THE CP/M USER'S JOURNAL

$2.95

Microsystems

Volume 4 / Number 10

October 1983

Local Area Networks—an alternative to multiuser time-sharing systems?

Local Area Networks

An Introduction to Local Area Networking
A Review of Digital Research's CP/NET
Build a simple low cost/low speed Local Area Network

A Step Toward the Wrist-Watch Communicator

Don't read on the train! Write an article for us on your portable Radio Shack Model 100—then transfer the article into your CP/M system for final editing.

"Bubble, Bubble, Toil and Trouble?"

Not any more! Intel has taken away the toil and trouble. Randy Reitz shows you how to build a simple, inexpensive S-100 bubble memory, using an Intel subsystem for inexpensive, nonvolatile disk buffering or program storage.

Product Reviews

S-Basic . . . A new interpreter for CP/M systems
QBAX . . . A CP/M file backup utility.

Other Feature Articles

A Machine Code Loader for Microsoft Basic
How to Hide Machine Code in REM Statements
How to get a 61K North Star System
How to Set Printer Options from a Menu
A Z80 Random Number Generator That Really Works

Microsystems Tutorial

GRAPHICS $975

VectorScan 512

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

Features include:

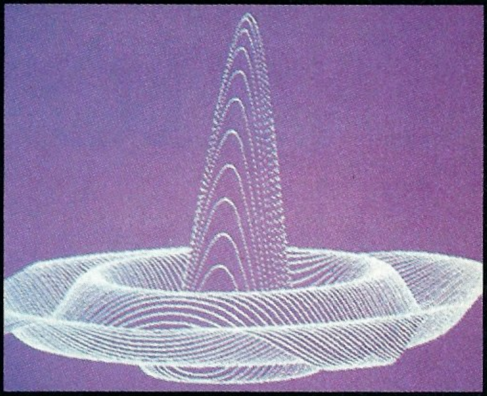
- High Resolution 512 × 480 Pixels × 16 Colors/Intensities.
- Color Lookup Table for each of the possible 16 pixel levels.
- Internal Character Generator for Horizontal and Vertical axes.
- Simple ASCII command structure allows system independence.
- Internal Line, Arc, Circle, Point, Programmable Shape 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.

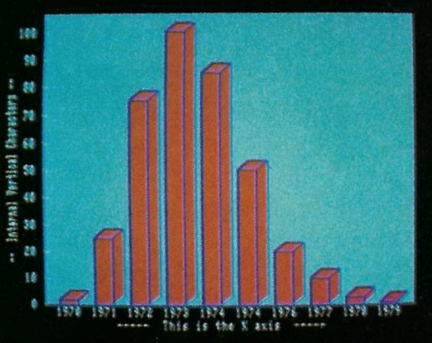
Applications:
CAD/CAM, Medical, Business, Slide Generation, Industrial Process Control, Laboratory Data Display, Mapping, Meteorological Studies, Graphic Artwork, Typesetting, CRT Strip Chart Recorders, Printed Circuit Artwork, Educational Graphics, Mathematical Modeling, Engineering, Bulletin Boards.

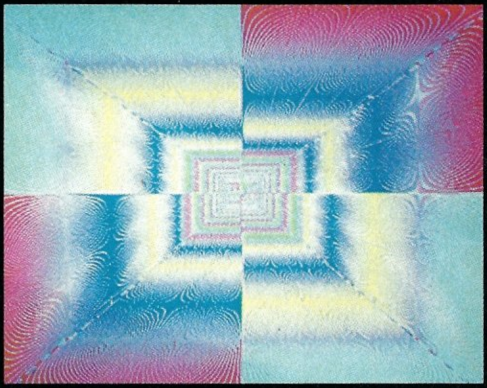
Computers:
IBM-PC, Apple, Epson, Morrow, North Star, Cromemco S-100, Radio Shack, Health, DEC, and most others.

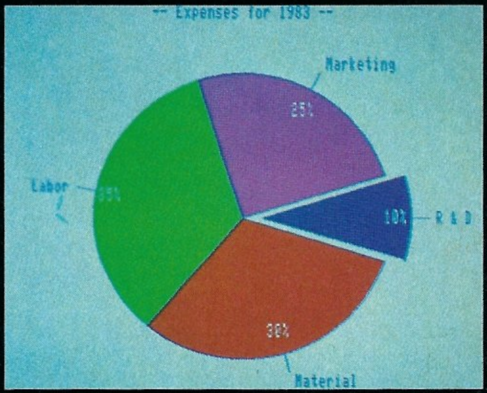
Operating Systems:
CPM-80, CPM-86, PC-DOS, MS-DOS, UNIX, OASIS, RT-11

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





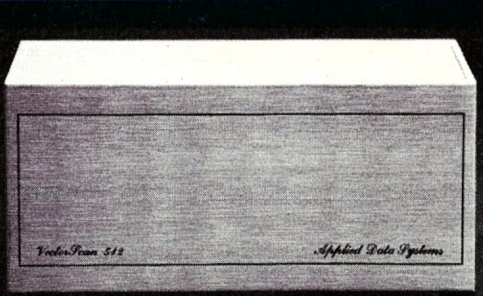




PRICES

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

Nine Thousand Nine Hundred & Ninety Dollars.

3 users. 21 Mbytes Winchester. 320K RAM. And you can only get it from Gifford.

Look at that price again. We were a little skeptical at first, too. But after checking out totals we knew we'd finally done it: developed a multiuser, 8-16 bit high performance, obsolescence-proof computer. Complete with dBase II,™ SuperCalc-86,™ and MP/M-86™—for under $10,000. Expandability's a breakthrough, too. You can add four additional users for just $600 each, plus terminal.

Is it 8 bit or 16 bit? Yes.

Like you would expect of a Gifford computer system, it can run any combination of 8 and 16 bit CP/M® programs simultaneously. So, on top of all the thousands of 8 bit CP/M programs available, you'll also be able to take advantage of any of the faster 16 bit programs... and run them at the same time. And when we say it works, it works. After all, we were the folks who invented 8-16 operating systems.

Stop looking. And start cutting.

Send us the coupon and we'll send you a brochure detailing *all* the advantages of going with Gifford. Such as the Gifford disk emulator (which *Byte* magazine called "One of the best things that happened to microcomputers in 1982."), our exclusive networking and multi-tasking telecommunications packages, and our two-year warranty. Plus the benefits of selecting an IEEE 696/S-100 bus-based system.

There's lots more to tell, but if you're a business or technical professional looking for a high performance, multiuser computer that features expandability and dependability, plus complete service and support at an incredibly low price, stop looking and cut the coupon. And by all means, wait for our brochure before you make any investment.

Gifford Computer System 321 Includes:

320K Static RAM Memory

Two 8″ DS/DD Floppies 2.4Mb Formatted

Gifford F5-21 Winchester Drive 21 Mb Formatted (27 Mb Unformatted)

CompuPro® Enclosure 2 With:
IEEE 696/S-100 bus
9 I/O Ports
20 bus Slots

dBASE II

SuperCalc-86

MP/M-86

Terminals not included. Compatible with all standard RS-232 terminals and printers.

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

Terminals not included.

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________

M

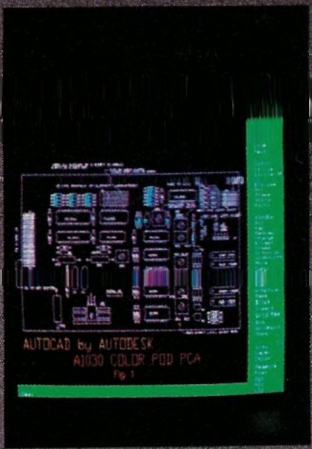
GIFFORD COMPUTER SYSTEMS □ SAN LEANDRO, CA (415) 895-0798 □ SAN FRANCISCO, CA (415) 391-4570 □ LOS ANGELES, CA (213) 477-3921 □ MIAMI, FL (305) 665-9212 □ HOUSTON, TX (713) 877-1212 □ AMHERST, NY (716) 833-4758 □ TELEX: 704521 □

CIRCLE 181 ON READER SERVICE CARD

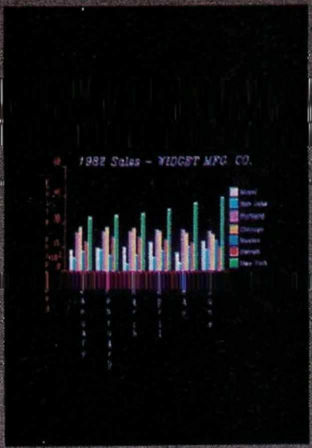
GDL's A-1000™

Why Pay More For Less?

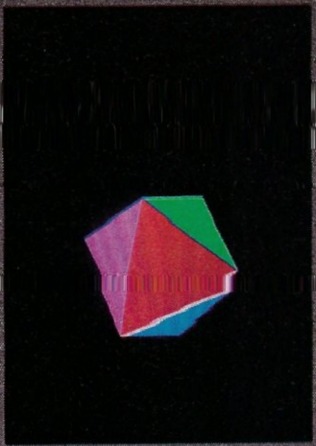
N.M.R. Brain Scan
J. Libove, Dual Systems



Autocad by Autodesk
R. H. Hymes, GDL

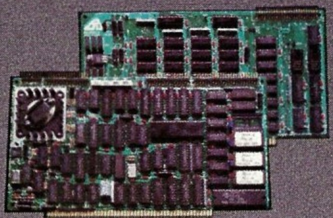


Supercalc Data Graph
Pacific Basin Graphics

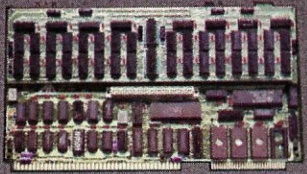


Octehedron With Hidden
Surfaces Removed
Dr. C.L. Morgan,
Calif. S.U. Hayward

- 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 com-

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

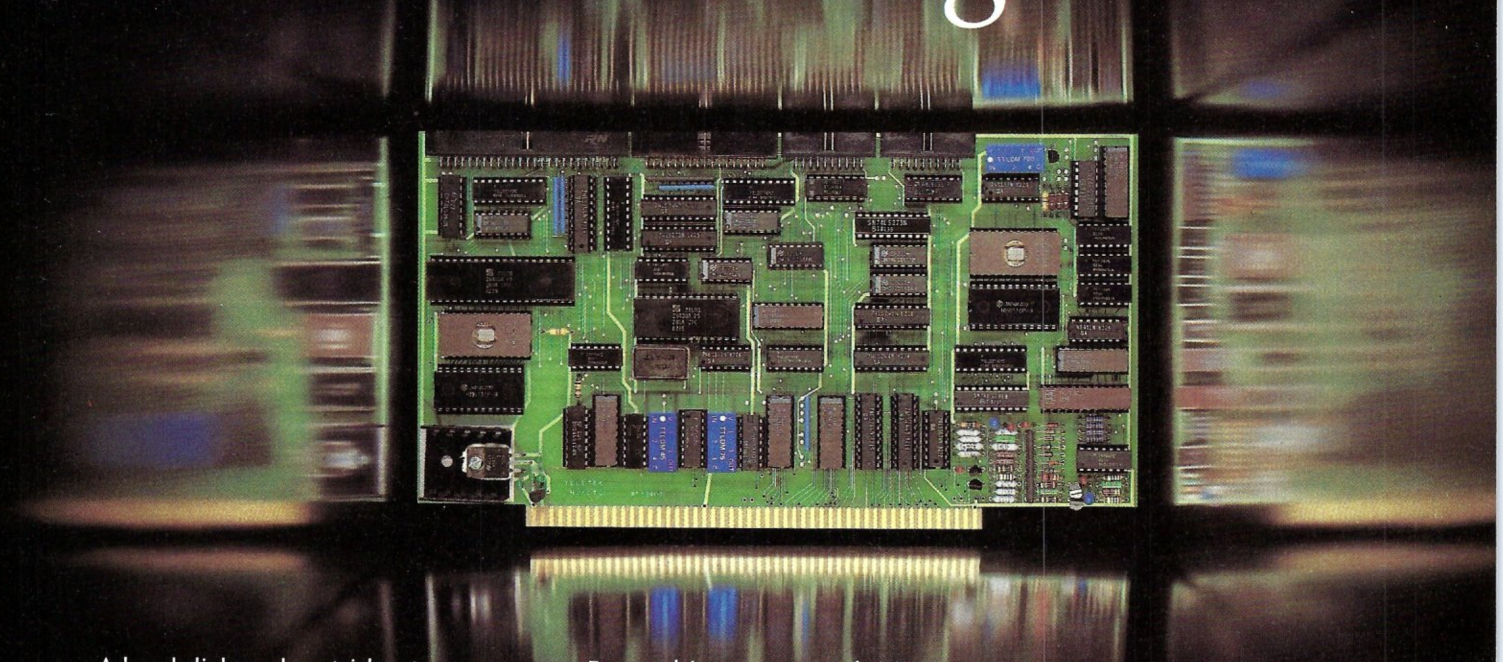
W. W. COMPONENT SUPPLY
1771 Junction Ave.
San Jose, Ca. 95112
(408) 295-7171

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

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



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

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

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

TELETEK

Contents

Microsystems

Volume 4/Number 10
October, 1983

DEPARTMENTS

Microsystems

STAFF

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

ADVERTISING SALES OFFICES

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

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

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

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

Southern California, Southwest
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 Leder, *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	J. Scott Briggs
Vice President Circulation	Carole Mandel
Vice President General Manager	Eileen G. Markowitz
Creative Director	Peter J. Blank

ZIFF-DAVIS PUBLISHING

Richard P. Friese, *President* Albert S. Traina, *President, Consumer Magazine Division* Furman Hebb, *Executive Vice President* Paul H. Chook, *Executive Vice President, Marketing and Circulation* Phillip T. 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.

Editor's Page

by Sol Libes

Most of the very early developments in the personal computer industry are the result of work done by computer amateurs. Therefore, for this issue, which emphasizes networking, I would like to take a look at what amateurs are doing in the networking area.

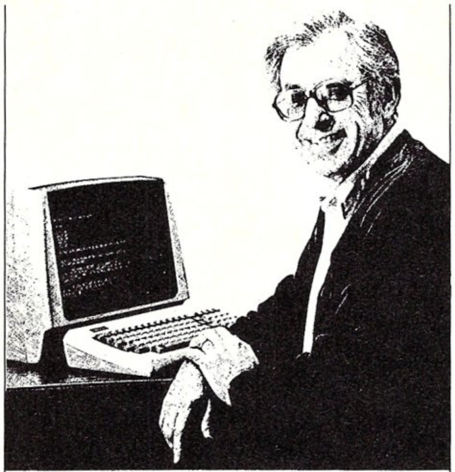
The Amateur Radio Research and Development Corporation (better known as AMRAD), located in McLean, Virginia, is a group of experimenters that has been around since about 1973, involved in digital radio communications. They are leading a coordinated effort to develop computer networking via radio.

Bulletin boards

Several years ago, computer hobbyists began creating Computerized Bulletin Board Systems (CBBS) that allowed their computers to communicate via modem devices and the telephone line. This movement has grown and spread across the country. The CBBS allows computerists to communicate with one another via a bulletin board type of operation. A serious effort was made to link together many of the CBBSs, using different machines, in a network arrangement under the banner of "PCNET." However, very few of the CBBSs participated in this effort because of the high cost of the telephone line and the question as to who was to pay for message transfers between CBBSs in the network.

RTTY systems

Amateur radio operators were communicating with one another using the airwaves and teletypes connected to transceivers. These systems used a 5-level code known as "Baudot" and became known as "Radio Teletype" or "RTTY" systems. When personal computers became available, hams started replacing their teletypes with personal computers in these systems. Today an estimated 80% of the over 70,000 hams are using personal computers in these RTTY systems. Several companys make and sell interfaces for personal computer/transceiver systems. Generally these systems are on higher frequency



bands and are used in the chat mode.

A few years ago the Federal Communication Commission changed the rules to permit the use of the ASCII code in these systems. Hams disconnected 300-baud modems from their telephone lines and instead connected them to their transceivers. Immediately problems started to develop. Protocol problems developed as to the use of originate/answer tones, especially in connections that involved more than two hams at a time. The problem was resolved by the use of one set of tones instead of the two sets of tones used in telephone modem use. Additionally, performance was found to be impaired when noise was encountered during program transmission.

Packet protocols

These problems were attacked by the adoption of bit-oriented protocols (called "packet"). Actually, such protocols had been developed earlier as part of the Ethernet and NBSnet systems. A terminal node controller for packet radio was developed. In this system, short messages called "packets" are sent. A personal computer could be used to convert the standard serial transmission into these packets at the transmitting end, and the opposite is done at the receiving end. The ham radio operators worked at developing the protocols and standards for these transmissions. The development of a standard node controller meant that interfacing of different personal computers became easier. Thus, at this point in time, both the node controller and personal computer software approaches are being used.

The other thing that hams have introduced is the use of repeaters. Hams had already created FM voice repeaters, with an estimated 3,000 in current use. They are all over the country and virtually within reach of all ham operators. The first packet repeater went into operation in Vancouver, Canada, followed by one in San Francisco about two years ago. AMRAD put one into operation shortly afterward on their 2-meter FM repeater. Both voice/digital and digital-only repeaters are in operation today. However, the trend today is to set up repeaters exclusively for digital use; some people call them "digipeaters."

In 1981, the packet repeater developers held their first conference, sponsored by the ARRL at the National Bureau of Standards in Washington, D.C.; 85 people showed up for the gathering.

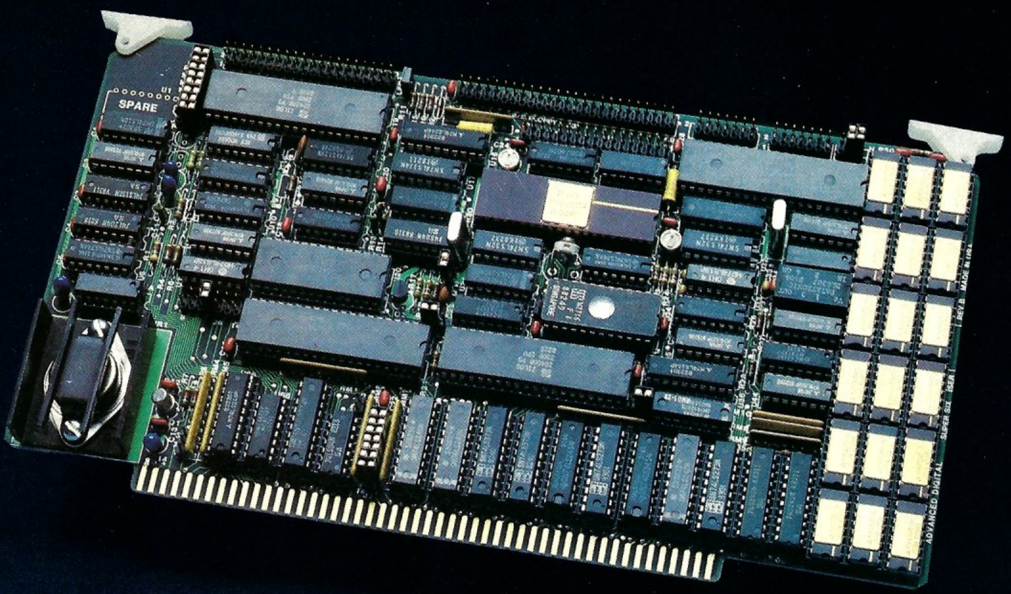
Currently there are several hundred amateurs testing different types of packet controllers and associated software. This beta test effort has now been going on for close to two years, and most of the bugs have been wrung out of these systems. This testing period is just about over, and the hardware and software should shortly become available for general use.

"Internetworking"

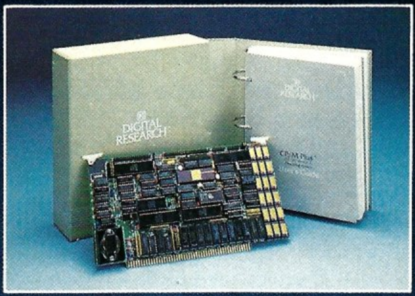
Work is being done in the development of protocol standards for the local networks and for networks of local networks, or what some call "internetworking." Internetworking is still in a very early stage of development. Radio Amateurs have already put several satellites into orbit, and it is likely that this will be used as the transmission path for internetworking. A special satellite specifically for packet radio work is already in the development by a group of amateurs.

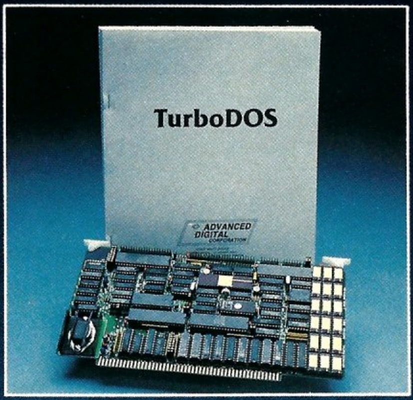
The amateur radio operators are already looking forward to the development of an international packet radio network that will link computer users throughout the world.

SUPER SIX™



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





SUPER SIX FEATURES:

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

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

SUPER SIX & CP/M™ 3.0 A PERFECT MATCH

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

SUPER SIX & TurboDOS™ ANOTHER PERFECT MATCH

When you combine the TurboDOS multi-user operating system with the 6 MHz SUPER SIX, you'll find your system running 1-1/2 times faster than before. Add the 4 MHz or 6 MHz SUPER SLAVE™ processor board(s) and you will have the fastest multi-user, multi-processor system available today.

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

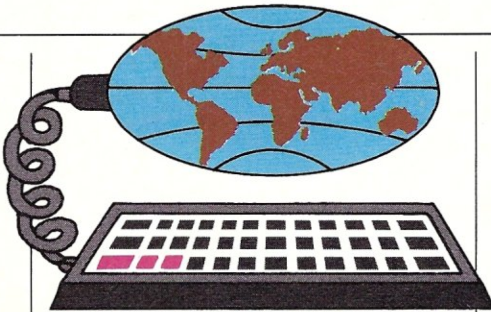
ADVANCED DIGITAL CORPORATION

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

CIRCLE 148 ON READER SERVICE CARD

News and Views

by Sol Libes



Random rumors

Intel is rumored to be working on putting UNIX on a chip (ROM) for their 80286 microprocessor the same way they did it for CP/M. Word is that the actual porting is being done by Microsoft. And you can also expect a 68000 ROM version from Motorola (rumor is that the porting is being done by UniSoft) and a 16032 version on a chip from National with the port being done by HCR . . . More rumors are seeping out about Intel's new 32-bit micro called the 80386. Look for it to be upward compatible with the 80286 so that it can run 8086 software. It will have a writable control store that allows you to design your own instructions in microcode. It should have a 32-bit segment address as well as 32-bit instruction address. It will have paging support.

Public domain software releases

The C User's Group has released five new volumes of public domain software. First is a "small tex" package designed for use with an Epson MX-80 printer and the Fancy Font program from SoftCraft. Second are two volumes of utilities and games. Third is a disk containing tools transliterated from RATFOR to BDS C. Last is a disk of patches and utilities for users of Mince and Scribble (Amethyst word processor). CUG also publishes an irregular newsletter (last issue was 16 pages). For information on obtaining this software or joining the group write: CUG, Box 287, Yates Center, KS 66783 or call 316-625-3554.

The $600 CP/M system

Spectravideo, a game manufacturer who previously introduced a low-cost home computer, has introduced a second system using CP/M (in ROM) as its operating system. For only $600 you get a Z80A-based system (3.6 MHz) with 48K ROM and 80K RAM (expandable to 144K), Microsoft Basic and a ROM cartridge slot. There is also a word processor and terminal program in the ROM. The keyboard has 87 keys plus 10 function keys. It produces a color display (40 columns × 24 lines) and has sound capability. The display output is composite or TV modulated. The unit has cassette I/O.

New user group

The Australian North Star Users Association has been formed. For more details contact: ANSUA, P.O. Box 194, Wangaratta 3677, Australia.

An independent user group for Victor 9000 users has been started. For information send a stamped self-addressed envelope to: SIVic-9000, 362 Peachtree Ave, NE, Atlanta GA 30305.

Western electric enters software business

Western Electric has introduced its first two software application packages. Naturally they run under UNIX. WE's only previous software effort was to license UNIX to OEMs. Now WE is expected to be a powerful force in the UNIX software marketplace.

The packages are called UNIX Writer's Workbench and UNIX Instructional Workbench. Both run under UNIX version V, are basically word processor packages, and sell for $4,000 and $2,500 respectively.

S-100 standard available

The Institute of Electrical and Electronic Engineers has finally published the S-100/IEEE 696-1983 bus standard. The 40-page document can be obtained from the IEEE Computer Society Order Dept., PO Box 80452, Worldway Postal Center, Los Angeles CA 90080. Price is $6.75 for IEEE members or $7.50 for non-members, plus $2.00 shipping and handling (California residents add 6% sales tax). The document can also be obtained from the IEEE Service Center, CP department, 445 Hoes Lane, Piscataway NJ 08854 (NJ residents add sales tax).

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

Intel announces 32-bit wide multibus

Intel has announced that it is developing a 32-bit wide version of the Multibus to be called "Multibus II." Multibus is one of the most popular microcomputer buses along with the S-100 and VME bus for the 68000. Multibus was created in 1976 as an 8-bit bus system and later upgraded to a 16-bit wide bus.

The latest upgrade is no doubt made in anticipation of Intel's expected introduction of a 32-bit version of its 80286 microprocessor. The new version of Multibus is also expected to support several hierarchies of data transfers as well as a hierarchial processor structure for multiprocessor operation.

Intel is also rumored to be working on a VLSI chip that implements all of the bus interface circuitry for Multibus II and a second chip that will interface the Multibus II to the new desktop VAX rumored under-development by DEC.

The VME bus for the Motorola 68000 16-bit processor already has 32-bit wide capability and Motorola is known to be working on a true 32-bit wide version of the 68000.

THE EASY CHOICE

Best Separate Compilation — Best Error Handling — Best Implementation on a Small Computer

Comments From 1983 LA AdaTEC Compiler Faire

". . . JANUS/Ada encompasses at least 5 times as much of Ada as Supersoft, including many of the more exotic features of the language." Creative Computing

". . . The compiler performed exceptionally well . . . the linker performs flawlessly . . . the error handling is excellent . . . RR's support is the best I've ever encountered." InfoWorld

"The compiler breaks new ground in the microcomputer field with its excellent runtime error-checking code and its excellent compiler error messages." Microsystems

"They're (RRS) honest to a fault, so they call their language JANUS; but its more nearly Ada than any other microcomputer implementation I know of . . . I watched some of the demonstrations and it works."
Jerry Pournelle, Byte

Encouragement of this kind deserves a just reward; we now intend to complete our compiler and submit it for validation.

Our Ada line is available on the following operating systems: CP/M, CP/M-86, MS-DOS, PC-DOS and CCP/M-86

Available from the following distributors:

Westico, Inc.
25 Van Zant, St.
Norwalk, CT 06855
(203) 853-9880

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

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

Internationally

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

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

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

RR SOFTWARE, INC.

specialist in state of the art programming

P.O. Box 1512 Madison, Wisconsin 53701

(608) 244-6436

CIRCLE 190 ON READER SERVICE CARD

VEDIT

CUSTOMIZABLE
FULL SCREEN EDITOR

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

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

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

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

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

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

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

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

COMPARE VEDIT'S FEATURES

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

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

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

CompuView
PRODUCTS, INC.

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

CIRCLE 35 ON READER SERVICE CARD

In the Public Domain

by Chris Terry

This month I shall be talking about disk utilities and diagnostics in the public domain. There are many of both, and it is not always clear how they should be classified. For this issue I shall be considering only programs that give the user direct access to data stored on the disk as an aid in reconstructing damaged files, speed up disk accesses, or test the manner in which data is stored with a view to finding bad spots on the medium or hardware malfunctions. Utilities that are primarily concerned with file management and archiving will be discussed in a future installment of this column.

Disk viewers

One of the most reliable of the early disk viewers was S.J. Singer's DUMP (CPMUG Vol. 24). A .COM file is supplied on the disk as well as the source code, which requires the Digital Research macroassembler MAC and the MACRO.LIB library to assemble. The user can select a file, a block, or a sector on a specified track at which to start viewing. The display is in hex, with ASCII equivalents alongside, as in DDT. An address offset from the start of the sector is also shown.

With a sector image displayed on the screen, the user can edit the image (using hex values) and can then write the patched image back to the same location on the disk. A minor disadvantage is that you can scan only forward from the sector at which viewing started. You can also display the allocation bit map of the disk instead of a sector, but you cannot change it.

An updated and enhanced version of DUMP was contributed by Software Tools of Australia and is available as DD6 in SIG/M Vol. 73. Some of the routines are hardware-dependent, but instructions for customizing these are given in the documentation.

The most powerful and comprehensive disk view/patcher is DU by Ward Christensen. The command set is not particularly consistent or easy to remember, but the excellent documentation contains a command summary that is handy to keep in front of one when using the program. You can move forward or backward from the current viewing point, patch the memory image and rewrite it, examine the allocation bit map, and do many other complex operations that help file recovery after a system crash. It was originally issued in CPMUG Vol. 40, but several updates have appeared. The latest version I can trace is Version 77, available in SIG/M Vol. 86 and CPMUG Vol. 68. Versions DU-V65 and later run under CP/M 2.2 as well as 1.4, and have customizable I/O for many controllers.

Diagnostics

A popular diagnostic for a quick test of the medium is FINDBAD, published first in *Interface Age* and subsequently made available in the SIG/M library. This is a nondestructive test that attempts to read all sectors on the disk. Any unreadable sector in the directory area causes the program to abort with a warning message. A bad spot in the system tracks results in a warning message, but the program continues. An unreadable sector in the data area results in the entire block being isolated in a file named [UNUSED].BAD. A count of bad blocks isolated in this manner is displayed when the last block has been tested. Versions up to 3.8 are for 8″ SSSD disks only. The latest version is in SIG/M Vol. 86, and this handles both $5^1/_4$″ and 8″ disks. A CP/M-86 version is also available in SIG/M Vol. 96 (CP/M-86 Utilities).

A more comprehensive medium diagnostic for 8″ SSSD disks is DISKTES1 (CPMUG Vol. 8). This overwrites existing data and should be used with care. It fills all sectors first with 00, then with FF, and finally with E5, testing as it goes. In between each data test the program executes a seek test, checking to see that every track is accessible. On my system, it has revealed "sticky bits" that were not noticed by FINDBAD.

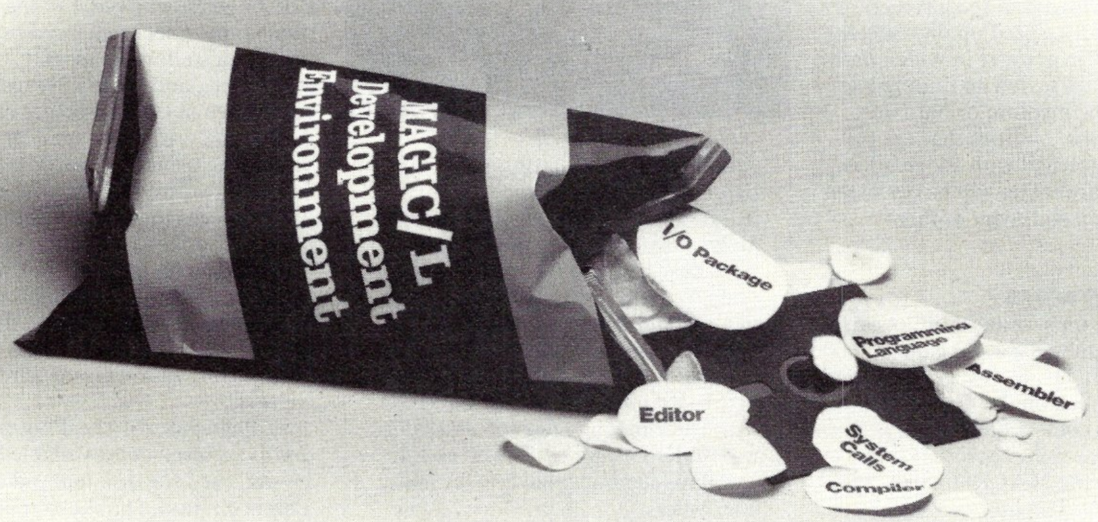
A more recent diagnostic is DTST.Z80 (SIG/M Vol. 62), contributed by Laboratory Systems of Los Angeles, who put it in the public domain for noncommercial purposes. This allows read-only, read/write, and write-only testing. Any range of tracks/sectors can be tested, and error messages can be sent to the printer. A particularly valuable feature is a test with random patterns that can show up problems overlooked by simpler tests. The results can indicate hardware problems, but the checking is mainly of the medium itself. Again, the documentation gives instructions for customizing hardware-dependent routines.

A diagnostic for checking the drives themselves is available from Dysan, Inc., the disk manufacturers. This is *not* in the public domain, but was being offered at a special CP/M-83 show price of $30 last January. The package consists of a specially recorded precision diagnostic disk and a second disk containing the ASM source code of a program running under CP/M that exercises the drive under test which has in it the diagnostic disk.

Disk accelerators

This group of programs consists of utilities that speed up disk accesses by reading one or more tracks at a time into a buffer and fetching sector requests from the buffer (if they are there). FAST (CPMUG Vol. 38) can substantially decrease the execution time of programs (such as PIP) that do a large number of sequential accesses. SPEED (also CPMUG Vol. 38) is a CCP replacement with track blocking. An extension of these techniques is COPYFST3 (SIG/M Vol. 63), which copies a complete disk to another drive. The destination disk is identical to the source disk when copying is complete. The program buffers as many tracks as possible into the available memory, and additional speed is obtained by the use of read skewing and track-to-track skewing. The author suggests some desirable enhancements such as better

USE MAGIC/L™ TO WRITE JUST ONE PROGRAM



Bet you can't stop with one.

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

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

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

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

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

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

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

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

LOKI ENGINEERING, INC.
55 Wheeler St., Cambridge, MA 02138 (617) 576-0666

CIRCLE 61 ON READER SERVICE CARD

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

Public Domain
continued . . .

error checking, CP/M version checking, and others, but (at least on my single-density system) the program provides a substantial increase in copying speed with no problems.

Communications update

I recently received from Frank Gaude COMM723, a communications program based on MODEM7 that has many new and important features, of which I can list only a few here. Files can be erased from the main menu. A comprehensive disk file manipulation utility (UTL) allows copying single or multiple files, renaming files, logging in new drives and user areas, printing, and viewing files on the terminal, all without leaving the communications program. A "Softkey" feature allows commonly used strings (such as "XMODEM S" or "XMODEM R") to be sent by striking ESC and a numeral. This version appears to have made another breakthrough in power and convenience, and Frank is planning to convert it for 16- and 32-bit processors.

The CP/M Bus

by Anthony Skjellum

In previous CP/M Bus installments, we have discussed various aspects of CP/M, as well as possible enhancements to CP/M2, and specific public domain software. With the advent of 16-bit microprocessors, new versions of CP/M have become available. Specifically, CP/M-86 has become a major operating system for 8086/88 microcomputers. Therefore, beginning with this installment, we will discuss various aspects of CP/M-86. For now, we will not include Concurrent CP/M-86 (CCP/M-86) in the discussion.

CP/M-86 for CP/M2 Users

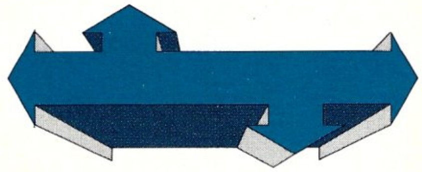
Most **Microsystems** readers have experience with the CP/M2 operating system. For those readers, the transition to CP/M-86 will be extremely smooth. This is the major advantage of CP/M-86. It provides 8080/Z80 programmers with a familiar operating environment which lends itself to the quick translation of existing software. The number of new utilities is minimal, and even the system call numbers have been maintained between the two operating systems.

Code translation from CP/M-80

Translation of 8080 assembly language may be effected via the Digital Research XLY-86 processor. This program runs on CP/M-80 (and under VAX/VMS for the Digital Equipment (DEC) VAX-11 series computer) and converts 8080 assembly language into the 8086/88 assembly code. The conversion is usually less than optimal, but it is a way to get the bulk of the translation effort done mechanically. XLT-86 is a separate utility priced at $150 for the microcomputer version and at $8000 for the VAX version. The advantage of the VAX version is its ability to convert larger programs.

Background information

Before discussing specifics about CP/M-86, an introduction to the 8086 registers and memory organization is in order. This is necessary to understand how assembly language programs work on the 8086/88



microprocessors. The 8086/88 provide 20-bit addressing. However, there are no 20-bit registers in the design. Instead, there are segment registers (16 bit) which are used in conjuction with the general/special purpose registers in forming memory addresses. When forming addresses, segment registers are always interpreted as having the form XXXX0 (hexadecimal). Thus, the full megabyte of memory may be accessed by the microprocessor.

A few definitions are needed before we can continue. First, data objects that begin on a 16-byte boundary (XXXX0), are said to be paragraph-aligned. Secondly 16-byte quantities that are paragraph-aligned are called paragraphs. Finally, segments are paragraph aligned objects of up to 64K bytes.

A special address notation is often used when discussing the 8086 microprocessor family. This is the notation xxxx:yyyy (hexadecimal). This notation represents a 20-bit address in the form of a segment (xxxx) and an offset (yyyy). The actual 20-bit address is formed by adding xxxx0 to 0yyyy. Thus the notation FFFF:000F represents the absolute address FFFFF, which is the highest memory location possible in 20-bit addressing. Furthermore, the location FFFFF resides within paragraph FFFF.

8086/88 register set

The 8086 has four segment registers: CS (code-segment), DS (Data-segment), ES (extra-segment), and SS (stack-segment). The CS register defines the starting address of the data segment. The SS register defines the segment in which the stack will reside. Finally, the ES register defines the starting position for extra segment. Note that the extra segment is used in different ways by various programs and usually has not set function, except with a few special instructions.

In addition to the segment registers, the 8086 has various general purpose registers. They are as follows:

```
AX   (AH, AL) [A]   --   16 bit
     accumulator
BX   (BH, BL) [HL]  --   16 bit
     index register
CX   (CH, CL) [BC]  --   16 bit
     counting register
DX   (DH, DL) [DE]  --   16 bit
     register
SI   [IX or IY]     --   16 bit
     source index
DI   [IX or IY]     --   16 bit
     destination index
BP   [no analog]    --   16 bit
     stack index register
```

where the parenthesized letters indicate the ability of the register to be used as 8-bit parts. The register names listed in square brackets are the closest analogs for the 8080/Z80 microprocessors. In addition to the general-purpose registers are the special-purpose registers. They are as follows:

```
F    [F]  -- 16 bit flags
SP   [SP] -- 16 bit stack
             pointer (SS:SP form
             the address)
PC   [PC] -- 16 bit program
             counter (CS:PC form
             the address)
```

where the square brackets again indicate the 8080/Z80 analogs.

With this introduction, we may now begin discussing the format of CP/M-86 BDOS (basic disk operating system) calls.

BDOS calling conventions

BDOS calls for CP/M-86 are analogous to their CP/M-80 counterparts. For example, console string output is effected as follows under CP/M-86:

```
mov  cl,9               ; command
to print
mov  dx, offset mesg ; point
to message (ds relative)
int  224                ; do the
bdos call
```

The CP/M-80 equivalent would be as follows:

```
mvi  c,9                ; command
```

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

A Remarkable Program For CP/M Users.

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

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

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

Over 55 Housekeeping Utilities.

