMOTE CONSOLE STATUS
        ANI     2               ;IS THERE A CHARACTER READY
        JZ      CONINL          ;IF NOT CHECK LOCAL CONSOLE
        IN      20H             ;ELSE GET CHARACTER
        ANI     7FH             ;DESTROY PARITY BIT
        RET                     ;RETURN WITH CHARACTER

CONINL: IN      11H             ;ELSE CHECK LOCAL CONSOLE
        ANI     2               ;IS THERE A CHARACTER READY
        JZ      CONIN           ;IF NONE LOOP AGAIN
        IN      10H             ;ELSE GET CHARACTER
        ANI     7FH             ;THROW AWAY PARITY BIT
        CPI     'R'-64          ;DID WE GET A CONTROL R
        RNZ                     ;IF NOT RETURN
        LDA     VIDBIT          ;IF SO GET VIDBIT
        XRI     1               ;FLIP REMOTE BIT
        STA     VIDBIT          ;REPLACE VIDBIT
        MVI     A,7FH           ;SAY WE GOT A DELETE
        RET                     ;AND RETURN
```

listing to show intercept of control r and setting of remote output switch

Listing 6

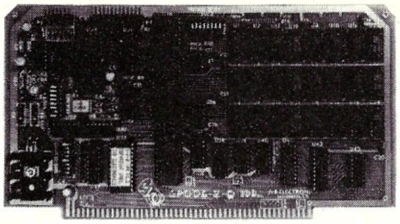
```
;VIDBIT - THE CONSOLE CONTROL BYTE IS A BYTE WITH EIGHT ONE
;        BIT SWITCHES.  AT PRESENT THE BITS USED ARE

;        BIT 0    - REMOTE COPY MODE [ 0 MEANS NO REMOTE COPY ]
;        BIT 1    - QUIET MODE [ 0 MEANS SYSTEM SEES NO CHARS ]

VIDBIT: DB      0FFH

;READ A CHARACTER FROM CONSOLE
;        CHARACTER IS RETURNED IN A REG. WITHOUT PARITY BIT

CONIN:  PUSH    H               ;SAVE HL REGS
        LXI     H,VIDBIT        ;SET UP ADDRESS OF VIDBIT
CONINR: IN      21H             ;GET REMOTE STATUS BYTE
        ANI     2               ;IS A CHARACTER READY ON REMOTE ?
        JZ      CONINL          ;IF NOT CHECK LOCAL CONSOLE
                        ;THERE IS A CHARACTER ON REMOTE CONSOLE
        IN      20H             ;GET CHARACTER FROM REMOTE
        ANI     7FH             ;REMOVE POSSIBLE PARITY BIT
        JMP     CONINO          ;GO TO COMMON EXIT ROUTINE
                        ;NONE ON REMOTE, SO CHECK LOCAL
CONINL: IN      11H             ;GET LOCAL CONSOLE STATUS BYTE
        ANI     2               ;IS A CHARACTER READY ON LOCAL ?
        JZ      CONINR          ;IF NONE, LOOP AGAIN
                        ;THERE IS A CHARACTER ON LOCAL CONSOLE
        IN      10H             ;GET THE CHAR
        ANI     07FH            ;REMOVE POSSIBLE PARITY BIT
                        ;NOW CHECK FOR SPECIAL CASES
        CPI     'R'-64          ;CHECK FOR A CONTROL R
        JNZ     TESTQ           ;IF NOT CHECK FOR QUIET
                        ;A CONTROL R MEANS FLIP REMOTE OUTPUT BIT
        MVI     A,1             ;GET BIT TO FLIP
        JMP     RETDEL          ;GO FLIP AND RETURN
                        ;SECOND TEST IS FOR QUIET MODE
TESTQ:  CPI     'Q'-64          ;IS THIS A CTL-Q
        JNZ     CONINO          ;IF NOT GO TO COMMON RETURN
                        ;A CONTROL Q MEANS GET QUIET MODE BIT TO FLIP
        MVI     A,2             ;FLIP QUIET MODE BIT
                        ;AFTER SPECIAL CASES, RETURN A DELETE
RETDEL: XRA     M               ;FLIP PROPER BIT
        MOV     M,A             ;RESTORE VIDBIT
        MVI     A,7FH           ;REPLACE CHAR. WITH DELETE
CONINO: MOV     C,A             ;TEMPORARILY SAVE CHARACTER
                        ;BEFORE WE RETURN, CHECK FOR QUIET MODE
        MOV     M,A             ;GET VIDBIT
        POP     H               ;RESTORE HL REGS
        ANI     2               ;ARE WE IN QUIET MODE ?
        MOV     A,C             ;IF NOT RESTORE A REG.
        RNZ                     ;AND RETURN
                        ;IN QUIET MODE, DO NOT RETURN, JUST PRINT
        CALL    CONOT           ;PRINT THE CHARACTER
        MOV     A,C             ;GET CHARACTER BACK INTO A REG.
        CPI     0DH             ;WAS IT A CARRIAGE RETURN
        JNZ     CONIN           ;IF NOT LOOP THROUGH CONIN
        MVI     C,0AH           ;ELSE GET LINE FEED
        CALL    CONOT           ;PRINT IT TO AVOID OVERPRINTING
        JMP     CONIN           ;LOOP THROUGH CONIN AGAIN
```

List to show console input from either one of two consoles with intercepts for control Q and control R

Listing 7

```
;VIDBIT - THE CONSOLE CONTROL BYTE IS A BYTE WITH EIGHT ONE
;        BIT SWITCHES.  AT PRESENT THE BITS USED ARE

;        BIT 0    - REMOTE COPY MODE [ 0 MEANS NO REMOTE COPY ]
;        BIT 1    - QUIET MODE [ 0 MEANS SYSTEM SEES NO CHARS ]

VIDBIT: DB      0FFH

;CHECK CONSOLE INPUT STATUS
;        ON RETURN IF NOT READY A = 0, IF READY A = FF

CONST:  IN      21H             ;GET REMOTE STATUS BYTE
        ANI     2               ;IS A CHARACTER READY ON REMOTE ?
        JNZ     CQUIET          ;IF THERE IS CHECK FOR QUIET MODE
        IN      11H             ;GET LOCAL CONSOLE STATUS
        ANI     2               ;IS A CHARACTER READY ON LOCAL ?
        RZ                      ;IF NONE RETURN
                        ;IF THERE IS A CHARACTER
CQUIET: LDA     VIDBIT          ;GET CONSOLE CONTROL BYTE
        ANI     2               ;IS QUIET MODE SET ?
        JZ      QUIETM          ;IF SO PROCESS IT HERE
                        ;IF NOT QUIET MODE
```

256K S-100 PRINTER BUFFER



SPOOL-Z-Q 100 - $119

(BARE BOARD WITH EPROM AND MANUALS)

Spool-Z-Q 100 is now available as a BARE BOARD WITH EPROM AND MANUALS! In addition, all of the parts needed to build a Spool-Z-Q 100 are available in partial kits. Complete kits at significant savings are also available. If you choose to "roll your own" with just the bare board option, you will find that no impossible to find parts are used.

Spool-Z-Q 100 is an S-100 board based hardware printer buffer which has an on-board computer and hardware features which allow it to send to either a serial (RS-232) or parallel (Centronics standard) printer. Spool-Z-Q 100 is available with 32K to 256K characters memory installed. Automatic internal space compression will allow even more storage for reports or listings containing "white space."

TECHNICAL DETAILS

SERIAL OUTPUT — RS-232 compatible. Baud rates- Switch selectable 19.2K, 9600, 4800, 2400, 1200, 600, 300, & 150 baud.

PROTOCOLS — Switch selectable XON/XOFF, ETX-/ACK, ENQ/ACK, Reverse Channel (Busy/Ready) either polarity, or parallel.

PARALLEL OUTPUT — Standard Centronics interface signals, 8 Data, Busy & Strobe.

S-100 (IEEE 696) INTERFACE — No wait states required on any system. Switch selectable I/O address can be set to ANY one of the 256 possible addresses. Extremely simple to use. Simply monitor the Busy status bit and send data to Spool-Z-Q when not busy. All protocols, etc. are taken care of already.

MEMORY TYPE AND EXPANSION — Spool-Z-Q 100 uses industry standard 4164 type 64K RAM chips. Sizes available are 32, 64, 128, 192, and 256K characters. Every Spool-Z-Q 100 is fully socketed for 256K and may be expanded by just plugging in chips.

AUTOMATIC SPACE CHARACTER COMPRESSION — Although the maximum size is 256K (60-120 pages of print) the space compression feature allows Spool-Z-Q to effectively hold much more printing which contains many spaces (listings, reports, etc.).

OTHER CAPABILITIES — Spool-Z-Q 100 has the same Pause-on-Formfeed, Clear Buffer, Copy, and Self-Test abilities as our stand-alone Spool-Z-Q. Signals are available on an 8 pin DIP socket to allow control of these functions via a simple external switch panel which will be available as an option.

PRICES: (Including shipping)

DOCUMENTATION ALONE - (No C.O.D.)	$5
KIT 1. BARE BOARD, EPROM & DOCS	$119
KIT 2. Additional Parts to make a parallel only board, except plug in chips.	$60
KIT 3. Plug in chips for a 32K parallel only board.	$65
KIT 4. Additional parts for serial capability.	$40
32K PARALLEL ONLY KIT. Kits 1, 2, & 3. Priced separately - would be $244	$209
32K SERIAL/PARALLEL KIT. Kits 1, 2, 3, & 4. Separately - $284	$239

ASSEMBLED AND TESTED WITH 15 DAY TRIAL PERIOD OFFER. FULL SERIAL AND PARALLEL.
32K - $319 64K - $349
128K - $409 192K - $469
256K - $529

CHECK WITH YOUR LOCAL S-100 DEALER, OR, ORDER DIRECT FROM FACTORY. We accept MC, VISA, AMEX, and COD orders. No extra charge for COD. Calif. Res. add 6% tax.

1601 Fulton Ave., Suite 10A
Sacramento, CA 95825
(916) 483-0709

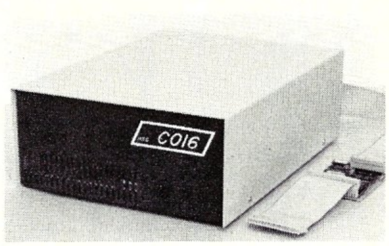
jvb electronics

DEALERS WANTED!

CIRCLE 28 ON READER SERVICE CARD

SAVE YOUR 8 BIT SYSTEM
Join the 16/32 Bit Revolution Through Evolution, For Under $600

APPLE 2E™
RADIO SHACK TRS80™
ZENITH™
OSBORN™
ALTOS™
KAYPRO™
COLUMBIA™
TELEVIDEO™
DEC™
or OTHER Z80 BASED SYSTEM



Would you like to run CPM86™, CONCURRENT CPM86™, CPM68K™, MS-DOS™, IBM™ PC APPLICATIONS, DEVELOP and TEST 16/32 BIT SOFTWARE, or add an INTELLIGENT HIGH SPEED RAM DISK to YOUR SYSTEM?

You can with the HSC CO16 ATTACHED RESOURCE PROCESSOR:

CO16 with either 8086, 80186, or 68000 16 bit micro-processor and up to 768K Bytes of parity checked RAM may be connected to virtually any Z80 based computer system or APPLE 2E computer system.

An 8 bit system equipped with CO16 retains all of its original capabilities PLUS it has the added ability to run most 16 bit operating systems and applications.

Prices start at $595.00 which includes the CO16 processor with 128K bytes of memory, two volume user manual, and the HSC Software Development System. Hardware options include memory expansion to 768K bytes, attractive desk top enclosure with power supply, and the INTEL™ 8087 Math Co-processor (available on CO1686 only). Software options include CPM86, CPM68K, Concurrent CPM86, or MS-DOS™ operating systems. The UNIX™ operating system will be available in the second quarter of 1984.

For Information on CO16 Contact
HSC INC.
Box 86
Herkimer, NY 13350
(315) 866-2311

HSC INC.

Dealer and OEM inquiries are invited.

CIRCLE 45 ON READER SERVICE CARD

VOCS

MOPI™

The Define Your Own Instructions Assembler / Compiler (Universal Cross Assembler)

A new type of software development package.
Generates machine code for any 8 bit microprocessor using user defined instructions.
Program text data insertable at time of assembly.
Unique serial number capabilities.
Divide large programs into overlay segments.
No Separate linking loader required.
Generates both relocatable and absolute code.
Produces user documentation from source file data.
Expandable to include new features as required.

MOPI software: $150 + options Manual: $35
Currently available for CP/M with 56 K memory.
Inquire concerning availability on other systems.
For more information & ordering instructions, write

Voice Operated Computer Systems
P.O. Box 3705, Minneapolis, MN 55403

MOPI is a trademark of Voice Operated Computer Systems
CP/M is a trademark of Digital Research Inc.

CIRCLE 113 ON READER SERVICE CARD

```
         MVI     A,0FFH          ;SET CHARACTER READY SWITCH
         RET                     ;AND RETURN
                        ;THIS IS IN CASE WE ARE IN QUIET MODE
QUIETM:  PUSH    H               ;SAVE HL REGS
         LXI     H,VIDBIT        ;GET ADDRESS OF VIDBIT
         MOV     A,M             ;GET VIDBIT
         XRI     2               ;FLIP QUIET MODE BIT
         MOV     M,A             ;RESTORE VIDBIT
         CALL    CONIN           ;GO GET CHARACTER
         MOV     C,A             ;MOVE CHAR TO C REG. FOR CONOT
         CALL    CONOT           ;LET CONOT DISPLAY IT
         MOV     A,M             ;GET VIDBIT
         XRI     2               ;RE-FLIP QUIET MODE BIT
         MOV     M,A             ;RESTORE VIDBIT
         POP     H               ;RESTORE HL REGS
         JMP     CONST           ;RUN STATUS CHECK AGAIN
```

list to show handling of console status routine in quiet mode

Listing 8

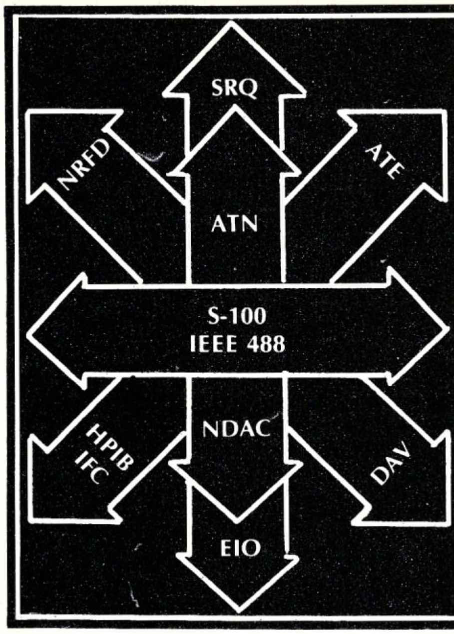
```
;VIDBIT - THE CONSOLE CONTROL BYTE IS A BYTE WITH EIGHT ONE
;        BIT SWITCHES.  AT PRESENT THE BITS USED ARE

;        BIT 0   - REMOTE COPY MODE [ 0 MEANS NO REMOTE COPY ]
;        BIT 1   - QUIET MODE [ 0 MEANS SYSTEM SEES NO CHARS ]
;        BIT 2   - DUPLEX MODE [ 0 MEANS HALF DUPLEX ]

VIDBIT: DB       0FFH

;DUPLEX - THE BYTE USED TO STORE REMOTE INPUT CHARACTERS SO THEY
;        ARE NOT ECHOED IN HALF DUPLEX MODE.