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

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

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

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

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

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

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

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

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

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

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

Special Password Protection, Too.

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

At $169., It's A Bargain.

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

InfoWorld
Software Report Card
Power Version 2.55

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

© 1982 by Popular Computing, Inc. A subsidiary of CW Communications Inc., Framingham MA. Reprinted from InfoWorld.

POWER! Is Better Than Ever!

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

Take The POWER! Trip Today!

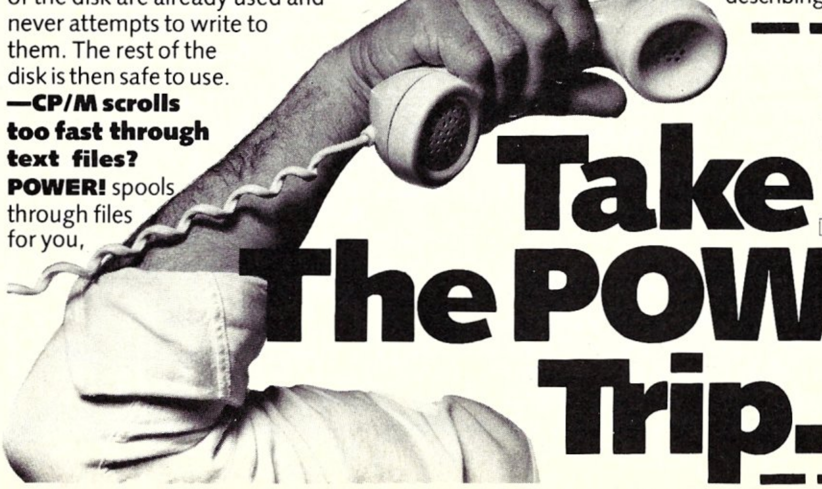
POWER! will operate in any standard CP/M or MP/M system, including CP/M-86, IBM PC, Apple (Z80 card), Osborne, Kaypro, HP, TeleVideo, TRS-80 conversions, S100's including NorthStar Vector, Morrow, CompuPro, etc. Up and running immediately, no configuration necessary—for hard disks and floppies.

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

NOW AVAILABLE FOR MS-DOS, TOO.

ONLY $169. Money Back Guarantee.
Charge & COD Orders Welcome.

TOLL FREE (800) 428-7825 Ext. 96AM
IN CA: (800) 428-7824 Ext. 96AM
DEALERS AND OEM'S (415) 567-1634 Ext. 96AM



Take The POWER!™ Trip.

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

ONLY $169. Calif. add 6½% sales tax

☐ **CP/M $169.** ☐ **CP/M-86 $169.** ☐ **MP/M $249.**

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

CIRCLE 14 ON READER SERVICE CARD

```
to print
lxi   d, mesg            ; point
to message
call 5                   ; do the
bdos call
```

The above example illustrates several points. First, BDOS function numbers are passed in the CL register. Second, 16-bit quantities (DS-relative) are passed in the DX register. Finally, BDOS calls are performed via a software interrupt instruction (int 224). The 8086/88 microprocessors support software interrupt instructions to 256 locations in low memory. CP/M-86 uses an interrupt to provide convenient access to BDOS from any part of the 20-bit address map.

In the above, we saw a specific example of register usage in BDOS calls. Here is the full specification for information passing:

```
1. function mumber is passed
   in CL.
2. byte parameters are passed
   in DL.
3. word parameters are passed
   in DX.
4. DS points to the data
   segment.
```

Note that when addresses are passed to the BDOS, the address used is formed with DS:DX.

Most BDOS calls also return information in registers. Here is the return value specification:

```
1. byte values are returned in
   Al
2. word values returned in AX
   and BX
3. double word values with
   segment in ES, offset in BX
```

It should also be noted that BDOS calls preserve all segment registers except ES.

With the information presented above, we can now discuss the BDOS calls themselves. In what follows, we will only discuss BDOS functions which differ from those of CP/M2. We'll complete this discussion in the next installment of CP/M Bus.

BDOS calls

SYSTEM RESET (#0): The System Reset function provides a way to exit from transient programs with reactivation of the console command processor (CCP). This function has no abort code which is passed in DL. When DL=0, memory allocated to the program is de-allocated and control is returned to the CCP. When DL=1, the program remains in memory with its memory allocation(s) intact.

DIRECT CONSOLE INPUT-STATUS-OUTPUT (#6): On entry, the DL register is set to one of several values to indicate the operation to be performed. OFFH indicates an input request, OFEH indicates a status request. Any other value indicates an output request with that value as the output character. For input requests, the AL register is returned with the character input. For status requests, AL is returned as OFFH when data is available, and as zero when none is available.

ADDR OF ALLOC VECTOR (#27): This function returns the address of the disk allocation vector for the currently selected disk drive. The information is returned in ES:BX. The allocation vector is used by programs such as STAT to ascertain the amount of free storage available on the disk.

RESET DRIVE (#37): This function is used to reset one or more system drives to the read/write state in which they have not been logged-in by CP/M. The DX register is used in a bitwise fashion in order to indicate the drive or drives to be reset. Bit zero of DX corresponds to the A: drive, bit 15 to P:, and so forth. For example, placing a one bit 2 of DX will cause the C: drive to be reset by this call.

CHAIN TO PROGRAM (#47): The default DMA buffer is filled with a command line to be handled by the CCP. This function causes the parsing of that command line and the execution of the program, if possible. Furthermore, any memory allocation belonging to the calling program is released before the chain is performed.

GET SYSDAT ADDRESS (#49): This function returns the address of the system data area in ES:BX. This data area is described on page 2 of the enhancments section of *CP/M-86 Operating System System Guide*. It allows direct access to the following variables:

```
1. user DMA address, user DMA
   segment
2. current disk
```

CP/M Bus continued . . .

3. current user number
4. list toggle flag (set using ctrl-P)
5. console width
6. printer width
7. current console column
8. current printer column

DIRECT BIOS CALL (#50): This function permits direct access to CP/M-86 basic input-output system (BIOS) functions. DS:DX points to a 5-byte BIOS descriptor on entry. The first byte of the descriptor is the BIOS function number. The following two entries are word data, to be placed in the CX and DX registers respectively. This information is placed in CX, DC before the BIOS call is initiated.

SET DMA SEGMENT (#51): This function permits the selection of the paragraph boundary for the DMA address. The value for the DMA segment is transferred in the DX register.

GET DMA SEGMENT (#52): This function returns the DMA SEGMENT and DMA BASE in ES:BX.

Letters to the Editor

Dear Mr. Libes,

Ralph Janelli's program in the August 1983 issue is an excellent way to give some security to CP/M systems. There are three improvements I would suggest to make the program more usable, and the system more secure:

(1) During program execution, the "ENTER USER NAME:" prompt comes on the screen. If the user types Control-C, the program will abort, leaving the user in user 0. This allows anyone to easily access any files in user 0, which includes the file containing all the user names and passwords.

To cure this, use BDOS function #6 instead of function #10. This disables the Control-C key. Specifically, under the LOGON: code, the lines:

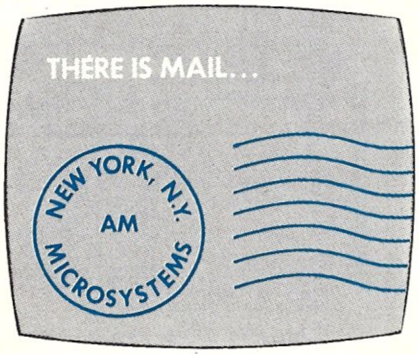
```
          LXI D,NAMBUF ;INPUT
                       ;USER NAME
                       ;TO NAMEBUF
          MVI C,10
          CALL BDOS
```

Would become:

```
          LXI H,NAMBUF+2 ;START OF
                         ;PASSWORD
                         ;STORAGE
          MVI B,OAH
          MVI E,OFFH
          PUSH B
          PUSH H
CHARIN:   MVI C,6
          MVI E,OFFH
          CALL BDOS
          ORA A
          JZ CHARIN
          CPI CR
          JZ FILL
          POP H
          MOV M,A
          PUSH H
          MOV E,A
          MVI C,6
          CALL BDOS
          POP H
          POP B
          INX H
          DCR B
          JNZ CHARIN
FILL:     MVI A,OAH
          SUB B
          STA NAMBUF+1
```

(2) Another way that any user can circumvent the security program is by simply turning the system off and on again. This will cold boot the system into the area for user 0, again allowing any user to read or alter the password file.

To cure this, use the auto-



command feature of CP/M. When you use MOVCPM to alter your system to remove the USER command, also place in CCP+7 the number of letters in BYE, then the hex code for BYE. The command string in DDT should look something like:

```
-S987
00 03   (number of letters in
         command)
20 42   (hex for B)
20 59   (hex for Y)
20 45   (hex for E)
20 00   (signals end of command)
```

Now the system is relatively secure. No user can get into any user space without the proper name and password combination. The problem that now occurs is that at the end of any program, when a warm boot occurs, the BYE program will automatically execute. In other words, at the end of every program execution in user 0, the user must log again. This rapidly becomes tiresome.

The solution to this problem would be to have the autocommand entry (here the command "BYE") execute only after a cold start, rather than after very warm start. This can be accomplished by having the cold start code jump to CCP+0 and the warm start code jump to CCP+3. Many BIOSes have common code for cold and warm start prior to entering the CCP. If this is the case, have the unique warm start code zero the byte at CCP+7. This will negate the autocommand feature on all but a cold start.

(3) When matching the entered password with the stored password, the program will not correctly match lowercase with uppercase. Changing the password prompt to "ENTER USER NAME (PLEASE CAPITALIZE):" ended most of the confusion.

Ralph Janelli's program comes a long way toward addressing a shortcoming of CP/M. It is well written, and, with the additions described above, should give the user a very secure system.

David C. Robertson
Systems Specialist
Harris Communications Div.
16001 Dallas Parkway
P.O.Box 400010
Dallas, TX 75240

Dear Messrs. Libes and Terry,

While I am aware that one purpose of computer magazines is to purvey puff jobs about new products, printing Dave Hardy and Ken Jackson's review of the Computime SBC-880 (August 1983) shows a failure of editorial judgment.

Consider the product in the context it is presented. From the second paragraph, the authors have to face the problem of a single-board computer which is actually a three-board computer. This design was state-of-the-art only six years ago. The CPU is strapple at 2 or 4 MHz in an era of 6 and 8 MHz. The authors note the serious serial handshaking problem in a "on the one hand this, on the other hand that" sentence; they never mention that having only one serial port requires a video or serial I/O board for a workable computer. The authors do seem to treat seriously the implications of the disk parameter required on each diskette.

I cannot understand why you permitted the authors to say that the SBC-880 is "IEEE-696 compatible." In the article itself, the following failures to conform are noted: (1) no 16-bit data path; (2) no extended addressing; (3) "pseudo-DMA" using wait states; (4) on-board memory not accessible to temporary bus masters (really a nonstandard phantom arrangement). For the only magazine edited by the chairman of the IEEE-696 committee, this is pitiful. In the context of "the *Microsystems* series of reviews on S-100 single-board computers," the product is a turkey.

Considering the product in the context of a potential user, from where I sit there are only three potential uses for SCBs: (1) as the core of a single-user general-purpose computer; (2) as a dedicated processor in a Turbodos-like multiuser system; (3) as the brains of a special-

This One Decision Saved our Business and Grossed over $1,000,000. in Sales

In 1979, our recreational manufacturing business was booming. And we had developed a new product that looked like a real winner. The new product was extremely important, in that it appeared to be the answer to a seasonality problem associated with our other product lines.

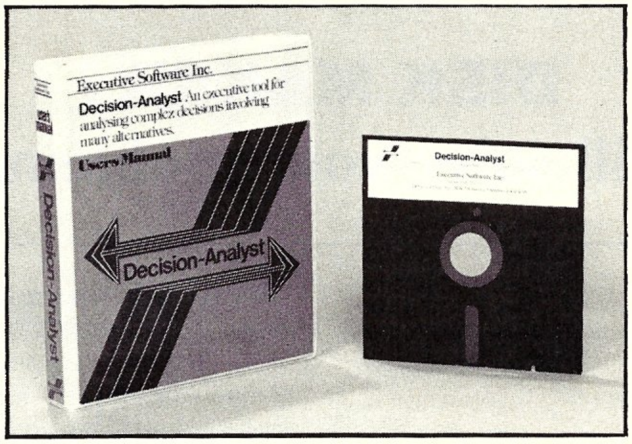
By early summer, our order book was bulging. It really looked as if our off-season sales and production problems were over. Then just as quickly the roof fell in. Gas shortages devastated the recreational vehicle market overnight. And our order book for over two million dollars worth of the new product disintegrated.

Faced with a fall and winter of virtually no sales, many thousands of dollars of unneeded parts and excess production staff, I had no choice but to shut down the production lines. And if a solution to our problem could not be found, the business itself was in jeopardy.

A life saving decision

I spent many sleepless nights trying to come up with a solution to this nightmarish situation. Then I remembered a course I had taken in decision analysis. I spent the rest of that night reviewing course material and other books I had bought on the subject. The next day, I called an emergency meeting.

Using the decision making techniques I had learned, we spent the rest of the week searching for and analysing potential solutions. The net result was that not only was the company pulled back from the brink of destruction, but we added over $1,000,000.00 in gross sales during that off-season.



A way of life

From that point on, almost every critical decision (and there were many) regarding new products, marketing channels, pricing, advertising, production equipment, engineering projects, received this same type of analysis.

Although the process was very time consuming and clumsy, because it had to be done by hand, our decisions were much improved. And there were some real benefits that we had not anticipated.

1. Our understanding of each problem was greatly increased.
2. We uncovered opportunities that we would not have thought of in any other way.
3. Our decisions were documented, preventing us from slipping off the selected path or 'rehashing' the same things over and over.
4. Consensus became easier because we were forced to focus and resolve each part of the problem, one piece at a time.

The birth of Decision~Analyst™

Decision~Analyst was created because the process of evaluating complex decisions with multiple alternatives and many criteria is very tedious and time-consuming if you do it on paper. And doing it in your head is virtually impossible.

Any complex decision usually requires multiple revisions to criteria, alternatives, weights and values assigned to them. If done manually, the ordeal of rewriting, recalculating and redocumenting tends to discourage revisions, thus producing poor results.

Decision~Analyst overcomes these problems by asking for the minimum input possible from you, in the correct sequence. It leads you step by step through the decision making process, then does all the necessary calculations and produces polished reports without any further effort. And all of your input is stored on your disk so that revisions and updates can be made easily at any time.

Comprehensive but easy to use

Decision~Analyst is probably the easiest program you'll ever use. You can literally learn to operate the program using only the 'help' screens. But it comes with a thoroughly indexed manual which includes many pages of examples plus a step by step guide to the decision making process.

And Decision~Analyst is no fly-weight. It uses over 100,000 bytes of fast compiled code and a 40,000 character help file. The program is extremely 'bullet proof' and does all the work . . . you do the thinking. And there is no danger of missing a critical step because your analysis is guided through each of the eight menu-driven sections.

Endless opportunity for use

If you're saying to yourself that you really don't have any earth shattering applications for Decision~Analyst . . . then consider this.

If you're in business, chances are that the most important thing you do is make decisions.

Decision~Analyst can help you select key personnel, decide on new machinery or equipment, prioritize major projects and allocate resources, choose a new product or a better price strategy, select new offices or plant locations, select the most profitable marketing channel or the best piece of computer hardware or software. Use it to analyse any decision which has more than one viable alternative.

In your personal life, it can help you choose the best job, select the right business, career or franchise, determine the best field of investment or even the right home, boat or car. In fact, improving your decisions is probably the most rewarding thing you can do.

Let us help you make a $139⁰⁰ decision

If there's a chance that Decision~Analyst could help you improve even one decision, would it pay for itself? Chances are it would. In fact, it's likely that it could save you the price many times over!

So why not take advantage of this special $139.00 introductory price (plus $5.00 shipping) by calling us today? We'll even refund your money, within 30 days, if you're not completely satisfied.

Decision~Analyst requires a CP/M or MS-DOS operating system, 52K of RAM, a 24 × 80 column screen and an 80 column printer for your reports. Order by calling collect to . . .

Executive Software Inc.

705-722-3373

Dept. no. 1025 DA5

Two North State Street, Dover, Delaware 19901

purpose computer (e.g., robotics). In the first application, the slow speed, poor serial I/O, and large number of slots used make it a poor choice. In the second, the IEEE-696 incompatible features make it a poor choice. I certainly wouldn't trust it to work with active devices (in contrast to passive memory boards) on the IEEE-696 bus on the basis of the review. In the third, the three-board configuration and poor serial I/O rule it out.

I am sure there are other features either unmentioned in the review or which are not apparent to one with my level of hardware and systems integration knowledge that affect the utility of the SBC-880. Certainly the article's conclusion is much too mild. But an editor's job is to ensure that the content of a publication is geared to the readership, and this is the real failing the article reveals.

Roger Friedman
581 Greenwood Ave., NE
Atlanta, GA 30308

Sol Libes replies:

First of all, I am not the chairman of the IEEE-696 committee; Mark Garetz is. Second, it is apparent to me that you got a great deal of information from reading the review by Dave and Ken, and on the basis of this information decided that "the product is a turkey." Therefore I do not understand how you can say that this review was a "a puff about a new product."

Gentlemen:

After reading one of your recent articles on and adapting word processing programs to specific terminals, and recalling a recent article debating the merits of the Xerox 820 system, I am prompted to write in. I was one of the early purchasers of the Xerox 820 system, buying it in November 1981 through a dealer who is no longer handling the system. The machine has been upgraded to the 820-II and has been running under heavy use in our law office since then, with an excellent service record requiring only one very minor service call in this time period.

The system has very many good points, but has a significant drawback I have not been able to overcome. I am seeking your advice and the advice of any of your readers who may be able to help. Xerox modified WordStar, customizing it for this machine and, in my opinion, greatly improving the commands and screen layout. The system is much easier to use, and the cursor controls are fully functional, together with delete and backspace keys. No requirement to use the cumbersome "magic diamond!" However, and I strongly emphasize *however*, the program does not come with any mailmerge. This is a great pity, since the machine is severely limited without the mailmerge capacity.

I have searched high and low for a solution to this problem. I have purchased regular MicroPro WordStar and Mailmerge, but you cannot use the cursor controls or full delete and backspace. The secretarial staff turns up its nose at the "magic diamond." If you or any reader can supply a patch enabling the cursor controls and the delete and backspace keys, I would pay dearly for it. One problem is that the down arrow key sends a ^B, which is also the MicroPro format command where

Xerox uses a 6 instead. One would think that the best-selling word processor could be customized for this popular machine, but it has not happened yet, and MicroPro gives you a very cold shoulder when you ask about it.

Curiously enough, the Xerox version of the program seems to have the potential mailmerge capability. Although not listed on the screen at the directory menu, entry of the J command will produce a prompt "Error E47 MERGEPRIN.OVR not found". This was a clue that the capability might exist. Being industrious, I changed the name of my MicroPro MAILMERG.OVR overlay program to MERGEPRIN.OVR and put it on the Xerox disk. A J command then produced the mailmerge prompts and took me clear through the print run questions, only to get a response "Error 343 wrong version, overlay file" at the last carriage return. The mailmerge program I purchased was version 3.0—the only version currently available.

This experiment convinced me that somehow mailmerge should work on the Xerox version. I made countless calls to Xerox and to MicroPro, with no helpful response at all. Xerox treated me like the plague, and I even wrote to the president and called the executive offices without result. I did finally talk with a tech rep for a distributor in California, who told me that Xerox used version 2.26 of WordStar and that if I could get a hold of a copy of mailmerge in this version, it probably would work. However, I have been unable to find this version, and MicroPro apparently will not sell it direct. I have heard the Osborne also uses this version. If any one out there has any knowledge about this, please contact me as soon as possible and call collect if you wish. I find it hard to believe that neither MicroPro nor Xerox will provide this capability, since virtually every 820 owner would probably be interested. Any help you or your readers can give me would be very greatly appreciated.

Edwin H. Bideau III
Attorney at Law
123 West Main St.
Chanute, Kansas 66720

Dear Mr. Terry,

SIVic-9000 is an independent information exchange for Victor 9000 users. A monthly newsletter is planned, with the first issue scheduled for mid-August publication. It will contain a list of compatible software available, new product information, and hopefully, a review of new word processing software. Anyone wanting further information can write SIVic-9000 at:

SIVic-9000
362 Peachtree Ave. NE
Atlanta, Georgia 30305

They should enclose a stamped, self-addressed envelope. A contribution of a dollar or two to help defray production costs would be appreciated, but is certainly not required.

We also *welcome* articles, notes, applications, compatible peripheral information; in fact, any information that will help Victor-9000 users get the most out of their equipment.

Bob Czajkowski
Editor/SIVic-9000

APPLE IIe 64K RAM CARD

80 column x 24 lines

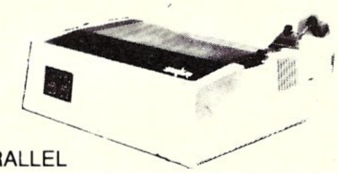
64K RAM

Compatible with Apple IIe Software

$149.00

COEX 80-FT DOT MATRIX PRINTER

- 9x7 Dot Matrix, 80 CPS, Bi-Directional Printing
- 2K Buffered Memory
- 80, 96, 132 Columns, Graphics and Block Printing
- Selectable Char Pitch, Line Spacing and Feed



PARALLEL

COEX Interface Card to Apple . . **$49.95**

Commodore Interface Card to VIC, 64, PET. **$79.95**

$240.00

PARALLEL INTERFACE
EPSON TO APPLE

New From COEX **$49.95** CABLE INCLUDED

for APPLE 16K RAM CARD

Language Transparent

COEX FACTORY WARRANTY **$49.95**

FLOPPY DISK DRIVE

Apple IIe Compatible

with
- Track Zero Micro Switch
- DOS 3.2.1 & DOS 3.3
- CP/M and PASCAL

DESIGNED FOR YOUR APPLE™ $235.00

Controller Card for above. $75.00

5¼″ Floppy DISKETTES

All Certified–100% Guaranteed

BOX of 100 . . . **$149.00**

Above with Hub Rings. **$169.00**

EXTENDER CARDS

for APPLE. . . **$16.95**

for I.B.M. **$19.95**

BMC 12″ Green Monitor $95.00

DO YOU HAVE A COMMODORE?

NEW ROM for COEX 80 & DP8480

Allows Full Graphics Compatibility with All Commodore Computers **$29.95**

Now You Can Afford Another 64K . . .

Especially when it's less than a half cent per bit!

Specifications:

- Fully Static Operation
- Supports S-100 IEEE–696 Standards
- Uses Popular 2716 Pinout Type Static RAM's
- Board Access Time Under 200nS
- 150nS RAMS Standard
- No Wait States Needed at 6.000MHz
- High Quality FR-4 Type PC Board
- Switch Selectable Phantom Line
- All Data, Status and Address Lines Fully Buffered
- Gold Plated Contact Fingers for Low Contact Resistance and Long Life
- Switch Selectable Extended Address Lines For Up To 16 M-bytes
- Extreme Low Power Dissapation (<500mA Typical
- Top 8K May Be Switched Disabled and/or Interchangeable with 2716 Type EPROM's

COEX 64K S-100 CMOS STATIC RAM BOARD

only **$299.00**

Assembled & Tested

"Have You Kissed Your Computer Lately"

Components Express, Inc.

VISA

1380 E. Edinger • Santa Ana, Calif. 92705 • 714/558-3972

master charge

TWX 910-595-1565 • ADVACON SNA • International Orders Welcome

Terms of Sale: Cash, Checks, Credit Cards, M.O., C.O.D. Calif. residents add 6% sales tax.

CIRCLE 226 ON READER SERVICE CARD

A MICROSYSTEMS TUTORIAL

An Introduction to Local Area Networks: Part 1

by William G. Wong

Personal computers are quickly becoming common items in business, and many of these are being connected together to form a Local Area Network (LAN). The purpose of a LAN is to provide sharable resources to the computers within the LAN. There is such a proliferation of different LAN implementations that the initial exposure to LANs can be confusing and overwhelming.

This article is designed to reduce the confusion by examining the International Standards Organization (ISO) network model. This generalized model for computer networks, which applies both to Local Area Networks (LANs) and to Global Area Networks (GANs), is described first. A LAN will typically have the computer network spread thoughout one room or building, while a GAN network is spread thoughout a city or a country.

The second section deals with LAN topology, access methods and protocols. The topology of a network can be viewed from several angles: 1) layout of the nodes and their interconnection; 2) the way the connection is made with some media; or 3) the routes along which messages move through the LAN. All these aspects are discussed along with the different combinations currently being used.

The third section addresses the standards being considered by the Institute of Electrical and Electronics Engineers (IEEE) 802 Committee, which is working on standards for local area networks. These standards deal with the physical connections to LANs and the associated communication protocols and access methods.

The fourth section presents a description of some services which can be provided by nodes within a LAN and names associated with such nodes.

The last section presents an overview of the hardware and software used to implement some existing local area networks. While a complete listing is beyond the scope of this article, this list should give a flavor of the different types of LANs available.

ISO network model

The ISO network model is an Open System Interconnection (OSI) and consists of seven layers. It is an ideal model of the types of functions provided by the hardware and software of a computer in a network. This is a generalized model and does not exactly match any particular implementation; it is, however, an excellent base for designs and comparisons. A particular system may implement only certain parts of the model, depending upon the functionality required. Also, the model does not place any restriction on which parts may be implemented in hardware and which parts in software.

The model of a network consists of a number of "nodes" connected together by a common communication system. The term *node* is used instead of *computer*, because a node may itself be a system consisting of one or more internal computers.

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

The ISO model is layered to allow for a modular implementation of the hardware and software. Each layer is designed to provide a distinct service related to the network. A layer has an interface to the layer above and below it. The exception is the Physical Layer, which provides the connection between the layers on different nodes. All layers are found on each node within a network. Figure 1 shows a logical representation of the ISO layers in a two-node network.

The user interface to the system is through the Application Layer. Communication across the network is between two application programs in the Application Layer. One example of an application program is a real-time message service which would allow you to send messages to a friend on another node that is running the same type of program.

In this example, your message would be passed from the Application Layer on your node to the Presentation Layer, and then to the Session Layer, and so on, until the Physical Layer is reached. At this point, the information would move to the other node and then be passed to the Data Link Layer. The information is then passed to the Network Layer and so on, until it reaches the other application program, where it can viewed by your friend. A response would travel over a similar path but in the reverse direction. The purpose of each layer is described in more detail in the remaining portion of this section, but first a word about why the layered approach is used in the ISO model.

Modular design is the basis for the layered approach. Each layer insulates the layers on one side of its interface from the particular implementation of the layers on the other side. For example, a particular Data Link Layer could be implemented for an associated Physical Layer. Use of a new Physical Layer implementation would require modification of the Data Link Layer implementation only, which has the same interface to the Network Layer. No other layers would be affected by this change.

Combining layers into one hardware or software module is also possible without affecting the structure of other layers. For example, the Presentation and Session Layers may be implemented as one or more logical modules. From the Application Layer view, there is no difference.

The interface at each layer and the functions of each layer constitute an area where standards can be set. This standardization will allow the creation of interchangeable network parts from various vendors, including networks with nodes from different vendors.

The rest of this section deals with the functions provided by various layers in the ISO model and the interface to the adjacent layers. Remember that a particular implementation of a layer may not include all the operations listed here, and some system implementations may exclude entire layers. The layers are presented from the most sophisticated layer (Application) to the least sophisticated layer (Physical).

Application Layer. The Application Layer provides the user with an interface to the network. The typical interface device to the Application Layer is a terminal display and keyboard. Applications deal with local resources only, and communicate with remote resources through the Presentation Layer.

Operations provided by this layer include password and logic procedures for the network, electronic mail, remote print spoolers, network file transfers, remote job entry, and downloading of files to another node in the network. This layer may also provide a Virtual Terminal Service where the local terminal is logically connected to an application located in a remote node. The Application Layer programs also serve other nodes in the network by providing shared resources such as a hard disk, a data base, or printer.

The Application Layers interface with the Presentation Layer and consist of a very high-level protocol where communication is assumed to be secure and reliable. Connections to a remote Application Layer are through a logical entity such as a file, mailbox or port. Information sent from a node to a remote mailbox will arrive without errors if possible. Any format conversion, retransmission, or security is handled transparently by the other layers. Receipt of an error indicates that a network failure has occurred.

Presentation Layer. The Presentation Layer is responsible for converting data from the Application Layer into a format used by the Session Layer along with the reverse operation. The simplest form of the Presentation Layer occurs when the two layers use the same data format. There are, however, many reasons for using different formats.

One reason for different formats in the Application and Session Layers is to provide data compaction. This reduces the size of the message sent across the network, thereby increasing the overall bandwidth of the system. Many data compaction methods exist, including Huffman coding of English text.

Another reason for the format conversion is to provide data security. Physical security can usually be provided within a node, but providing a secure communications channel is sometimes difficult. Data encryption techniques can be used to provide secure information exchange between secure nodes even if the communications channel is not secure. The Presentation Layer is responsible for performing the encryption of network-bound data and decryption of application-bound data.

The last reason for format conversion is more obvious. The modular approach to the network model design allows for separate implementation of the various parts. Plugging them together may not always work if the data format differs either on a remote node or within the layers on one node. A simple example would be a network in which one node uses ASCII while another node uses EBCDIC for character representation. In the simple case, the lower network layers would use the ASCII format and the Presentation Layer on the remote node would provide the EBCDIC to ASCII translation. It may also happen that both Application Layers use ASCII, whereas the lower layers in the network implementation use EBCDIC.

Character translation is not the only format conversion possible; however, it gives a flavor of one type of operation provided by the Presentation Layer. The Presentation Layer does not perform any other control function.

Session Layer. The Session Layer provides a very stable and reliable interface to the Presentation Layer. It protects the Presentation Layer from network failures by using alternate routes to communicate with remote nodes, if possible. It is responsible for initiating and terminating tasks within the nodes, along with making and breaking logical connections. This makes the Session Layer analogous to a typical multitasking operating system for a node.

The Session Layer allows tasks to communicate with other tasks, using high-level interfaces called virtual circuits and datagrams. Virtual circuits are used with pipes, ports and files, while datagrams are used with mailboxes and semaphores. These are the logical entities mentioned in the Application Layer. The Session Layer is responsible for mapping these logical entities into physical operations and data structures within a node.

A quick note on the difference between the operation of a virtual circuit and a datagram facility: A virtual circuit has a semipermanent port at the local node, with a corresponding port in the remote node. The virtual circuit is the logical connection between the two ports. Information placed at one port is moved to the other port. No routing information is required once the connection is made. No information can be sent when the connection is broken.

A datagram requires a mailbox at each node; however, the mailboxes are not logically connected as with a virtual circuit. Datagrams are messages that include the address of the destination mailbox along with some data. The address of the sending mailbox is usually included. Application programs must determine where datagrams come from by looking at the sender's address or the data. A datagram may have originated at any node in the network, whereas a message in a virtual circuit must have originated at one particular source.

Transport Layer. The Transport Layer is responsible for mapping the logical network addresses of the Session Layer to physical network addresses. For example, a logical address may be Print-server-1, while the physical address might consist of Port-3 of Process-5 on Node-4. Note that the physical address does not indicate the actual route over which data might travel to get to this point.

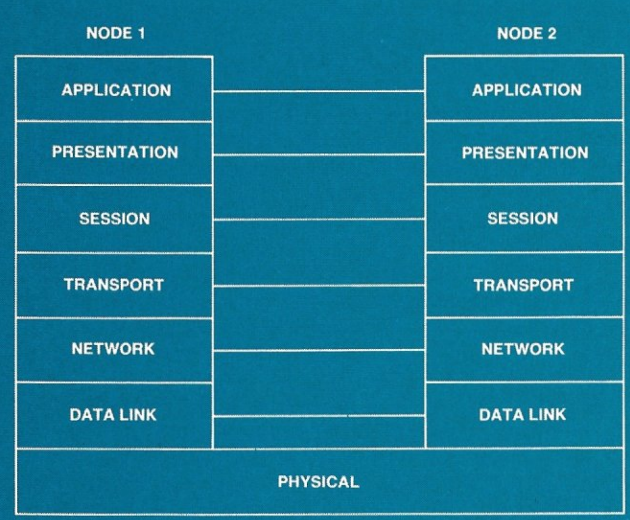


Figure 1. Logical Connection between layers.

The interface at—and functions of—each layer constitute an area where standards can be set; this allows the creation of interchangeable network parts from various vendors.

The communication link between a node and a remote node must be a quality connection with end-to-end data integrity. Information from the Session Layer is disassembled and sent to the Network Layer. The Transport Layer also performs the corresponding assembly of messages from the Network Layer into information which is passsed to the Session Layer. Messages may contain an arbitrary amount of data.

Any errors detected during assembly of messages or by the Network Layer cause the Transport Layer to attempt retransmission. This may involve sending additional coordinating messages to the appropriate Transport Layer at the remote node. Retransmission attempts continue until the information is exchanged intact or a set number of attempts prove unsuccessful. The Session Layer is notified of a network failure if all attempts fail.

The Transport Layer is the obvious choice for monitoring the quality of the network message service and the usage of the network. This information is typically available to the Application Layer through the intervening layers of the model and can be used to detect problems in the physical network links and the associated hardware. The information on the network usage can also be used for billing purposes.

Network Layer. The Network Layer receives messages from the Transport Layer, which include the data to be sent and the physical network address of the destination. The Network Layer is responsible for converting each message into one or more packets that can be sent by the Data Link Layer. At the receiving end, the Network Layer reconstructs the message by reassembling the packets in the proper order. The conversion between message and packets is required because the Data Link Layer usually places a limit on the size of a packet, whereas a message can be of any size and may be very large.

The Network Layer is also responsible for setting up the route by which a packet will reach its final destination. This routing function is easy if the Data Link Layer can broadcast the packet to all nodes in the network; but if this is not feasible, more sophisticated routing procedures are required. These may consist simply of selecting the channel to be used by the Data Link or, in more complex systems, may require the layer to generate the list of nodes through which the packet must pass before it gets to the destination.

The Network Layer keeps track of the network status and reports this information to the Transport Layer. The reports include information on any data channel failure or hardware interface failure.

The control of message priority and flow control resides in the Network Layer. Messages from the Transport Layer may include a priority, and it is the job of the Network Layer to see that high-priority items are sent first, possibly over a high-priority channel. High-priority incoming messages are also delivered to the Transport Layer before lower priority messages.

The flow control operation of the Network Layer prevents other nodes from sending too much information at one time. Without such control, the node could become overwhelmed by the incoming data, with possible loss of data or degradation of node performance. Flow control is usually accomplished by the exchange of flow control messages with the Network Layer on other nodes. These messages include information such as acknowledgement of receipt of data, the number of free message buffers, or the fact that the node is too busy now to receive any more information.

In more complex systems the Network Layer provides a "gateway" function. A gateway is the means by which two different networks are interconnected. Nodes in one network can send information through the gateway to reach nodes on the other network. The gateway function resides in a node with a connection to the other network. This connection may be direct to the other network or to a node in the other network that is also acting as a gateway.

Nodes in the network may have to make special requests to the node performing the gateway function in order to access nodes on the other network. Alternatively, the gateway node may provide a predefined set of services that are actually performed by nodes in the other network. In either case, the local nodes do not have to use a special protocol once the connection is made. Communication with the other network proceeds if the operations were being performed by the gateway node. The gateway node simply performs a routing and protocol translation function transparent to the local nodes.

A network may have more than one gateway, and it is possible to have a number of networks connected together with many gateways. In this case, a message may go through many gateway nodes before getting to its final destination. The message will usually go from the Network Layer on one gateway node to the Network Layer on an associated gateway node. The message may go through the Data Link and Physical Layers of a remote network to get to the destination; however, the message will usually not go through the other layers except on the destination node.

The information passed between the Network Layer and the Data Link Layer usually consists of a packet of information with a physical network destination address. The address of the node is usually included so the destination node can know the identity of the node which sent the packet. The Network Layer is also responsible for error recovery if problems occur during transmission of a packet.

Data Link Layer. The Data Link Layer is responsible for taking a packet from the Network Layer and adding the appropriate prefix and suffix information. This information is then converted to the format required by the Physical Layer. The information in the Data Link Layer is usually a number of bytes, while the Physical Layer often requires one bit at a time.

The prefix and suffix information usually provides synchronization between nodes connected by a common Physical Layer. This information may also provide error detection and correction codes to facilitate the detection of errors by other nodes. The additional information is usually coupled with an access method and protocol to coordinate the use of the Physical Layer.

These access methods and protocols are described in the

Combining layers into one hardware or software module is possible without affecting the structure of other layers.

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

Local Area Networks continued . . .

next section, since this is one area where standards are being developed. An access method is the procedure used to gain exclusive use of the Physical Layer, assuming that nonexclusive use of the Physical Layer results in errors. The protocol is the low- and high-level signaling procedure used in the Physical Layer. Examples of a low-level signaling procedure are the start and stop bits used in an asynchronous serial transmission of byte of data. High-level signaling protocols deal with the format of a packet as it is sent through the Physical Layer.

Physical Layer. The Physical Layer is the only layer in the model that is common to more than one node in terms of the actual implementation. That is, the other layer may be implemented differently in other nodes within a network. However, all nodes must have the same Physical Layer if they are physically connected.

The Physical Layer, another area in which standards are being developed, deals with the electrical and mechanical characteristics of the interconnection subsystem, including the electrical encoding and decoding of the information from the Data Link Layer. This information is typically bit-serial in nature. The Physical Layer also specifies the electrical characteristics such as voltage, current and duration. The corresponding physical media currently include twisted-pair wire, coaxial cable, fiber optics, and many others.

Part 2 of this article, dealing with LAN topology, access methods, and protocols, will be published next month.

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

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

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

RENT SOFTWARE BEFORE YOU BUY!

from our

SOFTWARE RENTAL LIBRARY

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

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

SPECIAL INTRODUCTORY OFFER

There are now 2 different plans to choose from:

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

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

(complete with manuals in original manufacturers' packages)

To Immediately Order, or for more information:

UNITED COMPUTER CORP.
Software Rental Library
Culver City, California

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

Canadian Orders Welcome

MasterCard AMERICAN EXPRESS VISA

*Plus postage and handling.