DUPLEX: DB       0FFH            ;IMPOSSIBLE TO GET AS INPUT

;READ A CHARACTER FROM CONSOLE
;        CHARACTER IS RETURNED IN A REG. WITHOUT PARITY BIT

CONIN:   PUSH    H               ;SAVE HL REGS
         LXI     H,VIDBIT        ;SET UP ADDRESS OF VIDBIT
CONINR:  IN      21H             ;GET REMOTE STATUS BYTE
         ANI     2               ;IS A CHARACTER READY ON REMOTE ?
         JZ      CONINL          ;IF NOT CHECK LOCAL CONSOLE
                        ;THERE IS A CHARACTER ON REMOTE CONSOLE
         IN      20H             ;GET CHARACTER FROM REMOTE
         ANI     7FH             ;REMOVE POSSIBLE PARITY BIT
         STA     DUPLEX          ;SAVE BYTE FOR DUPLEX CHECK
         JMP     CONINO          ;GO TO COMMON EXIT ROUTINE
                        ;NONE ON REMOTE, SO CHECK LOCAL
CONINL:  IN      11H             ;GET LOCAL CONSOLE STATUS BYTE
         ANI     2               ;IS A CHARACTER READY ON LOCAL ?
         JZ      CONINR          ;IF NONE, LOOP AGAIN
                        ;THERE IS A CHARACTER ON LOCAL CONSOLE
         IN      10H             ;GET THE CHAR
         ANI     07FH            ;REMOVE POSSIBLE PARITY BIT
                        ;NOW CHECK FOR SPECIAL CASES
         CPI     'R'-64          ;CHECK FOR A CONTROL R
         JNZ     TESTQ           ;IF NOT CHECK FOR QUIET
                        ;A CONTROL R MEANS FLIP REMOTE OUTPUT BIT
         MVI     A,1             ;GET BIT TO FLIP
         JMP     RETDEL          ;GO FLIP AND RETURN
                        ;SECOND TEST IS FOR QUIET MODE
TESTQ:   CPI     'Q'-64          ;IS THIS A CTL-Q
         JNZ     TESTH           ;IF NOT CHECK FOR H
                        ;A CONTROL Q MEANS GET QUIET MODE BIT TO FLIP
         MVI     A,2             ;FLIP QUIET MODE BIT
         JMP     RETDEL          ;GO FLIP AND RETURN DELETE
                        ;THIRD TEST IS FOR DUPLEX MODE CHANGE
TESTH:   CPI     'D'-64          ;IS IT A CONTROL D
         JNZ     CONINO          ;IF NOT GO TO COMMON RETURN
                        ;A CONTROL H MEANS GET DUPLEX MODE BIT TO FLIP
         MVI     A,4             ;FLIP DUPLEX MODE
                        ;AFTER SPECIAL CASES, RETURN A DELETE
RETDEL:  XRA     M               ;FLIP PROPER BIT
         MOV     M,A             ;RESTORE VIDBIT
         MVI     A,7FH           ;REPLACE CHAR. WITH DELETE
CONINO:  MOV     C,A             ;TEMPORARILY SAVE CHARACTER
                        ;BEFORE WE RETURN, CHECK FOR QUIET MODE
         MOV     M,A             ;GET VIDBIT
         POP     H               ;RESTORE HL REGS
         ANI     2               ;ARE WE IN QUIET MODE ?
         MOV     A,C             ;IF NOT RESTORE A REG.
         RNZ                     ;AND RETURN
                        ;IN QUIET MODE, DO NOT RETURN, JUST PRINT
         CALL    CONOT           ;PRINT THE CHARACTER
         MOV     A,C             ;GET CHARACTER BACK INTO A REG.
         CPI     0DH             ;WAS IT A CARRIAGE RETURN
         JNZ     CONIN           ;IF NOT LOOP THROUGH CONIN
         MVI     C,0AH           ;ELSE GET LINE FEED
         CALL    CONOT           ;PRINT IT TO AVOID OVERPRINTING
         JMP     CONIN           ;LOOP THROUGH CONIN AGAIN
```

code showing complete console input routine including store of remote input for duplex check and duplex intercept

Listing 9

```
;VIDBIT - THE CONSOLE CONTROL BYTE IS A BYTE WITH EIGHT ONE
;        BIT SWITCHES.  AT PRESENT THE BITS USED ARE

;        BIT 0   - REMOTE COPY MODE [ 0 MEANS NO REMOTE COPY ]
;        BIT 1   - QUIET MODE [ 0 MEANS SYSTEM SEES NO CHARS ]
;        BIT 2   - DUPLEX MODE [ 0 MEANS HALF DUPLEX ]

VIDBIT: DB       0FFH

;DUPLEX - THE BYTE USED TO STORE REMOTE INPUT CHARACTERS SO THEY
;        ARE NOT ECHOED IN HALF DUPLEX MODE.

DUPLEX: DB       0FFH            ;IMPOSSIBLE TO GET AS INPUT

;WRITE A CHARACTER TO CONSOLE DEV
;        ON INPUT CHARACTER IS IN C REG.

CONOT:   EQU     $
         IN      11H             ;GET LOCAL CONSOLE STATUS
         ANI     4               ;CHECK FOR READY
         JZ      CONOT           ;NOT READY - TRY AGAIN
         MOV     A,C             ;MOVE CHAR. FROM C TO A
         OUT     10H             ;OUTPUT TO LOCAL CONSOLE
                        ;NOW CHECK FOR REMOTE OUTPUT NECESSARY
         LDA     VIDBIT          ;GET SWITCH BIT
         ANI     1               ;CHECK REMOTE OUTPUT SWITCH
         RZ                      ;IF OFF DON'T DISPLAY REMOTELY
                        ;NOW CHECK FOR HALF DUPLEX
         LDA     VIDBIT          ;GET SWITCH BIT
         ANI     4               ;ARE WE HALF DUPLEX
         JNZ     CONOTR          ;IF NOT CONTINUE
                        ;ELSE SEE IF THIS IS ECHO
         LDA     DUPLEX          ;GET LAST INPUT
         CMP     C               ;ARE WE ECHOING
         RZ                      ;IF WE ARE DON'T BOTHER
                        ;AT LAST THE REMOTE OUTPUT CODE
CONOTR:  EQU     $
         IN      21H             ;GET REMOTE CONSOLE STATUS
         ANI     1               ;CHECK BIT #2
         JZ      CONOTR          ;NOT READY - LOOP
         MOV     A,C             ;MOVE CHAR FOR OUTPUT
         OUT     20H             ;OUTPUT IT
         RET                     ;RETURN
```

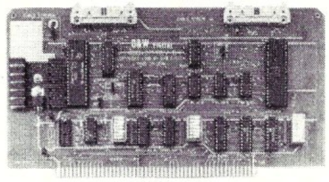
code for console output routine with check for half duplex as well as for no remote output



THE 488+3

IEEE 488 TO S-100 INTERFACE

IEEE-488



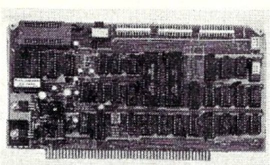
S-100

- Controls IEEE 488 (HPIB) Instruments with an S-100 computer
- Acts as controller or device
- Basic and assembly language drivers supplied
- Meets IEEE 696 specification
- Industrial quality burned in and tested
- Up to 125K bytes/sec under software control
- 3 parallel ports (8255-5)
- $375

D&W DIGITAL
20655 HATHAWAY AVENUE
HAYWARD, CA 94541 • (415) 887-5711

CIRCLE 66 ON READER SERVICE CARD

Extended Processing S100 Boards



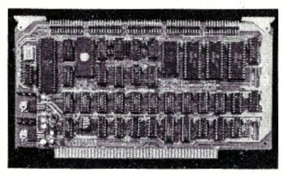
BURNER I/O

The BURNER I/O is a full feature multifunction S100/IEEE-696 board. It has a complete EPROM programmer, 2 Serial ports, parallel output port, parallel input port and memory management. Programs 2704 thru 2764, 2508, 2516 and TMS2716 EPROMS. Menu driven disk based software supplied in 4K EPROM that is easily loaded on your disk. Zero insertion force programming socket. Totally I/O mapped. Serial ports are independent and support hardware handshaking. Baud rates from 50–19,200. Serial ports are software programmable. Seperate parallel input and output ports with full handshaking plus 4 direct sense bits. Memory management controls the S-100 address lines A16-A23.

The board is offered in various configurations. In all versions it is fully assembled and tested.

Option A: Complete board with programmer, I/O and memory management, $354.95
Option B: Programmer only, $219.95
Option C: I/O only (2S + P), $219.95
Option D: Options B and C, $329.95
Option E: Memory management only, $109.95

Memory management may be added to Options B and C for $25.00



POWER I/O

The POWER I/O is a high performance slave computer that is a complete system including all hardware, software and documentation. It is guaranteed to operate properly with *any* S-100/IEEE-696 system. The POWER I/O sets a new standard in flexibility, ease of integration, expandability and performance. It is designed with the future in mind. All I/O and memory may be doubled with the addition of our POWER I/O add-on. Memory can be quadrupled when the 256k rams are available. The standard software supports 6 serial ports, 2 parallel ports and 512K of ram.

The basic board consists of a 4MHz Z80A, 64K DRAM, 3 serial (8251A) I/O ports, baud rates to 19,200. Hardware, ETX, ACK and X-on/X-off handshaking, 1 parallel port, timer, 6 programmable delays, high speed parallel I/O to host, complete POWER I/O software in 4K EPROM (expandable to 8K). On board software is designed for maximum flexibility and ease of use. Supports user written programs. Complete documentation including source code, schematics and many software interface examples.

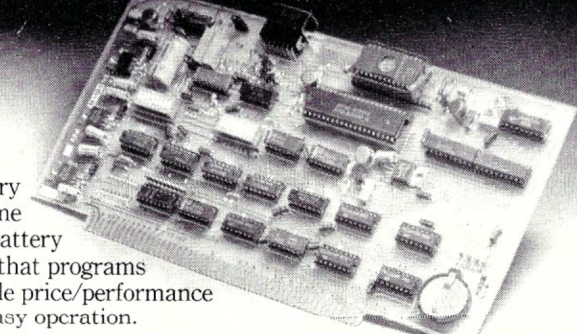
The POWER I/O add-on has 3 additional serial ports, 1 parallel port and 64K DRAM.

POWER I/O: $375.00
64K RAM ADD-ON: $175.00
3 SERIAL, 1 PARALLEL ADD-ON: $195.00
RAM AND I/O ADD-ON: $295.00

ep Extended Processing 3861 Woodcreek Lane ■ San Jose, California, 95117 ■ (408) 249-8248

CIRCLE 142 ON READER SERVICE CARD

WE GAVE YOUR DRIVES THE FIRST BREAK THEY EVER HAD AND NOW WE GIVE YOU TIME TO BURN...



TimeEPROMmer, the S-100 CP/M* compatible programmer that's useful every second of every day. A real time calendar/clock with lithium battery and an EPROM programmer that programs all popular eproms. Unbeatable price/performance ratio. Features designed for easy operation.

Eprom Programmer: Port addressable. Read, Verify, Program, and Disk transfer. Handles up to 28 pins. Power generated and controlled on board. All software and documentation included. Assembled units tested with burn in.

Real Time Calendar/Clock: Complete time counting functions with CMOS LSI. Allows up to 6 months power down use. Independently port addressable.

TimEPROMmer BB & software & manual $75
TimEPROMmer Kit & software & manual $195
TimEPROMmer A & T software & manual ... $295

Our DISK CONTROL UNIT that turns 8" drives off when not being accessed. State drive.

DCU, kit & manual $29.95
DCU, A & T & manual $49.95

P & H $2. NYS add tax.
CP/M is T.M. of Digital Research

OPTRONICS TECHNOLOGY
P.O. Box 81 Pittsford, NY 14534 (716) 377-0369

CIRCLE 188 ON READER SERVICE CARD

A MICROSYSTEMS TUTORIAL

Relocating Assemblers and Linkage Editors: Part 3

by Andrew L. Bender

Parts 1 and 2 of this tutorial discussed the basic concepts of relocation, collection of relocatable code modules, and the problems of address resolution within these modules. The topic of communication between the various modules by means of a common memory area was discussed. Part 2 concluded by examining two methods for introducing non-relocatable code into a relocatable program. This part of the article will discuss the concept of program libraries: First, the macro library facility used with the assembler; then the relocatable library used with the linkage editor.

A macro library is a sequence of statements kept in a file on disk. The macro library usually contains definitions of certain operations—statements that define macro instructions and EQU directives are the most common statements in a macro library, but any assembly language source statements can be filed in a macro library. The text of a macro library may be included in the source language of the assembly or compilation by a programmer request. Usually, the programmer places an INCLUDE directive in his source code, followed by the filename of the macro library.

Although the primary focus of this tutorial has been relocating assemblers, we mentioned that compilers also produce relocatable code modules. Many compilers such as C, Fortran, PL/I, and Cobol have facilities to include macros from libraries in their source language. The diverse nature of the various source languages makes it difficult to discuss the macro facilities of each compiler in detail. You should be aware that many of the techniques that will be discussed in connection with assembler macro facilities are also applicable to compilers.

Macro instructions, or "macros," are defined in an assembly language program by a macro definition. A macro definition, sometimes called a "model," contains a series of source program statements called "model statements." Listing 1 illustrates the INCLUDE facility with a program that calls for a macro model from the library (MACRO.LIB) and a set of equate statements defining the CP/M functions and addresses (CPMDEF.LIB).

The assembler will read the input file above. Each time it encounters an INCLUDE statement it will examine the file directory on the default disk, find the proper file, and incorporate the text from the file into the assembly as if the textual material were a part of the source program. When a macro statement is encountered in the source program, the assembler looks for the corresponding model and expands the macro according to the model statements. The result is shown in the Listing 2.

The letter C appearing to the left of the assembler listing of the source program means that the text on that line of the source language was obtained from an INCLUDE file.

Andrew L. Bender, M.D., Neurological Services, Inc.
336 Center Avenue, Westwood, NJ 07675

The plus symbol appearing to the left of each line of source language means that that line of source programs was synthesized by the assembler's macro generator.

When the same EQU statements and COMMON storage definitions are used by a set of programs, it is best to put these definitions into a file that will be processed by an INCLUDE statement. The possibility of an error is reduced and, if it is necessary to change a data structure, only the INCLUDE file that contains the definition of the data structure need be changed. All the programs that contain an INCLUDE reference for the changed file are simply reassembled, and the entire program complex is link edited, resulting in a new .COM file.

Just as the macro library was useful at the source language level, a relocatable library is useful at the object code level. The relocatable library can contain subroutines that are used frequently by many programmers. Subroutines that are in general use, once developed and tested, can be put in the relocatable library so that everyone can use them. This saves time because these library subroutines need not be assembled or compiled every time you need them. The need to know what disk contains what set of subroutines is eliminated because they are all in a single library file. Last, but not least, the library file has only one filename, and less disk space is needed than if each routine were filed separately.

The linkage editor is responsible for searching and reading the relocatable library file. The format of the relocatable library file is matched to the linkage editor's requirements. These requirements prevent the interchange of relocatable libraries between different linkage editors. Despite the fact that the relocatable format of Microsoft is almost the same as that of Digital Research and relocatable binary modules can be interchanged between LINK-80 (Digital Research) and L80 (Microsoft), the libraries are of different formats and cannot be interchanged without some reformatting operation.

In most cases the linkage editor is supplied with a library manager program. Programs to be incorporated into the library are first processed by the library manager, which produces a single library file containing all of the programs in a format acceptable to the linkage editor.

Certain linkage editors are not supplied with a library manager and use directives in the assembly language program to produce a library. For example, the CDL MACRO2 assembler requires that programs placed in a library file be assembled so that they are separated by .PRGEND directives instead of .END directives. Under this condition, the assembler writes one long output relocatable file instead of separate ones. This becomes the library file.

Once the library file exists, to reference subroutines in the library, you mention them as external symbols in the source program. After the last operator-specified relocatable module has been read by the linkage editor, the library will be searched for entry points that will match the unresolved external symbols. The library search may be automatically requested or may require an operator command to initiate it. In different linkage editors, there are different commands that cause a library search. In general, these commands specify a library file name and the disk it resides on. There is usually a default library name and a default library search command that is invoked when the module being linked contains unresolved external symbols and the operator has informed the linkage editor that he has no more commands to input.

Certain assemblers provide the programmer with the ability to enter the name of a library file that is to be searched when the relocatable code module is being link edited. In M80 this is the .REQUEST directive. The operand of the .REQUEST directive consists of the names of files that are to be searched by the linkage editor for the entry points that would satisfy any unresolved external references.

Finding the entry points in a library of relocatable modules depends on the implementation of the linkage editor. Some linkage editors will search the entire library module by module, looking for an entry point with the proper name. When this entry point is found, the module in which it exists is link edited. The process is repeated until all of the desired modules have been link edited or there are no modules with the desired name. In the last case the link editor usually returns control to the operator, hoping that the needed routine is available.

This library search scheme usually works well, but it can lead to problems when a library routine calls for a module in the library that was encountered previously. Since the entry point name will not be encountered as the remainder of the library is searched, the linkage editor will report that there is an unresolved external symbol. In order to correct this problem, it is necessary to ask for another library search to collect the needed module. If the newly loaded module again contains another reference to a previously encountered entry point, it will be necessary to search the library again. All of these searches waste time and are not necessary. If the library is arranged with some thought to the problem of how the linkage editor does its library searching, only a single search need be done.

Some libraries contain a directory of entry points and the modules in which they reside. In addition, the external symbols for each module are also part of the directory. The linkage editor reads this directory and consults it to organize the order in which modules will be collected.

The last topic to consider in a discussion of libraries is how to maintain the "latest copy" of each library on the disk. In CP/M 2 and older versions, there is no "time stamp" on any item. Because multiple copies of the same file can exist on different disks, it is important to know which is the latest. The date and time the file was created and the date and time it was last updated are invaluable for program development. Since the operating system did not support this prior to CP/M 3.0, many programmers wrote comments inside the source program such as "fixed bug so and so 12/5/81 LATEST." This didn't help the relocatables that could not be identified. I use a system of appending a serial number to the filename of my files. The serial number corresponds to an entry in a text file that gives the history of each change. Frankly, it is a lot of work; and sometimes I make an error and do not enter the last set of changes in the text file. The text file for my relocatable library is shown in Listing 3.

Each library is identified by three digits following the dash. These three digits correspond to an entry in the text file that describes which changes were made and why. This scheme can work with any library not referred to by a .REQUEST directive or a linkage editor search that implicitly requests a library file with a fixed name. In these two cases, the library filenames are relatively fixed.

To be sure, there are other schemes. The CDL assembler and linkage editor include the date and time of assembly or compilation in the relocatable modules automatically if you are running under their TPM operating system. No matter what scheme you use, don't discard or overwrite a library that you are updating. My advice here is to keep one floppy with old libraries on it. You might have to refer to an old subroutine for a number of reasons, and if the old subroutines are gone, it may cause major problems when a program is relinked using the new library. There are definite cases when older is better, and "new and improved" may be worse. Take, for example, a program that implicitly depends on a lack of precision in a required library subroutine. When the new, more precise, subroutine is link edited with the old program, the increased precision might cause the program that always worked to loop or "bomb" because of the nature of the numerical calculations.

There are many options available for specifying a search of a relocatable library file to the linkage editor. Sometimes, if you are not specific, the linkage editor will include every subroutine in the library without regard to what you really needed. Other commands are very selective, including only those routines specifically required. You need to read the documentation carefully because it is usually fuzzy in regard to this topic. Sometimes, as before, experimentation is the best teacher.

Once your programs are prepared and you are ready to link them together with the linkage editor, there are certain commands that may be used to control the linking process. You may set the initial address of any program's CSEG or DSEG by telling the linkage editor exactly where you want the CSEG or DSEG placed. This has the same effect as the ASEG assembler directive discussed in the last part of this serial. Any module can have the DSEG and CSEG segment positioned anyplace in memory, but to control the placement of COMMON blocks special tricks may be needed, because every linkage editor handles these blocks in a different manner.

One trick is to create modules consisting only of the COMMON block you wish to lead in a specific place, then requesting the linkage editor to load these modules at the address you specify as the base of either the data or code segment. Certain linkage editors may want one byte of DSEG or CSEG in the module, as shown in Listing 4.

For example, suppose you wanted this block loaded at

Subroutines that are in general use, once developed and tested, can be put in the relocatable library so that everyone can use them. This saves time because these library subroutines need not be assembled or compiled every time you need them.

4CO7H. You would make this module the *first* module containing the common block BASED that you would load. Remember that common storage is allocated only once, and if this block BASED was encountered previously by the linkage editor, it would already have been allocated space in memory, probably not where you wanted it. To continue with this problem, you would direct the linkage editor to load the module at 4CO6H. This would allocate the initial dummy byte of zero to 4CO6H, but the common block would be loaded next at 4CO7H. Certain linkage editors allow the user to specify the positioning of common blocks by explicit commands; as, for example:

```
/P400/D700/BASED-900
```

This would set the CSEG to 0400H, the DSEG to 0700H and the common block BASED to 0900H. When in doubt, consult the manuals or try it. Don't be afraid to experiment, since nothing will be broken if your attempts fail.

A useful command is to have the linkage editor output a .COM file or a .HEX file. Most linkage editors have these options built in with a file designator to specify the name of the receiving file.

In the next installment, we will conclude this series with a examination of parameter transmission, calling sequences, debugging, and some hints on constructing relocatable programs. We will also discuss interfacing assembly language relocatables with relocatables generated by Basic-80 and Fortran-80.

Listing #1

```
;DEMONSTRATE MACRO INCLUSION
        INCLUDE CPMDEF.LIB
        INCLUDE MACRO.LIB
        ENTRY   WATSYS
WATSYS: PUSH    D
        PUSH    B
        CPM     RETVER
        POP     B
        POP     D
        RET
        END
```

Listing #2

```
          C ;DEMONSTRATE MACRO INCLUSION
          C         INCLUDE CPMDEF.LIB
          C ;;CPM DEFINITIONS
          C ;;MACRO LIBRARY 1/3/83
          C ;;...................
0005      C BDOS    EQU     0005
005C      C FCB     EQU     005CH
0080      C TBUF    EQU     0080H
0001      C CIN     EQU     1
0002      C COUT    EQU     2
          C ;ETC...
000C      C RETVER  EQU     12
          C ;ETC...
          C         INCLUDE MACRO.LIB
          C ;;MACRO LIBRARY
          C ;;1/3/83
```

```
                        C   ;;CONTAINS COMMON MODELS
                        C   ;;
                        C   ;;
                        C   CPM        MACRO    ?F,?P,BDH
                        C              IF       NOT NUL ?P
                        C              LXI      D,?P
                        C              ENDIF
                        C              IF       NOT NUL ?F
                        C              MVI      C,?F
                        C              ENDIF
                        C              CALL     BDOS     ;CALL SYSTEM
                        C              ENDM
                        C    ;;...OTHERS FOLLOW
                                                      ENTRY   WATSYS
  0000'   D5                 WATSYS: PUSH     D
  0001'   C5                         PUSH     B
                                     CPM      RETVER
  0002'   0E 0C       +              MVI      C,RETVER
  0004'   CD 0005     +              CALL     BDOS     ;CALL SYSTEM
  0007'   C1                         POP      B
  0008'   D1                         POP      D
  0009'   C9                         RET
                                     END
Macros:
CPM
Symbols:
BDOS    0005    CIN     0001    COUT    0002    FCB     005C
RETVER  000C    TBUF    0080    WATSYS  0000I'
```

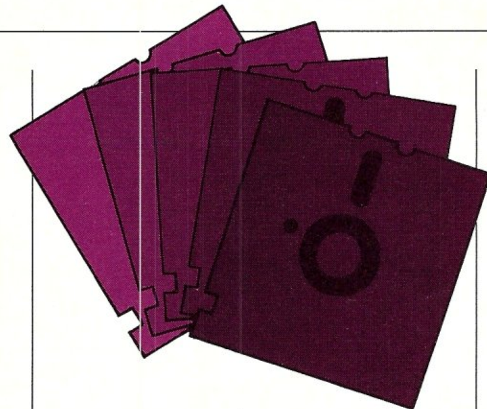
Listing #3

```
FILE=RLIB-NNN.LIB
FILE NAME          DESCRIPTION
--------------------------------------------------------------
RLIB-047           Change MTFCB to allow for creating FCB of 36
12/23/82           bytes instead of 33.
                   Change FILL.  Added entry point FILLZ to fill
                   memory with zeroes.
                   Correct FAD.  When -0.0 was added it gave an
                   unnormalized answer.
                   Add module HEXOUT.  This contains routines to
                   output 2 or 4 byte hex numbers to console.

RLIB-046           Correct AVLSTR to look at TBUF and bias top
11/28/82           of memory by 4096 words if ZDM, DDT or ZDMZ
                   are loaded (corrects problem with dynamically
                   allocated memory during debugging).

RLIB-045           Modify MTFCB so that read/write flag is kept
9/7/82             on the stack instead of in a fixed cell.
```

Listing #4