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

CIRCLE 4 ON READER SERVICE CARD

SOFTWARE

c h e a p c h e a p r e a l l y c h e a p

to-go

IBM PC SOFTWARE (PC/MS DOS)

Business	LIST	SALE
American Training International		
Power for PC DOS	75	56
Ashton-Tate		
Financial Planner	700	489
Friday	295	202
Aspen Software		
Grammatik	75	56
Proofreader 32K	50	38
Proofreader 50K	50	38
Proofreader 80K	50	38
Best Products		
Personal Financial Pgm	95	66
Comprehensive Software Support		
PC Tutor	80	60
Continental Software		
FCM (First Class Mail)	125	79
Home Accountant Plus	150	99
Eagle Software Publishing		
Money Pack Vol. 1 & 2	400	299
Emerging Technology		
Edix	195	149
Wordix	195	149
Edix + Wordix	390	279
Edix/Wordix/Spellix		325
Howard Software		
Real Estate Analyzer	250	185
Information Unlimited Software		
Easy Planner	250	188
Easy Speller 2	225	135
Easy Writer II	350	229
Lexisoft		
Spellbinder	495	259
Lifetree		
Volkswriter	195	129
International Volkswriter	205	139
Mark of the Unicorn		
The Final Word	300	223
Mince	175	139
Metasoft		
Benchmark Mail List	250	184
Benchmark Word Processor	500	367
Micro Pro		
Mailmerge	250	165
Spellstar	250	165
Word/Mail	695	426
Word/Spell	695	426
Word/Spell/Mail	845	558
Wordstar	495	327
Microsoft ®		
Multiplan™	275	189
Multitool Budget		109
Multitool Financial		69
Oasis		
Punctuation & Style		109
The Word Plus	150	112
PBL Corporation		
Personal Investor	145	98
Peachtree		
SERIES 4 PAK (GL,AR,AP)	595	349
Select Information Systems		
Select Word Processor	595	339
Software Arts		
TK!Solver		224
Sorcim		
Supercalc	195	129
Supercalc 2	295	185
Visicorp		
Visicalc	250	169
Visispell	225	183
Visiword	375	298

Database	LIST	SALE
Alpha Software		
Mail List	95	67
Applied Software Technology		
Versaform	389	259
Ashton-Tate		
dBASE II	700	398
dBASE II w/user's guide	729	419
Conceptual Instruments		
Desk Organizer		245
Condor Computer Corporation		
Condor 3	650	398
Fox & Geller		
Quickcode	295	185
dGRAPH	295	195
dUTIL	99	59
FYI, Inc.		
Superfile	195	129
Information Unlimited Software		
Easy Filer	400	279
Innovative Software		
TIM III	495	279
Link Systems		
Data Fax	299	224
MDBS		
Knowledge Man	500	327
Micro Pro		
Infostar	495	327
North American Business Systems		
<<Answer>>	250	159
Software Publishers		
PFS: File	140	105
PFS: Report	125	94
Supersoft		
Personal Data Base	125	93
Visicorp		
Visifile	300	219

Home/Education/Games	LIST	SALE
Automated Simulations/ Epyx		
Curse of Ra	20	15
Temple of Apshai	40	30
Upper Reaches of Apshai	20	15
Avalon Hill		
Computer Stocks & Bonds	25	19
Midway Campaign	21	16
Bible Research		
THE WORD Processor		146
Blue Chip		
Millionaire	100	75
Broderbund Software		
Apple Panic	30	22
Infocom		
Deadline	50	37
Starcross	40	30
Suspended	50	38
Zork I	40	30
ISM		
Mathemagic	90	65
Lightning		
Master Type	50	38
Microsoft ®		
Flight Simulator™	50	38
Sierra On Line		
Crossfire	30	22
Frogger	35	26
Strategic Simulations		
Warp Factor	40	30

Language	LIST	SALE
Lifeboat		
Lattice C Compiler	500	415
Microsoft ®		
C Compiler™	500	400
Mumath/Musimp™	300	225
muLisp/MuStar™	250	188
Ryan-McFarland		
RM/Cobol Full Dev System	950	713
RM/Cobol Runtime Only	250	188

Utility/Comm/Graphics	LIST	SALE
Alpha Software		
Apple-IBM Connection	195	135
Typefaces	125	87
Byrom Software		
BSTAM	200	149
BSTMS	200	149
Central Point		
Copy II PC	40	35
Hayes		
Smartcom 2	119	90
Hercules		
Graphics Card	499	399
Innovative Software		
Fast Graphs	295	159
Insoft		
Data Design	225	169
ISM		
Graphmagic	90	65
Graphmagic Combo	150	119
Microstuf		
Crosstalk/ Smartmodem	195	135
Peter Norton		
Utilities	80	60
Versa Computing		
Versawriter Graphics Tabl	299	249
Visicorp		
Visitrend/Plot	300	215
Woolf		
Move It	150	99

IBM PC HARDWARE

Hardware	LIST	SALE
Corvus		
PC Interface, Cable, Man.	300	239
Hayes Microcomputer Products		
Smartmodem 1200B	599	479
Kraft		
Joystick	70	52
Microsoft ®		
Mouse™	195	149
Programming International		
PC-Hayes Cable	35	29
QCS		
Big Blue	595	449
Quadrum		
256K QUADBOARD	595	435
64K Quadboard	395	289
Symtec		
Light Pen	150	126
TG Products		
Joystick	65	49
Tandon		
TM100-2 Drive (5¼" DS)	650	249

PC SOFTWARE — CP/M-86

	LIST	SALE
Ashton-Tate		
dBASE II	700	398
Fox & Geller		
Quickcode	295	195
dGRAPH	295	195
dUTIL	99	59
Digital Research		
CBASIC 86	200	150
CP/M-86	60	45
Pascal MT+86 With SPP-86	600	450
Woolf		
Move It	150	99

™ = Manufacturer's trademark
® = Indicates a manufacturer's registered trademark

TELEPHONE ORDERS

Inside CA **(800) 631-4400**
Outside CA **(800) 222-8811**
Local (415) 324-3730

Order lines are manned 6:30—6 Monday thru Friday and 9—5 Saturday. Other lines are open 9—5 Monday thru Friday.

Technical Support	*(415) 324-0311*
Order Status	*(415) 324-0306*
Sales Manager	*(415) 324-0305*

Orders must be **PAID** by **October 31** to quality for these special prices

More CPM®, APPLE®, IBM PC™, UCSD p-System™ software, hardware, etc.: call for quote.

Programming International

505 Hamilton Avenue • Suite 301
Palo Alto, California 94301

TERMS: All prices subject to change without notice and availability. Cashier's check/MO/bank transfer. Allow time for company or personal checks to clear. Prices reflect cash prepaid discount. VISA/MASTERCARD/COD/PO's + 3%. CA residents add sales tax. All sales final for games & special orders.

SHIPPING: $3 per item for UPS surface ($6 for Blue Label) within Continental USA, except where shipping cost is specified in square brackets. UPS does not go to Canada, Alaska, APO's, FPO's; call for ship charge or add 15%—we will refund/credit difference.

RETURNS: All returns subject to managers approval, must have authorization number, obtained at 415-324-0305. Unauthorized returns will be refused; damaged goods will be refused. All returns subject to 15% restocking fee. No return after 30 days.

CIRCLE 73 ON READER SERVICE CARD

A Low-Cost Local Network for Microcomputers

by William G. Wong

Today microcomputers dominate many aspects of computer networks. XEROX developed one such network called Ethernet. It has a data transfer rate of 10 MHz, but a network interface unit costs hundreds of dollars. In contrast, the interface design presented here costs less than $10, but has a data transfer rate under 100 KHz.

A computer is connected to the serial communication bus through a network interface unit. These network interface units can be placed at any point along the bus, thereby simplifying the network topology.

This low-speed interface is based upon an Intel 8251 Universal Asynchronous Receiver/Transmitter (UART) and an Intel 8253 Programmable Timer. The timer supplies the UART baud rate clock and contains a retriggerable one-shot pulse used for bus access. In low-cost applications, the timer can be replaced with a fixed baud rate generator and a 74LS123 retriggerable one-shot.

The serial bus can be implemented with twisted-pair wire or with coaxial cable. The network interface units are then connected directly to the serial communication bus.

Serial communication bus

The serial communication bus is a common interconnection method used in computer networks such as the XEROX Ethernet. In such networks, the computers are attached to a common bus through network interface units (Figure 1). Communication between computers is done with messages sent across the bus. Although many messages are sent between computers, only one message can be sent across the bus at any one time. The bus itself is typically coaxial cable or twisted-pair wire with a tap at each bus interface point.

The format of a message is similar to IBM's Serial Data Link Control (SDLC) protocol, except that the message format used with this bus is byte oriented instead of bit oriented. The message consists of an address byte, a control byte, an optional sequence of information bytes, and an error check byte. Figure 2 shows the general message format. The beginning and end of a message are marked by a bus available condition detected by the bus interface unit.

Each bus interface unit is wired with a unique address. Any message sent to a bus interface unit will have this address as its first byte. The control byte describes what the message is and how the information bytes are used.

The information bytes typically contain the address of the sender and parameters for the function described by the control byte. Parameters may be a file name, a record number, and a data record for a disk write function request sent to a disk computer in a network.

The error check byte is typically a block parity character or a cyclic redundancy check (CRC) character used to detect errors that may have occurred when a message was sent. The character takes into account all the data in a message. A message is usually discarded if an error is detected, and the sender must retransmit the message.

All computers must listen to the bus to receive messages. A computer can recognize messages destined for it because the address byte corresponds to the address of the computer. The computer must decode the control byte and process the information bytes. It then sends a response to the computer that was the source of the original message.

This is the normal method used to communicate on the serial bus. When errors do occur it is up to the sending computer to resend the message. Errors can occur because of external noise or because of the bus access method used.

The multimaster collision access method used here is similar to the one used with the XEROX Ethernet. Each computer on the bus is a master, which means there is no single bus arbitrator unit. A computer can send a message across the bus when it determines that the bus is not in use. Unfortunately, more than one computer may see that the bus is not in use and send a message. This simultaneous transmission is called a collision, and recepton of this message results in an error. These messages are discarded.

The collision access method assumes that collisions occur infrequently. A message sent during a collision must be re-sent after a timeout period in which no response is received. The serial interface presented can detect when the bus is not in use, and can also detect collisions through byte and block parity errors.

For further reliability, the sending computer can use the UART to receive the message it sends. The message received will not match the one sent if a collision occurs. In this case the sending computer would send a short sequence of illegal characters that would have invalid parity so as to force parity errors at other computers receiving the invalid message, thereby making the collision known as soon as possible. This would free the bus sooner and allow more messages to be sent. Not that the hardware can accommodate this feature, but the sample routines listed here do not implement this option.

Serial interface logic

The serial interface unit is based upon a UART, a retriggerable one-shot, a bus driver, and a twisted-pair cable. The UART is an Intel 8251A, and the retriggerable one-shot is an Intel 8253 timing element that also supplies the baud rate for the UART. The bus driver is an open-collector TTL driver (7407) and the receiver is a TTL buffer (74LS08). The basic logic diagram is shown in Figure 3.

The UART is used to send and receive messages via the bus, while the timer is used to determine when the bus is in use (or not in use). An open-collector driver is required because of the collision access method used. A totem-pole TTL driver would burn up if a collision occurred.

The bus is a twisted pair that has a pull-up resistor (required by open-collector drivers) to 5 volts at each bus interface. This configuration will accommodate about eight interface units with a bus length of 20 feet or less. The number of units and the bus length can be increased by using additional drivers in parallel at each bus interface.

The UART and timer are connected to the serial bus through the open-collector driver and receiver. They are

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

connected to the microcomputer through the internal microcomputer bus. Three interrupts are used with the interface: two from the UART and one from the timer. The UART interrupts indicate when data can be sent or received, and the timer interrupt indicates that the bus is not in use. The interrupts can be enabled and disabled by the microcomputer.

These elements are the only pieces of hardware needed to implement the serial bus interface. The parts are inexpensive and readily available; however, this is not the only way to implement a compatible interface. Single-chip microcomputers like the Motorola M6802 and interface chips such as the Intel 8256 Multifuncton Controller have these elements as part of the chip.

The interface can be further simplified if the baud rate is fixed instead of programmable. In this case a fixed baud rate generator can be used, along with a 74LS123 retriggerable one-shot instead of the Intel 8253.

Serial bus timing

Messages are sent from one microcomputer to another as a stream of bits; the UART converts each byte in the message into a bit stream using the standard asynchronous start-stop protocol shown in Figure 4. A byte in a message is sent as a stream of 11 bits. Note that the date on the bus is inverted with respect to the UART, so a bus zero is a high voltage and a bus one is a low voltage. The bit stream consists of a start bit (always zero), the eight data bits, a parity bit, and a stop bit (always one). This protocol synchronizes the receiving UARTs and indicates bus usage.

The timing element is used to detect the bus usage. It operates under the assumption that a message consists of a continuous bit stream bounded by at least 16 zeros. This means that the message stream cannot contain 16 consecutive zeros. Since the start-stop protocol inserts at least one bit that is a zero (the start bit) and one bit that is a one (the stop bit) the data bits can be any value. This greatly simplifies the message transmission.

The retriggerable one-shot mode of the Intel 8253 is used to determine when the bus is either available or in use. A one-shot generates a pulse of fixed length, beginning with some trigger condition such as a rising edge to a trigger input. A retriggerable one-shot operates in the same way, except that each subsequent rising edge at the trigger input resets the timing period, thereby causing the pulse to be lengthened. In this case the pulse will end the fixed time after the last rising edge that occurs before the pulse ends. A sample timing diagram is shown in Figure 5.

The pulse length is set to 16-bit times. This is the number of zeros that must be sent before and after a message. The output pulse of the one-shot will last as long as the message because the bit stream of a message will contain at least one rising edge for each byte of data in the message, and a byte is sent in 11-bit times. The 16 zero bit time was chosen to allow a 5-bit timing margin between data bytes.

A sample message and one-shot timing diagram are shown in Figure 6. Note that the timer generates an inverted output pulse and that each byte (B0, B1, etc.) is actually the sequence shown in Figure 4.

The microcomputer uses the one-shot pulse to generate interrupts when it must determine the state of the bus. Any microcomputer can determine when the bus is in use without a centralized arbitrator, because each microcomputer has its own timing element. This multiple master bus arbitration method can result in collisions (multiple messages at the same time), but they should be infrequent. The arbitration method also means that any unit can be added or removed without affecting the bus arbitration scheme.

Software

This section describes a sample set of routines that are used to send and receive messages via the serial bus interface. The messages may be of arbitrary length, where the first byte is assumed to be the address of the receiving microcomputer. All other bytes are user-definable. This includes the control byte and block check character (block parity).

Each interrupt routine is initiated when the corresponding hardware interrupt occurs. It is assumed that the state of the microcomputer will be saved before the routine is executed, and restored after the routine is done. The interrupts should be level-triggered and the timer interrupt should have higher priority if possible.

The system is initialized with the proper baud rate and timer settings. These are typically 9600 baud and 16-bit times, respectively. The UART should be disabled, but the one-shot should be enabled. The interrupts should be enabled after the interrupt vectors are set. The routines are presented in a structured form similar to Pascal.

Receive Message function

The Receive Message function takes an array as a parameter. The size of the array is the maximum number of bytes

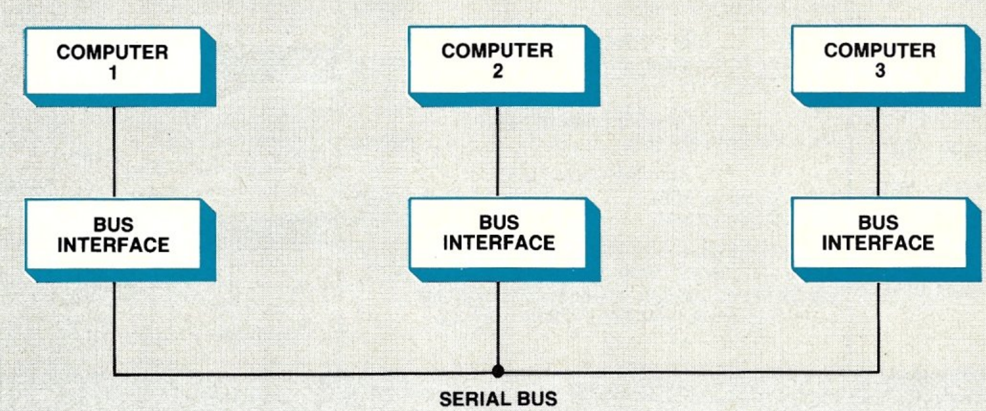


Figure 1. Local computer network.

Eastern States Save on Freight —

MINIMUM SHIPPING $3.00 - NO SURCHARGE ON VISA/MASTERCARD

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

DISK DRIVES

SHUGART

5¼" SA400 (35 TR) 160K 150.00
5¼" SA400L (40 TR) 190 K 175.00
8" SA801R(SS/DD) 1.2 MG 359.00
8" SA851R(DS/DD) 2.4 MG 479.00

QUME

8" DT8 (842) 490.00

MITSUBISHI

5¼" M-4853 ½ size 410.00
96 TPI same as TM 100-4
5¼" M-4854 ½ size 439.00
1.6 MG w/8" electronics formats same as IBM 1 & 2D 8" std

8" M-2894-63 (110V) 399.00
STD 8" DS/DD 1.2 MG

M-2894-63(SS/DD) 220V 449.00
8" M-2996-63 .... 439.00
Thinline 8" DS/DD 1.2 MG

REMEX

5¼" RFD 480-DS/DD (For IBM-PC) 235.00

TANDON

5¼" TM100-1 SS/DD 160K 150.00
5¼" TM100-2 FOR IBM-PC 239.00
DS/DD (320 K)
8" TM848-2 (DS/DD) 1.2 MG 450.00

PERTEC

5¼" FD200-S 160 K 139.00
SS/DD 40TR
5¼" FD250-5 320 K 195.00
DS/DD 40TR

SIEMAN'S

8" FD100-8 (SS/DD) 110V 801 R 179.00
8" FD100-8 (SS/DD) 220V Compatible .. 199.00

MPI

5¼" B-51 *40TR SS/DD* 180 K 175.00

DISK DRIVE CABINETS

8" CABINETS

8" DDC88V28 w/PS vertical for 2-8" drives 300.00
8" DDC88T-1 w/PS vertical-for 2 or 4-8" thinline drives ... 300.00
8" DDC88T-2 w/PS vertical for 2 8" thinline drives 225.00
8" DDC8V w/PS vertical for 1 8" drive 249.00
8" DDC88H w/PS horizontal for 2-8" drives 300.00

5¼" CABINETS

5¼" DDC5H w/PS horizontal-for 1-5¼" drive 59.00
5¼" DDC55V w/PS vertical-for 2-5¼ drives 89.00

DISK SUB ASSEMBLY

OUR BEST BUY'S

8" SUB ASSEMBLY

DDS + O 2EA SS/DD Siemens FD100-8 Drives w/Cabinet 595.00
DDS + 2 2EA DS/DD Mitsubishi M2894-63 w/cabinet 1,075.00
DDS + 4 2EA DS/DD 8" Thinline Drives w/Cabinet 1,150.00
Specify—Vertical or Horizontal Cabinet

5¼" SUB ASSEMBLY

DDS + 5 1EA SS/DD Shugart SA400 Drive 200.00
DDS + 6 2EA SS/DD Shugart SA400 Drive 369.00

DISKETTE STORAGE

ADVANCE ACCESS

AA-5¼ (Holds 82 Disks) 17.00
AA-8 (Holds 82 Disks) 26.00
Smoked Plexiglass Disk Tubs

LIBRARY CASES

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

DATA CABLES

8"DSC 88-2SKT-for 2-8" drvs w/skt. conn. 16.95
5¼"DSC55-2SKT-for 1-5¼" dvs w/skt. conn. 15.00
RS232MM-5' (male to male) 19.00
IBM to PAR 32.00
Osborne to PAR 32.00
Kaypro to PAR 32.00
OKI-Data Serial 24.00

DISKETTES

COLOR CODE YOUR FILES

Diskettes by 3M with lifetime warranty, 5 colors. Prices are for package of 10.

5¼" Sgl side/dbl den 21.00/10
5¼" Dbl side/dbl den 28.00/10
5¼" 10 sector 22.00/10
5¼" 16 sector 22.00/10
8" Sgl side/dbl den 29.00/10
8" Dbl side/dbl den 39.00/10

MODEMS

Anchor 89.00
Hayes Smart 300 .... 199.00
Hayes Smart 1200 .... 499.00
Multi-Tech MT 212 AD (1200/300) 499.00
Novation J-Cat 300 .... 119.00
Novation Apple Cat 269.00
SSM AMC -300 (For Apple) Auto Dial .. 259.00
U.S. Robotics 212A Auto Dial 469.00

E-PROM ERASERS

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

VIDEO DISPLAY MONITORS

GREEN

BMC 12AV (15 MHZ) 80 Col/12" 80.00
Dynax GM 120 (20 MHZ) Hi-Res/80 Col/12" . 129.00
Sanyo DMZ 112 (15 MHZ) 64 Col/12" 80.00
Sakata SC-100 (18 MHZ) 80 Col/12" 119.00
USI PI-2 (18 MHZ) Hi-Res/9" 129.00
USI PI-4 (18 MHZ) Hi-Res/12" 139.00
Zenith ZM121 (18 MHZ) Hi-Res/80 Col/12" ... 94.00

AMBER

Dynax AM121 (20 MHZ) Hi-Res/80 Col/12" . 139.00
USI PI-1 (18 MHZ) Hi-Res/80 Col/9" 139.00
USI PI-3 (18 MHZ) Hi-Res/80 Col/12" 149.00

COLOR

Amdek I-12" Composite (For Apple) 299.00
BMC 9191-12" Composite (For Apple) 269.00
Sakata SC-100-13" Composite (For All) 279.00
Amdek II-12"-RGB (For IBM-PC) w/audio ... 479.00
Princeton HX-12-RGB (For IBM-PC) 489.00
Sakata SC-200-RGB (For All) 489.00

PARTS

TR1602B	2.50 ea.	TMS2716(5 + 12V)	$3.95 ea.
2114L2	$1.50 ea.	TMS2732	$4.95 ea.
TMS2532	$5.95 ea.	TMS4164(200NS)	$5.95 ea.
2716 (5V)	$3.95 ea.	TMS116-4(200NS)	$4.75 ea.

MANY OTHERS IN STOCK

CONNECTORS

RS232 Connectors
SOLDER TYPE

DB25P 2.50
DB25S 3.00
DB25 Hood 1.00
S-100 Connectors .. 10 for 25.00
DE9P 2.00

FLAT RIBBON TYPE

IDC25P 6.25
IDC25S 6.60
IDC25 Hood 1.60

Socket

		QTY 100
IDC10SKT	1 90	1 00
IDC16SKT	2 50	1 20
IDC20SKT	2 75	1 30
IDC26SKT	3 50	1 60
IDC34SKT	4 50	2 20
IDC50SKT	6 50	3 20

8Card Edge

		QTY 100
CEC26	5 00	2 70
CEC34	6 00	3 50
CEC50	7 25	4 90

AC SURGE ELIMINATORS

Lemon (6AC outlets-3 prong) ... 44.00
Lime (5'-3 prong pwr cord w/on-off switch 69.00
Orange-Ac surge + EMI filter (6 outlets) 100.00
Peach (6 outlets) 70.00

SOFTWARE

FOR APPLE — **ASHTON-TATE D-BASE II $409.00** — FOR IBM

S-100 PRODUCTS

GIGANTIC REDUCTIONS IN PRICE!

CARD CAGES/MOTHER BOARDS

**IEEE-696-No termination required*

Slots	Bare Bd	A + T	w/card cage	bare card cage
4	15.00	40.00	60.00	20.00
6	20.00	48.00	70.00	22.00
8	25.00	69.00	100.00	31.00
12	30.00	99.00	140.00	41.00
18	45.00	150.00	200.00	50.00
22	60.00	185.00	—	75.00

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

MAINFRAMES

For 2 Standard 8" Drives

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

For 2 Thinline 8" Drives

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

For 2-5¼" Disk Drives

MF + MD12 (12 slot M/B) 575.00

Standard Plain Front

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

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

SIERRA DATA

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

BOARD SETS

Best Bare Board Set Available

QTC-SBC 2/4 CPU
QTC-EXP + III 256K Memory bd.
QTC-FDC 5/8 Floppy disk controller
Bare Board Set $150.00

1) Includes manuals & assembly instructions
2) Parts available
3) Monitor & B1OS available. Add $30.00.

CLOCK/CALENDAR

S-100 Clock/Calendar by QT/Computime

CCS-BB Bare Bd 45.00
CCS-A Assembled and Tested .. **$95.00**

COMPUPRO

S-100-all assembled and tested

Disk I w/CPM 399.00
System 816A 4,000.00
20 Slot MB 210.00
CPU 8085/88 .... 389.00
Ram 17-64K 410.00
Ram 21-128K 900.00
Disk II 610.00
System Support I 350.00
Interface 4 .... 350.00
Ram 16 .... 400.00
CPU 8086/8087 .... 599.00
Active Terminator 50.00
Enclosure 2 (desk) 675.00

CALL FOR OTHERS

CPU-MEM-I/O

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

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

I/O Tech I/O + 2 Ser 3 Par Bare Bd 75.00
I/O Tech I/O + 2 Ser 3 Par A & T 300.00
I/O Tech ADA Converter Bd 400.00
I/O Tech Dual GP1B Interface Bd 575.00
I/O Tech S-Ram 128K Static 16 bit 795.00

SEE AD IN ***BYTE***

FREE DISKETTE WITH EVERY ORDER OVER $35.00

The Great Salt Lake Computer Company, Inc.

1-801-363-3314

TOLL FREE 1-800-545-2633 CONTINENTAL U.S.

Western States Save on *SALES TAX*

CALIFORNIA RESIDENTS SAVE 6% SALES TAX

8 YEARS (1976) EXPERIENCE IN COMPUTER MAIL ORDER BUSINESS

PRINTERS

BROTHER

HR-1A Par-17CPS Daisy Wheel ... 695.00
HR-1A Serial-17CPS Daisy Wheel ... 785.00

DAISYWRITER

Daisywriter 2000-16K Buffer/ 200CPS/40CPS LTR/Par ... 1,025.00
Daisywriter 2000-48K Buffer/ 200CPS/40CPS LTR/Par ... 1,050.00
Daisywriter Cable ... 32.00

DIABLO

620 (25CPS/Serial) ... 940.00
630 (40CPS/Multi-IF) ... 1,899.00

DYNAX

Dynax-15 Par-13CPS Daisy Wheel 2 color PTG-3x buff ... 475.00
Dyanx 15 Serial-13CPS Daisy Wheel ... 525.00

C, ITOH

Gorilla ... 209.00
Pro-writer I (8510A) Par 120 CPS ... 365.00
Pro-writer I (8510A) Serial 120 CPS ... 529.00
Pro-writer II Parallel-15" ... 669.00
8600 (180CPS) Par or Serial- 18 PIN 90 CPS LTR ... 1,099.00
F-10 40CPS/Diablo/Par or Serial ... 1,149.00
F-10 55CPS/Diablo/Par or Serial ... 1,425.00

EPSON — CALL

JUKI

6100-18CPS/Diablo Compatible Par/Daisy ... 565.00

MANNESMAN-TALLY

160L (160CPC-40CPS LTR 10") ... 635.00
180L (160CPS-40CPS LTR 15") ... 869.00

MPI

MPI-99G (9") Par ... 499.00
MPI 150 BI (15") Par 2K Buff w/Graphics ... 675.00
MPI 150 AI (15") Par 4K Buff w/Graphics ... 799.00
MPI 150 AI (15") Par 16K Buff w/Graphics ... 899.00

NEC

NEC7715 (w/Diablo Emulation) ... 2,025.00
NEC8023A (100CPS-Par-Graphics) ... 419.00
Serial Card ... 139.00

OKI-DATA

Microline 82A (SER & PAR-120CPS 10") ... 389.00
Microline 83A (SER & PAR-120CPS 15") ... 629.00
Microline 92 (PAR-160CPS-LTR-10") ... 519.00
Microline 93 (PAR-160CPS-LTR-15") ... 899.00
Microline 84P (PAR-200CPS-LTR-15") ... 969.00
Microline 84S (SER-200CPS-LTR-15") ... 1,059.00

STAR MICRONICS

Gemini 10X (new version) (PAR-100CPS-10") ... 299.00
Gemini 15-(15" carriage) ... 425.00

SEIKOSHA

GP100-30CPS-Par ... 199.00

SILVER REED

EXP 55OP-17CPS Daisy Wheel-PAR ... 670.00
EXP 550S-17CPS Daisy Wheel-Serial ... 690.00

TOSHIBA

P-1350 (192CPS-120CPSLTR PAR or Serial) ... 1,500.00
P-350 (100 CPS-PAR-15") ... 399.00

ALL RIBBONS AVAILABLE

MULLEN COMPUTER

TB-4A Extender BD w/Logic Probe ... 80.00
ICB-10 8 Channel Low Voltage Controller BD ... 180.00
ZB-1 ZIF Extender Test BD ... 130.00

PRINTER ACCESSORIES

DIABLO

Diablo 620-Uni-Direct Tractor ... 110.00
Diablo 620-Sheet Feeder ... 720.00
Diablo 630-Bi-Direct Tractor ... 275.00
Diable 630-Tractor Sound Cover ... 29.00
Diablo 630-SGL Sheet Feeder ... 775.00

JUKI-6100

Serial Interface ... 55.00
Bi-Dir Tractor ... 125.00
Uni-Dir Tractor ... 90.00

MANNESMAN-TALLY

Factory Quick Tear ... 230.00
Auto Front Feed ... 1,150.00

NEC

Thimbles ... 15.00
Vertical Tractor ... 219.00
Horizontal ... 152.00
Bi-Dir. Tractor ... 260.00
Cut Sheet Feeder ... 1,100.00

OKI-DATA

84 Cut Sheet Feeder ... 515.00
82/92 Tractor ... 55.00
Serial Intf. w/2K Buffer (For 82A) ... 125.00
Serial Intf. w/2K Buffer (For 92 & 93) ... 110.00
Okigraph I 82A or 83A Graphics ROM ... 35.00
Okigraph II 82A or 83A Disk for Apple ... 55.00

STAR-MICRONICS

Serial Intf. Bd ... 70.00
Serial Intf. Bd w/2K Buffer ... 120.00
Commodore 64 Intf. Bd. ... 72.00

TOSHIBA

P-1350 Cut Sheet Feeder ... 1,125.00
P-1350 Bi-Dir. Tractor ... 210.00

OUR SALTIEST DEALS

IBM Add-On Drive

* Direct Drive * Quiet (Teac Equiv.) *
*320 K/48 TPI * 5ms Track to Track *
REMEX RFD 480 ... 235.00

DISKETTES

SEPTEMBER SPECIAL

5¼" Soft Sector ... 18.00/10
SS/DD 5 Yr. Warranty
Qty/100 for ... 150.00

I/O PORT CONVERTERS

SPC-Serial to Parallel Converter ... 99.00
*Allows the use of any port
*Baud to 19.2K * Handshake Signals
PSC-Parallel to Serial ... 99.00

IBM ACCESSORIES

AST

Combo Plus (SPC 64K) ... 299.00
Mega Plus (SC, 64K) ... 299.00
I/O Plus (SC) ... 125.00

MAYNARD

Floppy Controller ... 155.00
Floppy Controller (Serial) ... 225.00
Floppy Controller (PAR) ... 205.00

QUADRAM

Quadboard 64K ... 289.00/459.00
Quadboard II 64K/256K ... 289.00/459.00
Quad Link (Allows IBM PC to use Apple II Software) ... 589.00

KEYTRONICS

Enhance your PC-with a superior keyboard ... 200.00

KRAFT

Joy Stick ... 65.00

64K UPGRADE KIT — 48.00

RANA

Add-on 320K Drive and Case ... 269.00

APPLE/FRANKLIN ACCESSORIES

ALS

CPM 3.0 Card ... 319.00

ASTAR

RF Modulator ... 19.00

EVERTEK

EV16K (16K Add-on Memory) ... 45.00

GENERIC

Disk Controller for II E ... 60.00
80 Column Video (Videx Compatible) ... 99.00

KENSINGTON

System Saver/Fan & Surge Protector ... 69.00

KRAFT

Joystick for II E ... 52.00

MICRO-MAX

View Max 80 Col/64K Mem (III) ... 149.00
Micro-Max 80 Col/128K Mem (II) ... 169.00

MICROTEK

Dumpling 64K/Interface and Graphics 64K Buffer ... 235.00
Dumpling GX-P/Par Interface Card and Cable ... 99.00
Parallel Interface Board ... 61.00
Memory 128K ... 299.00

ORANGE MICRO

Grappler & Graphics Interface ... 121.00
Buffer Bd w/16K Buffer Exp. ... 139.00
Grappler + 16K Buffer Exp. ... 179.00

TG PRODUCTS

Joy Stick ... 41.00
Paddles ... 29.00
Selecta Port ... 39.00

VISTA

A800 Floppy Controller for 8" Drives ... 300.00
A-800-1 Cable ... 27.00

QT/COMPUTIME

CCA-A Clock/Calendar ... 100.00

APPLE DISK DRIVES

Micro-Sci A-2 (35TR) ... 225.00
Micro-Sci A-40 (40TR) ... 269.00
Micro-Sci A-70 (Quad) ... 329.00
Micro-Sci Controller ... 70.00
Rana Elite I ... 259.00
Rana Elite II ... 409.00
Rana Elite III ... 529.00
Rana Controller ... 85.00
"THINLINE" ½ size 163K 40TR ... 225.00

SYSTEMS

Apple II E Starter ... 1,625.00
Columbia PC ... CALL
Compupro System 816A ... 4,000.00
Eagle PC-2 (16BIT) 2 ea. 320 K FD/64K RAM ... 2,700.00
Eagle 1620 ... 3,400.00
Eagle PC-XL ... 3,500.00
Franklin 1000 ... 895.00
Franklin 1200 Starter ... 1,625.00
Kaypro II (w/2400 software) ... 1,495.00
NEC PC800/A ... 750.00
Pied Piper - 1 ea 256 K FD/64K ... 1,139.00
Sanyo MBC 1000 (w/$2,000 software) ... 1,495.00
Televideo TS-803 ... 1,900.00
Televideo TS-1603 ... 2,475.00
Teletote I (new) w/software by Televideo ... 1,695.00

TRY US FOR OTHERS

TERMINALS

ADDS

Viewpoint-A1 (White) ... 540.00
Viewpoint-A2 (Green) ... 599.00
Viewpoint-3A + (Green) ... 540.00
Viewpoint 60-Same as Televideo 925 ... 715.00
Viewpoint 90-Same as Televideo 950 (132 Col) ... 925.00
Viewpoint Color ... 995.00

TELEVIDEO

TV910 ... 529.00
TV910 + ... 565.00
TV925 ... 715.00
TV950 ... 925.00
TV970 ... 1,095.00
RG1000/TV60 Graphics Upgrade for 925/950 ... 1,100.00

WYSE

WYSE-100 ... 725.00
WYSE-300 (Color) ... 1,125.00

MasterCard

Retail Sales
123 East 200 South
Salt Lake City, Utah 84111
Retail Hours
Monday-Friday 10 AM to 6 PM
Saturday 10 AM to 5 PM

The Great Salt Lake Computer Company, Inc.

VISA

Mail Orders
P.O. Box 3150
Salt Lake City, Utah 84110
Mail Orders
Monday-Friday 8 AM to 6 PM
(sometimes much later)
Saturday 10 AM to 5 PM

CIRCLE 39 ON READER SERVICE CARD

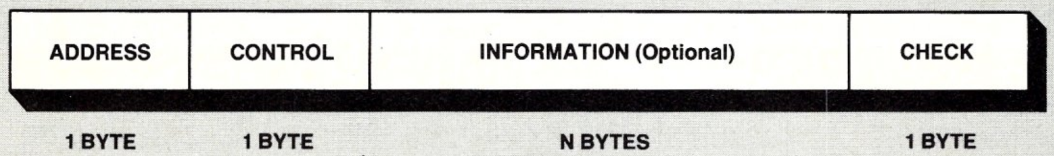


Figure 2. Message format.

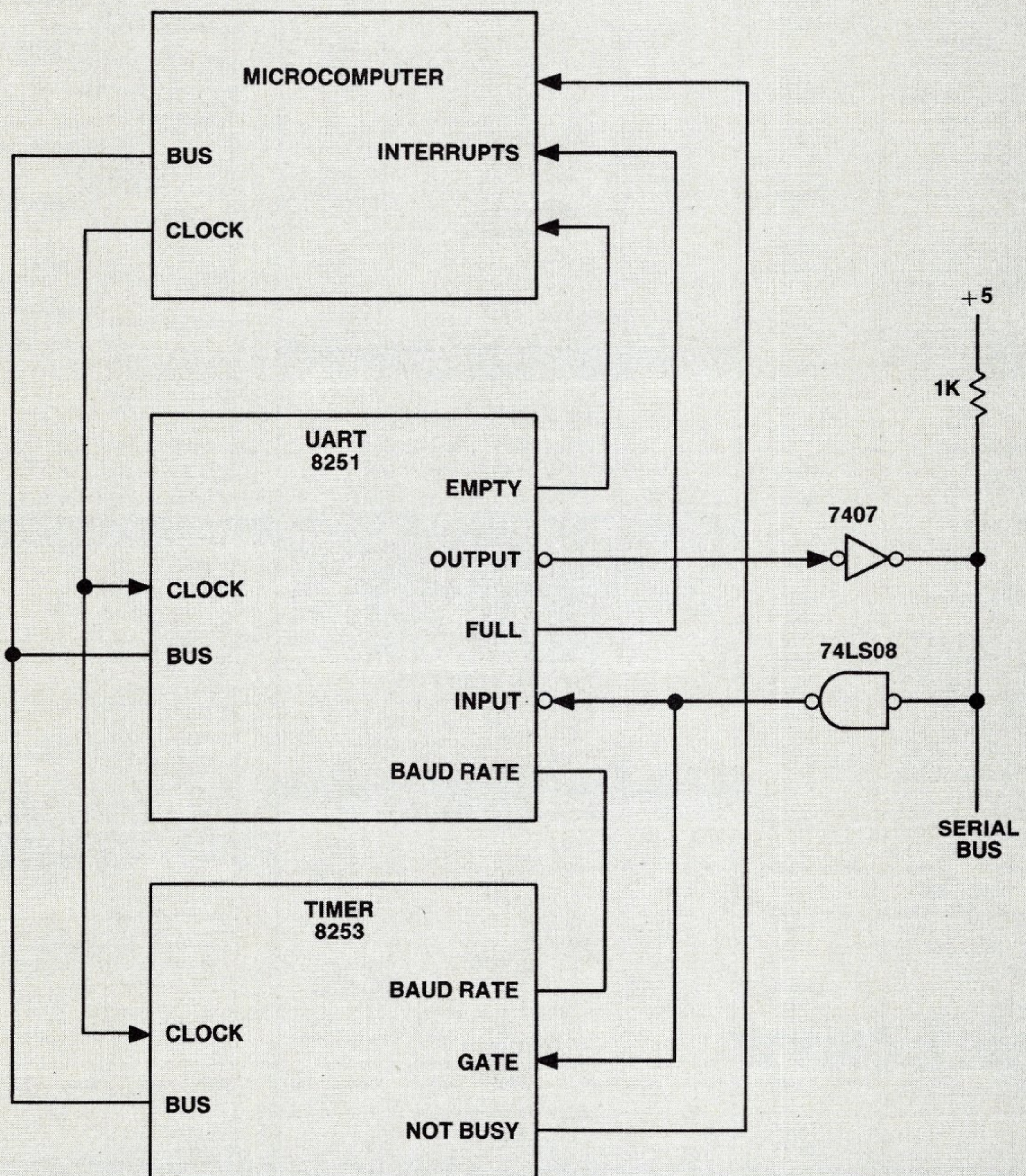
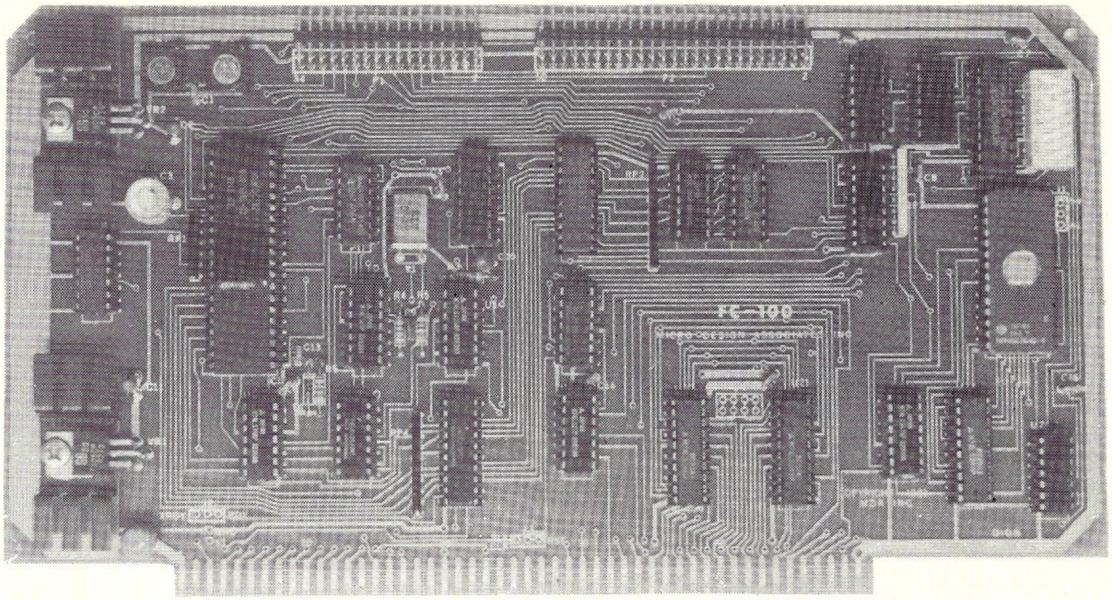


Figure 3. Serial bus interface.