```
                       ;FORCE LOADING OF COMMON BLOCK
                       ;AT K+1 WHERE K IS ADDRESS OF
                       ;/P LINKAGE EDITOR DIRECTIVE
  0000'                        CSEG
  0000'      00                DB       0      ;FORCE CSEG LOAD
                               COMMON   /BASED/
  0000!      00            A:  DB       0
  0001!      00            B:  DB       0
                           ;ETC,ETC
  0002!                        DSEG
                               END
Symbols:
A         0000!    B        0001!    BASED    0002C
```

Software Directory



Program name: CO-SORT
Hardware system: any microcomputer running CP/M or MS-DOS
Minimum memory: 12K
Language: 8080 assembler
Description: CO-SORT's co-routine capability for sorting and merging supports interactive processing between the applications program and a very high speed sort/merge routine. The applications program calls CO-SORT: at completion, the program continues processing. This allows complete selection and combination of input/output records and permits unlimited flexibility in key comparisons.

The rapid execution speed of CO-SORT is due to the following factors: data and control information are exchanged dynamically, so it is not necessary to build and move intermediate files; there is no break in the flow of the program and no operator intervention, thereby reducing throughput time substantially. Included in the CO-SORT package is COUP, a program that interacts with the user and makes calls to CO-SORT. COUP provides a general off-line utility sort/merge facility. Written in assembly language, it can serve as a teaching model for assembly language programmers.
Price: $200
Included with price: disk and complete documentation
Available from:
Information Resources
Box W
Manhasset, NY 11030
(516) 365-7629
CIRCLE 315 ON READER SERVICE CARD

Program name: Solomon Tutorial System
Hardware system: Any microcomputer running CP/M
Minimum memory: 64K
Description: The Solomon Tutorial Demonstration System is designed to teach the implementation and day-to-day operation of the Solomon Series Accounting Software in 10 hours through actual use of the software. The transactions of a fictional company are presented to provide hands-on training, allowing users to see how easy Solomon Series Software is to use, even if the user has had no previous computer experience. A unique management overview section shows how Solomon generates meaningful business management reports.
Price: With the Solomon Accounting Software; $60; separately: $260.
Included with price: Tutorial manual, user manual, wall charts and demonstration disks. A dealer demo package is also available.
Available from:
Computech Group Inc.,
Main Line Industrial Park,
Lee Blvd.,
Frazer, PA 19335
or:
24160 Haggerty Rd.
Farmington Hills, MI 48024
(215) SOLOMON
CIRCLE 316 ON READER SERVICE CARD

Program name: EMACS screen editor and LEX68 word processor
Hardware system: Tandy TRS80-16 under Xenix
Minimum memory: 0.25 MB
Language: C
Description: EMACS, a full screen text editor, features multiple windows that allow several files to be edited simultaneously, great extensibility through the use of macros and the built-in MLISP programming language, automatic indenting and parenthesis checking. In addition, EMACS communicates directly with Xenix, so that users can execute system commands and programs from within one EMACS window and the output will be captured in another window.

LEX68 is a powerful, menu-driven word processing package that handles document preparation using full screen editing and scrolling, rulers to format text, extensive cut and paste facilities, automatic column formatting, and a built-in four-function calculator. A mass mailing and a database are also included, as is a 100,000-plus word spelling system.
Price: EMACS, $395 for binary; $995 for source code; LEX68, $750.
Available from:
UniPress Software, Inc.
1164 Baritan Avenue
Highland Park, NJ 08904
(201) 985-8000
CIRCLE 317 ON READER SERVICE CARD

Program name: TCS™ Client Ledger System
Hardware system: Any Z80, 8080 or 8085 microprocessor with CP/M 2.2 or equivalent.
Minimum memory: 56K
Language: Microsoft Basic
Description: The TCS Client Ledger System is a multicompany general ledger system that has achieved "Big 8" accounting firms' recognition. TCS is a write-up package designed for CPAs and others who require such features as ledger consolidation, amortization and depreciation schedules, a fully integrated passive payroll, and over 50 financial reports. TCS accepts reporting from Total Receivables, Total Payables, and Total Payroll in the TCS Total Accounting System, furthering TCS's versatility.
When released: August 1981
Price: Price to the end user is determined by dealer installation, training, and support.
Included with price: Complete documentation, TCS pen and pad, and box for storage.
Availble from:
TCS Software, Inc.
3209 Fondren Road
Houston, TX 77063
CIRCLE 318 ON READER SERVICE CARD

Program name: RP/M
Hardware system: 8080 or Z80 running CP/M 2.2.
Minimum memory: 24K
Language: 8080 assembly language
Description: RP/M is functionally equivalent to CP/M 2.2. A new resident console processor named RCP and a new resident disk operating system named RDOS replace the CCP and the BDOS of CP/M 2.2. The new resident offers certain operational enhancements, such as user number made visible in the system prompt, cross drive command file search, user 0 files available to all users, file size and user assignment displayed by **DIR**, paged **TYPE** display, and an automatic disk flaw

BDS C

The *fastest* CP/M-80 C compiler available today

Version 1.5 contains some nifty improvements:

The unscrambled, *comprehensive* new User's Guide comes complete with tutorials, hints, error message explanations and an index.

The *CDB* symbolic debugger is a valuable new tool, written in C and included in *source form*. Debug with it, and *learn* from it.

Hard disk users: You can finally organize your file directories sensibly. During compilation, take advantage of the new path searching ability for all compiler/linker system files. And at run-time, the enhanced file I/O mechanism recognizes user numbers as part of simple filenames, so you can manipulate files located *anywhere* on your system.

BDS C's powerful original features include dynamic overlays, full library and run-time package source code (to allow customized run-time environments, such as for execution in ROM), plenty of both utilitarian and recreational sample programs, *and speed*. BDS C takes less time to compile and link programs than any other C compiler around. And the execution speed of that compiled code is typically lightning fast, as the Sieve of Eratosthenes benchmark illustrates. (See the January 1983 BYTE, pg. 303).

BD Software
P.O. Box 9
Brighton, MA 02135
(617) 782-0836

8" SSSD format, $150
Free shipping on pre-paid orders
Call or write for availability on other disk formats

CIRCLE 202 ON READER SERVICE CARD

'EASI - *Draft 2000*'

Computer Aided Drafting System

EASI - Draft 2000 is a menu driven 2D graphics package written in MC68000 assembly language. It provides FAST user friendly interactive display functions that are found typically on systems costing several times our (under $20,000) introductory price.

Hardware includes:

TRS-80 Model 16B
Vectrix high resolution (672 x 480) color monitor
Summigraphics digitizing tablet
Houston Instruments plotter (24" x 36" paper size)

Software features:

Multiple active drawing viewports
Multi-layered drawings
Graphic symbols and figures with transformations
Stroked text (can be user definable)
Circles, arcs, ellipses, and curve generation
Pan and zoom windowing
Background plotting
Dynamic rubber band line
Powerful drawing editing functions

EASI *Software, Inc.*

3425 East 117th Drive
Denver, Colorado 80233
(303) 451-6484

2891 Livonia Center Rd.
Lima, New york 14485
(201) 367-5735

CIRCLE 219 ON READER SERVICE CARD

Six Times Faster!

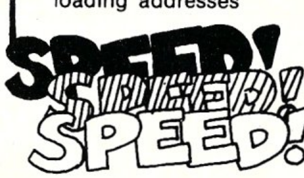
Super Fast Z80 Assembly Language Development Package

Z80ASM

- Complete Zilog Mnemonic set
- Full Macro facility
- Plain English error messages
- One or two pass operation
- Over 6000 lines/minute
- Supports nested INCLUDE files
- Allows external bytes, words, and expressions (EXT1 * EXT2)
- Labels significant to 16 characters even on externals (SLR Format Only)
- Integral cross-reference
- Upper/lower case optionally significant
- Conditional assembly
- Assemble code for execution at another address (PHASE & DEPHASE)
- Generates COM, HEX, or REL files
- COM files may start at other than 100H
- REL files may be in Microsoft format or SLR format
- Separate PROG, DATA & COMMON address spaces
- Accepts symbol definitions from the console
- Flexible listing facility includes TIME and DATE in listing (CP/M Plus Only)

SLRNK

- Links any combination of SLR format and Microsoft format REL files
- One or two pass operation allows output files up to 64K
- Generates HEX or COM files
- User may specify PROG, DATA, and COMMON loading addresses
- COM may start at other than 100H
- HEX files do not fill empty address space.
- Generate inter-module cross-reference and load map
- Save symbol table to disk in REL format for use in overlay generation
- Declare entry points from console
- The FASTEST Microsoft Compatible Linker available



- Complete Package Includes: Z80ASM, SLRNK, SLRIB - Librarian and Manual for just $199.99. Manual only, $30.
- Most formats available for Z80 CP/M, CDOS, & TURBODOS
- Terms: add $3 shipping US, others $7. PA add 6% sales tax

For more information or to order, call:

1-800-833-3061

In PA, (412) 282-0864

Or write: SLR SYSTEMS
1622 North Main Street, Butler, Pennsylvania 16001

SLR Systems

CIRCLE 53 ON READER SERVICE CARD

processing mechanism. However, the significant advantage of running under RP/M is provided by the system document. Source program listings of RCP and RDOS appear in the RP/M user's manual. RP/M is self installing on any computer that is currently running CP/M 2.2. The RP/M release disk contains **RPMGEN** for creating a relocated RP/M resident to match the existing CBIOS, and **GETRPM** for loading the relocated resident into location 0980H for **SYSGEN**, or equivalent system "put" utility.
When released: 1983
Price: Manual alone, $55; manual with 8″ SSSD disk, $75.
Included with price: Manual contains installation instructions and complete RCP and RDOS source code listings in 8080 assembly language. The standard CP/M 8″ disk contains **RPMGEN.COM** and **GETRPM.COM** for automatic installation of RP/M on any computer currently running CP/M 2.2. Most 5¼″ formats available; specify and add $15 for conversion.
Available from:
microMethods
P.O. Box G
118 SW First St.
Warrenton, OR 97146
(503) 861-1765
CIRCLE 319 ON READER SERVICE CARD

Program name: UTL™
Hardware system: Any CP/M system.
Language: Assembler
Description: A universal utility for use on any CP/M system. It consolidates six commonly used file functions into a single program with a concise menu. All commands are accessed from the menu with a single keystroke, are self-typing, and provide extensive operator prompting. There is absolutely no command syntax to remember. Each function has extensions and features not generally available with other utilities.

The directory command has full selectivity, an extremely fast recursive sorting algorithm and an optimized columnar display with file sizes and attributes. It provides remaining disk space available even on selective directories while automatically accounting for the different techniques required for implementation under CP/M 2.2 and bank-switched CP/M-Plus systems.

The TYPE command is fully buffered, has screen pauses and features an exclusive bidirectional capability to display the next screen or previous screens from a 10,000 character buffer.
When released: January 1984
Price: $29.95
Included with price: disk and manual.
Available from:
EWDP Software, Inc.
P.O. Box 40283
Indianapolis, IN 46240
(317) 872-8799
CIRCLE 320 ON READER SERVICE CARD

Program name: TCS™ Total Accounting System
Hardware system: Any Z80, 8080 or 8085 microprocessor with CP/M 2.2 or the equivalent.
Minimum memory: 52K
Language: Microsoft Basic
Description: The TCS Total Accounting System is a complete family of sophisticated yet easy to use business programs that includes TCS Total Ledger, Total Receivables, Total Payables, Total Payroll, Total Inventory, Total Utilities, Simple, and Q/LABEL. Each package uses the built-in database manager to integrate with all other TCS packages, and to provide fast database access, data file protection from system or power failure, and customized reporting.

TCS Simple is the key to the TCS Total Accounting System for customized displays and reports using information stored in any of the TCS accounting data files. It quickly and easily accesses information you need to make immediate and accurate business decisions.
When released: September 1981
Price: Price to the end user is determined by dealer installation, training, and support.
Included with price: Complete documentation, TCS pen and pad, and box for storage.
Available from:
TCS Software, Inc.
3209 Fondren Road
Houston, TX 77063
CIRCLE 321 ON READER SERVICE CARD

CIRCLE 14 ON READER SERVICE CARD

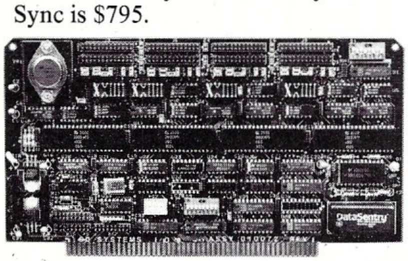
New Products

Serial I/O expansion boards

SDSystems, Inc. has released a family of serial I/O expansion boards for the IEEE-696/S-100 bus. Family members include the I/O 4 Async, with four asynchronous serial channels, the I/O 8 Async, with eight asynchronous serial channels, and the I/O 4 Async/4 Sync, with four asynchronous and four synchronous/asynchronous serial channels. All three boards support the full 24-bit address space of the IEEE-696 specification.

All three boards use only eight S-100 I/O ports, addressable to any eight-byte boundary in 64K with user-selectable switches. A realtime clock/counter is provided, with 56 bits of battery backed-up RAM that may be used for storage during power-downs. The I/O family may operate in polled I/O or interrupt-driven operation. Three interrupt types are generated on board: serial I/O, realtime clock, and standby realtime clock. Software-programmable baud rates from 50 to 19,2000 are supported.

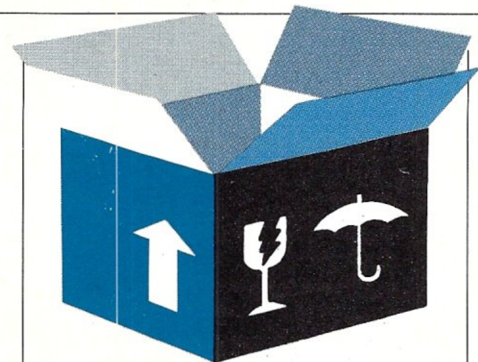
All asynchronous channels have RS-232C drivers and receivers supporting RXD, RTS, and CTS. These channels are configured for DTE only. The four asynchronous/synchronous channels have RS-232C drivers and receivers supporting TXD, RXD, RTS, CTS, DSR, and DTR as well as modem transmit and receiver clocks. Full DCE/DTE strapping is provided for modem and terminal look-alike pin-outs. The I/O 8 family has multilayer cards with full power and planes for superior noise immunity. The I/O Async/4 Sync is $795.



SDSystems, Inc., *P.O. Box 28810, Dallas, TX 75228; (214) 340-0303.*

CIRCLE 322 ON READER SERVICE CARD

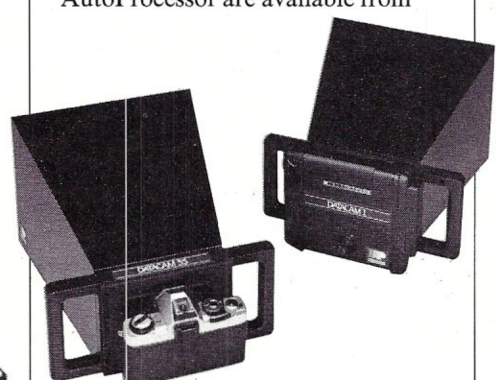
Color graphic recording cameras



DATACAM I and DATACAM 35 are two color graphics recording cameras that can record CRT images from personal and industrial computers. Each comes preassembled; no customizing for individual video screens is required. Just place a DATACAM on any CRT monitor and press a button to shoot.

DATACAM 35 comes in two sizes—12″ x 13″ and 19″ x 21″—to fit almost every available screen. It can also record CRT images in a slide format on Polachrome 35 mm film. This film produces color slides that can be processed on an AutoProcessor and mounted in a matter of minutes. Conventional 35 mm color film may be used as well and sent out for developing.

Polachrome film and AutoProcessor are available from

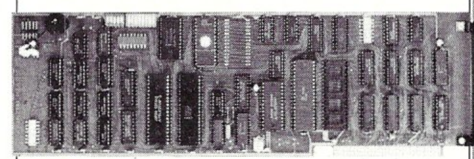


Photographic Sciences Corp., *770 Basket Rd., Webster, NY 14580; (716) 265-1600.*

CIRCLE 323 ON READER SERVICE CARD

IBM PC, XT compatible 8088-based CPU board

CP-88, the new Electro Design CPU, features a highly advanced ROM-resident BIOS with MS-DOS-compatible calling conventions and extensions for scientific and industrial applications. The CP-88 can support a no-disk, ROM-based system without either a video board or a keyboard and come up as an RS-232 board. It can therefore be used along with a RAM memory board and a six-slot Electro Design Expansion chassis to serve as a remote control box at the end of a host computer in a process control application. For applications requiring high-speed complex math processing, the CP-88 may be enhanced with a Numeric Processing Extension (NPX). The CP-88 is available in single-lot quantities for $476.



Electro Design, Inc., *690 Rancheros Drive, San Marcos, CA 92069; (619) 471-0680.*

CIRCLE 324 ON READER SERVICE CARD

External monitor adaptor and 12″ keyboard cable

Advent Products, Inc. has introduced two new quality accessories for the Kaypro computer: an external monitor adaptor and a 12″ keyboard cable.

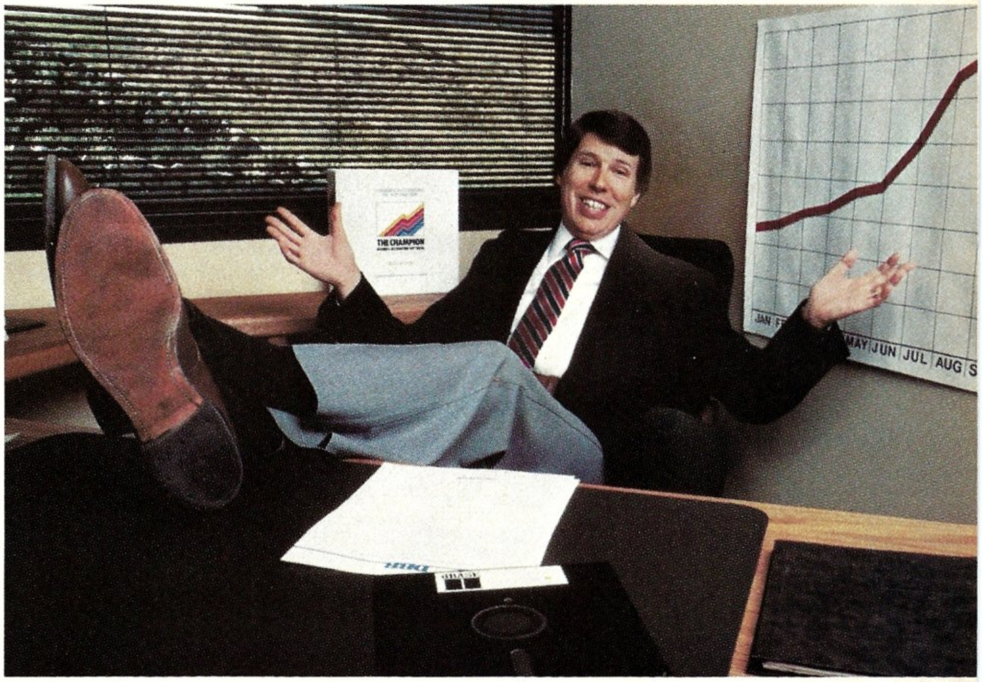
The external monitor adaptor allows the Kaypro II or 4 to drive an external video monitor. The size or manufacturer of the monitor does not matter. The external monitor adaptor will even work with large-screen color TVs that are equipped with a composite video input jack. A larger monitor can be extremely useful in training sessions, demonstrations, aiding the visually impaired and just making the Kaypro easier to read. Unlike other currently available adaptors, installation requires no soldering, desoldering, or wiring. In addition, installation in no way alters or damages any part of the motherboard. Removing the cover from the Kaypro, then removing one IC from its socket and plugging in the adaptor and video cable is all that's needed.

The external monitor adaptor is available for immediate delivery at a cost of $59.95.

The 12″ keyboard cable is twice the length of the original cable and allows additional flexibility in placing the keyboard and computer. Using the same type and quality of ca-

"dBASE II® is far, far better than a shoehorn."

Rusty Fraser
President
Data Base Research Corp.



"We laughed when our customers asked us to put our minicomputer-based real-time accounting system, The Champion,™ on a micro.

"No way was it going to fit, we thought.

"We'd have to create our own database management system and, even then, it'd be a tight squeeze.

"Then we discovered dBASE II, the relational database management system for microcomputers from Ashton-Tate."

"dBASE II was a perfect fit."

"dBASE II is a program developer's dream come true. The dBASE II RunTime™ module quickly provided us with the powerful text editing, data entry speed and other 'building block' capabilities we needed to develop and deliver a new Champion to our customers—the leading real-time on-line accounting system available for a micro."

The short cut to success.

The dBASE II RunTime module has helped a lot of program developers like Data Base Research become successful software publishers.

For more about dBASE II and RunTime, contact Ashton-Tate 10150 West Jefferson Boulevard, Culver City, CA 90230, (800) 437-4329, ext. 217. In the U.K., call (0908) 568866.

For more about The Champion, call Data Base Research at (303) 987-2588.

ASHTON·TATE

CIRCLE 15 ON READER SERVICE CARD

This page is for people who want to buy a great multi-user system.

And who don't want to pay an arm and a leg for it.

For you we have the Octagon 8/16.™ At only $7350,* the Octagon 8/16 has the guts to handle any job you and three other people throw its way. And then some.

The multi-tasking Octagon 8/16 includes all the operating software you need to simultaneously run any of your 8-bit CP/M 80† applications and any of your 16-bit Concurrent CP/M 86† applications. (Or MP/M 86† if you're in a multi-user environment.)

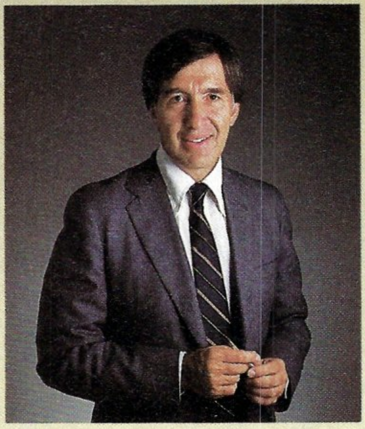
It will also execute software written to run under MS-DOS.† And UNIX† will be available by December '83.

So all your 8-bit software is as useful and productive as ever. But you still reap all of the advantages of a 16-bit system. Simultaneously.

All this performance is due to the Octagon 8/16's dual processor architecture. Its 8-bit NSC-800 CPU executes the full Z80† instruction set. Its 8088 CPU gives it the muscle to plow through 16-bit programs in a flash.

If you need a lot of number-crunching, there's an optional 8087 math co-processor.

Every Octagon 8/16 includes your choice of at least 256K of static RAM or 512K of dynamic RAM. Either is expandable to 1 MB: No lack of power here.



It also includes RAM disk software for the fastest possible execution speeds.

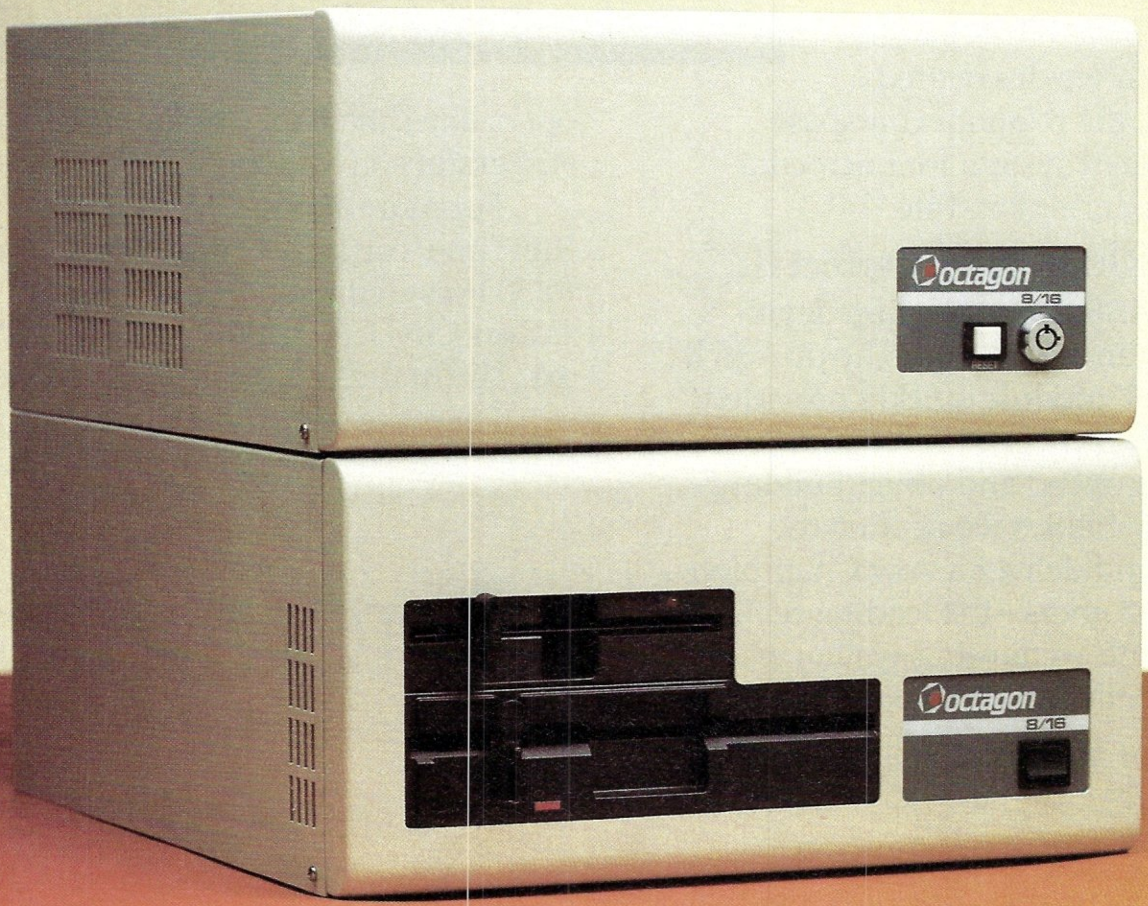
And for unheard-of versatility, each Octagon 8/16 includes a 5¼″ floppy (315K, IBM PC formatted), an 8″ floppy (1.4 MB formatted), and a 5¼″ Winchester (19.2 MB formatted). And there are other configurations to choose from, too.

So you not only get the kind of mass storage you need, you also get the kind of cost-effectiveness you can't get anywhere else. Call us today for complete details.

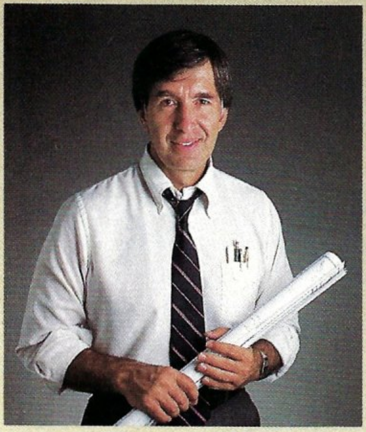
The Octagon 8/16. All the performance you need from a single- or multi-user system.

At a price that won't bust your budget.

*Basic configuration.



This page is for people who want to build a great multi-user system.