THE ULTIMATE DISK CONTROLLER

STATE OF THE ART



ALL SIGNALS IEEE 696 (S-100)

$299
Assembled and Tested

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

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

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

MasterCard

VISA

TO ORDER, PHONE:

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

CIRCLE 85 ON READER SERVICE CARD

1 YEAR WARRANTY!

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

MANUAL — $15.00 — Deductible From Purchase of Board that can be contained in a message. Array subscripts begin with "s." The Receive.State is set to Waiting.For.Bus. Available, and the timer interrupt is enabled.

The timer interrupt routine is called when the bus is available. The interrupt routine then enables the UART and the receiver interrupt. The receiver interrupt then checks the next message on the bus and changes the Receive.State to Done if the message is destined for this microcomputer; otherwise it waits for the following message.

The function will wait until the Receive.State is Done. It will then return the number of bytes received and the Receive.Result, which will be Message.Received or Buffer.Too.Small.

Send Message function

The Send Message function also takes an array as a parameter. This array contains the message; the first byte is the address of the recipient. The function also assigns the Transmit information and enables the timer interrupt.

The timer interrupt routine will be called when the bus is available and will enable the UART transmitter and associated interrupt. Next, the transmitter interrupt routine will set the Transmit.State to Done when the message is sent.

The function will wait until the Transmit.State is Done, and then return the Transmit.Result. The result will be either Message.Sent or Premature.Termination. The latter occurs when the bus becomes available before the entire message is sent.

Receiver Interrupt routine

The Receiver Interrupt routine is enabled by the timer interrupt routine when the Receive.State is Waiting.For. Bus.Available. If the UART indicates an error, then the current message is ignored and the timer interrupt routine is used to wait for the next one. If the current buffer cannot hold the current message, then the Receive Message function will return with the result Buffer.Too.Small.

Otherwise the current byte of the message is placed into the receive buffer. If the Receive.State is Waiting.For.Address, then the current byte is also the address byte of the message. If the address byte matches My.Address, the message is destined for this microcomputer and the Receive.State is set to Waiting.For.Data, allowing all subsequent data bytes to be placed into the buffer. The timer interrupt will terminate the reception since that indicates the end of a message. If the message is not destined for this microcomputer, the routine will disable the receiver in the UART and wait for the end of the current message via the timer interrupt. In either case, the timer interrupt is enabled after the first byte of a message has been received.

Transmitter Interrupt routine

The Transmitter Interrupt routine is called when the UART can send another byte, and it is enabled when the time interrupt routine detects a Transmit.State of Waiting-.For.Bus.Available. The transmitter interrupt routine sets Transmit.State to Done if all bytes in the message have been sent; otherwise, the next byte of the message is indexed within the buffer and sent to the UART, which

Figure 4. Serial bus data byte timing diagram.

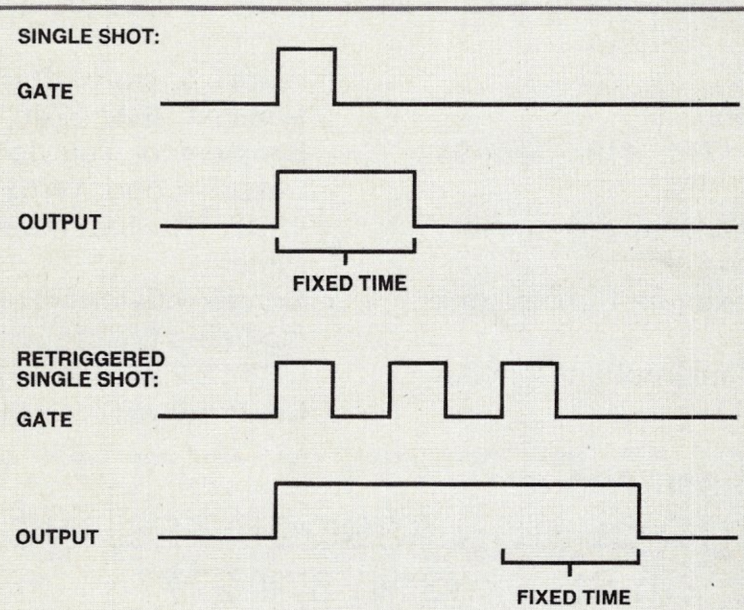


Figure 5. Single shot timing diagram.

Move up to Microsystems

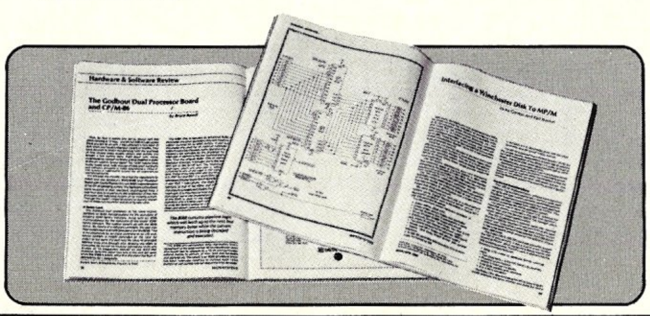
THE JOURNAL FOR SOPHISTICATED MICROCOMPUTER USERS.

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

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

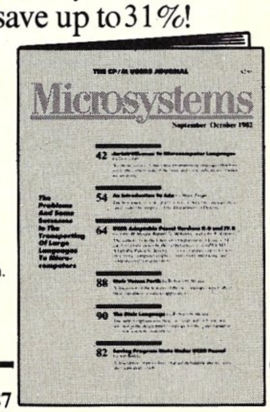
In a few recent issues of Microsystems you would have seen:

- Reviews of four C compilers: BDS C, Small C, tiny c TWO™ and Whitesmiths C.
- A programming style comparison: Digital Research PL/I-80™ vs. Microsoft BASIC.
- Twenty-seven 16-Bit DOS compared.
- Virtual segment procedures under UCSD Pascal.
- An in-depth look at MP/M-80 II.™
- A simple 6-byte hexadecimal ASCII conversion routine.
- Using CP/M's undocumented "Autoload" feature.
- An explanation of double density disk controllers.
- A comparison of five popular S-100 disk controller cards.
- OS-1—a UNIX-like DOS with a CP/M adapter.



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

To get the high-level systems information you want, join the club of serious microcomputer professionals who subscribe to *Microsystems*. Use the coupon below to order, and save up to 31%!



PROFESSIONALS PLEASE NOTE:

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

Trademarks:
XENIX: Microsoft; tiny c TWO: tiny c assoc.; PL/I-80, MS-DOS, MP/M-80 II: Digital Research.
Registered Trademarks:
CP/M, MP/M: Digital Research; UNIX: Bell Laboratories.

Microsystems CN 1987
Morristown, New Jersey 07960

YES, enter my subscription to *Microsystems* for:

- ☐ 12 issues at $21.97—I save 19%.
- ☐ 24 issues at $40.97—I save 24%.
- ☐ 36 issues at $55.97—I save 31%.

Savings based on full 12-issue subscription price of $26.97 4S259

Mr.
Ms. ______________________
(please print full name)

Company ______________________

Address ______________________

City ______________________

State ____________ Zip ____________

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

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

NEW SUBSCRIBERS ONLY

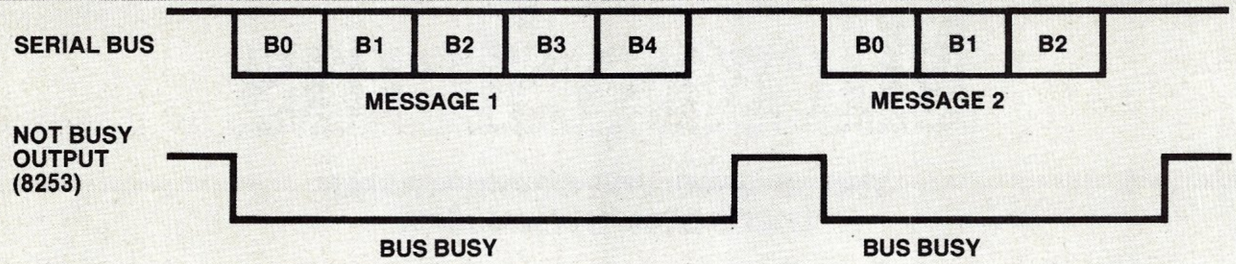


Figure 6. Serial bus timing.

sends it out on the serial bus with the start-stop protocol.

Timer Interrupt routine

The Timer Interrupt routine is used to mark the end of a message and bus availability. The routine first checks the Receive.State and sets it to Done if it is Waiting.For.Data, thus marking the end of the current message.

The routine then checks the Transmit.State. If it is Waiting.For.Bus.Available, then a message is to be sent, in which case the Transmit.State is changed to Sending.Data and the UART transmitter is enabled. If the state is Sending.Data, then an error has occurred because there are data bytes in the current message yet to be sent, even though the end of message marker has been sent. When this happens, the transmitter is disabled and the Send.Message function is notified.

The final case is when the Transmitter.State is Done. The timer interrupt is then disabled and the Receiver.State is checked. If the receiver is waiting for the bus to become available, then the receiver information is initialized and the UART receiver is enabled in preparation for receiving the first byte of the next message on the serial bus.

Summary

This serial bus interface design can be built with a minimal number of parts while providing the advantages of the multiple master collision access method. The routines presented support the hardware and the access method while still allowing user extensions. These features and its low cost make it particularly suitable for home and microcomputer applications by providing reliability and expansion unavailable in single computer systems. This interface would also prove useful for small business systems where low speed is tolerable because the traffic volume is small.

References

Intel Component Data Catalog, 1981.

IBM Synchronous Data Link Control General Information, IBM, GA273093-1.

Guidebook to Data Communications. Training Manual. Hewlett-Packard. 5955-1715.

R.M. Metcalfe and D.R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks." *Communications of the ACM,* vol. 19, no. 7, pp. 395-404, July 1976.

Receive Message Function

```
Receive Message Function (Array)
  Receive.Buffer       := Array
  Receive.Buffer.Size  := Size(Array)
  Receive.State        := Waiting.For.Bus.Available
  Enable Timer Interrupt
  Wait until Receive.State = Done
  Return (Receive.Buffer.Index, Receive.Result)
End of Receive Message Function
```

Send Message Function

```
Send Message Function (Array)
  Transmit.Size          := Size (array)
  Transmit.Buffer        := Array
  Transmit.Buffer.Index  := 0
  Transmit.State         := Waiting.For.Bus.Available
  Enable Timer Interrupt
  Wait until Transmit.State = Done
  Return (Transmit.Result)
End of Send Message Function
```

Receiver Interrupt Routine

```
Receiver Interrupt Routine
  If Input(UART.Receiver.Status)=Receive.Error Then
     Disable UART Receiver
     Receive.State := Waiting.For.Bus.Available
  Else
  If Receive.Buffer.Index=Receive.Buffer.Size Then
     Receive.Result := Buffer.Too.Small
     Receive.State :=Done
     Disable UART Receiver
     Disable Timer Interrupt
  Else
     Increment Receive.Buffer.Index
     Receive.Buffer(Receive.Buffer.Index) := Input(UART.Data)
     IF Receive.State=Waiting.For.Address Then
       If Receive.Buffer(1)= My.Address Then
         Receive.State :=Waiting.For.Data
       Else
         Disable UART Receiver
         Receive.State := Waiting.For.Bus.Available
       Enable Timer Interrupt
End of Receiver Interrupt Routine
```

Transmitter Interrupt Routine

```
Transmitter Interrupt Routine
  If Transmit.Buffer.Index=Transmit.Buffer.Size Then
     Tansmit.Result := Message.Sent
     Transmit.State := Done
  Else
     Increment Transmit.Buffer.Index
     Output to UART(Transmit.Buffer(Transmit.Buffer.Index))
End of Transmitter Interrupt Routine
```

8087 Number Cruncher

for CompuPro® and other microcomputers.*

*North Star, NEC, and Zenith versions by 3rd quarter, 1983.

"This will be a boon to those of you who wish to take advantage of **8087** for fast floating-point math but don't want to trash your present CPU board . . ." —Ray Duncan, ***DR. DOBB'S JOURNAL***

". . . the result (from the **Hudson 8087 Support Board**) is well worthwhile." —Jerry Pournelle, ***BYTE***

Perform math operations at the speed of a mainframe computer. The **Hudson 8087 Support Board**™ plugs right on your processor board, permitting you to perform numeric operations **UP TO 300 TIMES FASTER!**

Additionally, the **8087** is being supported by virtually every major language house, including Digital Research and Microsoft.

NOW 8-BIT LANGUAGE SUPPORT FOR THE 8087 SUPPORT BOARD

Typical double precision (64-bit) **benchmarks:****

Operation (5000 iterations)	FORTRAN-80	FORTRAN-87/BASIC-87 (5 MHz)	(8 MHz)
multiplication	32 sec.	2.4 sec.	1.5 sec.
division	62	2.5	1.6
sine or cosine	380	3.1	2.0
logarithm	390	2.6	1.7
square root	500	1.7	1.1

FORTRAN-87™ and BASIC-87™ put the **8087** to work for you using Microsoft's popular BASIC and FORTRAN compilers. Use your regular CP/M 2.2® and FORTRAN-80/BASIC-80 compilers on your CompuPro, Zenith, or North Star dual-processor systems interfaced with FORTRAN-87/BASIC-87 and utilize the floating point power of the **8087 SUPPORT BOARD**, increasing your math processing up to several hundred fold.

8087 SUPPORT BOARD: $495.00

(Includes 5 MHz 8087 processor; 8 MHz 8087 $95 extra)

FORTRAN-87: $200.00 BASIC-87: $200.00

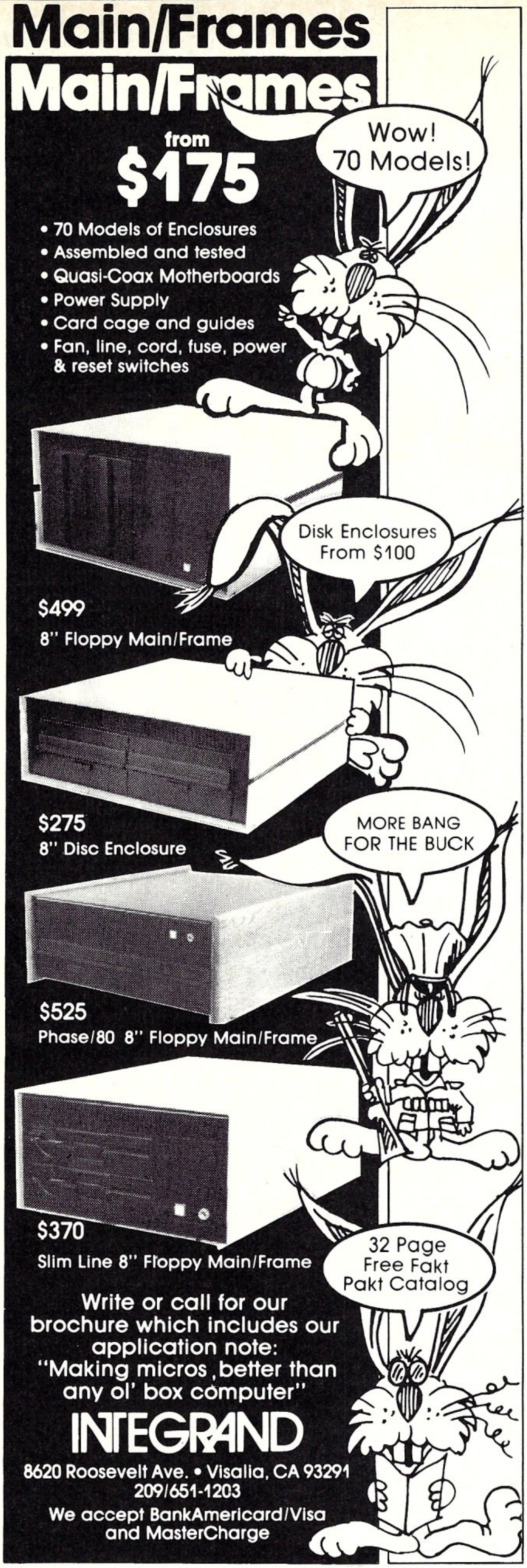
Postage and handling included. Prepaid or C.O.D. orders accepted. Give street address for UPS delivery, and allow 3-6 weeks. Dealer inquiries welcomed. Specifications subject to change without notice.

**Results were obtained on a CompuPro 8/16 computer with an 8085 running at 6 MHz and an 8088 running at 5 MHz and 8 MHz respectively.

Hudson Marketing

829 Rosewood Court, Walnut Creek, CA 94596 — **(415) 944-9680**

CIRCLE 28 ON READER SERVICE CARD



Low-Cost Local Network continued . . .

Timer Interrupt Routine

```
Timer Interrupt Routine
If Receive.State=Waiting.For.Data Then
  Disable UART Receiver
  Receive.Result := Message.Received
  Receive.State := Done
Case Transmit.State of
Waiting.For.Bus.Available:
  Transmit.State := Sending.Data
  Enable UART Transmitter
Sending.Data:
  Disable UART Transmitter
  Transmit.Result := Premature.Termination
  Transmit.State := Done
Done:
Disable Timer Interrupt
If Receive.State=Waiting.For.Bus.Available Then
  Receive.Buffer.Index := 0
  Receive.State := Waiting.For.Address
  Enable UART Receiver
End of Case Statement
End of Timer Interrupt Routine
```

CP/NET: The CP/M Network Operating System

by William G. Wong

CP/NET version 1.2 is the latest networking version of the popular 8-bit CP/M operating system from Digital Research, Inc. (DRI). It offers a number of improvements over its predecessor. It is a multiuser alternative to MP/M II, also a DRI product. CP/NET provides each user with a dedicated processor and memory, and common shared resources of a server processor for devices such as disks and printers. This approach can give significantly better performance than MP/M, since CP/NET can supply more processing power and memory per user. For example, a four-user CP/NET system would have one processor for each user, while a four-user MP/M II system would have a single processor for the whole system.

This version of CP/NET has a number of improvements, including better documentation. Performance has been increased and record locks compatible with MP/M II are supported. Basic password protection is added when accessing common server based disk resources. A simple electronic mail program is also included. Even support for a banked MP/M II server is supplied with CP/NET.

This article presents an overview of CP/NET architecture, the system programs supplied with CP/NET, the additional CP/M functions available to programmers, and a brief description of how CP/NET is implemented with some comments on system performance.

CP/NET overview

A CP/NET system typically consists of a number of CP/NET nodes connected to a CP/NET server. Each node has its own processor, memory, and network interface to the CP/NET server. The node may also have local peripherals such as a printer or disk drives. Figure 1 shows the general CP/NET architecture with one server and one requestor. A CP/NET system can also have many servers as well as many requestors.

DRI supplies for the requestor a standard Network Disk Operating System (NDOS) and a skeletal Network I/O System (NIOS), which is customized by the system implementor. These are similar to the BDOS and BIOS of CP/M. A corresponding network interface skeleton is provided for the server. These parts map into the International Standards Organization (ISO) model for computer networks, as shown in Figure 2. The operation and structure of the CP/NET requestors and servers are discussed in the rest of this section.

CP/NET requestors can be divided into two types: those with local disk drives and those without. The first type is normally referred to as a CP/NET node, while the later is called a CP/NOS node. They differ only in terms of initial loading of the CP/NET system. A CP/NET node loads the network support by running the CPNETLDR.COM program, which is located on a local drive. CP/NOS nodes usually have a ROM that contains about 4K. This ROM can either contain the CP/NOS operating system or it may act as a bootstrap loader which can load CP/NOS from the server.

In either case, the resulting memory model is shown in Figure 3. The CP/NET NDOS examines all I/O calls, including direct BIOS calls. Local device access is done through the normal CP/M BDOS and BIOS, while all remote accesses are forwarded to the CP/NET server through the CP/NET NIOS.

A CP/NET node has a smaller TPA than CP/M because of the added CP/NET NDOS and NIOS, but the reduction is usually less than 3K. A CP/NOS node usually has a larger TPA than CP/M because the BDOS and BIOS are smaller. The CCP.SPR replaces the normal CP/M console command processor (CCP). It is loaded into the top of the TPA at each warm boot and does not reduce the size of the TPA when a program is loaded.

CP/NET servers come in two flavors: those based on MP/M and those implemented under other operating systems. Both types are available from various vendors. Figure 4 shows the basic CP/NET server architectures. The number of server and interface processes is a function of the implementation, which varies depending upon the design constraints. The documentation describes the various considerations, and examples are provided. In any case, the server process at the host performs functions for the requestor and returns the results to the requestor upon completion.

An MP/M II server can be implemented using modules and guidelines supplied with the CP/NET package. In this case, it is a matter of creating a network interface routine to support the particular hardware interface, and of setting the appropriate table values. This version of CP/NET optionally allows the designer to place parts of the server and network interface processes in banked memory, thereby providing better memory utilization on the MP/M II server.

Servers implemented under other operating systems must be modeled after the MP/M II flavor. Designing this type of server may require assistance from DRI. Both implementations are described in detail in the documentation.

CP/NET documentation

The CP/NET documentation is a vast improvement over the previous version. It includes a table of contents, an index, and an excellent set of appendices, but references to other sources of information are missing. The presentation

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

is very good, with excellent figures and tables placed throughout the document.

The document is divided into three basic sections: the utilities, the programmer's guide, and the systems guide. The first two of these have been greatly improved; only the systems section needs more work. The systems section explains the various options for requestors and servers, and examples are provided for each area. There are assembly language listings for three existing implementations, but the comments are very sparse. More comments should be added to the source code and a commentary needs to be included.

CP/NET utilities

Figure 5 lists the set of utility programs supplied with CP/NET. The MAIL program has been added to the original list. It is also the only program that runs on the server as well as on the requestor. All other utilities change the logical network configuration or provide network status. Their operation is consistent with the previous release of CP/NET.

The MAIL program is worth mentioning in more detail, since it provides a method of communication between users on different nodes. It is the only real application program supplied with CP/NET.

The MAIL program can run on either the requestor or the server and uses files on the temporary disk of the server. Each node using the mail system has a file named xxMAIL.TEX, where xx is the node identification number. The MAIL program can send messages to a file or read messages from a file. The messages can contain text up to 1.7K in length, and can be deleted after they are read.

The program is menu driven, but menus are presented by scrolling the display—a simple customization for screen erase would make presentation much nicer. Data to be sent as mail can be entered from the console or can be read from a previously created file. The console entry is line-oriented. Again, a simple screen editor would be a great improvement. Another minor difficulty arises in accessing different mail files, because the mail program uses the node address to select the mail file.

The only major quibble with MAIL is that all addresses are hex values; it makes things quite confusing. However, the next release of CP/NET is slated for real names instead of numbers. It will then be possible to send mail to J. Doe instead of number 54.

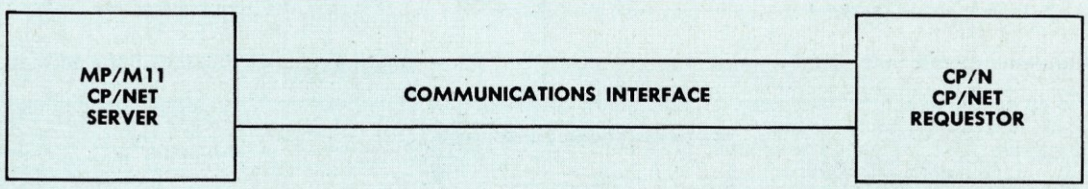


Figure 1. General CP/NET architecture.

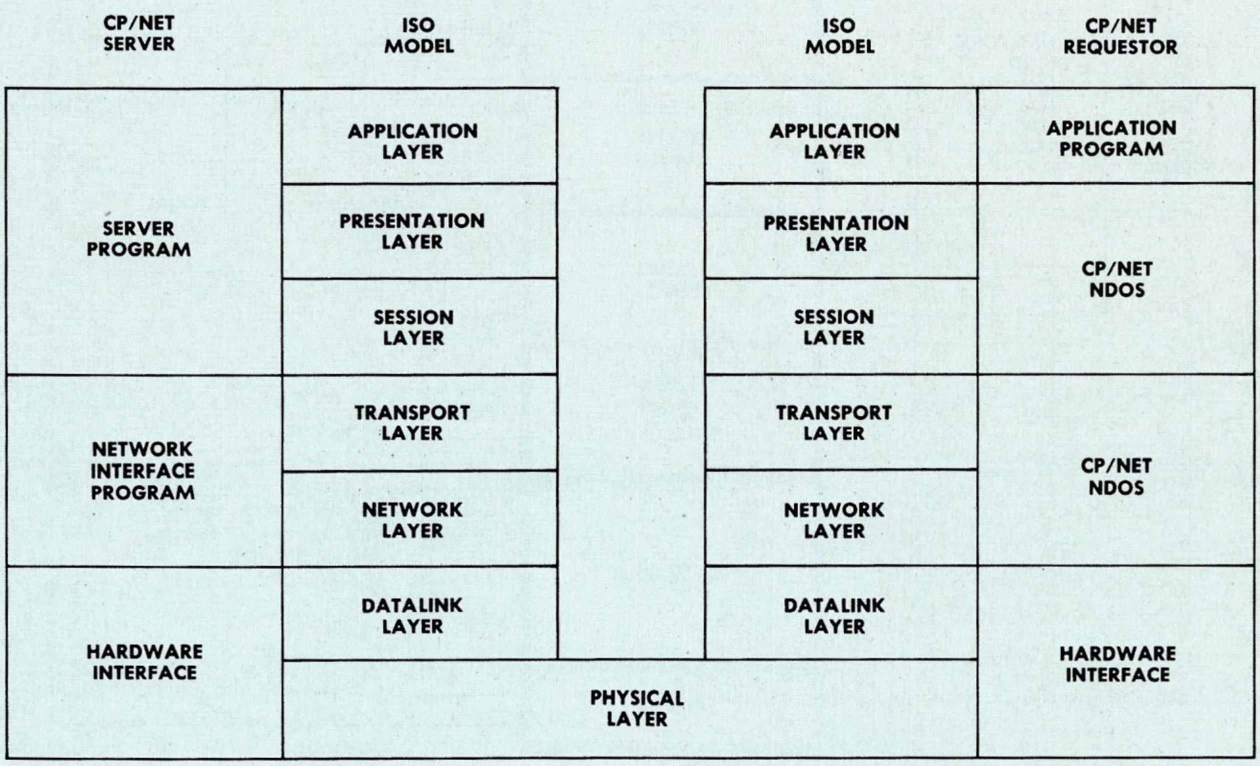


Figure 2. CP/NET and the ISO network model.

Programmer's interface

The programs running on the requestor have the standard interface to CP/M, except that a number of additional BDOS functions are recognized. These new functions are used by the programs supplied with CP/NET and are available for general use. Each function is described in more detail in the CP/NET documentation.

The new function codes can be divided into two categories. The first category contains functions that are defined under MP/M II. These deal with device, file, and record locking, along with controls for the enhanced MP/M II error message control and file password protection. The MP/M II compatible functions are:

BDOS Function	Description
38	Access Drive
39	Free Drive
42	Lock Record
43	Unlock Record
45	Set BDOS Error Mode
106	Set Default Password

These functions operate in the same manner whether the logical device maps to a physical device on the requestor or to one on the server. The functions actually do nothing if the operations access a resource on the requestor, since this can run only one program, which will have exclusive access to any local resources. However, operations which take place on the server operate in the same fashion as they do under MP/M II.

Functions in the second category are unique to CP/NET and are available only on a requestor. The following functions are used by the NDOS and the CP/NET support programs.

BDOS Function	Description
64	Login
65	Logout
66	Send Message on Network
67	Receive Message from Network
68	Get Network Status
69	Get Configuration Table
70	Set Compatibility Attributes
71	Get Server Configuration Table Address

These functions change the state of the network and also

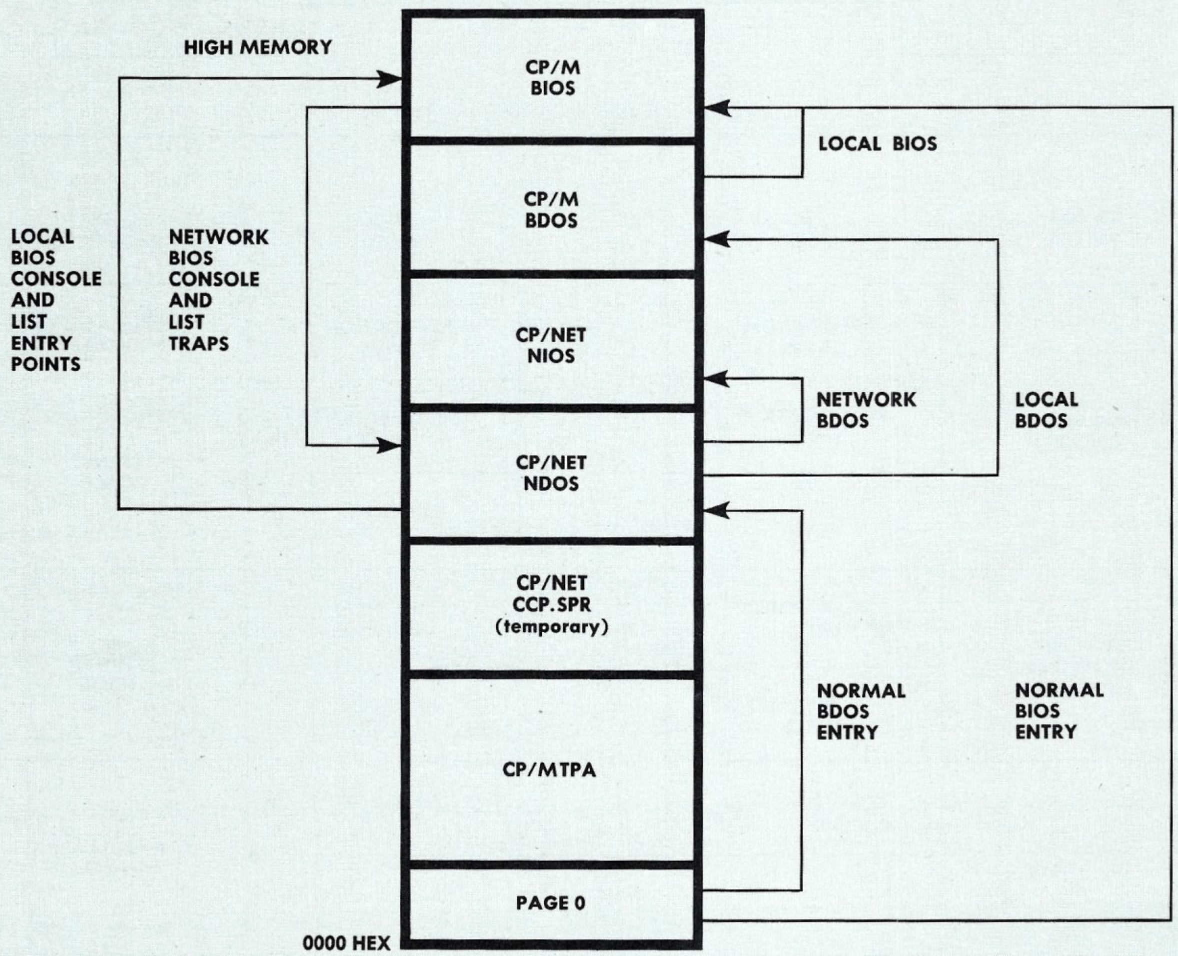


Figure 3. Requestor memory model.

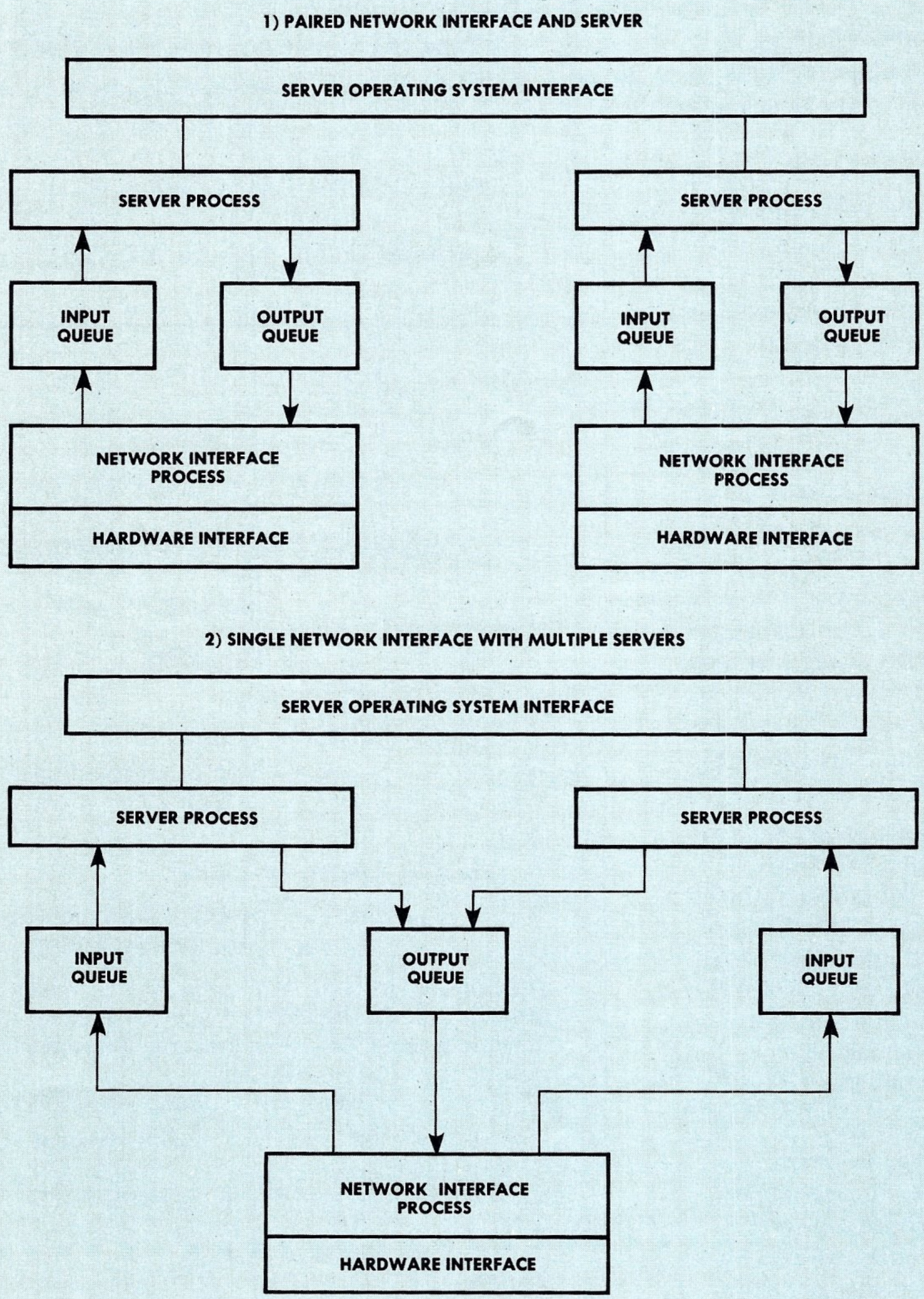


Figure 4. Server models.

provide access to the servers. The operation of these functions should be apparent from the description. The notable exception is function 70.

It seems that some programs written under CP/M may not run as expected when accessing disk resources on the server, since this is similar to a multitasking operation. For example, a program accessing a file on a server will have exclusive access to that file. This may be the proper mode of operation, but it makes simultaneous sharing of data files or overlays difficult. The compatibility attributes can be used to select the proper access mode as required.

CP/NET also performs another useful function with regard to temporary file names. It will translate any use of $$$ as a file name or file type to $xx, where xx is the requestor number, thereby allowing common applications programs to use $$$ in temporary file names. The CP/M SUBMIT program is a notable example.

The manner in which a requestor communicates with a printer attached to a server differs from the CP/M method. Although programs send characters one at a time to the logical printer device, CP/NET collects them in a 128-character buffer at the requestor. Only when the buffer is full does CP/NET transmit the entire buffer to the server. This buffering reduces the amount of network traffic due to printer output.

Implementing CP/NET

Implementing CP/NET is a two-part project. The first part consists of customizing the NIOS for the requestor; the second is to bring up the network interface on the server. Several examples are provided for both parts. Even so, implementing a CP/NET system is still the domain of a good systems programmer, especially if high performance is required. A good background in CP/M is also a prerequisite.

Program	Description
LOGIN.COM	Notifies a server that the node may want to use some of common resources. These resources are selected using NETWORK.COM.
LOGOFF.COM	Indicate that the resources of the server are no longer required.
NETWORK.COM	Indicates that a logical device is located on a server which has been logged in. For example: NETWORK B:=D:[02]
LOCAL.COM	Indicates that a logical device is located on this node.
ENDLIST.COM	Indicates that output to the server's printer is finished.
DSKRESET.COM	Resets a specified disk drive so a new disk can be mounted.
CPNETLDR.COM	This program loads the CP/NET system which consists of the SNIOS.SPR and NDOS.SPR files. It is run only ONCE to initialize the node.
CPNETSTS.COM	Prints the current network status including the physical location of all logical devices.
MAIL.COM	A menu driven electronic mail system for sending and receiving mail.

Figure 5. *CP/NET utility programs*

Putting together a CP/NET requestor requires the creation of the SNIOS.SPR file (Slave NIOS). This program supplies the hardware-specific interface to the communications network and the configuration tables used by CP/NET. There is no need to link this file with the other CP/NET files—the CPNETLDR.COM file performs this function. Thus, the task is actually simpler than creating a CP/M system. Debugging capabilities have also been placed into CPNETLDR.COM to allow testing of the NIOS on the slave.

Building a CP/NOS requestor has been simplified, too, though it is more complex than the CP/NET requestor. In fact, the recommendation is to generate the CP/NET version first, even if it is not used in the final product, because debugging is easier. More sophisticated tools such as in-circuit emulators and logic analyzers may be required for debugging a NIOS in a CP/NOS requestor.

Actually, the NIOS for CP/NOS differs only slightly from the CP/NET SNIOS.SPR file. Conditional assembly can allow one source file to generate both. The implementation process changes because the NIOS file plus all the CP/NOS modules must be linked together into one program. This is typically placed into a ROM. The documentation indicates that a 4K ROM is sufficient for most implementations.