With the right ingredients, you can build almost anything. When you use advanced S-100 boards from Octagon, all you have to add is your imagination.

Start with the most powerful multiprocessing CPU board available: The CPU Board 8/16.™ For only $895 you get an 8-bit 4 MHz NSC-800 (which executes the full Z80 instruction set) and a 16-bit 8 MHz 8088. An optional 8087 math co-processor is available, too.

Together they let you simultaneously run 8-bit CP/M 80 and 16-bit CP/M 86. (MP/M 86 in a multi-user environment.)

The board's 8272 floppy disk controller governs up to four 5¼″ or 8″ floppies at once in any combination.

An 8K PROM monitor boots the operating system and contains several key memory debugging routines.

Two serial ports feature software-selectable baud rates up to 19.2K baud.

And when you buy the operating system from Octagon—be it CP/M 86 for $150, Concurrent CP/M 86 for $195, or MP/M 86 for $495—you also get a full CP/M 80 emulator at no extra charge.

As for memory, your multi-user system wouldn't be state-of-the-art without our 256K static RAM memory board.

The first of its kind, this board—for only $1850—accepts either 8-bit or 16-bit bus requests. So it makes a perfect match for the CPU board. (If 128K is all you need, it's all yours for just $1095.)

Both versions feature a handy time-of-day clock with battery back-up that keeps track of seconds, day, month, and year.

If your tastes run more toward dynamic RAM, you need our unique 521K DRAM board. Use it as a standard memory card. Or flip a switch and it becomes RAM disk. $1400 gets you the 512K version; $800 for the 256K version.

The last major ingredient in your system is the hard disk controller. This board handles up to four 5¼″ Winchester disk drives.

It includes automatic seeks and retries after error, both CP/M 80 and CP/M 86 bios, two serial ports, and one Centronics-compatible parallel port.

Plus, it will detect and correct single-bit errors and detect double-bit errors.

There's not another S-100 hard disk controller like it. Not at any price, let alone $595. Or for $2295, we'll include a 19.2 MB Winchester, complete with power supply, cables, and enclosure.

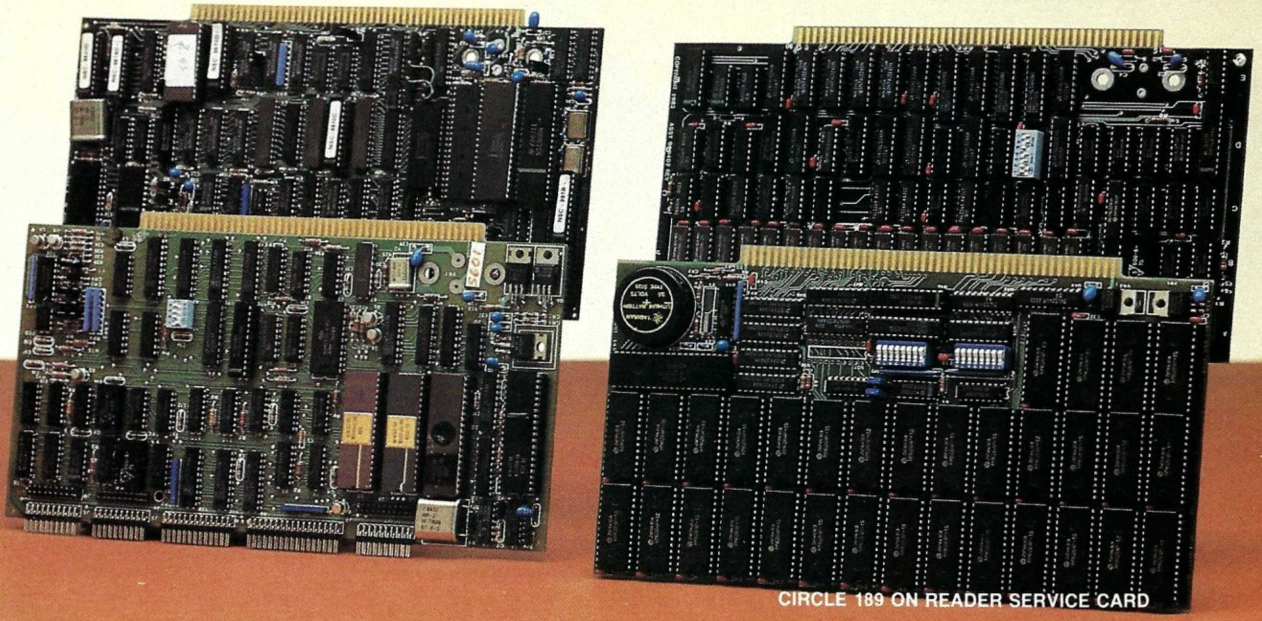
Call us today for all the mouth-watering details. Because with these ingredients, you'll cook up a terrific single- or multi-user system in no time.

A system you can really call your own.

Octagon

2960 North First St., San Jose, CA 95134
408-262-7777

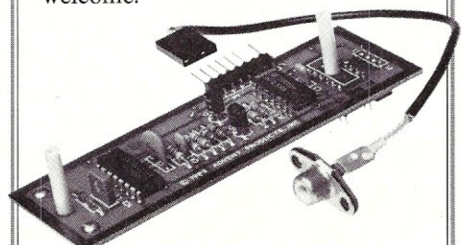
Octagon 8/16 and CPU Board 8/16 are trademarks of Octagon Corp. † CP/M 80, CP/M 86, Concurrent CP/M 86, and MP/M 86 are trademarks of Digital Research, Inc. Z80 is a trademark of Zilog Corp. MS-DOS is a trademark of Microsoft Corp. UNIX is a trademark of Bell Labs.



CIRCLE 189 ON READER SERVICE CARD

New Products
continued . . .

ble, the 12″ keyboard cable is a direct replacement for the original and works with the Kaypro II, 4 or 10. Offered with a money-back guarantee, the 12″ keyboard cable is available for $12.95. Dealer inquiries are welcome.



Advent Products, Inc., *965 North Main St., Orange, CA 92667; (714) 997-0800.*

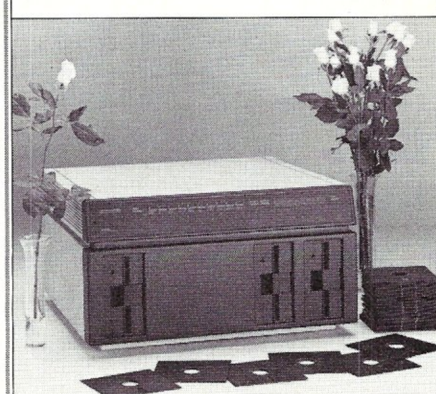
CIRCLE 325 ON READER SERVICE CARD

New office disk duplicator

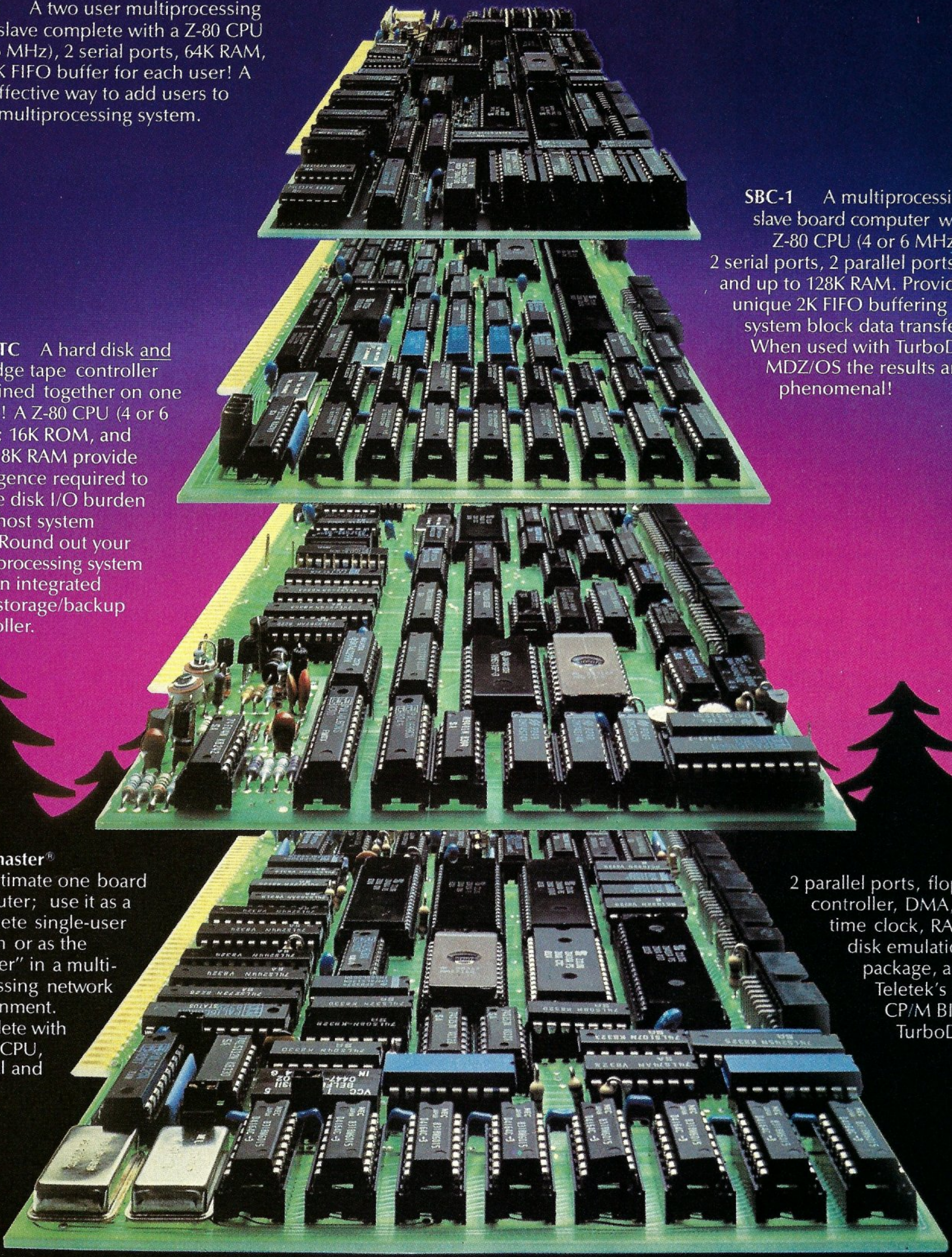
Formaster Corporation has introduced Series Two, a desktop copier that can produce fully verified duplicates at the rate of 120 units per hour. This is about 10 times the speed possible by manual duplication on a personal computer. The self-contained unit includes three disk drives, one for the source disk and two for duplicating. The source disk remains in place until the desired number of copies has been produced on the two "copy drives." Signal lights chart the process from source to completed copies. The unit can copy floppy disks in all popular formats, including IBM, Apple, Commodore, TRS-80 and DEC Rainbow. The Formaster Series Two will be available in February 1984 at $12,700 per unit.

Formaster Corporation, *1983 Concourse Drive, San Jose, CA 95131; (408) 924-1771.*

CIRCLE 328 ON READER SERVICE CARD



Our Family Tree Is Growing Again



SBC-II A two user multiprocessing S-100 slave complete with a Z-80 CPU (4 or 6 MHz), 2 serial ports, 64K RAM, and 2K FIFO buffer for each user! A cost effective way to add users to your multiprocessing system.

SBC-1 A multiprocessing slave board computer with Z-80 CPU (4 or 6 MHz), 2 serial ports, 2 parallel ports, and up to 128K RAM. Provides unique 2K FIFO buffering for system block data transfers. When used with TurboDOS or MDZ/OS the results are phenomenal!

HD/CTC A hard disk and cartridge tape controller combined together on one board! A Z-80 CPU (4 or 6 MHz); 16K ROM, and up to 8K RAM provide intelligence required to relieve disk I/O burden from host system CPU. Round out your multi-processing system with an integrated mass storage/backup controller.

Systemaster®
The ultimate one board computer; use it as a complete single-user system or as the "master" in a multi-processing network environment. Complete with Z-80A CPU, 2 serial and 2 parallel ports, floppy controller, DMA, real time clock, RAM drive disk emulation package, and Teletek's advanced CP/M BIOS or TurboDOS.

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

TELETEK

CIRCLE 174 ON READER SERVICE CARD

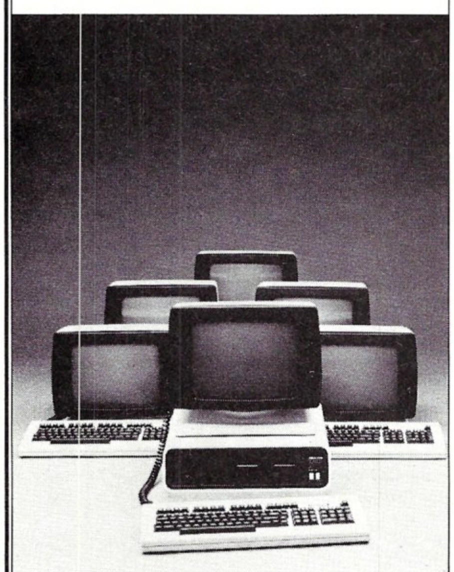
Micro Five Series 1000 multiuser expansion

Micro Five Corporation has released two new models that expand the capabilities of the 16-bit Series 1000 multiuser computer family to up to 80 MB of hard disk storage and 10 users.

The new model 1640 configuration features 40 MB of disk storage, one MB disk, 256K of memory, six I/O ports and 20 MB high-speed streaming tape for backup. Both of these models are fully compatible with the Micro Five Series 1000 users and allow an upgrade path for existing Series 1000 users.

The Micro Five Series 1000 family of multiuser business computer systems provides 16-bit processing power with 512K of main memory and up to 10 I/O ports. Both floppy and hard disk models are available and are fully upgradeable. The Series 1000 supports industry-popular operating systems, including CP/M-86, MP/M-86, SMC Business Basic, MicroCobol, and STARDOS. These operating systems provide the availability of a wide range of field-proven general accounting and vertical market applications software.

The list price of the model 1640, including the operating system, is $16,495. The new model 1740, including the operating system, is priced at $19,995.



Micro Five Corporation, *17791 Sky Park Circle, Irvine, CA 92714; (713) 957-1517.*

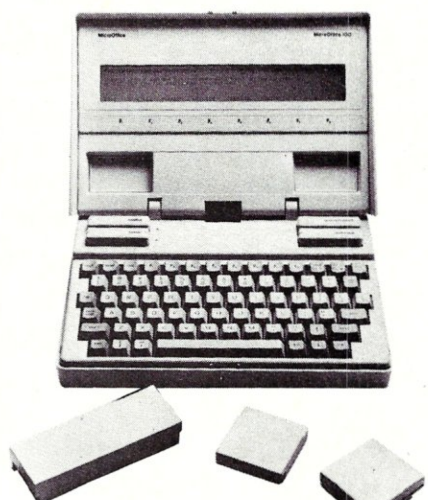
CIRCLE 326 ON READER SERVICE CARD

MicroOffice 100 portable computer: the RoadRunner

The RoadRunner is a battery-powered computer that fits into half of a standard briefcase and measures only $7^3/_4''$ x $11^1/_2''$. The RoadRunner permits users to perform mathematical calculations, analyze financial models, and prepare and transmit reports while on the road. It features a standard-sized, 73-key typewriter-style keyboard with 18 function keys, eight of which are used for single-key menu selection, an eight-line, eight-character-wide liquid crystal display, and 2″ x 2″ removable, reusable data and program cartridges that provide 32K of storage each.

The RoadRunner is equipped with an RS-232C interface and 300 baud (auto-dial, auto-answer) modem as direct sending and receiving links when accessing databases and other computers. Its cover, containing the display, powers the unit when opened. The main memory power supply of the RoadRunner is a removable battery pack that performs for more than eight full hours.

Software packages, now available in cartridges, include text editor, spreadsheet, Basic language, phone list, appointments manager and terminal communications. CP/M software will soon be available on cartridge as well. The RoadRunner comes with 64K RAM and ROM memory and is priced at $1,695. Manual is included.



MicroOffice Systems Technology, *35 Kings Highway East, Fairfield, CT 06430; (203) 367-2525.*

CIRCLE 327 ON READER SERVICE CARD

Hire performance.

Gifford gives you more.

When you go with Gifford, you get more than you ever thought possible from a multiuser computer system. Or a computer company. More productivity. More flexibility. More expandability. More speed. And more support.

It means *total performance* for every Gifford customer.

Gifford systems can run *both* 8 and 16-bit programs, giving you an incredible choice of over 5,000 CP/M® or MP/M™ applications. And we developed this feature, so when we say it really works, it really does.

In addition, our systems can run popular single user programs, while giving you multiuser benefits such as the ability to share resources like printers and hard disks, plus advanced security features to protect sensitive information. And you can run true multiuser programs with features like file lockout, record lockout and shared data bases.

All Gifford systems use CompuPro's proven S-100 based products, making it easy to accommodate any performance enhancements.

For example, expansion's a snap ... literally. Just snap in an expansion board, add a terminal, and you're ready to handle more. And since every system has twenty bus slots, there's plenty of power for everyone.

Gifford keeps on giving.

To make absolutely sure you get all the performance you hire, we support everything we sell. Should you ever need to call us, use our hotline to reach a knowledgeable representative.

And, if the problem can't be solved with words, it can be diagnosed via modem. If the problem is hardware related, we'll replace any defective CompuPro part within 24 hours *free* for two full years. Support continues with our two, three and five day hands-on seminars.

Meet our specs.

Just look at what our popular $9990 three user system includes: 320K static RAM memory, two 8″ DS/DD floppies, the Gifford F5-21 Winchester 21Mb drive, a CompuPro enclosure with IEEE 696/S-100 bus with 20 slots and nine I/O ports. Plus dBASEII,™ SuperCalc-86™ and MP/M-86.™

Other exclusive features include time accounting for users and projects, electronic mail, system scheduler, non-destructive memory test, MP/M II queue calls, and more. As well as expansion capabilities for up to six users.

Make the cut.

If you need our kind of performance, cut the coupon or give us a call. We'll send you a free brochure detailing our exclusive high performance features, our two year warranty, and the benefits of selecting an IEEE 696/S-100 bus-based system.

But if you don't need to hire us, there's only one possible reason. You already have.

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

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

GIFFORD COMPUTER SYSTEMS

1922 Republic Avenue, 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.

IW-P

GIFFORD COMPUTER SYSTEMS ☐ San Leandro, CA (415) 895-0798 ☐ San Francisco, CA (415) 391-4570 ☐ Los Angeles, CA (213) 477-3921 ☐ Miami, FL (305) 665-9212 ☐ Houston, TX (713) 680-1944 ☐ Amherst, NY (716) 833-4758 ☐ Telex: 704521 ☐

CIRCLE 181 ON READER SERVICE CARD

1983 ARTICLE INDEX

Microsystems Mart

ISIS ⟷ CP/M®

Full bi-directional file transfer capabilities are provided in the ISIS-CP/M utilities package. Written in machine language and running under CP/M, these utilities permit the CP/M user to read or write files direct to/from an ISIS Diskette. They will run under any version of CP/M without regard to diskette density. The complete package is $250.00 including user's manual. Write for free brochure on other CP/M software.

CP/M is a registered trademark of Digital Research
ISIS is a trademark of Intel Corporation

SOUTHERN COMPUTER SYSTEMS, INC.
P.O. Box 3373A
Birmingham, AL 35255
(205) 933-1659

The **DELPHI SYSTEM**
15 Megabyte **HARD DISKS**
for
KAYPRO, XEROX 820, Big Board
Televideo and S-100 computers

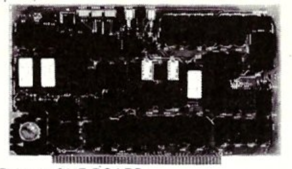
$2195 **complete**

- 15 megabyte hard disk drive with plated media
- switching power supply and enclosure
- controller, cables and CP/M software

DELPHI DEVELOPMENT
6273 19th Ave. NE
Seattle, WA 98115
(206) 524-5369

---

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 282 ON READER SERVICE CARD

ELF—The Statistical Package
Reads All . . . Writes All
For IBM, Apple, and CP/M

FEATURES
• Reads and writes from VisiCalc, Super-Calc, dBase II, DIF, etc. • Factor analysis and principal components • Stepwise regression • Stepwise discriminant analysis • Analysis of variance (1 and 2 way) • Mean, standard deviation, etc. • T-test on means • Crosstabs • Frequencies • Histograms • Create and revise a database • Complete transformation language • Report writer • Software numeric keypad • more . . .

CAPACITY
DATABASE: Observations: unlimited (multi-diskette). Variables: IBM 350+, Apple 250, CP/M 150. ANALYSIS (minimim) IBM 35+, Apple 25+, CP/M 15+. Note: some CP/M machines require smaller capacity version.

PRICE
IBM $350, MS DOS $350, Apple $200, CP/M $200, CP/M 86 coming soon. To order or for further information write or call:

The Winchendon Group, Inc.,
PO Box 10339, Alexandria, VA 22310
(703) 960-2587

MC/Visa

Trademarks: VisiCalc; VisiCorp; dBase II; Ashton-Tate; SuperCalc; Sorcim; ELF; The Winchendon Group, Inc.

Electronic Circuit Analysis

- AC and DC analysis
- Very fast, optimized machine language
- Infinite circuits on multiple passes
- Worst case, sensitivity analysis
- Sweep component values
- 64 Nodes, 127 branches
- Compare circuits
- Log or linear sweep
- Full file handling
- Full editing, error trapping
- Frequency response, magnitude and phase
- Complete manual with examples
- CP/M $150.00
- Now available for MSDOS (IBM PC) $150.00

Tatum Labs
P.O. Box 722
Hawleyville, CT 06440
(203) 426-2184

αβγδεξηθικλμνξοπρστυφχψωΓΔΘΛΞΠΣΤΦΨΩ∞+

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.

αβγδεξηθικλμνξοπρστυφχψωΓΔΘΛΞΠΣΤΦΨΩ∞+

BURN EPROMS USING CP/M*

EPROM PROGRAMMING SYSTEM RUNS UNDER CP/M

PROGRAMS: 2708, 2758, 2716, 2732, 2732A, 2764 and 27128.

– FEATURES –
- STAND ALONE BOARD
- ELECTRONIC SWITCHING OF EPROM TYPES
- ON BOARD SUPPLIES
- 28 PAGE MANUAL
- DESIGNED WITH EASY-TO-GET PARTS
- WORKS WITH ANY CP/M SYSTEM
- USES 24 VOLT XFMR FOR POWER

- INTERFACES THROUGH ONE 8 BIT INPUT PORT AND ONE 8 BIT OUTPUT PORT. 16 WIRES - NO SPECIAL HANDSHAKE LINES.

– CONTROL SOFTWARE COMMANDS –
- PROGRAM EPROM(S) FROM DISK FILE
- PROGRAM EPROM(S) FROM RAM
- READ DISK FILE INTO RAM
- COMPARE EPROM W/RAM
- VERIFY EPROM IS ERASED
- COPY EPROM
- DISPLAY/MODIFY RAM (MONITOR) (MONITOR HAS 11 SUB COMMANDS FILL-DUMP-XFER-MODIFY ETC.)

ALSO AVAILABLE FOR OSBORNE 1 (TM)

BARE P.C. BOARD WITH COMPLETE DOCUMENTATION AND SOFTWARE ON AN 8" SINGLE DENSITY DISKETTE **$69.**

TO ORDER SEND CHECK, MONEY ORDER OR CALL
WRITE OR CALL FOR MORE INFO. 513/752-7218

ADD $3.00 SHIPPING
Ohio res. add 5.5% tax
$3.00 for C.O.D.

VISA OR M.C. ACCEPTED

AndraTech
P.O. BOX 222
MILFORD, OHIO 45150

* CP/M is a trademark of Digital Research

CIRCLE 275 ON READER SERVICE CARD

RAM'S AND EPROM'S

4164	150ns	$4.95
4164	200	4.80
6116P-3	150	4.50
6116LP-3	150	4.75
2716	450	3.15
2732	450	3.95
2764	250	6.00
2764	300	5.75
27128	250	CALL
2532	450	4.25
2564	450	8.50

8088, 8748, 8255, 8253 AVAILABLE

☐ Minimum order is **$25.00.** Prices subject to change
☐ Distributors, Dealers, OEM inquiries invited
☐ !!Call/write for latest price quotes!!

FREE BROCHURE

Excel-Tec Industries, Inc.
Post Office Box 2205
Silver Spring, MD 20902
(301) 434-4123

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 5% 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) 325-5120

1983 ARTICLE INDEX

CONTINUED

1983 ARTICLE INDEX

C O N T I N U E D

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.

SAL/80® and SAL/86™

do for assembly language what RATFOR does for FORTRAN but emits **OPTIMALLY DENSE** code. SAL/8X includes console I/O primitives which trivialize the task of writing complex interactive user interfaces. Improves programmer productivity by a factor of two and program maintainability by an order of magnitude.

Extensively documented, available for all CP/M compatible disk formats. SAL/80 version 2.1, $59.00, requires 64K and MAC or RMAC.

CALIFORNIA RESIDENTS ADD 6% SALES TAX.

PROTOOLS®
"Software Tools for the Professional"
24225 Summerhill Avenue
Los Altos, CA 94022
(415) 948-8007

CIRCLE 278 ON READER SERVICE CARD

PROMPT DELIVERY! AND TRY TO BEAT THESE IC PRICES:

DYNAMIC RAM		
64K	200 ns	$ 5.65
64K	150 ns	5.95
64K	120 ns	6.27
16K	200 ns	1.56
EPROM		
27128	300 ns	$19.20
2764	250 ns	5.99
2732	450 ns	4.25
2716	450 ns	3.60
2532	450 ns	4.75
STATIC RAM		
6264P-15	150 ns	$38.00
6116P-3	150 ns	5.62
6117P-3	150 ns	4.20

Prices 10/18/83

MasterCard/VISA or UPS CASH COD
Factory New, Prime Parts µP∞
MICROPROCESSORS UNLIMITED
24,000 South Peoria Ave.
BEGGS, OK. 74421
(918) 267-4961

Prices subject to change. Please expect higher prices on some parts due to world wide shortages. Call for volume prices. Subject to available quantities. Shipping & Insurance extra. Cash discount prices shown. Federal Express Standard Air @ $5.99! Orders received by 5 PM CST can be delivered to you by the next morning.

---

eXchanger CP/M ⟷ ISIS Emulator

Now move files and programs between your CP/M-80 system and your Intel Series I or II MDS! The ICX package provides complete bidirectional file conversion capability, and even **allows execution** of ISIS-II programs under CP/M using the ISE emulator. The ICX Package is composed of the following two programs:

ICX A Deluxe bidirectional file conversion utility which works with your CP/M system and an 8" floppy drive to provide complete manipulation of an ISIS-II diskette. Takes directories, deletes files, and even initializes a blank disk with the ISIS file structure. Complete C source included. **$89**

ISE An ISIS-II Emulator which allows ISIS programs to run on any CP/M-80 system. Support for all ISIS-II system and monitor calls makes your CP/M micro look like an MDS! Supports banked memory. Complete MAC source included. **$89**

Complete ICX Package (ICX & ISE) **$175**

Supplied on Single Density 8" Disk
CP/M Digital Research, Inc. ISIS-II Intel Corp.

Western Wares
Box C
Norwood CO 81423
(303) 327-4898

CP/M-UNIX COMMUNICATION PROGRAM

CPNIX

- Transfers files between UNIX working directory and any CP/M disk.
- Works with UNIX utilities--no special UNIX program required.
- Terminal mode lets CP/M system act as UNIX terminal.
- Several commands and options include automatic login to UNIX, HELP command, and upper-to-lower case translation.
- Only communication program designed specifically for CP/M and UNIX.
- Entire source code and manual supplied for easy installation.

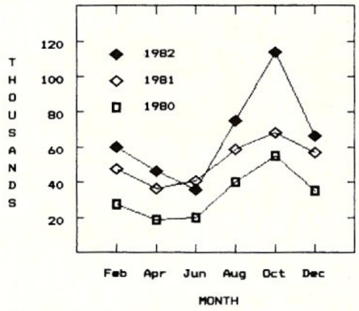
ANSCO
Box 24069
Minneapolis MN 55424
612-927-9018

CP/M is a trademark of Digital Research
UNIX is a trademark of Western Electric

PUBLISHABLE QUALITY GRAPHS on MOST DOT MATRIX PRINTERS

Requires no Graphics Screen
Line Graphs and Scatterplots
Easy to Use, Excellent Manual
CP/M 80, CP/M 86
Most disk formats.
Price: $50. Manual Alone: $10.

DataPlotter from Lark Software
7 Cedars Rd, Caldwell, N.J. 07006
(201) 226-7552 Visa, M/C



CIRCLE 279 ON READER SERVICE CARD

CREATIVE PROGRAMMING OPPORTUNITY

Innovative software company seeks experienced assembly and C language programmers (6502 preferred) interested in creative applications including:

- graphics
- animation
- music/sound effects
- artificial intelligence
- simulations
- systems programming
- and robotics.

Join the people who wrote PAINT and MovieMaker and participate in our growth.

Send resume to:
INTERACTIVE PICTURE SYSTEMS
c/o James Ehlers
2004 Spring Garden Street
Philadelphia, PA 19130

S-100 COLOR

GRAPHICS CONTROLLER BOARD

Using NEC 7220 Graphics Display Controller

One board, single jumper configurable for 8-color mode or monochrome mode. For color, configures to 3 planes of 32K bytes for output to R-G-B TTL monitors. For monochrome to a single large 96K image plane.

For use with standard TTL monitors with separate syncronization; R-G-B or monochrome, 15.75 KHz horiz., 60 Hz vert. A composite output is available on the board for no-interlace monochrome monitors.

Select positive or negative sync - Each bit (pixel) is individually addressable - Built in algorithms for arcs, lines, rectangles, area fill - Light pen input - 1x to 16x zoom - Occupies two contiguous addresses on the system I/O bus - Interlace or non-interlace modes.

Programmable display window: 640x408 interlaced color, or 512x512 interlaced color, or 640x240 non-interlaced color and monochrome, or 640x480 interlaced monochrome, and others.

Using pan techniques enables viewing image areas larger than the display window most likely in non-interlace or monochrome applications, or when zoomed.

$595 This is an extremely versatile, cost effective graphics controller for personal, home, and small business computers.

$460 for partially stuffed board with less memory for monochrome - upgrade later.

PIXELTRONICS

1050 Tulip Way
Palatine, Illinois, 60074
(312) 359-1442

S-100 COLOR GRAPHICS MUSIC SYNTHESIS DUAL I/O PORTS

SINGLE BOARD DESIGN USES ONLY 22 IC'S

COLOR GRAPHICS

-TMS 9918A Video Display Processor -On board RF Modulator