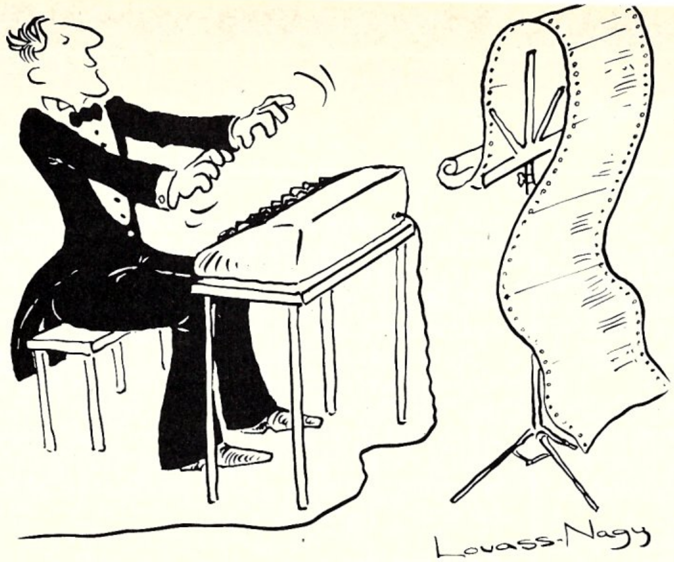
Servers are a bit more difficult to build, especially those not based on MP/M II. The documentation covers this approach, but it is best left to the experts. On the other hand, the MP/M II approach is much easier. It requires the creation of one or more network interface processes named NTWRKIPn, where "n" is the process number. These processes are very similar to the requestor's NIOS. The main difference is that these network interface processes are interrupt driven to improve efficiency and to provide better response time. Interrupts can complicate the debugging process significantly, but the results are well worth the work.

As for the CP/NET requestor, the server customization consists of a single program which contains the network interface code, the network configuration tables, and the queues necessary for communicating with the server processes supplied by DRI. Again, this customized program need not be linked to the other components; however, the MP/M II GENSYS procedure must be done each time the network interface program is modified.

The CP/NET documentation also addresses many important issues such as banked and non-banked MP/M II server, watchdog timers, and modifications of the MP/M II XIOS (eXtended I/O System) to enhance the CP/NET support. Most of these options can be added after the basic network is running.

Adding new types of requestor nodes or server nodes to an existing system is usually much easier if the existing nodes are already operating in a reliable fashion. If this is not the case, then keep the implementation simple and add functions only when the basic version is dependable.

The transparency of CP/NET indicates that a great deal of thought has been given to compatibility and flexibility.



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

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

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

The Well-Tempered Cross-Assembler

Development Tools That Work

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

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

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

Avocet Cross-assembler	Target Microprocessor	CP/M-80 Version	CP/M-86 MSDOS** IBM PC Versions
XASM05	6805		
XASM09	6809		
XASM18	1802		
XASM48	8048/8041	$200.00	$250.00
XASM51	8051	each	each
XASM65	6502		
XASM68	6800/01		
XASMZ8	Z8		
XASMF8	F8/3870		$300.00
XASM400	COP400		each
XASM75	NEC 7500	$500.00	$500.00
(Coming soon: XASM68K...68000)			

Turn Your Computer Into A Complete Development System

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

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

CP/M-80 version $150
CP/M-86 or MDOS version $195
(when ordered with any Avocet product)

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

From $495 depending on cabling and RAM installed.

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

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

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

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

CIRCLE 201 ON READER SERVICE CARD

Call Us

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

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

*Trademark of Digital Research **Trademark of Microsoft

AVOCET SYSTEMS INC.™
DEPT. 1083M
804 SOUTH STATE STREET
DOVER, DELAWARE 19901
302-734-0151
800-448-8500
TLX 467210

Performance

The overall system performance is limited by three factors: the speed of the communications network, the speed of the network server, and the load produced by the requestors. Obviously, the first two should be made as fast as possible and the third should be a low as possible to reduce the response time of a requestor.

In general, a four-requestor system gives very good response time when supported by an MP/M II server over a fast serial bus running at one megabit per second. Eight requestors can be supported, but response time then depends upon the loading of the system.

The communication link between the server and the requestors can take many forms. In general, point-to-point RS-232 serial links can provide performance close to a floppy-based system when running at 9600 baud or faster. Slower rates are possible but not recommended. High-speed serial point-to-point, bus, or loop architecture operating at about one megabit per second seem to be ideal for systems with four to 16 requestors. Very high-speed serial bus, shared memory, or a multiprocessor bus are attractive for large networks or those requiring the best response time with heavy loads.

There are several ways to increase the performance of the server. One is to build a customized server not based on MP/M—but this is a big task. MP/M-based systems can be improved by including additional message buffers in the network interface program. Substantial improvements can be seen when multiple disk data buffers are available; these reduce the likelihood of thrashing at the server, which tends to occur if only a single buffer is used. Unfortunately, multiple disk buffers usually require modification of the MP/M II XIOS, and such modification is not always advisable or possible.

Although some data base programs may provide better response time under MP/M, CP/NET generally provides better performance. This is very apparent with computation-bound programs or those using the local resources, such as the requestors console. The CP/NET architecture also provides better performance than MP/M as the number of users increases.

Summary

CP/NET is a unique product in Digital Research's product line. It provides an environment where existing CP/M programs can be run on a dedicated machine while allowing access to shared resources such as a hard disk. The transparency of CP/NET indicates that a great deal of thought has been given to compatibility and flexibility.

This new version of CP/NET is a vast improvement over the previous one. The documentation is superb, and the system is much easier to use. Also, the implementation details have finally been properly addressed.

The popularity of CP/NET has been growing along with the interest in local area networks. CP/NET is currently used by a number of major computer system manufacturers, including NCR and Corvus. These commercial systems use many different communication protocols and interfaces, but they have CP/M and CP/NET as common elements. Some systems even allow different types of requestor nodes on the same network. Digital Research may once again have provided the basis for a de facto industry standard with CP/NET, as it did with CP/M.

Things to come

Digital Research is currently working on the 8086 version of CP/NET, called CP/NET-86, with new enhancements including an improved electronic mail system. This includes server support for MP/M-86 and Concurrent CP/M-86. Servers will be able to support both CP/NET and CP/NET-86 requestors, thereby allowing 8- and 16-bit processors to run in the same network.

Concurrent CP/NET-86 is also in the works. Imagine—multiple virtual consoles, multitasking, plus the ability to share resources on the network. Digital Research should be taking the wraps off these systems soon after this article is published.

References

CP/NET Network Operating System Reference Manual, Digital Research, Inc. 1982.

CP/M Operating System Reference Manual, Digital Research, Inc. 1981.

CP/NET is available for $200 from **Digital Research, Inc.,** *P.O. Box 579, 160 Central Avenue, Pacific Grove, CA 93950, (408) 649-3896.*

Bubble Memory for the S-100 Bus

Building a magnetic bubble memory system's easier than you think. Here are the complete hardware and software details.

by Randy Reitz

The first patent for bubble memory was received by Bell Telephone Laboratories in 1966. Since that time, there have been expectations that bubble memory would take over and replace the now traditional forms of mass storage. Texas Instruments was first into the commercial bubble market with a small-capacity bubble that was used in a line of "memory" terminals. Rockwell International and National Semiconductor followed with their own devices. Now, all three of these manufacturers are out of the bubble business. The technology is too complicated and expensive.

In 1979, Intel Magnetics announced a high-capacity bubble memory, as well as a "family" of support devices. This "systems approach" distinguished Intel from the other "component" suppliers; it made the job of designing bubble memory for products easier. The Intel Magnetic Bubble Memory (MBM) is a 1-megabit device that is now available and used in several products, most notably portable terminals.

The Intel 7110 MBM is a large chip that exhibits a conservative design, but is manufacturable. The design must be conservative, since Intel has announced a 4-megabit device in the same size package. The 1-Mbit bubble chip features 320 storage loops that each hold 4096 bubbles (or bits). This represents a theoretical capacity of 1,310,720 bits—quite a bit more than the advertised 1,048,576 (1-Mbit). The "extra" bits (actually extra storage loops) represent built-in redundancy that increases the manufacturing yield as well as the device reliability.

A magnetic bubble is a magnetic domain (a region of common magnetic orientation) in an extremely thin (less than 0.001 inch) film of magnetic material. In such a thin film, the magnetic domain orientation can only be perpendicular to the surface of the film. When an external magnetic field is applied perpendicular to the film, the size of the domains shrinks until the length (determined by the thickness of the film) becomes approximately the same as the width. This makes each domain cylindrical and, when viewed from above the film, the domains appear round in shape. Hence the domains are called bubbles.

The magnetic bubbles can be made to move by using another magnetic field that is applied parallel to the film. This magnetic field affects a magnetic material that is overlaid on the surface of the film in a repeating pattern. By appropriately manipulating this magnetic field, the magnetic bubbles can be made to "follow" the pattern on the surface. Under the influence of this "rotating" magnetic field, the magnetic bubbles pass points on the film where they can be generated and detected. Hence, a magnetic bubble memory is a serial device, like a disk drive. The important distinction to bear in mind is that in a mechanical disk drive, the magnetic domains are stationary relative to the magnetic material (the disk), and the disk moves. In a bubble memory however, the magnetic material is fixed and the magnetic domains (bubbles) move (rotate) under the influence of the applied magnetic fields. No mechanical motion is involved.

Bubbles are stored in the memory in "storage loops." The storage loops are defined by deposits of magnetic material on the surface of the thin film. These deposits are called chevrons, because of their characteristic shape. The capacity of the bubble chip is determined in part by the number of these chevrons that can be deposited on the film. Each storage loop consists of two parallel rows of chevrons arranged so that at the ends the bubbles turn and continue back in the other row. Each storage loop is like a very long and skinny Indianapolis 500 speedway.

With no power applied to the MBM chip, the bubbles are held stationary by the "bias" field, a small magnetic field almost perpendicular to the chip that is created by permanent magnets above and below the chip. Field coils are also sandwiched between the magnets and the chip. The whole sandwich is wrapped in a magnetic shield that provides a return path for the bias magnetic flux. Current passed through these coils causes a magnetic field to be generated; by appropriate manipulation of the coil currents, the magnetic field is made to rotate. This causes the bubbles to "jump" from one position (chevron) on the chip to the next. The magnetic field is rotated at 50 kHz, so every 20 μsec the bubbles move one position.

Data is entered into the storage loops using an input track that touches all storage loops at one of the "turns." Another track (on the other turn of the loop) is used for output. Very simply, input data to be stored are converted to bubbles by a bubble generator at the start of the input track. Data 1 bits will cause a "seed" bubble in the generator to split in two; data 0 bits will not split the seed bubble. Bubbles that are generated serially in this fashion travel down the input track (the same rotating magnetic field that moves the storage loop bubbles also moves the track bubbles) until they all line up with the storage loops. Then a "swap" signal causes the data in the input track to be exchanged with the data in the storage loop. The swap signal causes all storage loops to exchange bubbles with the input track simultaneously; hence this is a parallel transfer as opposed to the serial bubble generation. The bubbles swapped out of the storage tracks continue to travel down the input track until they reach a "bubble bucket" at the end of the track. (I always suspected computers had bit buckets; now I have proof.)

Data is read by "replicating" bubbles in the storage loops at a replicate gate at the opposite end of the loop (the other turn) from the swap gate. To replicate a bubble, it is split in the same way that it was generated. The "replicat-

Randy Reitz, 26 Maple Street, Chatham Township, NJ, 07928

ed" bubbles are a copy of the bubbles in the storage loops and travel down the output track to a detector and eventually the bubble bucket. Since all storage loops are read and written in parallel, the data in the MBM is said to be organized as pages, each page being composed of 512 bits (64 bytes). The 512 bytes are obtained from 256 storage loops by using two bubbles per loop. So the 4096 bubble-per-loop capacity yields a 2048-page capacity. This technique increases the speed by increasing the page size. Hence each access of the MBM is done in chunks of 64 bytes.

This arrangement is called the major track-minor loop architecture. This architecture offers some advantages that are evident from the above description. First, data is read in a nondestructive fashion. Data is never removed from the storage loops when read; rather, a copy is made. This improves reliability in the case of power failure during reading. Second, data is written in parallel, so the data in the storage loop is in transition only briefly during a write. Finally, this architecture allows the chip to be designed with redundant storage loops. Although all loops, good as well as bad, have the appropriate gates connecting them to the tracks, only data from good loops can be selected for use. The chip is tested after manufacture and the good-loop information is recorded in the "boot" loop. This is a special loop that is read once when the MBM is initialized; the data is subsequently used to separate data in good loops from bad loops. The boot loop is also used to determine the location of page 0.

I said that this was a simplified explanation. Actually, the 7110 MBM is divided into 4 "quads," each with 80 loops. One boot loop serves two quads (i.e., there is one for each half of the chip); this arrangement provides redundancy for the important boot loop data. Each quad contains its own seed bubble, input track, output track and bubble detector. These quads are run in parallel to increase the data transfer speed of the device.

After this description, you may want to give up on bubble memory, since it is so complicated! Given a 7110 MBM all by itself, an engineer would need to design a board full of logic in order to manage all the bubbles in the MBM. However, Intel supplies a "family" of five types of support chips for the MBM. These special-function chips reduce a board full of logic to one corner.

Support logic

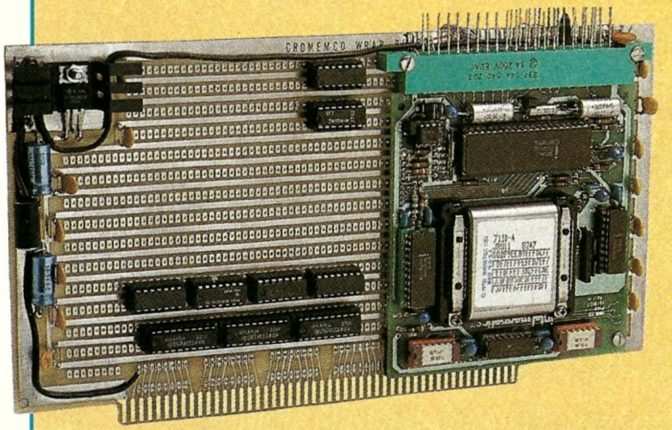
Figure 1 shows how the support chips in the Intel Magnetic Bubble Memory system interact. Here is what each one of these five chips does.

The Bubble Memory Controller (BMC) is the first member of the family. The BMC interfaces with the microprocessor bus and can control up to eight MBMs for a total bubble memory system capacity of 1 megabyte. The interface with the microprocessor that the BMC provides is very simple: the BMC appears as two 8-bit parallel ports, one for commands and status, the other for data input/output. The BMC contains a number of internal registers that are set prior to accessing the MBM system. These registers define the parameters of the MBM system for the desired access. Since the registers are set in much the same way as if the software were making a subroutine call, the registers are called the parametric registers. One unique feature is that, in addition to these parametric registers, the BMC contains a 40-byte FIFO used for buffering data transfers between the host microprocessor and the MBM system. The BMC supports three types of data transfer: 1) polled, in which the microprocessor constantly checks the status of the BMC by reading the status register; 2) interrupt driven, in which the BMC generates an interrupt signal whenever the FIFO is half full or empty; and 3) DMA, in which the BMC can handshake with a DMA controller so that the data transfer is transparent to the microprocessor.

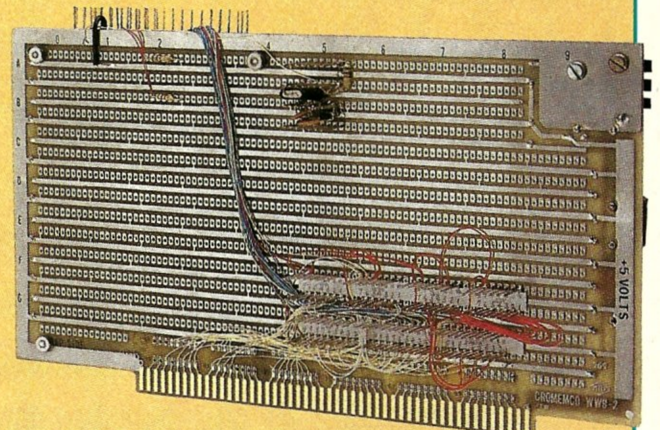
The next support chip is the Formatter/Sense Amplifier (FSA); one FSA is associated with each MBM. The FSA senses the small bubble detector signals when the MBM is read, and formats the data for a write. Formatting means that the boot loop data is used to separate data read from or written to good loops. Data from bad loops is ignored. The FSA communicates with the BMC over a 1-bit wide serial channel. Up to eight FSAs share this channel and are distinguished by time division multiplexing.

One of the unique features of the FSA is the error correction logic. If the error correcting code (ECC) option is used, 14 bits are stored in the MBM for each 256 bits of data. These 14 bits are used by the ECC algorithm to check for errors on all data transfers with the MBM, and if an error (up to 5 bits) is detected, it can be corrected. This feature significantly enhances the reliability of the whole MBM system.

The other support chips, needed for each MBM, are one Current Pulse Generator, one Coil Predriver and two coil drive transistor chips (each chip contains 4 high-power VMOS transistors). As you can guess, these devices supply the current pulses needed for the MBM's drive coils, bub-



Bubble Memory S-100 board, component side.



Bubble Memory S-100 board, wiring side, showing clock crystal (top center).

ble generate, bubble swap and bubble replicate signals. Thus, five support chips per MBM are required, in addition to one BMC to serve up to eight MBMs.

Hardware

Intel is offering a prototype kit consisting of a 1-Mbit 7110 MBM, the five support chips and a 7220 BMC. Intel has designed a 4″ by 5″ circuit card with a 44-pin connector for these chips, and supplies an application note to describe how to interface with a microprocessor bus. The interface with the S-100 bus is not very difficult, since the BMC can be considered as two parallel ports. The only requirements are to decode the S-100 address and control buses to determine when the BMC ports are to be accessed, and to split the BMC bidirectional data bus into the S-100 input/output data buses.

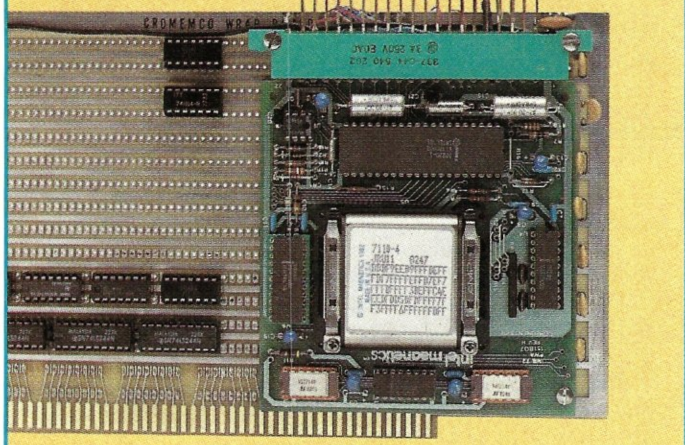
The S-100 interface I designed is straightforward. As the photos show, I simply mounted a 44-pin wire-wrap connector to an S-100 prototype card. The resulting board unfortunately requires three slot spaces: one for the board connector, one to allow space for the wire-wrap on the back, and one to allow clearance for the MBM chip on the front. The address of the BMC ports is hardwired into the address decode logic,seven S-100 address lines (A_7—A_1), as well as a combination of sIN and sOUT indicating that an input/output instruction is being executed (see Figure 2). The output of the eight-input NAND gate is used as the BMC chip select (CS*) and this, along with pDBIN, controls the gating of data to the S-100 input bus for an INP cycle. The RD* and WR* control lines for the BMC come directly from the S-100 control bus since the BMC ignores these signals without CS*. The 4-MHz oscillator (Figure 3) is copied from the Intel application notes. The extra 74LS244 to buffer the address lines is included in case additional functions are added to the card (otherwise it can be left out).

Software

Software for the MBM system reflects the simplicity of the hardware interface. The complexity of the MBM with the error correcting code, boot loop data and signal timing is insulated from the software by the BMC and the other support chips.

The first application I tried was programmed in FORTH. I use FORTH whenever I am testing new hardware since it is a very convenient language (interpretive and fast execution) for exercising hardware. I have the MVPFORTH (Mountain View Press) version of FORTH that updates Fig-FORTH to the FORTH-79 standard. MVPFORTH is in the public domain and is available from Mountain View Press.

FORTH programs are prepared and stored in units of "screens". FORTH screens are 1024 bytes and are stored on CP/M disks as 8 sectors of 128 bytes each. I have implemented the Intel MBM prototype system as a 128K disk drive and could use that interface for FORTH screen storage; however MVPFORTH provides for easy changes in the disk drives so a direct interface can be used. A direct



Bubble Memory S-100 board showing Intel subsystem.

interface will be more efficient than using the CP/M BIOS disk drivers, since the MBM system can be accessed in 1K chunks rather than 8 CP/M sector size chunks.

FORTH screen 50, (Listing 1) is the load screen for the Bubble I/O interface. This screen contains some utility words for creating lists of screens for loading and printing. Since the bubble screen I/O interface was developed subsequent to testing the MBM system with FORTH, many of these testing screens were usable in the I/O interface. Hence the list of screens needed to load the I/O interface is not sequential. The screen 50 words have nothing directly to do with the I/O interface.

Screen 51 contains some useful constants. The BMC ports are hard wired in the S-100 interface at addresses E0 and E1 hex. These ports are named BMDATA and BMSTAT. The constant REG1 is the address of the first parametric register in the BMC that needs to be set. When REG1 is written to the BMSTAT port, the Register Address Counter (RAC) is set with the value eleven (0B hex). This is the address of the Block Length Register (BLR). The BLR is two bytes wide and contains the number of sequential 64-byte "blocks" that are to be accessed. The constant PAG/BLK is 16, (10 hex); this is the value desired for the BLR in order to move 1K bytes (or one FORTH screen). The BLR controls the size of each access (number of bytes) in units of 64 bytes. This feature allows flexible organization of the bubble memory, 1024 bytes per access for FORTH screen and 128 bytes per access for CP/M disk emulation.

Once the RAC is set by writing the desired address to the BMC status port (BMSTAT), the parametric register selected can be read/written by the next access of the BMC data port (BMDATA). Each access of the BMC data port automatically increments the RAC. This means that a series of parametric registers with sequential addresses can be accessed with only one setting of the RAC. When the

The speed of the bubble BIOS is impressive; with WordStar and this article on the bubble disk, I got better than twice the speed of my double-density disk drives.

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

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

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

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

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

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

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

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

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

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

ProComp
systems inc.

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

CIRCLE 36 ON READER SERVICE CARD

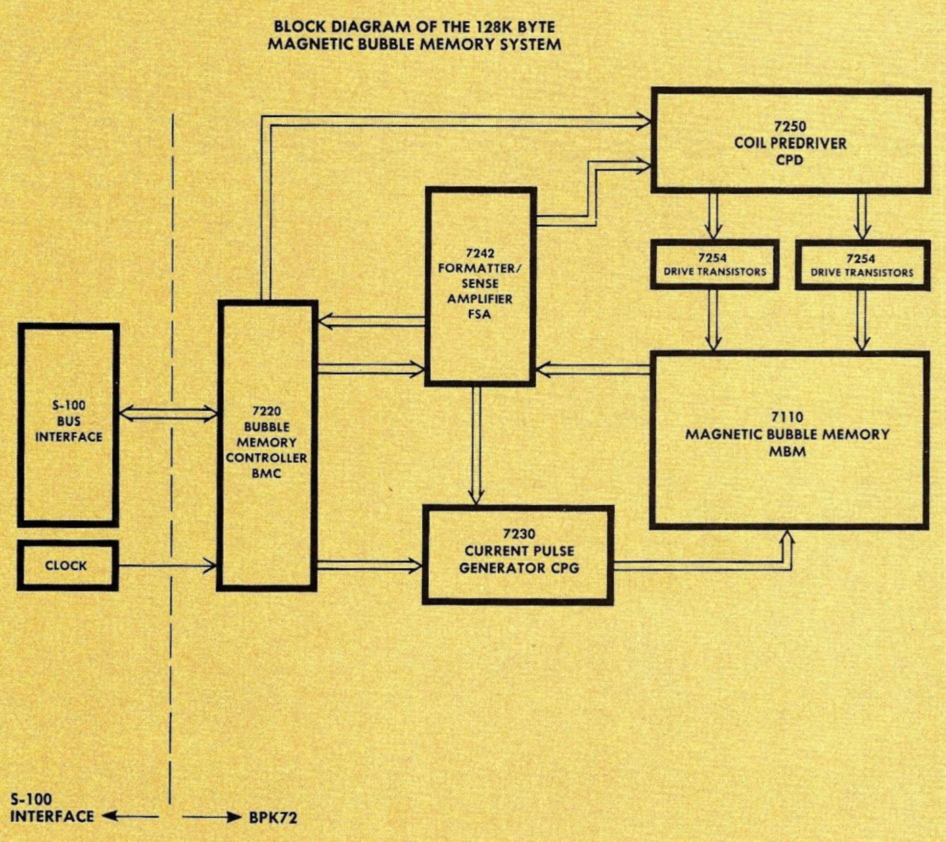


Figure 1. Bubble memory system, block diagram.

RAC reaches the value zero (the RAC is four bits wide), it stops the automatic incrementing with each data port access. The address zero in the RAC is the address of the FIFO. Hence, once the RAC reaches zero, the FIFO is accessed with subsequent reads/writes of the BMC data port. This technique simplifies programming for the MBM system.

Other constants in screen 51 are for the three BMC commands used: 1) the CIZ command for BMC initialization, 2) the CRD command for a read and 3) the CWD command for a write. Notice that the commands all have bit 4 set. When a value is written to the BMC status port, this bit distinguishes a BMC command from a value for the RAC. The constants FIFOBT and BUSYBT are used as masks when the status port is read. Finally, BPPUB is 127, the maximum FORTH screen number that can be accessed in a 128K MBM system.

Screens 35 and 36 are assembler code used to read and write to the BMC FIFO once an MBM read or write has been started. FORTH is a fast interpretor, but assembler code is needed to get the speed required to keep up with the MBM system data transfer rate. The read and write routines are similar: an inner loop transfers data whenever the FIFOBT is high (indicating that data is available in the BMC FIFO on a read, or that space is available in the FIFO on a write); the outer loop checks that the BMC is still busy with the command. If the BUSYBT bit drops before the requested number of bytes have been transferred, an error has occurred.

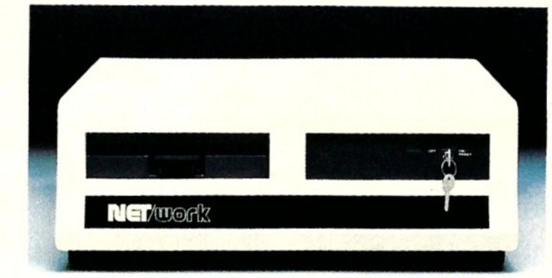
Screen 43 defines SETREG that is used to set the BMC parametric registers prior to a read, write or initialization command. As described above, first the RAC is set, then the BLR is set with the value that is on the top of the stack. The next parametric register is the ENABLE register that is used to select the error-correcting logic and type of data transfer protocol. The value contained in SETREG enables the ECC and selects a polled protocol. The last parametric register is a two-byte register that specifies the starting page address for the data transfer. To set the BLR, ENABLE and page address requires 5 bytes, since the RAC address for the first byte of the BLR is 11; after 5 writes to the BMC data port, the RAC "rolls-over" to zero and points to the FIFO.

Screen 44 defines the <MBM_R/W> word that is used by FORTH to access the MBM system. The arguments on the FORTH parameter stack are the same as those used by the disk <R/W> word. The requested screen number is checked against the maximum available (BPBUB), then the BMC parametric registers are set, and finally the flag determines whether a read or write command is sent to the BMC. The appropriate FIFO read/write routine completes the data transfer and TST_ERR checks for any problems. The BUBBLE and DISK words vector the appropriate driver into the 'R/W variable that FORTH accesses to do any mass storage I/O operation.

The complete MBM system interface for FORTH is

A reasonable hard error rate for the bubble memory is 10^{-8}; using the error correcting algorithm yields a rate of 10^{-16}—one hard error every 100 years!

May we interrupt the hoopla with a few facts?



Nothing in the noisy world of multi-user micros fits the requirements of business system integrators like our new desktop system — the MuSYS NET/work 8816™.

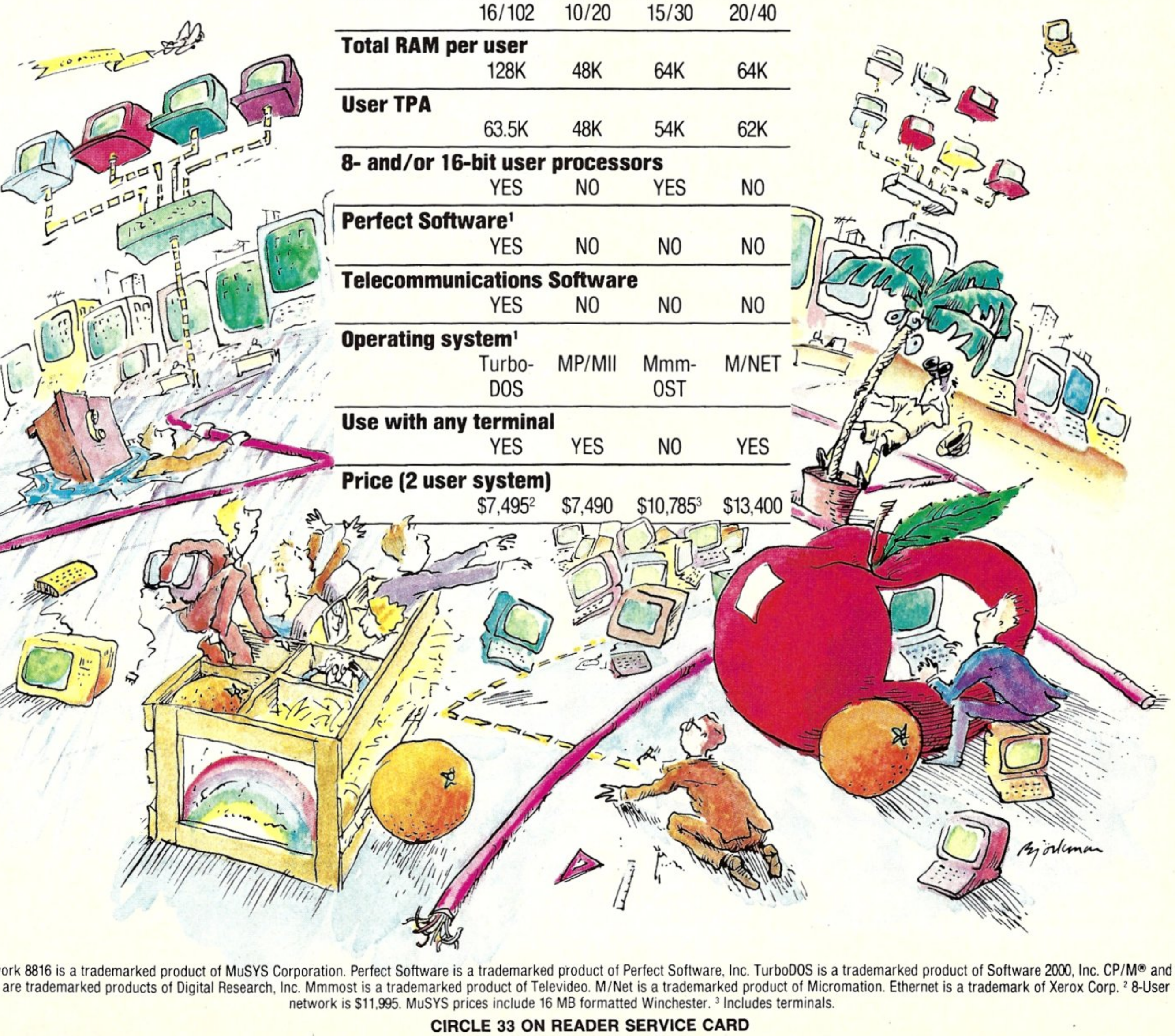
The MuSYS NET/work gives you multiprocessor architecture, extensive mass storage facilities (much greater than 8MB) and TurboDOS™, the high speed CP/M® compatible operating system. Gone are the bottlenecks that make shared-processor multi-user systems too sluggish for real world business applications. You'll also appreciate the 128K RAM per user, the local area networking capability (including Ethernet™), log-on security and other mainframe-like features that make the NET/work a truly professional, reliable business solution. Dealer and OEM customization programs available.

Get all the down-to-business facts! Call or write, MuSYS Corporation, 1752 Langley, Irvine, California 92714. (714) 662-7387 (toll-free outside California 1-800-852-5362). TWX: 910-595-1967. Cable: MUSYSIRIN.

MUSYS CORP.

We design the future.

The facts:	MuSYS NET/work 8816[1]	ALTOS 8000-10	Televideo 806/20	Micromation Mariner
No. users (min/max)	2/8	4/4	1/6	1/8
Winchester formatted Mbytes (min/max)	16/102	10/20	15/30	20/40
Total RAM per user	128K	48K	64K	64K
User TPA	63.5K	48K	54K	62K
8- and/or 16-bit user processors	YES	NO	YES	NO
Perfect Software[1]	YES	NO	NO	NO
Telecommunications Software	YES	NO	NO	NO
Operating system[1]	Turbo-DOS	MP/MII	Mmm-OST	M/NET
Use with any terminal	YES	YES	NO	YES
Price (2 user system)	$7,495[2]	$7,490	$10,785[3]	$13,400

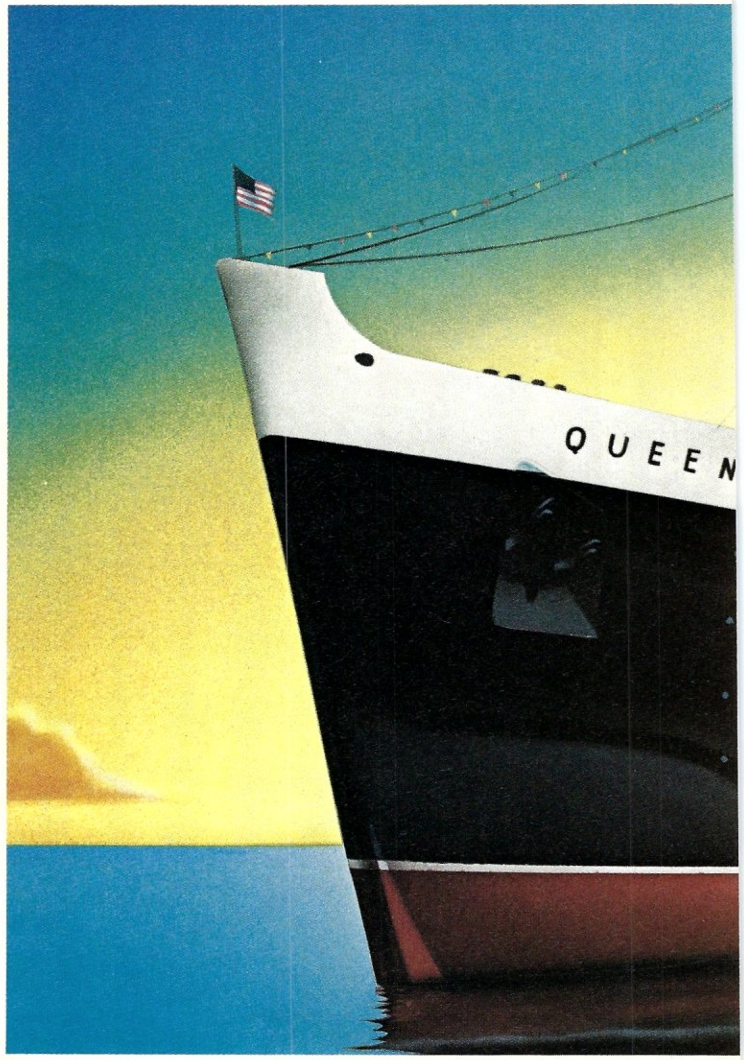


[1] NET/work 8816 is a trademarked product of MuSYS Corporation. Perfect Software is a trademarked product of Perfect Software, Inc. TurboDOS is a trademarked product of Software 2000, Inc. CP/M® and MP/MII are trademarked products of Digital Research, Inc. Mmmost is a trademarked product of Televideo. M/Net is a trademarked product of Micromation. Ethernet is a trademark of Xerox Corp. [2] 8-User network is $11,995. MuSYS prices include 16 MB formatted Winchester. [3] Includes terminals.

CIRCLE 33 ON READER SERVICE CARD

Good news, Your ship

Ashton-Tate proudly announces an unforgettable three-day informational extravaganza aboard the historic Queen Mary in Long Beach, California: The first dBASE II RunTime™ Conference.



For dBASE II old salts. And landlubbers, too.

If you're an old hand at dBASE II program publishing, our on-board entrepreneurs and experts will show you how to expand your base of operations.

And even if you've just written your first dBASE II command, we can help you launch a whole new career with dBASE II and RunTime, our applications development module.

You'll get invaluable business and technical insights from people like Ashton-Tate president David Cole. Wayne Ratliff, the man who gave the world dBASE II. SoftSel president Robert Leff. Meir Geller of Fox & Geller. Representatives from Apple and Northstar. And many, many others.

So don't miss the boat.

Over 150,000 software developers and end-users have found dBASE II to be the solution to just about every conceivable information handling need, in business, scientific and educational applications.

Join us on the Queen Mary and let us share their experiences and ours with you. It'll put some real wind into your sails.

Ashton-Tate, 10150 West Jefferson Boulevard, Culver City, CA 90230. To register, just send the coupon or call (213) 204-5570.

GUEST SPEAKER
Professor E. A. Feigenbaum, Stanford University, co-author of The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World.

OVERVIEW
Getting the most out of vertical markets.

BUSINESS BASICS.
Surviving the start-up.
Legal issues.
Advertising and publicity.

TECHNICAL WORKSHOPS
Basic programming techniques.
Advanced methods and short-cuts.
Maximizing dBASE II power.
The internals of dBASE II.
New features for dBASE II.
Documenting your applications.

THE DISTRIBUTION PROCESS.
Assessing marketability.
Choosing and using distribution channels.
How to talk to dealers.
Pricing and promotion.

DEVELOPING AND MARKETING APPLICATIONS PROGRAMS.
Entrepreneurs tell all.
Selling to OEM's and corporations.

MARKETING SPECIALIZED APPLICATIONS.
The financial market.
Products for professionals.
Record and library management.
The non-profit sector.

dBASE II® fans: has come in.

I don't want to miss the boat, so here's my registration:

Name ______________________________

Company ______________________________

Address ______________________________

City/State/ZIP ______________________________

I would like _____ registrations at $350 per person. ☐ Check

Card # ____________________ Expires ______ ☐ Money Order ☐ VISA

Signature ____________________ ☐ MasterCharge

Send to: Ashton-Tate, 10150 W. Jefferson Blvd., Culver City, CA 90230.
Or call (213) 204-5570, extension 300

ASHTON·TATE

dBASE II RunTime Conference
November 10-12, 1983.

CIRCLE 86 ON READER SERVICE CARD

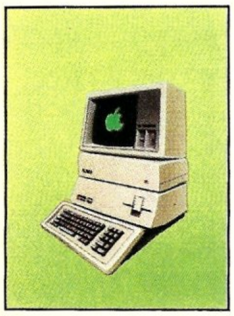
A NEW BREAKTHROUGH

SAVE UP TO 50% WITH OUR CHARTER SUBSCRIPTION OFFER!