-256 x 192 pixel resolution -NTSC composite video connector
-16 unique colors -1 text and 3 graphic modes
-16K of display RAM (4116's) -32 sprites for 3D simulation

COMPLEX SOUND GENERATION

-AY-3-8910 Prog. Sound Generator -16 level amplitude control
-3 programmable channels -Noise generator
-Produces 3-note chords -Envelope shape cycle control
-Tones from 33.5Hz to 125KHz -Special sound effects

The AMUSIGRAF system is offered as an affordable package which includes the following:

-Bare S-100 P.C. Board -AY-3-8910 data manual
-TMS 9918A data manual -Complete user's documentation
-CP M* compatible 8" disk with test and demo software

ORDER by mail through: FORESIGHT TECHNOLOGY Write or call for more information	ORDER by phone through: ADCOM MKTG. SERVICES (513) 831-1561

AMUSIGRAF SYSTEM
Introductory offer
Visa $95.00 Master Card

Add $3.00 for C.O.D.
Ohio res. add 5.5% sales tax
Send certified check or money order for immediate delivery
Personal checks allow 2 wks.

FORESIGHT TECHNOLOGY
1016 Huffman Ct.
Cincinnati, Ohio 45231

* CP M is a trademark of Digital Research

1983 SOFTWARE DIRECTORY

SOFTWARE	Vol.-No.-Pg.
8086 OS Converter	4-11-122
ACOM (telecom for N*)	4-2-91
ASCOM (Telecom)	4-7-122
BackRest (hard disk archiving)	4-5-139
BAKUP	4-10-134
BOBCAT (file cataloger)	4-4-107
BRAINSTORMER	4-10-142
CERES I	4-7-121
COMSTAR Basic compiler	4-1-98
CP/M recovery	4-10-136
CPMGREP	4-4-106
Database	4-11-122
DECISION-ANALYST	4-10-142
DES-Crypt	4-4-106
DMM1 utilities	4-1-101
DR Graph	4-10-140
EIS-110 (realtime OS)	4-11-128
EPM (EPROM programming)	4-1-98
FILEBASE	4-8-122
FILEFIX	4-1-99
FilePlan	4-8-124
Financial Utility I	4-12-145
Font-19	4-8-125
GrafTalk 2.0	4-6-128
GRAM-A-SYST (PC to Telex & TWX)	4-10-135
GRAPH (for Diablo w/Hyplot)	4-3-106
HASTE (Telecom w/HASP protocol)	4-7-121
HDIS (8080 disassembler)	4-7-121
HexPrintR (modifier to WS)	4-4-106
ICT 1.0 (Isis to CP/M)	4-10-135
IM/80 Information Manager	4-5-138
InfoShare	4-10-134
International Software Directory	4-4-107
INTRCEPT (CP/M call handler)	4-3-106
KBasic	4-8-123
KEYDISK (data entry)	4-7-124
LOCKPROG (security)	4-12-144
MagicIndex	4-12-146
MAILER	4-9-122
Mailer	4-8-122
Master Catalog	4-1-98
MCS-3 (mfg control system)	4-10-134

SOFTWARE	Vol.-No.-Pg.
MemoPlan	4-8-124
MIS (medical info svc)	4-10-140
MODEM 8-16	4-6-128
Modula-2 (compiler)	4-9-124
MTRANS (MBasic/Oasis)	4-11-122
MultI/NET	4-4-106
NAMOR (mail list)	4-7-121
Nevada FORTRAN	4-1-98
Optimizer (financial modeling)	4-2-90
Personal Finance Utility	4-3-107
Printing Package for MX-80	4-2-91
QUESTEXT III (text info mgr)	4-10-136
Quic-n-Easi AG	4-8-122
Quick Check (money management)	4-1-99
Real-Time C Runtime Library	4-12-146
Relocation Utilities	4-7-126
RHESUS (file recovery)	4-5-138
SCREEN MASTER	4-8-127
SDAM/II (data mgmt)	4-7-125
SM-MODEM (telecommunications)	4-2-90
Small/C & Small/VM	4-1-98
SOFTPLOT/BGL	4-8-122
SUB:MATH	4-9-122
Super-19	4-8-126
SUPERFILE	4-7-124
SYSTEM-TMS7 cross assembler	4-2-90
Tarbell CB-80 Database System	4-11-122
Tarbell Database System	4-4-106
TaxCalc	4-10-139
TECHTYPE	4-10-137
TLX-A-SYST (PC to Telex telecom)	4-10-135
TRAKSPLAYER and TUNINGS	4-11-122
UNE/CON (file recovery)	4-10-138
VersaForm	4-3-107
WASH (file utility)	4-5-138
WES word processor	4-2-90
Wiremaster	4-1-100
WS-Patch	4-11-127
Z-COM (BaZic to ASM)	4-7-124
Z80ASM	4-10-134
ZAS Z-8000 (assembler)	4-5-138
ZSTEM (terminal emulator)	4-9-122

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION *(Required by 39 U.S.C. 3685)*

1. Title of Publication: *Microsystems*.
   a. Publication No. 01997955
2. Date of filing: October 1, 1983.
3. Frequency of issue: Monthly.
   a. No. of issues published annually: 12
   b. Annual subscription price: $24.97
4. Complete mailing address of known office of publication *(not printers)*: One Park Avenue, New York, NY 10016.
5. Complete mailing address of the headquarters or general business offices of the publishers *(not printers)*: One Park Avenue, New York, NY 10016.
6. Full names and complete mailing addresses of the publisher, editor, and managing editor: Publisher, None. Editor, Sol Libes, One Park Avenue, New York, NY 10016. Managing Editor, None.
7. Owner: Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016; Ziff Corporation, One Park Avenue, New York, NY 10016.
8. Known bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages or other securities: None.
10. Extent and Nature of Circulation

	Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies of Single Issue Published Nearest to Filing Date
A. Total no. copies *(net press run)*	51,651	65,800
B. Paid Circulation		
1. Sales through dealers and carriers, street vendors and counter sales	17,032	20,900
2. Mail subscriptions	20,597	27,800
C. Total Paid Circulation *(sum of 10B1 and 10B2)*	37,629	48,700
D. Free distribution by mail, carrier or other means samples, complimentary, and other free copies	2,544	3,284
E. Total distribution *(sum of C and D)*	40,173	51,984
F. Copies not distributed		
1. Office use, left over, unaccounted, spoiled after printing	1,792	1,716
2. Returns from news agents	9,686	12,100
G. Total *(sum of E, F1, and 2—should equal net press run shown in A)*	51,651	65,800

11. I certify that the statements made by me above are correct and complete.

WILLIAM L. PHILLIPS,
Assistant Treasurer

classified ads

CLASSIFIED RATES: Per Word, 15 Word Minimum. REGULAR: $1.35. EXPAND-AD: $2.03. GENERAL INFORMATION: Prepayment discounts available. Payment must accompany order except credit card — Am. Ex., Diners, MC, VISA (include exp. date) — or accredited ad agency insertions. Copy subject to publisher's approval; must be typewritten or printed. First word set in caps. Advertisers using P.O. Boxes MUST supply permanent address and telephone number. Orders not acknowledged. They will appear in next available issue after receipt. Send order & remittance to: Classified Advertising, MICROSYSTEMS Magazine, 1 Park Avenue, New York, NY 10016. Direct inquiries to (212) 725-3927.

SOFTWARE

MCL—a music compiler. Runs Digital Research (of Texas) Stereo Sound Boards. Easy sheet music translation, sound effects, more. Requires 48k-TPA, CP/M-2.2. $25.00. 8-inch SSSD. Howard Peters, Box 3324, Rockford, IL 61106.

SPEAKER SYSTEM DESIGNERS! Analyze your woofer/cabinet/port/horn combinations and get response curves on your printer. CP/M diskette 8 inch SS/SD or 5 inch SS/SD for Kaypro, Osborne, Morrows, or Xerox. $39.00 diskette and booklet includes UPS 48 states. ($10.00 Foreign shipping). Prepay with check, Visa, MasterCard, or American Express. SONIX CO. INC., Rt. 2 Box 163-G, Indian Head, MD 20640 (301) 753-6432.

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.

FOR SALE

MAXELL DISKES-Single sided: $28. Double sided: $45. Write: Deco Software. 420 South Beverly Dr., Beverly Hills, CA 90212; or call: (213) 275-1403. Ask about other great prices!

BUSINESS OPPORTUNITIES

HACKER? PUBLISHER SEEKS micro computer experiences in hacking/crime/society twisting. Pay if used. AW Box 5115, Tahoe City, CA 95730.

YOUR Expand-Ad® is more outstanding on a BEN-DAY background like this! Call (212) 725-3927 for more information.

ADVERTISERS INDEX

Four Answers To Your S-100, Multi-User Problems.

Intercontinental Micro Systems makes everything you need for S-100 bus multi-user systems, networks or single user systems.

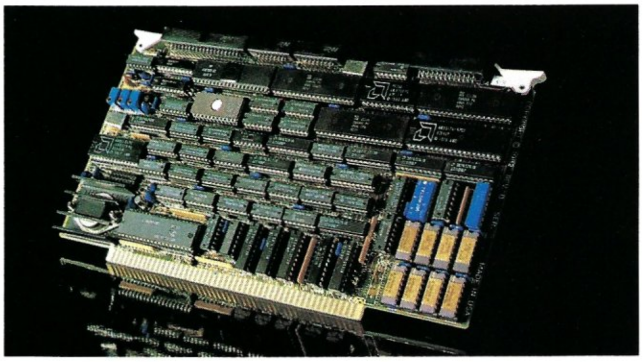
At a price that won't break your budget.

Quite simply, our single board computers, slaves, 256K memories and personality boards let you build a system *now*, not later. The hardware works, the software works, and the prices are what you'd expect from a company that uses the most advanced design, software and production techniques to keep costs down.

What you won't expect is the almost awesome sophistication of Intercontinental Micro System's products.

So stop messing around with multiple sourcing, hardware integration problems and software nightmares. Come to Intercontinental Micro and get it all — price, performance and delivery.

Read the specs, then call, write or circle the bingo number below. We'd be glad to send more information and help solve your S-100, multi-user system problems.

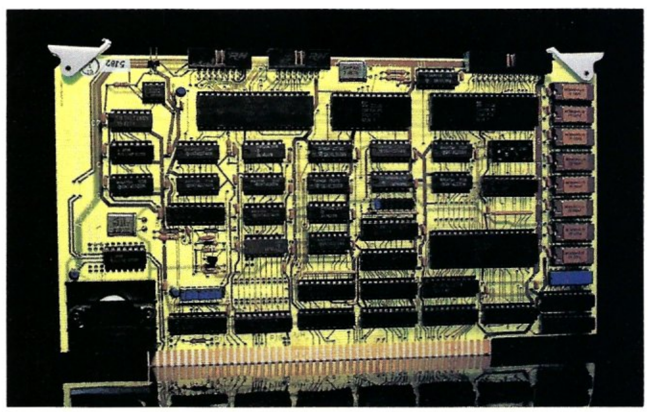


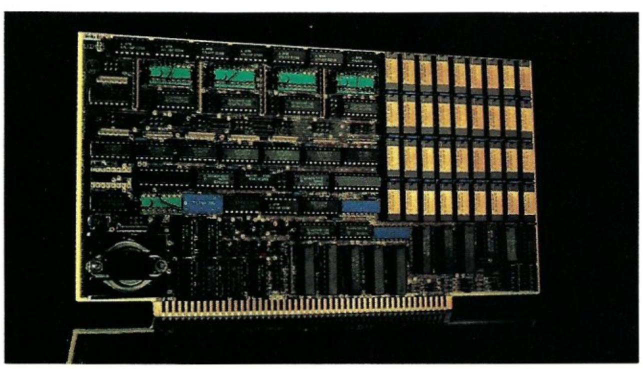
1 CPZ-48000 SINGLE BOARD COMPUTER.

□ IEEE 696.1/D2 S-100 compliance. □ Z80A™ 4MHz Operation. □ Floppy disk controller (FDC). Single or double sided. Single or double density. 8" or 5¼". □ Two synchronous or asynchronous serial I/O channels (SIO). □ Two parallel I/O channels (PIO). □ Four channel DMA controller. □ 64K on board RAM. □ Memory management unit (MMU). Addresses up to 16 megabytes of system memory. □ Eight Vectored priority interrupts. □ Provisions for 2K or 4K onboard EPROM. □ Software selectable baud rates. □ IBM Bisync, HDLC, SDLC and other protocols. □ CP/M,™ MP/M,™ and TurboDOS™ operating systems available. □ Turbo-Disk® implementation included.

2 CPX-MX SLAVES.

□ IEEE 696.1/D2 S-100 compliance. □ Compatible with CPZ-48000 SBCP, any Z-80A based CPU with extended address capability or 16 bit based CPUs complying with IEEE 696.1/D2 bus specification. □ Z-80B™ 6MHz (CPS-6X) or Z80A 4MHz (CPS-4X) operation. □ Two synchronous (CPS-MS) or asynchronous (CPS-MA) serial I/O ports. □ TurboDOS™ & CP/NET™ compatible. □ Master confiscation of slave memory for diagnostic purposes. □ Two parallel I/O ports; eight data bits + 2 handshake lines per port. □ 64 Kbytes of onboard dynamic RAM. □ Master/slave memory-to-memory transfers under DMA control @ 571 Kbyte/sec transfer rate when used with CPZ-48000 SBCP. □ Software selectable baud rates. □ Usable as an intelligent I/O processor in single user system.



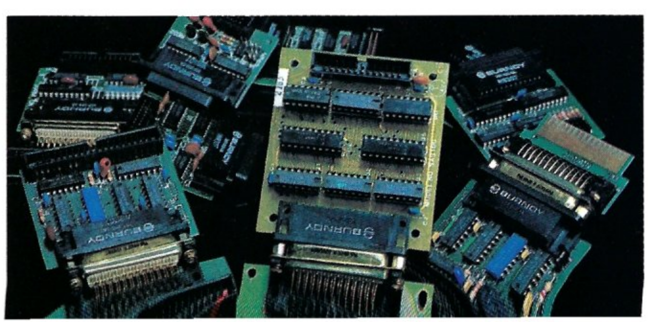


3 256KMB-100 256K MEMORY.

□ IEEE S-100 bus, spec 696.1/D2 compliance. The 256KMB-100 is compatible with most IEEE S-100 board products now on the market. □ Linear addressable to 2 megabytes. □ 225 nano-second access time, maximum, 160 nano-seconds, typical. □ 295 nano- second read-write time, minimum. □ Bank selectable 16K increments. □ I/O port address bank selection. □ Configures for phantom deselection. □ Parity error detection, visual and/or interrupts. □ Bank selection compatible with CROMIX,™ CP/M2.2,™ MP/M,™ Alpha Micro, and other major systems.

4 PERSONALITY BOARDS.

□ Centronics printer. □ 8 inch floppy disk. □ 5¼ inch floppy disk. □ RS232 serial communications. □ Synchronous/asynchronous modem. □ Priam smart/smart E hard disk. □ Long distance serial communication (2000 ft @ 9600 baud). □ Shugart Associates Systems Interface (SASI). □ Clock/calendar. □ Konan David, Jr.™ hard disk. □ Archive tape drive.



4015 Leaverton Court