Introducing A+, the Most Significant Advance in Apple History!

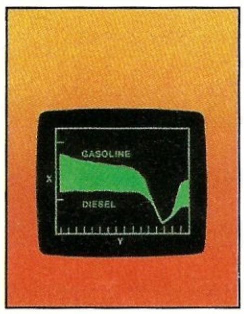
It's here! Big, bold and bursting with page after page of the latest in hardware, software, programming—and more! **A+** is the monthly magazine written exclusively for your Apple microcomputer. Each issue gets you involved with a host of fascinating features, probing updates, unique departments—and reaches to the depths of your imagination to challenge you with daring new strategies and adventures!

Here's just a sampling of what A+ is all about:



HARDWARE: From the latest advances in telecommunications and networking...to new ways to augment memory and access a wider variety of peripherals with greater ease...each issue of **A+** gets you involved.

SOFTWARE: What's new on the market and how you can make your system more powerful...plus full reviews and programming information.



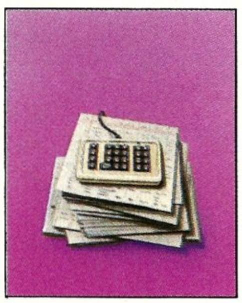
UTILITIES AND DIAGNOSTICS: In addition to language utilities, **A+** helps you troubleshoot problems... makes you faster and more efficient...with utilities ranging from record processing to data transfer.

BUSINESS AND FINANCE AIDS: Software for financial forecasting and analysis...general accounting programs...financial news retrieval services...and business graphics so you can see just where your money is going.

GAMES: Take the **A+** challenge—with hoards of innovative brain teasers...from games to help you learn spelling and arithmetic...to games that provide fun and adventure!

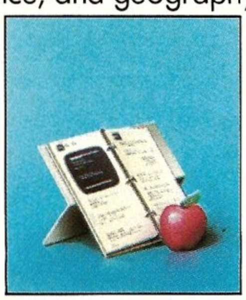
FOR YOUR APPLE™...

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



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

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



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

Plug into Savings of Up to 50% with Our Limited Time Charter Offer, Good Until Nov. 30, 1983

8H039

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

☐ One year (12 issues) only $15.49—38% off
☐ Two years only $26.49—47% off
☐ Three years only $37.49—50% off

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

Name____________________

Address____________________

City____________________

State__________ Zip__________

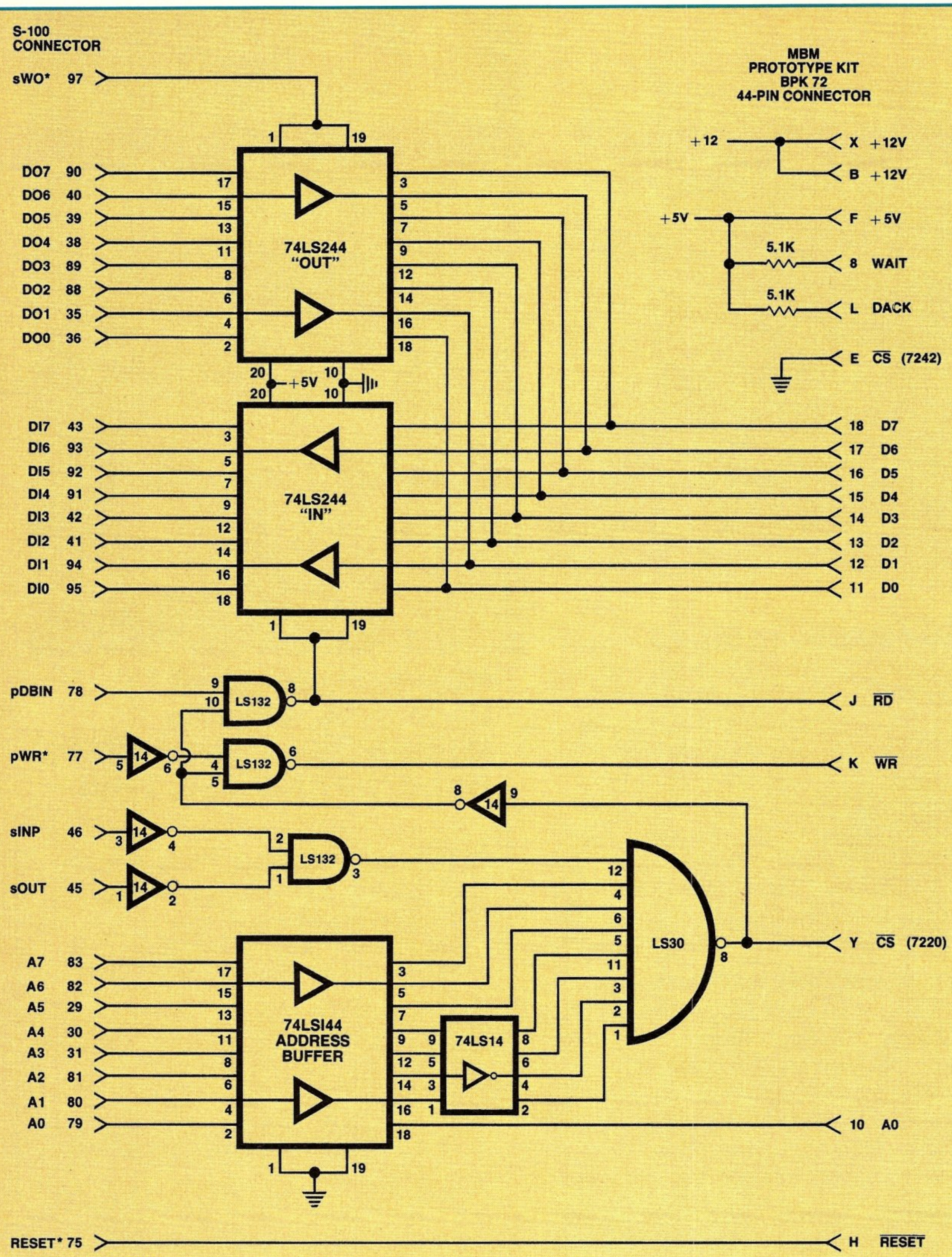
Check One: ☐ Payment enclosed. ☐ Please bill me.

Please charge my:
☐ MasterCard ☐ VISA ☐ American Express

Card No.________________ Exp. Date______

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

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



	PWR	GND
74LS244	20	10
74LS132	14	7
74LS14	14	7
74LS30	14	7

Figure 2. Bubble memory system interface.

contained in these four screens. This is possible because of the simple software interface provided by the BMC. Screens 52 and 53 contain more utility words for easily transferring screens between bubble and disk systems. BINDEX in needed since the MVPFORTH word INDEX does not use the 'R/W vector. The INDEX word is system dependent in order to speed it up. BINDEX is the corresponding word for the MBM system.

The speed of the MBM-FORTH screen interface is impressive. Of course, if the CP/M BIOS drivers were not used for disk access, a more efficient driver could be designed for disks as well.

The other application I have developed is the CP/M BIOS drivers for the bubble memory system that format the bubble memory as an 128K-byte disk drive. The assembly program in Listing 2 patches the CP/M BIOS jump table and allows CP/M to access the bubble memory as drive C: The bubble memory simulates a disk drive that has 1024 tracks, each with one sector. This organization is quite arbitrary; I chose it only to simplify the conversion of track/sector data to MBM page number. With this organization, the CP/M sector number can be ignored: it will always be 1 (one), and the MBM page number will be the CP/M track number times 2.

The code is as straightforward as the FORTH code above. Only minimal error checking is done because of the high reliability of the MBM system. I did not choose to patch the CP/M BIOS directly, since this would make the code system-dependent. Rather, the bubble BIOS is designed to run in high-memory, above the CP/M BIOS. If your BIOS is already at the top of memory, a 1K smaller CP/M system should be generated in order to provide space.

The first part of the program is a "mover" that locates the actual bubble BIOS code at its proper execution address. Since I used the CP/M ASM assembler, absolute address references in the first part of the program had to be adjusted, since it will run at 100H rather than where the ORG pseudo-op indicates. After the bubble BIOS code is moved, the BIOS jump table is patched to point to the bubble BIOS. Finally, the MBM is initialized and then the program returns.

Seven CP/M BDOS calls to the BIOS are monitored by the bubble BIOS. The bubble BIOS simply maintains a copy of the latest requested disk, track, sector and DMA address information. Whenever a home, read or write request is made, the bubble BIOS checks to see if the MBM is the selected device; if it is not, control is passed to the original BIOS disk drives.

If the MBM is selected, the bubble BIOS reads/writes two MBM pages to simulate a 128-byte CP/M disk sector. These reads/writes are similar to the FORTH code above. First, the BMC parametric registers are set using the copy of the desired track number to determine the MBM page number. The MBM block length is always 2. Next, the read or write command is sent to the BMC. Once the BMC is busy, the FIFO bit is polled and data transferred when FIFO space is available. A counter keeps track of the number of bytes transferred; when 128 is reached, the busy bit should drop. Finally the BMC status is checked.

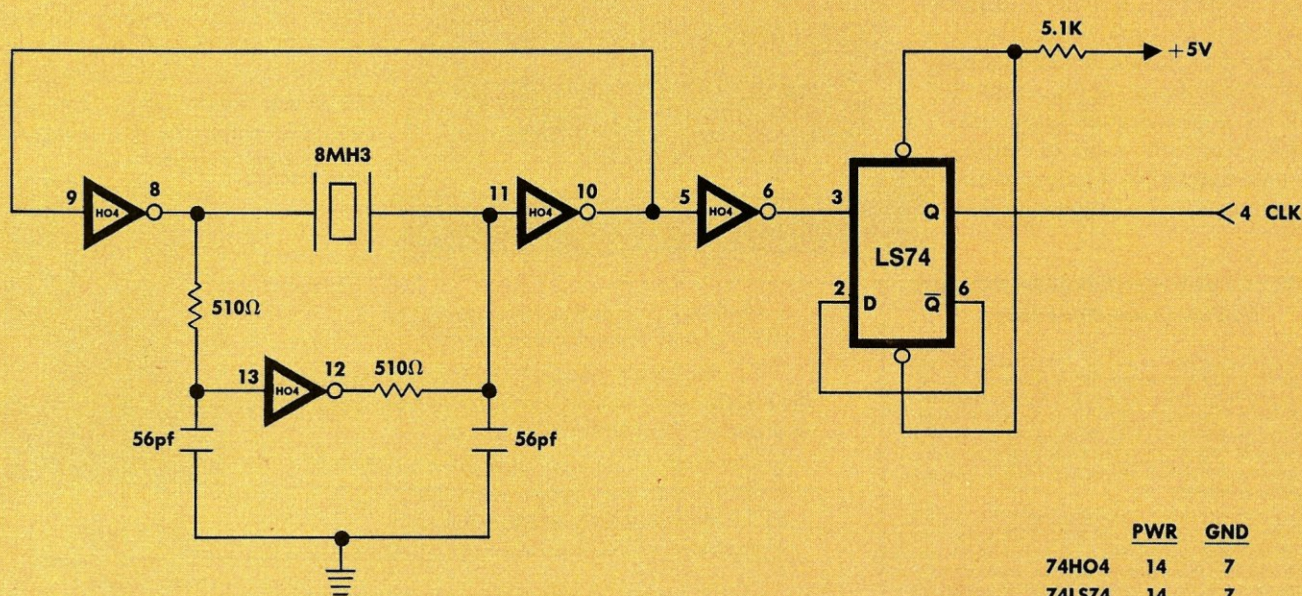
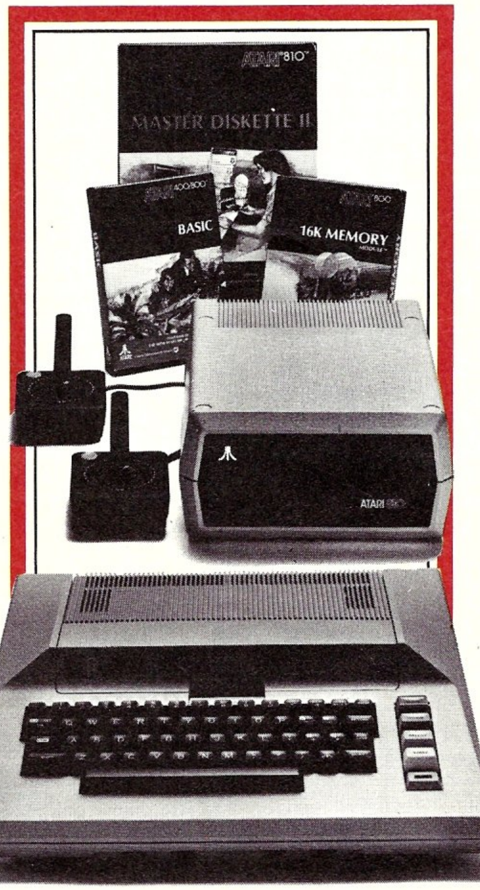


Figure 3. Clock circuit, copied from INTEL application NOTO (AP119) microprocess interface for the BPK72, June, 1981.

CREATIVE COMPUTING
WIN AN ATARI® SWEEPSTAKES

Save up to 33% on *Creative Computing*, too!

The Creative Computing Sweepstakes is open to all our readers. No purchase is necessary—and you'll receive a fantastic Atari home computer system worth over $1600 if you're the lucky winner!



How the Sweepstakes works

Just mail the attached card or the coupon below after filling in your name and address. Be sure to indicate whether you're also subscribing to *Creative Computing* at the special rates shown—you can save as much as 33%.

Then, if you win, you'll get a superb Atari personal computer system: the Atari 800 with 16K RAM and 10K ROM…an 810 Disk Drive… a 16K RAM Memory Module for extra memory… a pair of Joystick Controllers for game interaction. Use the Atari to handle small business and professional applications as well as sophisticated home processing functions—and, of course, for entertainment. Analyze your investments, learn languages, do business accounting, play video games. Programmable in Basic or Assembly, the 800 can use all Atari peripherals and can control up to four disk drives and a 60 lpm printer. In all, the 800 and its accessories form a package worth a full $1620 at retail!

You're sure to win with *Creative Computing!*

Whether you win our Sweepstakes or not, the hours you spend with a computer are certain to be winners when you subscribe to *Creative Computing*. It's the Number One magazine of computer applications and software—filled with in-depth evaluations of computers, peripherals and software, plus applications and software that you can run directly on your computer.

Why not enjoy a year or more of *Creative Computing* at our low introductory prices? You'll save up to 33% if you subscribe at the same time you enter our Sweepstakes!

Atari is a registered trademark of Atari, Inc.

OFFICIAL RULES

No Purchase Necessary

1. On an official entry form or a 3" x 5" piece of paper, hand-print your name, address and zip code. Enter as often as you wish, but mail each entry separately to Creative Computing Sweepstakes, P.O. Box 5215, Boulder, Colorado 80322. Entries must be received no later than December 31, 1983, and the drawing will be held by January 28, 1984. All entries become the property of Creative Computing, which reserves the right to reprint the name and address of the winner.
2. The winner will be selected in a random drawing from among all entries received, under the supervision of the publishers of Creative Computing, whose decision will be final. Only one prize will be awarded in this Sweepstakes. Winner will be notified by mail and may be required to execute affidavit of eligibility and release. Odds of winning will depend on the number of entries received. Creative Computing will arrange delivery of prize. Taxes are the responsibility of the winner. Any manufacturer's claims and warranties will apply, but Creative Computing makes no claims or warranties with regard to any prizes. Prize is not transferable. No substitutions for prizes.
3. Sweepstakes open to all residents of the U.S., its territories and possessions, except employees (and their families) of Creative Computing, its affiliates, and its advertising and promotion agencies. Void wherever prohibited or restriced by law.
4. For the winner's name, send a stamped, self-addressed envelope to Creative Computing Sweepstakes, Circulation Department, One Park Avenue, New York, N.Y. 10016.

OFFICIAL ENTRY FORM

Mail to: **Creative Computing Sweepstakes**
P.O. Box 5215, Boulder, Colorado 80322

☐ **YES!** Enter my name in the Creative Computing Sweepstakes and start my subscription to *Creative Computing* for the term checked:

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

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

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

☐ **NO** I don't wish to subscribe now, but tell me if I've won the Creative Computing Sweepstakes.

SH559

Mr./Mrs./Ms.____________________
(please print full name)

Address____________________ Apt.________

City____________ State________ Zip________

Offer valid only in the U.S., its territories and possessions. Please allow 30 to 60 days for delivery of first issue if you subscribe.

The location "bubdph" contains the CP/M data structure for the "disk parameter block." The sector translation table is at location "trans" and is particularly simple.

Performance

I am impressed with the speed of the bubble BIOS. I have put Wordstar and this article on the bubble disk and I get better than twice the speed of my double-density disk drives. The bubble memory system beats all disk drives in terms of access time (average 40 ms versus approximately 200 ms for a disk). However, the disk data transfer rate is higher than a single MBM (about 62 K/sec for a double-density disk versus 8.5 K/sec for a single MBM). Eight MBMs can be operated in parallel so that the data transfer rate for a 1-Mbyte bubble system would exceed the rate for a double density disk. CP/M emphasizes the access time over data transfer rate, so that a single MBM (with its slower data rate) will outperform the disk.

The reliability of the bubble memory system is also impressive. Although I have only one month's usage, it has been a pleasant experience. I assembled the BKP-72 prototype kit, wirewrapped the S-100 interface, and the MBM system came up the first time I powered up! Intel presents data that suggests the MTBF for the bubble memory is 180,000 hours. This compares to 5,000 to 10,000 hours for a floppy disk system. The Intel data shows that using the error-correcting algorithm has the effect of approximately squaring the error rate. A reasonable hard error rate for the bubble memory is assumed to be 10^{-8}; squaring this gives 10^{-16}. Hence, you should expect a hard error once every 100 years! I can live with this error rate.

One potential problem I can find in the bubble memory system is the power consumption. When data is being transferred, the single bubble memory system consumes approximately 4 watts total (counting both 5-volt and 12-volt supplies). If eight systems are operated in parallel, this power consumption rises to 28 watts. A "big" 8″ disk drive uses about 8 watts. Of course, the bubble memory system consumes this much power only when data is being read or written, since this is when the magnetic field is being rotated. The Intel literature discusses techniques for cycling the power on the MBM system to reduce the power needs.

Another potential problem is price. I estimated the price per bit of the bubble memory to be about three times more expensive than the 8″ full-height (not the new half-height) floppy disk drives. I was using the $300 price for the BKP-72 prototype kit; however, this may not be a representative price for a commercial bubble implementation. Also, Intel has announced a 4-Mbit version of this type of bubble memory; this should help reduce the price. However, the floppy and hard disk drive prices will no doubt continue to fall, so I don't expect bubble memory ever to be as cheap as disk. The advantages of bubble over disk (speed, reliability and size) will justify the price premium.

I am impressed with the Intel Magnetic Bubble Memory System. I suspect that it will find its way into many products. I don't know if any of these products will be for S-100 type machines; I guess that portable applications would predominate (S-100 machines are not noted for their portability). In addition to the fast disk drive for CP/M, there are other applications I would like to build. First, the bubble could be used to "dump core" (even though main memory is made from semiconductors, I still refer to it as "core"). It would take less than 8 seconds to dump 64K worth of memory to the bubble. The memory could then be analyzed by a debugger to retrieve data and find out what went wrong. Such a dump could be used when the system is powered down so that at the next power-up the system could be restored to the same state. This would be a clever way of handling power outages, since battery power for 10 seconds of operation enough to save everything. A more likely project will be to use the bubble memory for CP/M warm boots. Since these are done frequently, the speed of the bubble should improve system performance.

```
SCR #50
( MBM - DISK/BUBBLE I/O - LOAD SCREEN                    52983_RCR)
FORTH DEFINITIONS

: MAKELIST CREATE DEPTH DUP . 0 DO . LOOP ; \ Save list of #s

DECIMAL
SP! \ Clear stack for list of SCRs for BUBBLE words
51 35 36 43 44 52 53 MAKELIST SCR-LIST
50 51 35 36 43 44 52 53 MAKELIST PRT-LIST
: GETLIST OVER @ SWAP 2DUP > NOT ABORT" INDEX ERROR" - 2* + @ ;
: LOADLIST DUP @ 0 DO DUP I GETLIST DUP . LOAD LOOP DROP ;
: PRTLIST DUP @ 0 DO DUP I GETLIST CR LIST
  ?TERMINAL IF LEAVE THEN
  I 1+ 3 MOD NOT IF 'TITLE @ EXECUTE PAGE THEN LOOP DROP ;

SCR-LIST LOADLIST


SCR #51
( MBM - DISK/BUBBLE I/O - CONSTANTS                      52983_RCR)
BASE @      FORTH DEFINITIONS

HEX
50 CONSTANT BMDATA                 51 CONSTANT BMSTAT
10 CONSTANT PAG/BLK                11 CONSTANT CIZ
12 CONSTANT CRD                    13 CONSTANT CWD_
 1 CONSTANT FIFOBT                 80 CONSTANT BUSYBT
7F CONSTANT BPBUB  \ Max BLOCK number for BUBBLE memory
 B CONSTANT REG1    \ Address of 1st parametric reg (BLR)

BASE !
EXIT




SCR #35
( MBM - LOW LEVEL DEFINITIONS - READ FIFO                52683_RCR)
\ Read the BMC FIFO until count is reached
\ Returns flag. 0 means count bytes read
CODE RD_FIFO             ( addr count --- f )
  D POP                  \ get count
  H POP                  \ get address to load
  BEGIN                  \ loop until not BUSY
     BEGIN BMSTAT IN FIFOBT ANI \ loop until not FIFO
           0# WHILE BMDATA IN A M MOV H INX D DCX
           D A MOV E ORA 0= IF XCHG HPUSH JMP THEN
                         \ stop loop if count reached
     REPEAT
     BMSTAT IN BUSYBT ANI 0= IF XCHG HPUSH JMP THEN
  AGAIN                 \ stop above if not BUSY
END-CODE


        MOUNTAIN VIEW PRESS FORTH  VERSION 1.01.03

SCR #36
( MBM - LOW LEVEL DEFINITIONS - WRITE FIFO               52683_RCR)
\ Write the BMC FIFO until count is reached
\ Returns flag. 0 means count bytes read
CODE WR_FIFO             ( addr count --- f )
  D POP                  \ get count
  H POP                  \ get address of data to write
  BEGIN                  \ loop until not BUSY
     BEGIN BMSTAT IN FIFOBT ANI \ loop until not FIFO
           0# WHILE M A MOV BMDATA OUT H INX D DCX
           D A MOV E ORA 0= IF XCHG HPUSH JMP THEN
                         \ stop loop if count reached
     REPEAT
     BMSTAT IN BUSYBT ANI 0= IF XCHG HPUSH JMP THEN
  AGAIN                 \ stop above if not BUSY
END-CODE
```

```
SCR #43
( MBM - DISK/BUBBLE I/O - PARAMETRIC REGISTERS        52983_RCR)
BASE @   HEX

: SETREG      ( pageno blklen --- )   \ Set BMC parametric regs
  REG1 BMSTAT P!                      \ Set RAC
  DUP BMDATA P! 100 / 7 AND 10 OR BMDATA P! \ Set BLR and NFC
  60 BMDATA P!                        \ Set ENABLE reg
  PAG/BLK * DUP BMDATA P! 100 / 7 AND BMDATA P! \ Set ADDR reg
:

: TST_ERR     ( f --- )               \ Test for BUBBLE errors
  BEGIN BMSTAT P@ BUSYBT AND NOT UNTIL \ Wait for not BUSY
  BMSTAT P@ 34 AND OR ABORT" BUBBLE ERROR" ;

BASE !


SCR #44
( MBM - DISK/BUBBLE I/O                               52893_RCR)
BASE @   HEX
: <MBM_R/W>  ( addr blk f --- )     \ Replacement for DISK <R/W>
  SWAP OFFSET @ - DUP BPBUB > ABORT" BUBBLE PAGE OUT OF RANGE"
  10 SETREG                         \ Set-up the BMC regs
  400 SWAP IF CRD BMSTAT P! RD_FIFO \ do READ
   ELSE CWD_ BMSTAT P! WR_FIFO      \ do WRITE
  THEN TST_ERR ;

: BUBBLE                            \ Select MBM for screen I/O
  ' <MBM_R/W> CFA 'R/W ! ;

: DISK                              \ Selsect DISK for screen I/O
  ' <R/W> CFA 'R/W ! ;

BASE !

        MOUNTAIN VIEW PRESS FORTH  VERSION 1.01.03
SCR #52
( MBM - DISK/BUBBLE I/O - UTILITY ROUTINES            52893_RCR)

: COPY-DB             \ Copy a screen from DISK to BUBBLE
  DISK OFFSET @ + SWAP BLOCK 2- ! UPDATE BUBBLE FLUSH ;
: COPY-BD             \ Copy a screen from BUBBLE to DISK
  BUBBLE OFFSET @ + SWAP BLOCK 2- ! UPDATE DISK FLUSH ;

: M-COPY-DB           \ Multiple copy from DISK to BUBBLE
  1+ SWAP DO I . I I COPY-DB LOOP ;
: M-COPY-BD           \ Multiple copy from BUBBLE to DISK
  1+ SWAP DO I . I I COPY-BD LOOP ;

: INITIALIZE          \ Initialize the BUBBLE memory
  0 1 SETRES          \ Set-up the parametric regs
  CIZ BMSTAT P!       \ Do initialize cmd
  0 TST_ERR :         \ Test for init error


SCR #53
( MBM - DISK/BUBBLE I/O - UTILITY ROUTINES            52883_RCR)
BASE @   HEX
: .BINDEX
  DUP 1 SETREG
  PAD 40 CRD BMSTAT P! RD_FIFO
  TST_ERR
  CR 4 .R 2 SPACES
  PAD C/L -TRAILING TYPE :

: BINDEX
  CR 1+ BPBUB MIN SWAP BPBUB MIN
  DO I .BINDEX
  PAUSE ?TERMINAL IF LEAVE THEN LOOP :

BASE !
OK
```

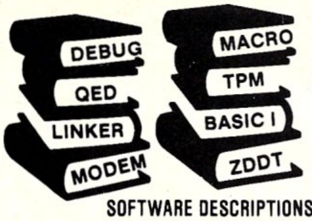
```
                    ; Patch for CP/M 2.x for Bubble Memory as a disk drive
                    ; Locate in high memory - above CP/M BIOS
                    ;
0005 =              bdos    equ     5
                    ;
FB00                        org     0fb00h  ;phony org, code really runs at 100h
FB00 =              phony   equ     $       ;used to adjust address below
0600 =              offset  equ     100h-phony ;used in DDT to "LOAD" command
                    ; First, move program to high memory
FB00 210002                 lxi     h,start-phony+100h
FB03 1100FC                 lxi     d,to
FB06 015001                 lxi     b,fin-to ;move this many bytes
FB09 CD6901                 call    move-phony+100h
                    ; Next, get original BIOS transfer table
FB0C 2A0100                 lhld    1       ;get address of 1st BIOS jump
FB0F 111500                 lxi     d,3*7   ;start with 7th entry from wboot
FB12 19                     dad     d
FB13 1115FC                 lxi     d,obios ;put copy here
FB16 011500                 lxi     b,3*7   ;move 7 entries
FB19 E5                     push    h
FB1A C5                     push    b
FB1B CD6901                 call    move-phony+100h
                    ; Next, patch BIOS transfer table with new jumps
FB1E C1                     pop     b       ;move 7 entries again
FB1F D1                     pop     d       ;put new jumps in BIOS
FB20 2100FC                 lxi     h,nbios ;these are the new jumps
FB23 CD6901                 call    move-phony+100h
                    ;
                    ; Finally, initialize the bubble memory and return to CCP
                    ;
FB26 210000                 lxi     h,0
FB29 22E1FD                 shld    track   ;initialize all RAM
FB2C 22E3FD                 shld    sector
FB2F 22E5FD                 shld    dmaad
FB32 22E7FD                 shld    diskno
FB35 CD13FD                 call    sndreg  ;set up parametric registers
FB38 3E11                   mvi     a,ciz   ;BMC initialize command
FB3A D3E1                   out     bmstat  ;send command
FB3C CDFDFC                 call    waitby  ;wait for command completion
FB3F E6B5                   ani     error   ;any errors?
FB41 C8                     rz              ;NO, go back to CCP
FB42 0E09                   mvi     c,9     ;send error message
FB44 114BFB                 lxi     d,errmsg
FB47 CD0500                 call    bdos
FB4A C9                     ret
                    ;
FB4B 427562626Cerrmsg: db      'Bubble initialization fails'
FB66 0D0A24                 db      0dh,0ah,'$'
                    ;
                    ; Move block of RAM from HL to DE for BC bytes.
FB69 7E             move:   mov     a,m
FB6A 12                     stax    d
FB6B 23                     inx     h
FB6C 13                     inx     d
FB6D 0B                     dcx     b
FB6E 78                     mov     a,b
FB6F B1                     ora     c
FB70 C26901                 jnz     move-phony+100h
FB73 C9                     ret
                    ;
                    ; This is the Bubble memory I/O for CP/M.
                    ; The Intel 7110 MBM looks as a disk drive to CP/M that has
                    ; one sector per track (128-byte sector) and 1024 tracks for
                    ; a total capacity of 128K bytes.
                    ;
0002 =              bubdsk  equ     2       ;MBM simulates drive C:
                    ;
                    ; These are the BMC commands that are used:
0011 =              ciz     equ     11h     ;the BMC initialize command
0012 =              crd     equ     12h     ;the BMC read command
0013 =              cwd     equ     13h     ;the BMC write command
                    ;
                    ; These are the BMC status bits that are tested:
0001 =              fifobt  equ     1       ;FIFO has/needs data
0004 =              uncerr  equ     4       ;an uncorrectable error detected
0010 =              timerr  equ     10h     ;a timing error has occurred
0020 =              opfail  equ     20h     ;the BMC command has failed
0080 =              busybt  equ     80h     ;the BMC is busy
00B5 =              error   equ     opfail+uncerr+timerr+busybt+fifobt
                    ;
                    ; These are the addresses of the two BMC ports:
00E0 =              bmdata  equ     0e0h    ;data
```

```
00E1 =           bmstat  equ     0e1h    ;status and commands
                 ;
                 ; The enable parametric register is set to:
0060 =           enable  equ     60h     ;enable error checking and correction
                 ;
                 ; The parametric register (BLR) starting address is:
000B =           reg1    equ     0bh
                 ;
                 ; Locate the Bubble I/O drivers here:
FC00             to:     org     0fc00h  ;desired place to run
FC00 =           start:  equ     $       ;to relocate this code
FC00 C32AFC      nbios:  jmp     home    ;new BIOS jumps
FC03 C33BFC              jmp     seldsk
FC06 C348FC              jmp     settrk
FC09 C352FC              jmp     setsec
FC0C C35CFC              jmp     setdma
FC0F C366FC              jmp     read
FC12 C3B4FC              jmp     write
                 ;
                 obios:                  ;place to save original BIOS jumps
FC15             ohome:          ds      3
FC18             oseldsk:        ds      3
FC1B             osettrk:        ds      3
FC1E             osetsec:        ds      3
FC21             osetdma:        ds      3
FC24             oread:          ds      3
FC27             owrite:         ds      3
                 ;
                 ; These routines intercept the BIOS calls, keep track of
                 ; the requested actions, and read/write the MBM if selected.
                 ;
FC2A E5          home:   push    h
FC2B 210000              lxi     h,0
FC2E 22E1FD              shld    track   ;to track 0 on a home
FC31 E1                  pop     h
FC32 3AE7FD              lda     diskno  ;see what is selected
FC35 FE02                cpi     bubdsk
FC37 C215FC              jnz     ohome   ;back to BIOS if not BUBBLE
FC3A C9                  ret
                 ;
FC3B 79          seldsk: mov     a,c
FC3C 32E7FD              sta     diskno  ;save selected drive
FC3F FE02                cpi     bubdsk  ;is it MBM?
FC41 C218FC              jnz     oseldsk ;NO, to BIOS
FC44 2130FD              lxi     h,bubdph        ;YES, return address
                 ;                       of Bubble "Disk Parameter Header"
FC47 C9                  ret
                 ;
FC48 E5          settrk: push    h
FC49 60                  mov     h,b
FC4A 69                  mov     l,c
FC4B 22E1FD              shld    track
FC4E E1                  pop     h
FC4F C31BFC              jmp     osettrk
                 ;
FC52 E5          setsec: push    h
FC53 60                  mov     h,b
FC54 69                  mov     l,c
FC55 22E3FD              shld    sector
FC58 E1                  pop     h
FC59 C31EFC              jmp     osetsec
                 ;
FC5C E5          setdma: push    h
FC5D 60                  mov     h,b
FC5E 69                  mov     l,c
FC5F 22E5FD              shld    dmaad
FC62 E1                  pop     h
FC63 C321FC              jmp     osetdma
                 ;
FC66 3AE7FD      read:   lda     diskno  ;see if MBM selected
FC69 FE02                cpi     bubdsk
FC6B C224FC              jnz     oread   ;NO, go to BIOS
FC6E CD13FD              call    sndreg  ;YES, prepare BMC parametric regs
FC71 3E12                mvi     a,crd   ;send read command
FC73 D3E1                out     bmstat  ;to BMC
FC75 2AE5FD              lhld    dmaad   ;put data read here
FC78 110000              lxi     d,0     ;counter so don't wait forever
FC7B 0E80                mvi     c,128   ;number of bytes expected
FC7D DBE1        rd1:    in      bmstat  ;wait for BMC to go busy
FC7F E680                ani     busybt  ;this means read has begun
FC81 C28DFC              jnz     rd2     ;OK, start reading
FC84 1B                  dcx     d       ;don't wait forever
FC85 7A                  mov     a,d
FC86 B3                  ora     e
FC87 C27DFC              jnz     rd1     ;keep waiting
```

```
FC8A C3ACFC              jmp     rwerr   ;it's an error
FC8D DBE1        rd2:    in      bmstat  ;is any data in FIFO?
FC8F E601                ani     fifobt
FC91 CAA5FC              jz      rd3     ;NO, check if still busy
FC94 DBE0                in      bmdata  ;read the byte
FC96 77                  mov     m,a     ;save it in RAM
FC97 23                  inx     h       ;bump pointer for next byte
FC98 0D                  dcr     c       ;any more expected?
FC99 C28DFC              jnz     rd2     ;YES, keep it up
FC9C CDFDFC              call    waitby  ;command should by complete
FC9F E6B5                ani     error   ;any errors?
FCA1 C8                  rz              ;NO, done
FCA2 3E01                mvi     a,1     ;CP/M error flag
FCA4 C9                  ret
FCA5 DBE1        rd3:    in      bmstat  ;check if BMC is still busy
FCA7 E680                ani     busybt
FCA9 C28DFC              jnz     rd2     ;OK if busy
FCAC DBE1        rwerr:  in      bmstat  ;save last status for analysis
FCAE 32E8FD              sta     status
FCB1 3E01                mvi     a,1     ;CP/M error flag
FCB3 C9                  ret
                 ;
FCB4 3AE7FD      write:  lda     diskno  ;see if MBM selected
FCB7 FE02                cpi     bubdsk
FCB9 C227FC              jnz     owrite  ;NO, go to BIOS
FCBC CD13FD              call    sndreg  ;prepare BMC parametric regs
FCBF 3E13                mvi     a,cwd   ;get write command
FCC1 D3E1                out     bmstat  ;request BMC write
FCC3 2AE5FD              lhld    dmaad   ;get data from RAM
FCC6 110000              lxi     d,0     ;counter
FCC9 0E80                mvi     c,128   ;bytes expected
FCCB DBE1        wr1:    in      bmstat  ;wait for BMC to go busy
FCCD E680                ani     busybt  ;to indicate write has started
FCCF C2DBFC              jnz     wr2     ;OK to start
FCD2 1B                  dcx     d       ;don't wait forever
FCD3 7A                  mov     a,d
FCD4 B3                  ora     e
FCD5 C2CBFC              jnz     wr1
FCD8 C3ACFC              jmp     rwerr   ;counter expired, error
FCDB DBE1        wr2:    in      bmstat  ;see if room in FIFO for more data
FCDD E601                ani     fifobt
FCDF CAF3FC              jz      wr3     ;be sure BMC is still busy
FCE2 7E                  mov     a,m     ;get data to write
FCE3 D3E0                out     bmdata  ;send to BMC
FCE5 23                  inx     h       ;point to next byte
FCE6 0D                  dcr     c       ;any left to do?
FCE7 C2DBFC              jnz     wr2     ;continue if so
FCEA CDFDFC              call    waitby  ;wait for command completion
FCED E6B5                ani     error   ;any errors?
FCEF C8                  rz              ;NO, done
FCF0 3E01                mvi     a,1     ;CP/M error flag
FCF2 C9                  ret
FCF3 DBE1        wr3:    in      bmstat  ;BMC must be busy
FCF5 E680                ani     busybt
FCF7 C2DBFC              jnz     wr2     ;OK, continue to wait for FIFO
FCFA C3ACFC              jmp     rwerr
                 ;
                 ; This routine waits for BMC to go not busy
                 ; Returns status in A-reg and saves status in RAM
FCFD 110000      waitby: lxi     d,0     ;don't wait forever counter
FD00 DBE1        wby1:   in      bmstat
FD02 E680                ani     busybt
FD04 CA0DFD              jz      wby2    ;it's not busy
FD07 1B                  dcx     d
FD08 7A                  mov     a,d
FD09 B3                  ora     e
FD0A C200FD              jnz     wby1
FD0D DBE1        wby2:   in      bmstat
FD0F 32E8FD              sta     status
FD12 C9                  ret
                 ;
                 ; This routines sets-up the BMC parametric registers
                 ; It uses the track number saved at the last settrk call.
FD13 3E0B        sndreg: mvi     a,reg1  ;address of first (BLR) register
FD15 D3E1                out     bmstat  ;register address, not command
FD17 3E02                mvi     a,2     ;BLR LSB always 2 to get 2x64=128 bytes
FD19 D3E0                out     bmdata
FD1B 3E10                mvi     a,10h   ;BLR MSB always 0001 (NFC) for two channels
FD1D D3E0                out     bmdata
FD1F 3E60                mvi     a,enable
FD21 D3E0                out     bmdata  ;set ENABLE register
FD23 2AE1FD              lhld    track   ;get desired track
FD26 29                  dad     h       ;MBM page is twice track
FD27 7D                  mov     a,l     ;address LSB
FD28 D3E0                out     bmdata
```

Bubble Memory continued . . .

```
FD2A 7C                mov     a,h         ;address MSB
FD2B E607              ani     7           ;set bubble number to 0
FD2D D3E0              out     bmdata
FD2F C9                ret
                ;
                ; This is the CP/M data structure needed to define a disk
                ; the "Disk Parameter Header"
FD30 4FFD0000   bubdph: dw      trans,0h
FD34 00000000           dw      0h,0h
FD38 50FD40FD           dw      dirbuf,bubdpb
FD3C E1FDD0FD           dw      csv0,alv0
                ;
                ; These are the "characteristics" of the bubble for CP/M
                ; the "Disk Parameter Block"
FD40 0100       bubdpb: dw      1           ;"sectors" per "track"
FD42 03                 db      3           ;block shift factor
FD43 07                 db      7           ;block mask
FD44 00                 db      0           ;extent mask
FD45 7F00               dw      128-1       ;disk size - 1
FD47 1F00               dw      31          ;directory maximum entries
FD49 80                 db      80h         ;alloc 0 - 32 directory entries
FD4A 00                 db      0           ;alloc 1
FD4B 0000               dw      0           ;check size - not removable
FD4D 0000               dw      0           ;track offset
                ;
                ; This is the sector translation table
FD4F 01         trans:  db      1
                ;
                fin:                        ;end of code and initialized memory
                ;
FD50            dirbuf: ds      128         ;scratch directory area
FDD0            alv0:   ds      17          ;allocation vector
FDE1            csv0:   ds      0           ;check vector
FDE1            track:  ds      2
FDE3            sector: ds      2
FDE5            dmaad:  ds      2
FDE7            diskno: ds      1
FDE8            status: ds      1
                ;
FDE9            end
```



Z80 Software

SOFTWARE DESCRIPTIONS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 sectcr/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

Use the New Radio Shack Portable Computer with a CP/M System

by Bill Machrone

No, this isn't a review of Radio Shack's TRS-80 Model 100. In fact, *Microsystems* may distinguish itself by being the only microcomputer magazine *not* to review it.

Most of you have probably read at least one review, but in case you've been living under a rock, let me summarize: It's a great machine, a benchmark of achievement in the current state of the art. It's unlikely that it will be a satisfactory "only" computer, but man, what an accessory!

The Model 100 is endlessly practical, allowing you to take some of your most needed computing capabilities along with you. For me, that's word processing. This article was written in an airport and on various means of public transportation, from a 737 to a subway. How's that for sublime to ridiculous? When I finally got home, I set the telecommunications rate to 9600 baud, plugged it in at the console of my CP/M system, booted, then said "PIP RS100.ART=CON:". I then pressed the "upload" function key on the Model 100 and, as if by magic, the article was transfered to disk. Then I used good old WordStar to put the finishing touches on it, and here it is.

The real purpose of this article, though, is to acquaint you with a few undocumented items that may enable you to increase the machine's utility in some application you may have in mind.

First off, there is a Z-19 terminal lurking within. The screen responds to the same escape sequences as the Z-19/VT-52, making it an ideal remote portable terminal for use on a timesharing system. Here's a list of the sequences I found operational:

ESC E	Clears the screen
ESC H	Homes the cursor
ESC J	Clears to end of screen
ESC A	Cursor up
ESC B	Cursor down
ESC C	Cursor right
ESC D	Cursor left
ESC L	Insert line
ESC M	Delete line
ESC K	Erase to end of line
ESC Y	(r,c) Direct cursor addressing
ESC l	Erase line
ESC p	Reverse video on
ESC q	Reverse video off

A formfeed (chr$(12)) will also clear the screen.

The function keys act locally in telecommunications mode; nothing goes out over the line. The "paste" key, used by the text editor, transmits the contents of the paste/delete buffer from the last editing session. Despite the Model 100's extensive auto logon capabilities, you could use this feature for some trick password insertion. The only keys that transmit and the codes they send are:

Pause	Control-S (DC3)
Left Arrow	Control-] (GS)
Right Arrow	Control-\ (FS)
Up Arrow	Control-^ (RS)
Down Arrow	Control-_ (US)

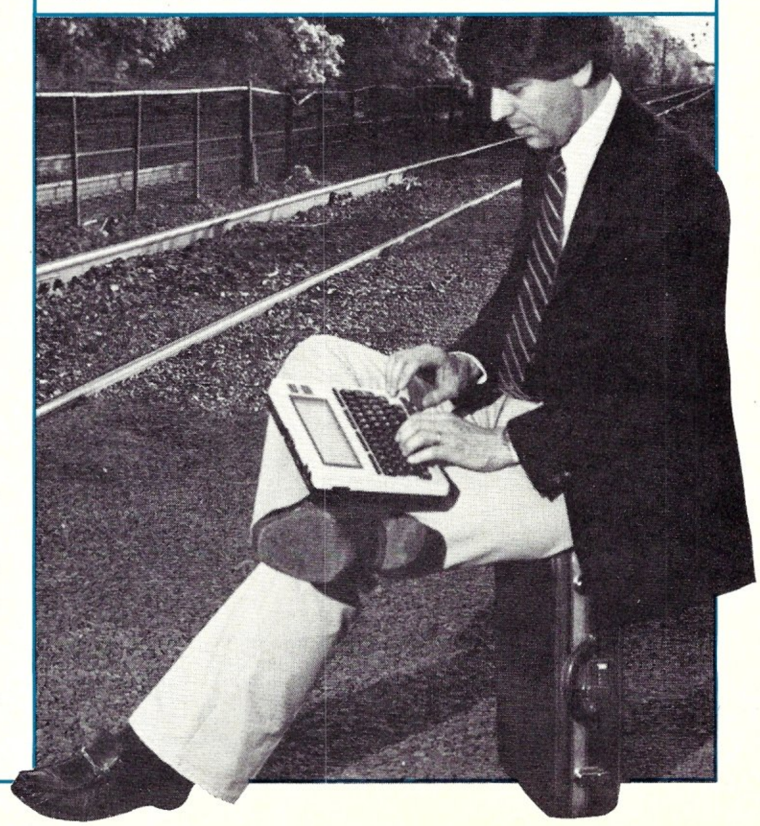
Using the above, I was able to define a termcap (terminal capabilities) file entry in UNIX and use it on the dial-up port. It's pretty much like the VT-52 entry except for the screen size and the reverse video. All my full-screen programs work, although they are somewhat cramped and there's a fair bit of screen paging.

Why didn't Radio Shack document these functions? Maybe Microsoft forgot to tell them they were in there.

Telecommunications

If you are going to do any remote-computing/timesharing applications with the Model 100, you had better be aware of just how slow the LCD display is. How slow is it? It drops characters at 1200 bud when XON/XOFF handshaking is not enabled. Fortunately, the TELCOM program supports this popular method of handshaking, which covers UNIX and MP/M, to name two possible host systems. On the other hand, the display scrolls very smoothly and is always legible. This won't be a problem if you use the internal 300 baud modem, but a direct hookup or fast modem needs the old handshake.

TELCOM's option setting can be tricky. For instance, you can't set it up for 8-bit transfer and "ignore" parity; it



Bill Machrone, 121 North Avenue, Fanwood, NJ 07023

insists on begin told there is no parity. If you attempt to give TELCOM parameters that it doesn't like, it leaves the old ones unchanged and beeps at you. You might agree that there's no room in the ROM for error messages, but how about the manual? TELCOM is about the only thing on the Model 100 (Why can't I bring myself to call this machine a TRS-80? It's too good for that!) that uses esoteric codes, here to set the baud rates. Rather than toting the voluminous (and good, except for some typos) manual with you, just set up a document file with the baud rates and codes and leave it in memory all the time. You'll never miss the couple of hundred bytes.

The first thing you'll discover about using the RS-232 port is that no cable with a hood fits. You either have to remove the hood, or use an Ansley-type plastic connector that doesn't have one. Radio Shack has a 10″ extension that will do the job for a mere 18 bucks. I think you could file some of the offending plastic away for a bit less than that.

Disks and extensibility

You have probably heard by now that the machine is destined for disk I/O. The instructions buried in the Basic ROM, DSKI, and DSKO seem to suggest some sort of serial access to the disk. Of course, with the additional plug-in ROM capability, it's anybody's guess what functions may be included. I understand that NEC's version of the machine, currently being sold in Japan, has a micro-disk accessory. No word yet on whether there are support software changes.

I said before that this wasn't a review, but there is one operational mode that is somewhat deficient in comparison to the performance you may be used to on a normal computer terminal. The insert mode during text editing is quite slow. This is because the software is continually updating the display, reforming the lines and rearranging memory. The keyboard obviously generates an interrupt, or dumps into a FIFO, or both, since you never lose a character. You have to take what you're typing on faith, though, since it takes a second or two for the display to catch up with your keying. An alternative, if you have large revisions to a document, is to go to the end of the document, enter your next text and use the cut and paste feature to move it to the desired position.

I'm sure that a lot of the slowness is due to the 2.4 MHz clock rate. It would be handy to have an "I don't give a hoot about the battery life" switch that would double the clock rate when you wanted fast operation. CMOS, of course, consumes more power when you switch it faster. That really isn't a cause for concern when you're running off the AC adaptor.

Finally, I can't wait to get my hands on Radio Shack's first machine language program so I can see whether there is an operating system with system calls to a standard address or if the technique is to call specific addresses in the ROM. The answer to that question will, I think, indicate whether the Model 100 is to be a truly extensible cornerstone to a product line or just an entity unto itself with limited upgrade possibilities. ■

SIMPLE.

StationMate.™ For uncomplicated computer connections.

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, microcomputers, 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.*

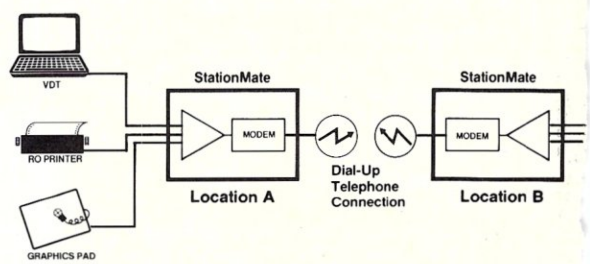
StationMate provides local area networking with XLAN™, the simple communications concept from Complexx. All you need is StationMate and an inexpensive shielded-pair cable.

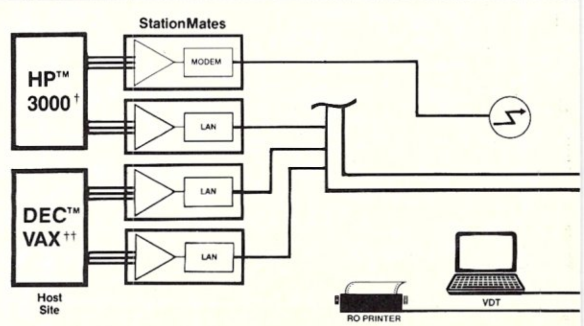
Besides local area networking, StationMate permits access to remote workstations via an internal dial-up modem. And it serves as a gateway for teleprocessing access to all LAN resources. You couldn't get all of this in one unit. Until now. And the suggested retail price is only $1450.00. StationMate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

Teleprocessing



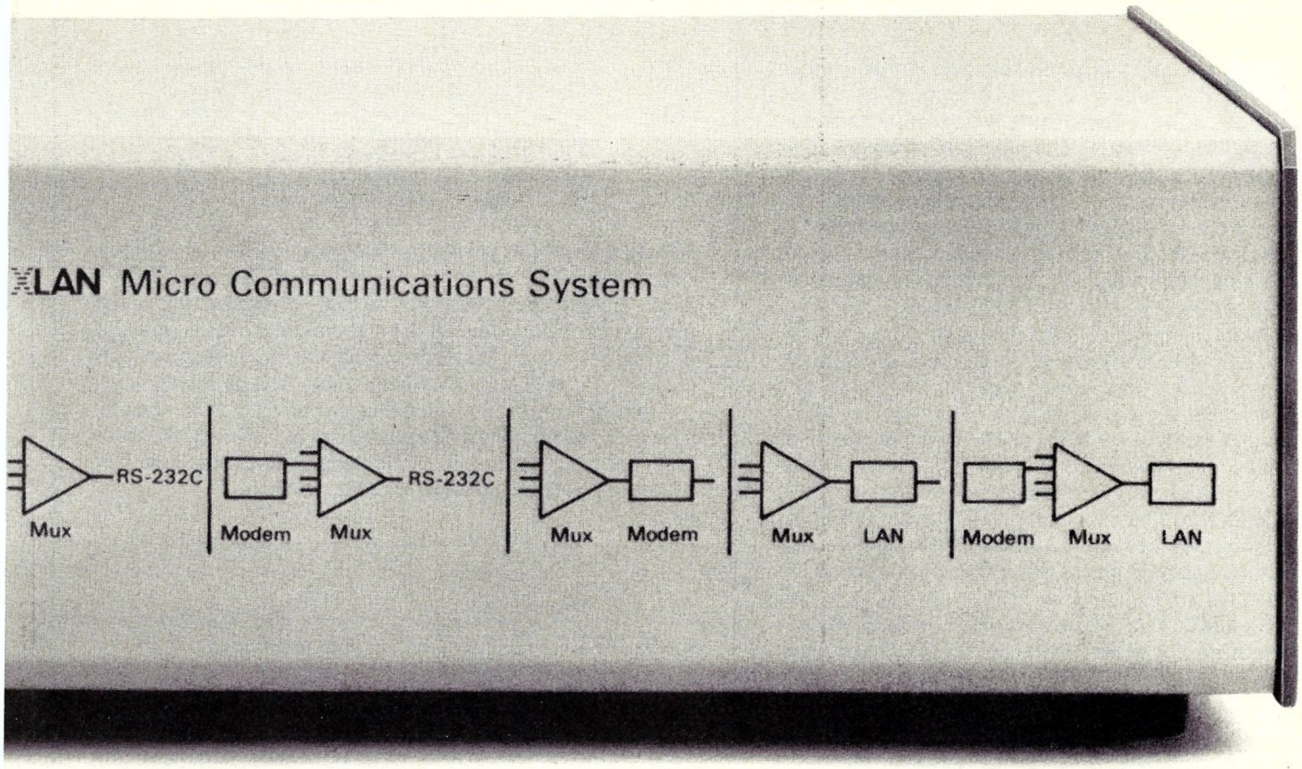


Local Networking

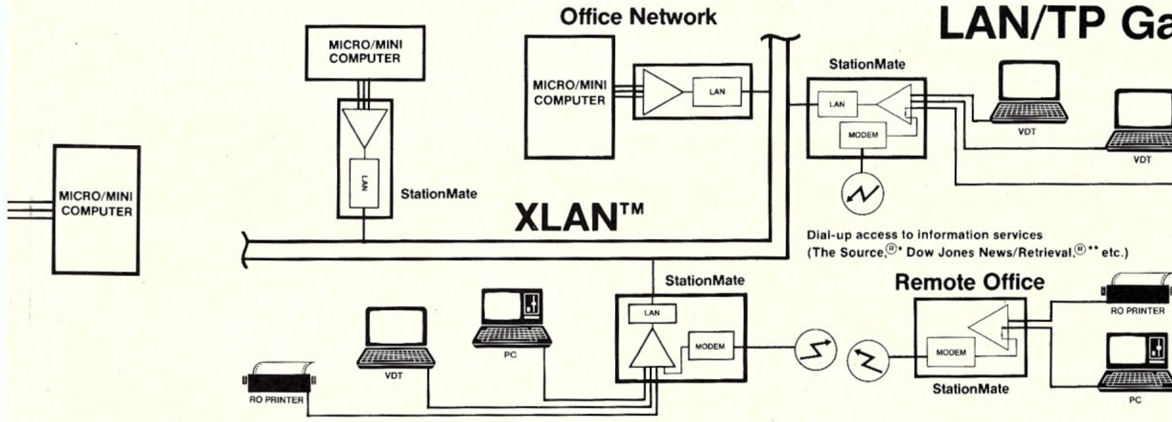
COMPLEXX™

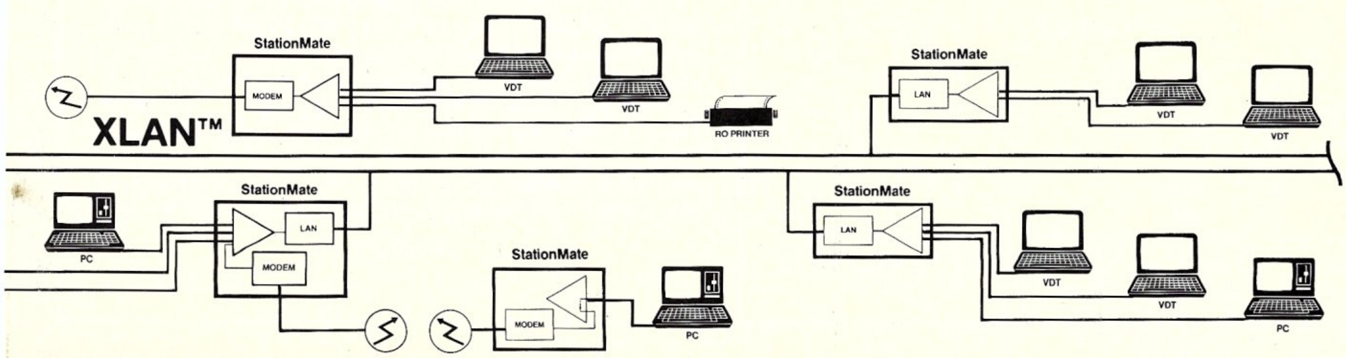
CIRCLE 51 ON READER SERVICE CARD

*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





Write a Menu-Driven Utility To Set Printer Options

by Robert L. LaFara

Most printers provide a number of options that can be selected by use of switches. Many of these options can also be selected via software. The purpose of this article is to provide you with an example program that you can customize to use with your own printer.

Why have such a program? It is usually inconvenient to use the switches on your printer. Some printers require that printer power be turned off and then back on again after changing the switch settings. This is because they read the switch settings only during the power-on phase. In other printers, the switches are inside the case and are difficult to reach (especially if the paper feeds directly over the switches.) Furthermore, it is difficult to remember which switch in which bank of switches controls which function. Some are not even labelled adequately enough to tell which way is on and which way is off. Because of these difficulties, how many times have you left your printer at 12 CPI when you would have rather had it set for 10 CPI?

The accompanying program was written to facilitate the selection of options for an IDS 560 printer; it will also work for the newer IDS Prism printer. You may wish to implement more (or fewer) options for use with your printer.

The program is written in 8080 assembly code and uses CP/M BDOS input/output calls. If you are unfamiliar with BDOS for input and output, this will provide examples of a few of these calls. The program has been divided into a series of listings to facilitate the discussion. The complete program is a simple concatenation of the separate listings.

In Listing 1, a number of equivalences are declared to provide for easier coding and to provide more understandable documentation. Only four different BDOS calls are required for this program. They are:

```
CONOUT-Output to the console
LSTOUT-Output to the LST: device
CONDIR-Direct input from the console, and
PRTSTR-Output a string of characters to the
        console.
```

In addition to the equivalences for the above BDOS calls, equivalences are given for BDOS itself (location 0005) and for various control codes that are to be sent to the printer.

Listing 1

```
CONOUT  EQU   2
BDOS    EQU   5
LSTOUT  EQU   5
CONDIR  EQU   6
PRTSTR  EQU   9
LF      EQU   0AH
CR      EQU   0DH
FSTR    EQU   0CH     ;FORM FEED CHARACTER
ESTR    EQU   81H     ;CHARACTER FOR ENHANCING
NSTR    EQU   82H     ;CHARACTER FOR NORMAL PRINT
ESC     EQU   1BH     ;ESCAPE CHARACTER
ASTR    EQU   1DH     ;CHARACTER FOR 10 CPI
BSTR    EQU   1EH     ;CHARACTER FOR 12 CPI
CSTR    EQU   1FH     ;CHARACTER FOR 16.8 CPI
```

Robert L. LaFara, Castle Oaks Computer Services, 10632 E. 79th St., Indianapolis, IN 46236

Listing 2 gives that part of the program concerned with initialization. First, the program must be originated at 100H. Next, a local stack is created for this program. Then, the first BDOS call is used to print the menu. This is done by putting 9 (function 9 means output a string to the console) in the C register and the address of the first character to be sent in the D and E registers. Then a call to BDOS is made. The string of characters, beginning at MENDAT and ending when a dollar sign is reached, is sent to the console. See Listing 6 for the menu used for this program.

Listing 2

```
        ORG    100H
        LXI    SP,STKTOP      ;SET UP LOCAL STACK
MENU:   MVI    C,PRTSTR       ;DISPLAY MENU
        LXI    D,MENDAT
        CALL   BDOS
```

Listing 3 gives the coding to input a character from the keyboard, store it for later use, and echo it back to the screen. First, BDOS function 6 is used for direct console input. (OFFH in register E specifies input.) The program keeps looping to CHARIN until the A register goes nonzero, which means a character has been received. The character is then saved temporarily, and echoed to the screen by means of BDOS function 2.

Listing 3

```
CHARIN: MVI   C,CONDIR    ;DIRECT CONSOLE INPUT
        MVI   E,0FFH
        CALL  BDOS
        CPI   0         ;TEST FOR CHARACTER RECEIVED
        JZ    CHARIN ;   IF A=0, NO CHARACTER YET
        STA   CHAR   ;SAVE CHARACTER TEMPORARILY
        MVI   C,CONOUT       ;ECHO CHARACTER
        MOV   E,A
        CALL  BDOS
```

Listing 4 shows the coding used to test for the various permissible options. The input character is tested against each possible option. If a match is found, a jump is made to the appropriate location to perform the requested function. If no match is found, BDOS function 9 is used to output an error message, the menu is re-displayed, and the program loops until a new option is entered. Before testing for any alphabetic characters, the lower case bit is discarded so that lower case input is converted to upper case for testing.

Listing 4

```
        LDA      CHAR      ;START OPTION TESTS
```

```
        CPI     '3'
        JZ      V3
        CPI     '4'
        JZ      V4
        CPI     '6'
        JZ      V6
        CPI     '8'
        JZ      V8
        ANI     5FH         ;DISCARD LOWER CASE BIT
        CPI     'F'
        JZ      FF
        CPI     'L'
        JZ      LINEF
        CPI     'E'
        JZ      ENHANC
        CPI     'N'
        JZ      NORMAL
        CPI     'A'
        JZ      P1
        CPI     'B'
        JZ      P2
        CPI     'C'
        JZ      P3
        CPI     'T'
        JZ      TEST
        CPI     'Q'
        JZ      QUIT
        MVI     C,PRTSTR    ;PRINT ERROR MESSAGE
        LXI     D,ERRMSG
        CALL    BDOS
        JMP     MENU
```

Listing 5 gives the coding to send the various option codes to the printer. Options 3, 4, 6, and 8 are used to control vertical spacing. These each require an escape sequence. A skeletal escape sequence, VSTR, appears in Listing 6. Each of the above options substitutes the appropriate values in the character string before it is sent to the printer using the module, TEST1. Since BDOS function 5 only sends a single character to the printer, the module, TEST1, keeps sending characters until a dollar sign is encountered in the string. Any other character could be used as a terminating character, but the dollar sign was chosen to be compatible with the BDOS function 9, which uses the dollar sign as a terminating character.

Each of the options F, L, E, N, A, B, and C require that only one control character be sent to the printer. In each case, a carriage return is also sent to cause the printer to dump its buffer.

The option, T, is provided as a test function. A string of test characters, TSTR, is shown in Listing 6. This string is sent to the printer via the routine, TEST 1.

Listing 5

```
V3:     MVI     A,'1'
        MVI     B,'6'
        JMP     VERT
V4:     MVI     A,'1'
        MVI     B,'2'
        JMP     VERT
V6:     MVI     A,'0'
        MVI     B,'8'
        JMP     VERT
V8:     MVI     A,'0'
        MVI     B,'6'
VERT:   STA     ARG1
        MOV     A,B
        STA     ARG2
        LXI     H,VSTR
        JMP     TEST1
FF:     MVI     E,FSTR
        JMP     OUTPUT
LINEF:  MVI     E,LF
        JMP     OUTPUT
ENHANC: MVI     E,ESTR
        JMP     OUTPUT
NORMAL: MVI     E,NSTR
        JMP     OUTPUT
P1:     MVI     E,ASTR
        JMP     OUTPUT
P2:     MVI     E,BSTR
        JMP     OUTPUT
P3:     MVI     E,CSTR
OUTPUT: MVI     C,LSTOUT
        CALL    BDOS
        MVI     E,CR
        MVI     C,LSTOUT
        CALL    BDOS
        JMP     MENU
TEST:   LXI     H,TSTR
TEST1:  SHLD    INDEX
        MOV     E,M     ;PUT CHARACTER TO PRINT IN E
        MOV     A,E     ;PUT IN A ALSO
        CPI     '$'     ;SEE IF IT IS A DOLLAR SIGN
        JZ      MENU    ;      IF SO, END TEST
        MVI     C,LSTOUT        ;PRINT CHARACTER
        CALL    BDOS
        LHLD    INDEX   ;PREPARE TO GET NEXT CHARACTER
        INX     H
        JMP     TEST1
QUIT:   MVI     C,0
        CALL    BDOS
```

To exit from the program, BDOS call function is used to re-boot the system. This program can be RUN from WordStar, and upon completion the return will be back to WordStar rather than to CP/M. Therefore, while wordprocessing, I can go to the NO-FILE menu and RUN my printer options program to: slew to top-of-page, change pitch, etc., all without shutting off the printer or even switching it off-line.

Listing 6

```
MENDAT: DB  CR,LF,'PRINTER OPTIONS MENU',CR,LF,LF
        DB  '3 - THREE LINES PER INCH',CR,LF
        DB  '4 - FOUR LINES PER INCH',CR,LF
        DB  '6 - SIX LINES PER INCH',CR,LF
        DB  '8 - EIGHT LINES PER INCH',CR,LF,LF
        DB  'F - FORM FEED',CR,LF
        DB  'L - LINE FEED',CR,LF,LF
        DB  'E - ENHANCED MODE',CR,LF
        DB  'N - NORMAL MODE',CR,LF,LF
        DB  'A - 10 CPI (5 CPI)',CR,LF
        DB  'B - 12 CPI (6 CPI)',CR,LF
        DB  'C - 16.8 CPI (8.4 CPI)',CR,LF,LF
        DB  'T - TEST',CR,LF,LF,
        DB  'Q - QUIT',CR,LF,LF
        DB  'ENTER OPTION  ','$'
ERRMSG: DB  CR,LF,'NO SUCH OPTION, TRY AGAIN!'
        DB  CR,LF,'$'
VSTR:   DB  ESC,'B,12,',CR,'$'
ARG1    EQU VSTR+3
ARG2    EQU VSTR+4
TSTR:   DB  CR,LF,'ABCDEFGHIJKLMNOPQRSTUVWXYZ',CR,LF
        DB  'abcdefghijklmnopqrstuvwxyz,CR,LF
        DB  '1234567890-='\;{,./',CR,LF
        DB  '!@#''%^&*()_+¯:}<>?',CR,LF,'$'
CHAR    DS  1
INDEX   DS  2
        DS  32          ;16 LEVEL LOCAL STACK
STKTOP:         :
        END
```

This program, written in 8080 assembly code, helps you select options for both the IDS 560 and the newer IDS Prism printer.

NorthStar Topics

A North Star Improvement

Making a 61K CP/M system out of a 56K North Star Horizon

by John H. Gillespie

The North Star Horizon was one of the first reliable S-100 systems to be produced. The fact that many of these systems are still in daily use is a tribute to their dependability. One of the few quirks in the design of the Horizon is that the disk controller is memory-mapped and occupies 1K of memory beginning at E800 hex in the standard configuration. Losing 1K of memory in such an awkward spot is bad enough, but the problem is compounded by the fact that the Horizon memory boards can be disabled only in 8K blocks, requiring that the top 8K of memory be disabled so that the memory boards and the disk controller do not occupy the same memory space. Thus, the Horizon is really a 56K machine even though supplied with 64K of memory. Today this restricted memory space seems like a silly design, but when the Horizon was first produced typical systems had as little as 16K of memory. A controller at E800 was viewed as safely out of the way of any foreseeable application. In this business, what was once unforeseeable very quickly becomes commonplace, and so it is that the 56K Horizon actually places some restrictions on the software that can be run on it.

This article will describe a method for increasing the horizon's total available RAM to 63K and for constructing a 61K CP/M system that uses the additional memory. The increase in memory space is accomplished by running an appropriate signal from the disk controller to the phantom line on the bus and by jumpering the memory boards to respond to the phantom signal. The 61K CP/M is so configured that the BIOS resides above the disk controller and the BDOS and CCP reside below it. The details of these modifications follow.

Hardware modifications

Step 1: Jumpering the disk controller. The disk controller must be modified so that it generates a PHANTOM* signal by pulling line 67 on the S-100 bus low each time the CPU does a memory fetch to the controller. An appropriate signal is provided by DI-GATE* that comes out of pin 6 of the NAND gate at locaton 7C on the board. DI-GATE* is used by the controller to enable the data-in buffers. This pin is easy to jumper because the trace from the pin is routed through the board just after leaving pin 6. Remove the disk controller from the computer and locate the IC at location 7C on the component side. Locate pin 6 and turn the board over; notice that the trace from pin 6 goes to the solder side of the board via a plated-through hole. Solder one end of a piece of thin wire in this hole and attach the other end of the wire to the top edge of the finger for line 67 on the edge connector. The fingers are counted beginning with number 51 on the right side of the board.

John H. Gillespie, 3335 Monterey Ave., Davis, CA 95616

That's all there is to it! Unless, of course, you are staring at a new disk controller and wondering how you are going to solder to the fiberglass. On these newer boards the plating for unused fingers was left off, presumably as an economy move. To solve the problem you will need to purchase some stick-on connector patterns such as those sold by Bishop Graphics, Inc., Westlake, CA. Attach a copper finger in position 67 and proceed with the soldering.

Step 2: Jumpering the memory board. Follow the instructions provided by North Star for adding a jumper so the that memory board that sits in the top 32K of memory will respond to the phantom signal. Then enable the top 8K of this board.

Step 3: Testing. Replace both boards in the computer and turn it on. Everything should behave in a completely normal fashion except that you now have 7K more memory than a few minutes ago! Run a memory test on locations E000-E7FF and F000-FFFF. These locations should not produce any error messages. The most visually exciting test is to use RAMTEST3 provided by North Star. If you require more gratification, use MOVCPM* to generate a new system. MOVCPM will inform you that the new system is a 58K system, a net gain of 2K. There is still an unused 5K of memory above the disk controller. The next section tells how to move the BIOS into that area.

Modifying CP/M

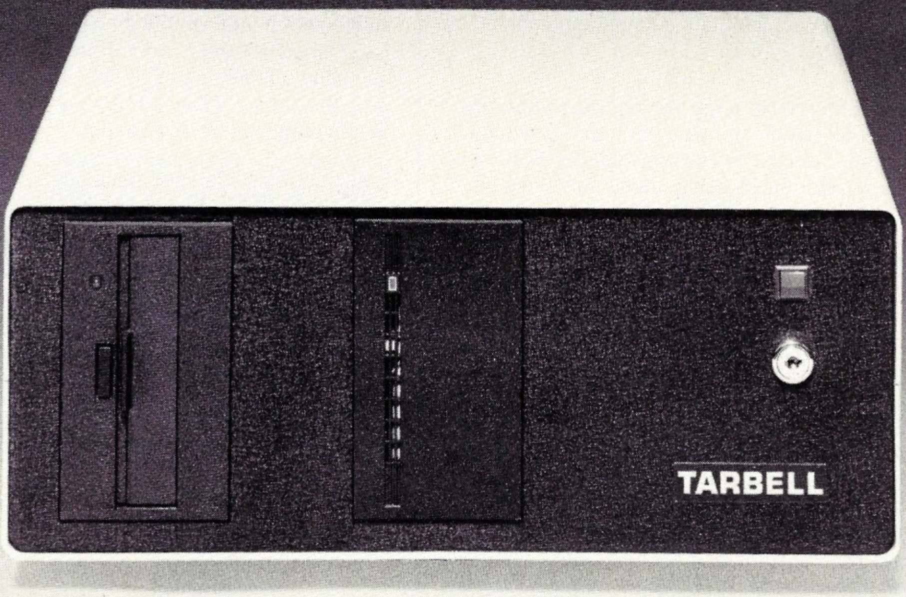
The modificatons to CP/M are straightforward, though somewhat tedious to perform. Basically, three different sized CP/M systems must be patched together to get the final product. The BIOS from a 64K system must be grafted to the BDOS and CCP of a 61K system, and both of these must be joined with a boot from a 24K system. The need for the first two components is obvious; the need for the 24K boot is less so. The CP/M boot loads the entire system into a contiguous piece of memory. If a 61K boot were used, the boot would merrily load the BIOS right on top of the memory space occupied by the disk controller! By using a boot from a 24K system, CP/M will be loaded at 3D00; then a special routine in the BIOS will move the various pieces to their final resting places. All of these modifications will be described in detail. The CP/M that is used is version 2.22 marketed by Lifeboat Associates. Other CP/Ms would be similarly modified.

Step 1: Modifying the user area. A short routine must be patched onto the end of the user area to move the CCP and the BDOS to D100 (the position for a 61K system) and the BIOS to F300 (the position for a 64K system). For a standard Horizon, add the code from Listing 1 to the end of the HORUSER.ASM file provided by Lifeboat. Otherwise add it to the end of the USER.ASM file configured for your system. Assemble the code using ASM and note the value of MOVSYS in the PRN file. This is the address to

JOIN THE REBELLION!

TURN TO TARBELL'S REBEL COMPUTER

19 MEGABYTES OF HARD DISK
6 MHZ SPEED
EXPANDABILITY
ALL FOR ONLY $4995



Do you need a computer that gives you more speed and expansion capability at an affordable price?

Then it's time to join the Rebellion . . . time to turn to the REBEL.

This new computer is the ultimate in design in its category in three important ways: more speed, more disk memory, and more capability for expansion of memory, disk, interfaces, and users.

REBEL hardware includes Z-80B CPU operating at 6 Mhz, 19 megabytes of hard disk, 372 kilobytes of floppy disk memory, 64 kilobytes of main (semiconductor) memory expandable to 1 megabyte, two RS-232 serial input/output ports for CRT and printer, plus cabinet which houses all of above and power supply.

Software includes MicroPro's WORDSTAR word-processing module, Tarbell Database System, Tarbell BASIC and Digital Research's CP/M® 2.2 disk operating system. The REBEL Series includes a range of computers from one 372K floppy disk to hard disks up to 200 megabytes.

REBEL is ideal for word processing, data bases, planning, accounting, inventory control, mailing lists, and other important business applications. And the REBEL gives you an unequalled advantage: Tarbell support — the after-sale backup unsurpassed by any other computer manufacturer.

So if you need a computer that can get the job done at an affordable price, join the Rebellion. Turn to the REBEL. See your nearby Tarbell dealer for a demo.

Tarbell Electronics
950 Dovlen Place, Suite B
Carson, California 90746
(213) 538-4251

CP/M is a registered trademark of DIGITAL RESEARCH INC.

CIRCLE 223 ON READER SERVICE CARD

which the boot will jump after loading the system into memory. From now on we will refer to the assembled file as USER.HEX.

Step 2: Creating the 64K BOS. Ordinarily, when MOVCPM is used to create a new system, the BIOS assumes that it is sitting directly on top of the BDOS and CCP. After a cold or warm boot the BIOS jumps to the CCP, which it expects to be in its usual place. In our case the CCP is much lower in memory, so a few bytes in the BIOS must be altered to reflect this fact. Use MOVCPM 64 to create a 64K system and execute a SAVE 40 NEWBIOS.COM. Type DDT NEWBOS.COM and use the SET command to change the following bytes

address	was	now	
204E	34	31	;in signon message
20B2	DC	DO	;a warm boot load address
20C3	E8	DC	;''
20F6	E5	D9	;base page of bdos
211C	DD	D1	;entry page into ccp
211F	DD	D1	;''

These bytes were located by tracing through the cold and warm start code of the BIOS. In other versions their locations will be different, but they are so conspicuous that one should have little trouble in locating them. Just look for jumps to locations below the BIOS. The first byte is to change the sign-on message from "64K" to "61K". After changing these bytes, read in the user file using the commands IUSER.HEX and R2D00. Finally move the BIOS to 100 with M2000,28FF,100 and exit DDT, saving the work with a SAVE 9 NEWBIOS.COM.

Step 3: Creating the boot. As mentioned above, the system can't be loaded in the 61K CP/M location because it will place the BIOS in the same location as the disk controller. One way around this would be to write a new boot that loads everything in the correct place. A much easier technique is to use the boot from a 24K system to load the entire code at 3C00 and then to use the routine in Listing 1 to relocate the pieces.

To accomplish this, use MOVCPM 24 and SAVE 40 NEWBOOT.COM to generate a 24K system called NEWBOOT.COM. Next, you need to enter DDT with DDT NEWBOOT.COM and locate the address where the boot jumps to the cold start location of the BIOS after it has finished loading the system. This byte is at address 964 in the Lifeboat version. At this point substitute for the jump a JMP MOVSYS using the 'A' command. The value for MOVSYS is obtained from USER.PRN. As before, move the boot to 100 using M900,9FF,100 and exit followed by a SAVE 1 NEWBOOT.COM.

Step 4: Creating the 61K BDOS. This is easy. Use MOVCPM 61 and SAVE 40 NEWBDOS.COM to bring it off.

Step 5: Putting it all together. It ought to be fairly obvious by now what to do. Here are the commands:

```
A>ddt newbdos. com
DDT VERS 2.2
-inewbios.com
-r
-m100,9ff,2000
-inewboot.com
-r
-m100,1ff,900
-g0
A>save 40 newcpm.com
```

Step 6: Putting it on the system tracks. Use SYSGEN NEWCPM.COM and give the appropriate drive.

A final note

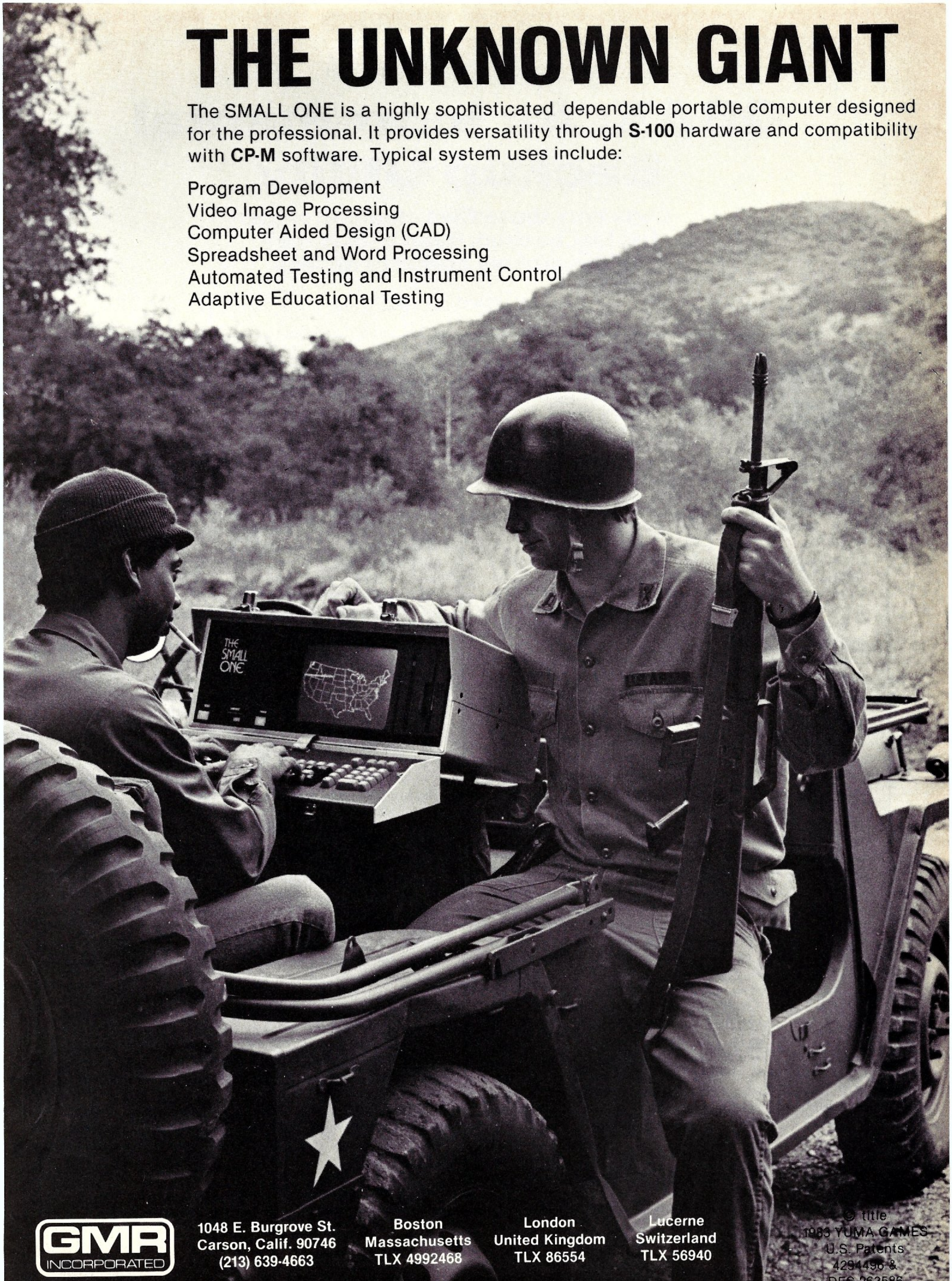
All of the above is written for version 2.22 from Lifeboat. You probably have some other version, so a little customization may be necessary. This is a fun project, and tracking down the appropriate bytes to alter should present very few difficulties.

If you want to have the slickest North Star on the block, get a copy of the replacement for Digital Research's CCP called ZCPR, available from the SIG/M library of public domain software. This user interface extends the set of built-in commands to include jumps and file loads to arbitrary addresses and a file search sequence that will automatically look for system files on the A: drive irrespective of the current default drive. To adapt this interface to the 61K system, equate CPRLOC to 0D100H and CPRR to 0A00H-CPRLOC, and follow the instructions for incorporating ZCPR into CP/M. The combination of ZCPR and the larger TPA will produce a quantum leap in the performance of your Horizon.

John H. Gillespie is a professor of zoology at the University of California at Davis, where he teaches evolution and genetics. He received his doctorate from the University of Texas at Austin. Dr. Gillespie recently developed a networking system specially designed to do Monte Carlo simulations, which he has used in simulations of population processes. His previous contribution to *Microsystems* was "A Hardware Random Byte Generator," July/August 1982.

Listing 1

```
;       routine to move CPM
;       add  to the end of USER.ASM OR HORUSER.ASM   just  before
;       the HORLENGTH equate

DIF     EQU     BIOS-5300H      ;address offset

MOVE:
        LXI     H,3D00H         ;start of BDOS for 24K cp/m
        LXI     D,0D100H        ;start of BDOS for 61K cp/m
        LXI     B,1636H         ;size of BDOS + (BIOS jump vector)
BDLP:   MOV     A,M             ;get byte of BDOS
        STAX    D               ;store in final resting place
        INX     H               ;bump pointers
        INX     D
        DCX     B               ;decrement counter
        MOV     A,B             ;check for zero count
        ORA     C
        JNZ     BDLP-DIF

        LXI     H,5300H         ;start of BIOS for 24K cp/m
        LXI     D,0F300H        ;start of BIOS for 64K cp/m
        LXI     B,900H          ;BIOS size
BILP:   MOV     A,M
        STAX    D
        INX     H
        INX     D
        DCX     B
        MOV     A,B
        ORA     C
        JNZ     BILP-DIF
        JMP     0F300H

MOVSYS  EQU     MOVE-DIF        ;entry from the cp/m boot
```


THE UNKNOWN GIANT
The SMALL ONE is a highly sophisticated dependable portable computer designed for the professional. It provides versatility through S-100 hardware and compatibility with CP-M software. Typical system uses include:
Program Development
Video Image Processing
Computer Aided Design (CAD)
Spreadsheet and Word Processing
Automated Testing and Instrument Control
Adaptive Educational Testing
THE SMALL ONE
GMR INCORPORATED
1048 E. Burgrove St. Carson, Calif. 90746 (213) 639-4663
Boston Massachusetts TLX 4992468
London United Kingdom TLX 86554
Lucerne Switzerland TLX 56940
© title 1983 YUMA GAMES U.S. Patents 4294496 & DES 263586

MICROSYSTEMS TESTS THE

QBAX: An Incremental Backup Utility

Copy only those files changed since the last backup; keep a sharp eye on the archive status of all files

by David Fieldler

QBAX will probably become one of those legendary programs that everyone eventually buys. It performs a function useful to anyone with a CP/M system, does it well and quickly, is understandable to the novice computer user, and is inexpensively priced at $30.

The function of QBAX is to copy files for backup. Big deal, you say, I have PIP, and a whizbang track-for-track copy program from the hardware manufacturer, and a UNIX-like utility called **cp** that accesses all my user areas. Who needs another copy program? (In case you just arrived from the planet Mongo, computer experts recommend keeping backup copies of all your files: data files, text files, .COM program files, and source code files. This is because experts *don't* keep copies, and realize their importance only after they erase all their files by mistake).

Everyone needs a copy program like QBAX. The usual method of backing up a disk relies on either brute force (copying an entire disk, even if some or most of the files on it are already backed up) or memory (using a copy program to back up all those files you know you changed). Unfortunately, human memory is more unreliable than a 16K dynamic RAM chip in a field of cosmic rays, and it is all too easy to "forget" a few files. How does QBAX help?

Every time you run QBAX, the program determines which of your disk files has been changed since the last time it was run. Then it copies these files, and *only* these files, to whatever disk you specify. This is called *incremental backup,* and is the backup method of choice on most large timesharing systems. It will work on any or all active user areas, and so is an absolute *must* for hard- or RAM-disk owners. Before I had QBAX, it would take me literally several hours to figure out which was the latest version of a backup file when I had to recover from a crash or accidental erasure. Now, I keep an entire series of backup disks, secure in the knowledge that even if my entire hard disk were to be wiped clean, I would just have to copy each of the backup disks back to the hard disk in turn, and it would be restored *exactly* as it was.

QBAX options allow you to keep multiple backup copies, report which files need backup, change the "backup status" of files (so files can be "marked" as ready for backup or not needing backup if you desire); and all this can be redirected to a disk file or the printer for later perusal. In fact, a data file containing names of other files can be used as input to QBAX, much like programs on UNIX can have their I/O redirected. Another nice extra feature. But for the novice, QBAX can be used simply by typing **qbax a b,** which will back up all newly changed files from disk **a** to disk **b.** And that's a lot easier to remember than the syntax for PIP! The manual is well-printed, accurate, complete, and gives plenty of examples.

For the more technically minded: QBAX works by fooling around with unused bits in the FCB area of the directory, so it doesn't take a long time to determine whether files have been changed—less than a minute on a hard disk full of hundreds of files. Its only true limitation is that if you have a program that updates a file "in place" (i.e. through random access) without closing the file, QBAX may not be able to detect that the file was changed. Regular "in place" random access programs will also fool QBAX with most copies of CP/M, so a program is provided with QBAX that automatically patches CP/M to permit files modified by such programs to be detected. In several months of using QBAX with many different programs from a variety of software houses, only Perfect Filer and RBBS31 (a public-domain "Bulletin Board" program) have required this patch. Due to my nonstandard CP/M system that uses several different density disks at once, I have encountered problems recovering backups of data files created by these programs. Other than this, QBAX has not glitched, bugged, or hiccupped in all that time.

The only complaint I have about QBAX is possibly in the user interface department. If the disk you're copying your files to fills up, you get an error message, and you then have to use a different disk. QBAX does keep track of itself, and knows enough to copy that last file (whose copy failed) to the new disk. But suppose you were copying a 600 KB file from a hard disk to a standard floppy? QBAX would keep failing as the floppy filled up! A better way would be for QBAX to mark *each disk extent* as it is copied, so that such a large file would be automatically split up, possibly across several floppy disks, and could then be recovered as easily. The next version of QBAX will solve this problem according to the vendor.

Personally, I am very happy with QBAX, as it has saved me numerous hours that could have been spent retyping programs or text. I frequently work until 4 A.M. and find that I just barely have the mental powers left to type the QBAX command line, much less remember the names of all the files I edited! It's this simplicity that is the best feature of QBAX.

QBAX runs on all CP/M 2.2 systems with a minimum of 32 KB of memory and costs $30. It is sold by **Amanuensis, Inc.,** *RD#1 Box 236, Grindstone, PA 15442; (412) 785-2806.*

David Fiedler, InfoPro Systems, P.O. Box 33, East Hanover, NJ 07939

S-Basic: A Structured Basic Compiler

by Timothy J. Parker

Basic as a language is in an unusual position. Most advanced programmers consider it limited. It tends to be slow and lacks the programming power of languages such as Lisp or APL. Yet Basic is the world's most popular small-computer language because of several factors. The majority of home computers have an interpretive Basic included either in ROM or as a loadable program. Basic is easy to learn: the syntax makes common sense for starting programmers, and it is relatively easy to follow a written program's purpose. (The wonderfully complex lines that can be constructed in APL, for example, can be almost impossible to decipher, while the function of a Basic line is usually apparent from its context.)

Basic can be mastered by a starting programmer in a fraction of the time required for other languages that depend on an abbreviated or simplified coding or command structure to increase speed and versatility. When used in program development, Basic allows a multitude of GOSUBs and GOTOs that would crash a highly structured language.

Because of its popularity, a great deal of effort has been applied to the problem of increasing Basic's attraction to seasoned computer users. The advances have occured in two areas: speed and programming power.

Speed has been increased dramatically by the use of compilers. Some Basic versions allow the programs to be written interpretively, then compiled for speed of execution. Others require compilation to be used before the program can be run.

Programming power has been increased by adapting many of the more attractive features of higher level languages into the Basic structure. Fortran (which has a steadily decreasing number of advocates) has given Basic the DO loops, WHILE/WEND conditionals, and others. Structuring similar to Pascal's has been appearing in the more advanced Basics.

Microsoft's Basic-80 seemed for a while to be becoming a de facto standard. It offered many attractive features that increased versatility and, when coupled with a compiler, offered better speed of execution. Other companies continued to offer enhanced Basic packages, and, while Basic-80 may be the most popular CP/M Basic currently available, it does have drawbacks. Digital Research's CB/80 (sometimes erroneously referred to as CBasic, which itself is pseudo-compiled) was introduced as a compiled Basic, and offered improvements on the "standard" versions of the language that were attractive to programmers.

Topaz Programming's introduction of S-Basic (structured Basic) gives the CP/M user a choice. It has options that for some applications are outstanding. It is also fast.

Timothy J. Parker, Univ. of Ottawa, Chemistry Dept., 365 Nicholas St., Ottawa K1N 9B4 Canada

For the precision involved, it is quite possibly one of the fastest Basics available.

S-Basic is a compiled Basic that generates binary programs which can be run directly by the computer. It is tailored specifically for the CP/M operating system.

S-Basic variables

Six variable types are supported. These are real, double precision real (called real double), fixed, integer, string, and character. Real and real double are floating-point numbers with approximately six and 14 digits of accuracy respectively. Fixed offers 11-digit precision and is ideal for business applications (as the decimal points can be formatted so they appear beneath each other). There are three digits to the right of the decimal place, although only two are printed. Rounding is done automatically by adding 0.005 to each number. This all yields a dollars-and-cents configuration. Exponential forms are not used in this variable type. Integer has a range of -32767 to +32767, and is stored in memory as a two-byte word.

The string variable can contain any ASCII character except null, as this is used to mark the end of the string. (Nulls can be used with the character variable.) The length of the string can be defined in a program, up to 255 characters. A default value of 80 is used if unspecified. Char (character) is a single ASCII character.

All variable must be defined in an S-Basic program. However, considerable flexibility is available. Variables can be located in three different locations in the compiled code (referred to as the "runtime package" by the manual). These locations are: the data storage area created by the compiler, the common storage area (where they will be untouched when chained to another program), or an area that is not assigned until the program is run.

Variables can be defined at the outset of a program—which effectively defines them as global variables—or they can be defined in each small program block. They can then be redefined in a later section. This allows local, global, and a compromise variable that is not global for the entire program, but only for a subsection.

S-Basic statements

Arrays are defined, as in most Basic versions, using a DIM statement. An argument can be added if the array is not to reside in the data storage area. If the array has been assigned no specific location in the program, it can be LOCATEd in a memory location with a statement.

Line numbers are not required in S-Basic programs, except for a specific branching instruction (i.e., GOSUB, GOTO, etc.). A line number is defined using a digit (0-9) followed by ASCII characters. Thus a program can be defined by the line number '1test', or '1format'. Numbers do not have to be in numerical order.

Any valid statement can be replaced by a BEGIN . . . END framework. For example, if an IF/THEN/ELSE were to be used, it could be replaced with the structure:

```
IF A=B THEN
    BEGIN
        [commands]
    END
ELSE
    BEGIN
        [commands]
```

Table 1 gives a list of the statements supported by S-Basic. Some of these are explained in more detail below, to highlight the more unusual additions.

Remarks can be added in two ways. The usual REM or REMARK statement is valid, as is COMMENT/END. This is used to identify a large block of comment lines, beginning with the statement COMMENT, and terminating in END. Both REM (and REMARK) and COMMENT/END are ignored by the compiler. This allows a programmer to be generous in his documentation.

The logical functions (Boolean operators) provided are NOT, AND, OR, XOR (logical exclusive OR), IMP (logical implication), and EQV (logical equivalence). A truth table is also provided in the manual.

Relational symbols are straightforward: =, <, >, < >, > = [or = >], and < = [or = <] are supported. A frequently forgotten feature, the use of # to symbolize "not equal to," is used in S-Basic.

An error-trapping routine can be added to programs using the ON ERROR GOTO statement for nonfatal errors. Naturally, a fatal error will revert control to CP/M. ON ERROR causes the runtime stack pointer to be cleared, with subsequent loss of any data in the stack. Errors cause an error message to be printed, and place an error code in a memory location (103H) for reference.

The REPEAT/UNTIL is always executed once, regardless of the UNTIL conditional. WHILE/DO is not executed if the WHILE conditional is false. These two are essentially the same except for the option of running the routine through at least once. As mentioned above, they can be decomposed into lengthy subroutines using BEGIN/END.

An option that allows the ability to select one statement or group of statements from a larger number is accessed by the CASE statement. The syntax used is:

```
CASE [expression] OF
  [expression 1]: statements
  [expression 2]: statements
        .
        .
        .
  [expression x]: statements
END
```

Here, the [expression] is evaluated and compared with each of the [expression x]'s. If they are equal, the relevant statements are executed, and the rest is ignored. This is a very convenient method of evaluating an input. RETURNs and GOTOs are not supported from outside a CASE statement.

Input/output is handled well in S-Basic. The IN/PUT statement can have arguments that specify the physical devices used. The question mark that usually signifies a prompt on an INPUT statement can be overridden, as can line feeds. An ECHO statement that "echoes" any input can be switched on or off.

Table 1. Statements supported by S-Basic

REPEAT/UNTIL	WHILE/DO
IF/THEN/ELSE	VAR/COM/BASE
DIM	BASE/AT
LOCATE/LOCATION	END
PRINT USING	ECHO ON/OFF
INPUT	TEXT
LPRINT	CONSOLE
COMMENT/END	REM/REMARK
ON ERROR GOTO	GOTO/GOSUB
RETURN	ON X GOTO/GOSUB
FUNCTION/END	PROCEDURE/END
FILES	OPEN/CLOSE/READ/WRITE/INITIALIZE
DELETE/RENAME/CREATE	CASE X OF
FOR X TO (STEP)/NEXT	TRACE
LINES	PAGE

Forced output to a printer is accomplished with LPRINT. (This does not imply that it is required for printer output. This can be specified by the device number.) CONSOLE can be used to give a device default.

Large volumes of text, such as instructions, can be manipulated with ease using the TEXT statement. In operation, TEXT is followed by a device number (e.g., 0 for console), a comma, and a delimiter such as %, $, or #. Any material between the delimiters is sent to the device. The statement:

```
TEXT 0,%
   instructions, diagrams, etc
%
```

will send everything between the two %s to the CRT. This is one command that should be standard in all Basics—no more PRINT statements or formats to worry about. (Naturally, PRINT is available for short texts.) PRINT USING is available for formatting of output.

FUNCTION and PROCEDURE are two carryovers from higher-level languages. FUNCTION is similar to a powerful DEF FNX() in most Basics. In use, a function is defined as one or more statements to be executed. Arguments can be defined for use in the FUNCTION. A function is called by using its name (and any arguments required). A function has a "type" defined, such as real, integer, etc. PROCEDURE differs from FUNCTION in that it has no "type" specified.

Chaining is supported by invoking the filename of the required files. The specified file is loaded and executed. Variables can be passed, depending on the way they were defined. Assembly language routines can be CALLed.

Both random and serial files can be used for storage or retrieval. Files can be CREATEd, DELETEd, RENAMEd, OPENd, WRITEn to, and CLOSEd. INITIALIZE will even initialize a new disk.

Compiling & debugging programs

The compiler is invoked by specifying the file to be compiled. Three other files are called for use by the compiler program on the language disk. Up to two resultant files can be obtained. One is a PRN file that contains the source.

S-Basic is fast. For the precision involved, it is quite possibly one of the fastest Basics available.

The location of the files is specified when S-Basic is called.

A TRACE option follows the execution, as does a LINE command. The difference between the two is that TRACE follows the actual execution procedure, allowing evaluation of IF statements. A PAGE command sends an ASCII form feed to the listing device. Traces can be executed at runtime for debugging purposes by typing control-t (toggle trace on/off).

The S-Basic compiler is fast by compiler standards. Unfortunately, that still translates to minutes of actual time. A typical 100-line program is compiled in approximately one minute, while an 1800-line program requires approximately 20 minutes. Naturally, the exact time varies according to the number of branches, subroutines, etc.

Documentation

The manual (over 100 pages, plus appendices) is the weakest link in the package. Although it is ideal as a reference work, it takes several readings to become acquainted with the features S-Basic has to offer. It is certainly comprehensive: All commands are shown with any possible arguments or limiters, and examples are given of most commands, especially those not encountered in standard Basics.

Appendices are added to summarize most instructions, list error codes, and add technical notes. Application notes are also included. A welcome addition is a section on idiosyncracies of the language (nothing that severely limits its use). An index is not included, although each appendix contains references to relevant sections of each command. Using the manual to look up syntax or uses of a statement tends to be slow. Acquaintance with standard Basics is assumed. A neophyte to the language would probably be lost with the documentation provided. Addition of a full index and a more "friendly" style of writing would be welcome.

Overall evaluation

On the whole, S-Basic certainly has lot to recommend it. While program development using a compiled language is notoriously slow (in case of an error, the source code has to be reloaded, changed, then recompiled), it has the primary advantage of execution speed. Although game design is not something to be attempted lightly in compiled Basic, a library of standard subroutines and functions in compiled Basic saves time and effort. S-Basic source code can be written with word processors or CP/M's ED (the former is preferable). The compilation is a two-step process: in the first step, the source code lines are numbered and checked for errors. Only if there are no errors will the second step (code generation) begin. This saves a geat deal of time that is wasted with some languages which compile whether an error is present or not.

S-Basic was easy to use once the manual was deciphered. The power of the language is undeniable. No problem was encountered despite several attempts to sabotage the programs. S-Basic combines the simplicity of Basic, the routine-handling of Fortran (and others), and the structure of Pascal into a pleasing package. Although compiled Basics are not for all conditions, when one is required, S-Basic is an excellent alternative to Microsoft's Basic-80, Digital Research's CB/80 or CBasic. When proficiency is attained, S-Basic can be the choice of the bunch.

Topaz Programming is now owned by Kaypro Corp.; S-Basic is available separately for $75, or bundled with the Kaypro 4 and 10 computers. For information, contact:

Kaypro Corp.
P.O. Box N
Del Mar, CA 92014
(619) 481-3920

CIRCLE 319 ON READER SERVICE CARD

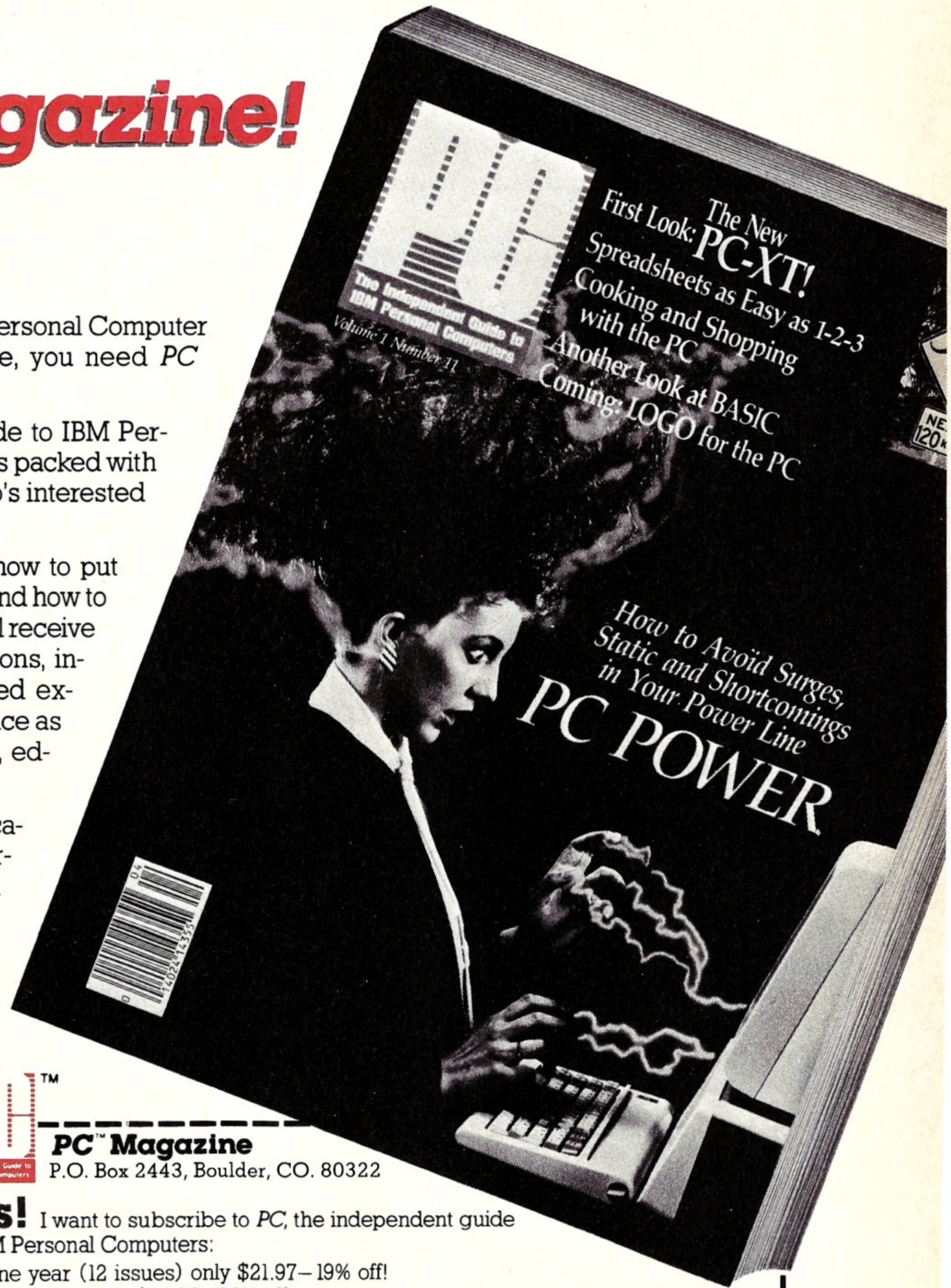
Table 2. S-Basic intrinsic functions and uses

ABS(X)	Returns absolute value of X
ASCII(X)	Return integer equal to first character of string
CHR(X)/ CHR$(X)	Returns 1 character string with ASCII value of X
COS(X)	Returns cosine in radians
EXP(X)	Returns e to power of X
FCB(X)/ FCB$(X)	Returns string equal to valid format of a FCB
FFIN(X)	Returns integer part of fixed type expression.
FINT(X)	Returns next lowest fixed integer.
FIX(X)	Returns integer of real X
FRE(X)	If X is false returns free memory If X is true returns number of used drive blocks
HEX$(X)	Returns a string of 4 characters
INP(X)	Performs input instruction from port X
INSTR (X,A,B)	Searches for B in A starting at Xth character
INT(X)	Returns next lowest real integer
LEFT$(X,I)	Returns leftmost I characters in X
LEN(X)	Returns integer equal to length of X
LOG(X)	Returns natural log of X
MID$(X,A,B)	Returns string of B characters starting at Ath
NUM$(X)	Returns string of characters representing X
STR$(X)	Similar to above
PEEK(X)	Returns memory location value
POS(I)	If I positive, returns print position If I negative, returns line count
RIGHT(X,A)	Returns character of X in Ath position
RIGHT$(X,A)	Returns rightmost A characters of X
RND(X)	Random number generator
SGN(X)	Returns sign
SIN(X)	Returns sine in radians
SIZE(X)	Returns size of disk file in blocks
SPACE$(X)	Returns string composed of X spaces
SQR(X)	Returns square root
STRING(X,A)	Returns string of characters of ASCII A, length x
STRING$(X,A)	Similar to above
TAB(X)	Returns string of spaces to move print head
TAN(X)	Returns tangent in radians
VAL(X)	Returns real number equal to numeric X
XLATE(X,A)	Returns translated string

S-Basic combines the simplicity of Basic, the routine-handling of Fortran (and other languages), and the structure of Pascal into one pleasing package.

Get all the facts about the IBM Personal Computer.

Get PC™ Magazine!



Whether you already own an IBM Personal Computer or you're thinking about buying one, you need *PC* Magazine.

PC Magazine is the independent guide to IBM Personal Computers. Each monthly issue is packed with the latest information for everyone who's interested in IBM Personal Computers.

This is *the* magazine for finding out how to put together the best IBM "PC" system... and how to get the most out of it. Every month you'll receive hundreds of colorful pages of evaluations, insights, and straight talk from respected experts—professionals in computer science as well as lawyers, businessmen, writers, educators and many others.

PC covers software, hardware, applications, and every other topic of importance to the thousands of IBM Personal Computer users who read it. To make sure that we give you the facts you need, we include a special "User-to-User" section, a "PC Wish List," and news about IBM Personal Computer clubs, events and publications.

Right now you can save up to 31% on an introductory subscription. And if you're ever dissatisfied with *PC*, just write and tell us—you'll receive a prompt refund for all the unmailed issues remaining in your subscription.

PC™ Magazine
P.O. Box 2443, Boulder, CO. 80322

Yes! I want to subscribe to *PC*, the independent guide to IBM Personal Computers:

- ☐ One year (12 issues) only $21.97—19% off!
- ☐ Two years only $40.97—24% off!
- ☐ Three years only $55.97—31% off!

Savings based on full one-year subscription rate of $26.97.

CHECK ONE: ☐ Payment enclosed. ☐ Bill me later. 8H336

Mr./Mrs./Ms.____________________
(please print full name)

Company____________________

Address____________________

City__________ State________ Zip________

Offer valid only in the U.S., its territories and possessions. Please allow 30 to 60 days for delivery of first issue.

A Z80 Random Number Generator

Get 16-bit random numbers from the Z80 refresh counter

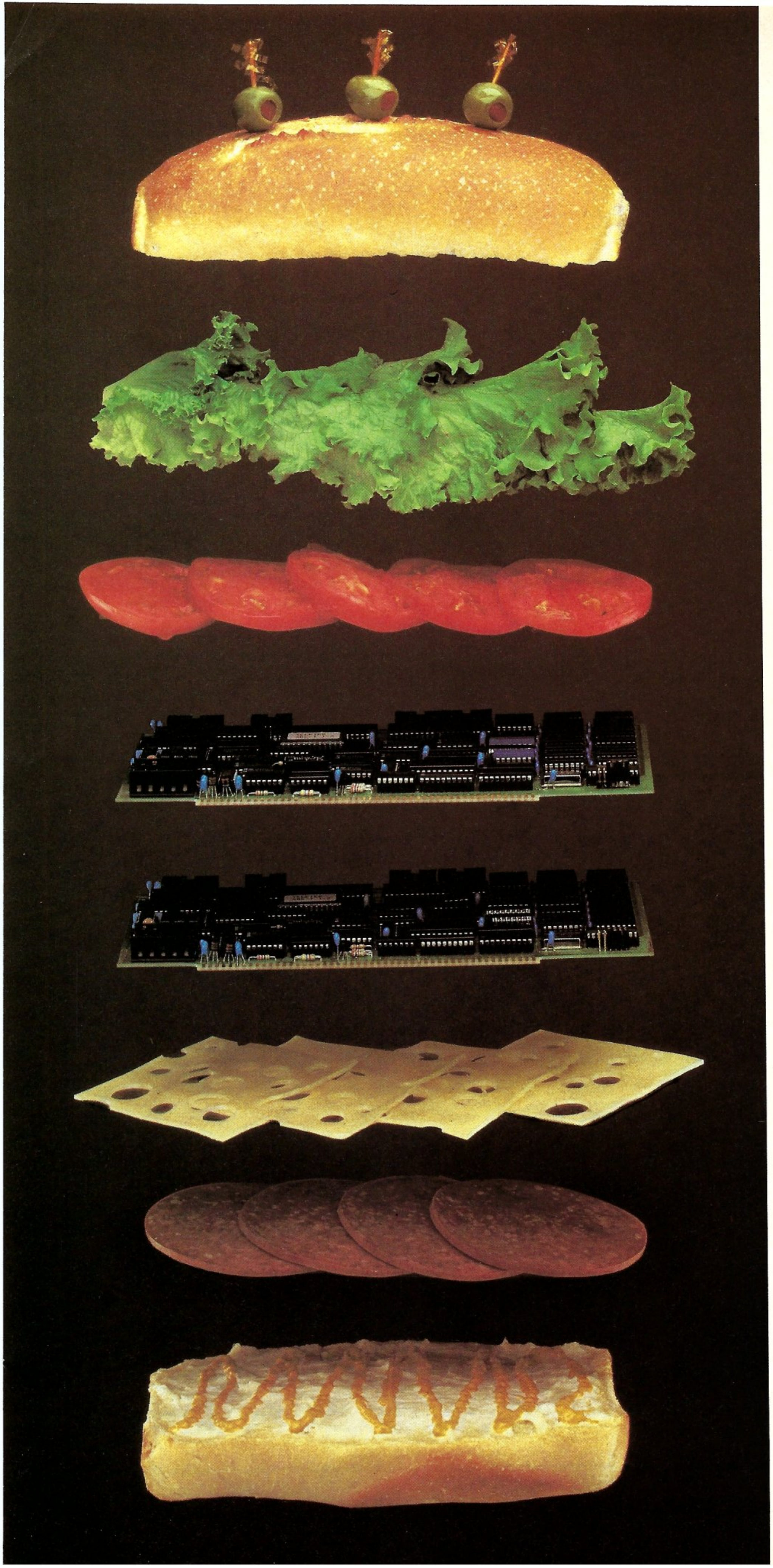
by Robert W. Zimmerer

The RAM refresh counter in the Z80 offers the basis for a very satisfactory random number generator. This 7-bit counter "clicks" away to provide the bonus RAM refresh feature of the Z80. An occasional reading of this counter will give an approximation of a 7-bit random number. If it is frequently read as part of a repetitive program, successive values may have too great a correlation to be useful as random numbers. In the course of playing with a random walk simulator, I worked out a simple routine to provide a good approximation to a 16-bit number. To test the randomness of my routine, I plotted points on a 75 X 160 point matrix—my TV display—and used my eye to look for patterns in dot placement. The *x* and *y* position of each dot was computed from the random 16-bit number generator by the usual method of dividing the 16-bit random number by either 75 or 160 and using the remainder as the *x, y,* coordinate.

Robert W. Zimmerer, 131 Second Ave., Port Angeles, WA 98362

My test procedure was certainly a highly repetitive routine. Looping through the program at high speed, the refresh counter was being read at nearly regular intervals. Using it as a random number produced a very obvious pattern of dots. I found that by using two 16-bit numbers and alternating my use of them "randomly," I succeeded in producing a dot distribution with no apparent pattern. The 75 X 160 matrix filled up slowly with dots appearing randomly everywhere until their density became so great that the appearance was more of randomly scattered non-dots slowly disappearing.

My routine reads the refresh counter when a random number is requested and constructs a 16-bit number in register DE, using 0 for the high byte D and the refresh counter value for the low byte E. The refresh counter byte is then shifted right one bit circularly with bit 0 going into



Teletek's New Combo Could Make You A Hero!

The SBC-II could be just the right ingredient for your latest concoction. The newest member of Teletek's family of multi-user, multi-processing S-100 products, the SBC-II essentially combines, or "sandwiches" two Teletek SBC-I's into one board. The SBC-II provides the capability to support two users from one standard size IEEE-696/S-100 slave board.

The SBC-II maintains full performance for each user with an independent CPU (Z80A or Z80B), 64K RAM, Serial I/O, and FIFO communications port to the system master. The system integrator benefits by getting complete support for two users for the price of one board.

TurboDOS and MDZ operating systems will support combinations of SBC-I's and SBC-II's offering system design efficiency and flexibility never before possible.

If you're hungry for value and efficiency, order an SBC-II from Teletek. You'll love every byte.

TELETEK

4600 Pell Drive
Sacramento, CA 95838
(916) 920-4600
Telex 4991834 TELETEK
Dealer inquiries invited.

CIRCLE 174 ON READER SERVICE CARD

carry, CY. If the carry is now 1, the first of the two 16-bit random numbers being maintained is selected for the computation. If carry is 0, the second 16-bit random number is selected and E—the low byte of DE—is replaced with the rotated value just produced.

The 16-bit random number thus selected is put into register HL and DE is added to it. The result, HL + DE, is then multiplied by DE to stir up the bits. The product is register HL now replaces the 16-bit random number used and is also returned as the new 16-bit random number requested.

Putting it into assembly language, we have the routine shown in Listing 1.

Listing 1

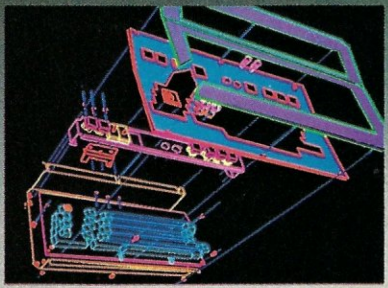
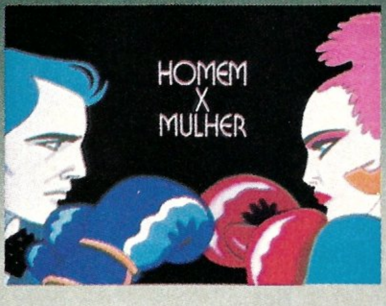
```
;       #1     Random number function
;       Return a random 16 bit number on TOS
;       using the Z80 refresh register R for a
;       random number seed.

RANFUN LD      A,R          ; get random byte
       LD      E,A
       LD      D,0          ; make a 16 bit number
       RRCA                 ; scramble and set carry
                            ; randomly
       JR      C,RAN1       ; use random number A if
                            ; carry = 1
       LD      E,A          ; else use scrambled value
       LD      BC,RANB      ; point at random number B
       JR      RAN2
RAN1   LD      BC,RANA      ; point at random number A
RAN2   LD      A,(BC)       ; move previous random
                            ; number into HL
       LD      L,A
       INC     BC
       LD      A,(BC)
       LD      H,A
       ADD     HL,DE        ; mix it up
       CALL    MULTIPLY     ; scramble bits again
                              ; HL = (HL+DE)*DE, BC = BC

       LD      A,H          ; and replace the previous
                            ; random number with this new one
       LD      (BC),A
       DEC     BC
       LD      A,L
       LD      (BC),A

       EX      (SP), HL     ; swap random number with RET
       PUSH    HL           ; push back RET address
       RET                  ; return to calling routine
                            ; with random number on stack

RANB   DEFW    1936H        ; initial random numbers
RANA   DEFW    1936H        ; just to get started
```

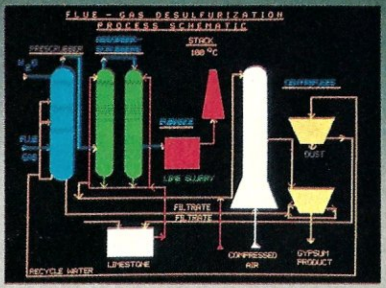
This Z80 routine provides a good approximation to a 16-bit random number. It could be extended to larger numbers if so desired. It serves me well, generating random integers between 0 and 160 for simple simulations. Perhaps others would find it as useful as I have.

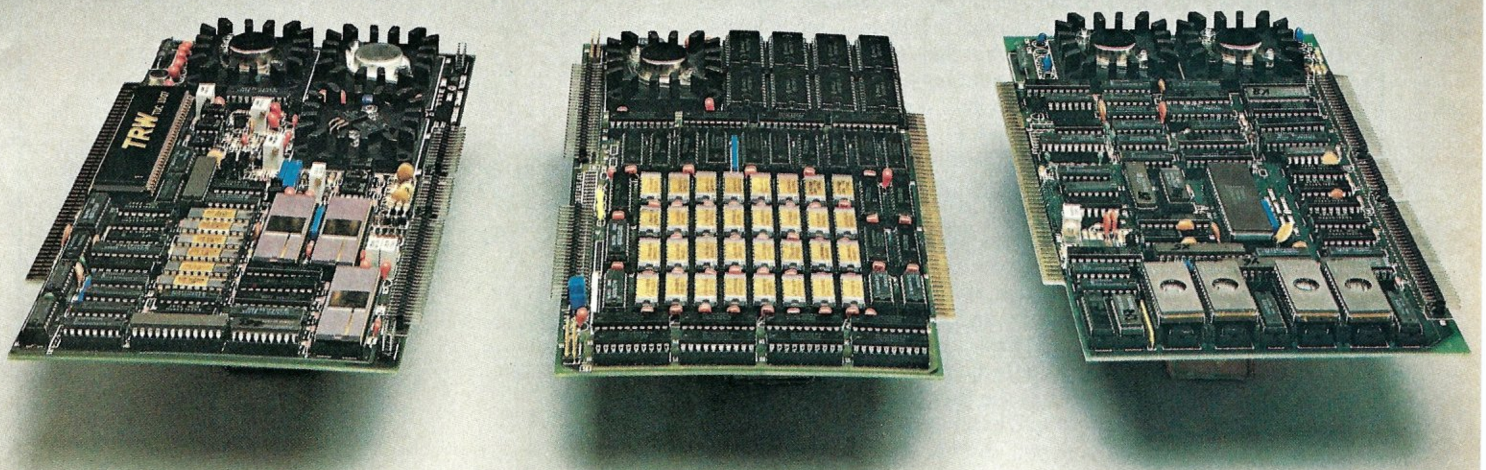
The Image Solution

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







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

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

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

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

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

DIGITAL GRAPHIC SYSTEMS, INC.

What You See Is What You Get.

CIRCLE 249 ON READER SERVICE CARD

Hiding Machine Code in REMs

by Dennis Brewer

Some Basic programs use machine language routines to do tasks that would otherwise be too difficult or too slow for Basic. These machine language routines are usually invoked by a "USR" or "CALL" command. One minor but annoying problem with these routines is that they are not SAVEd on disk (or tape) when the Basic program is SAVEd. It is necessary, usually, to SAVE the Basic program and the machine language routine in separate operations. An example of this is shown in Listings 1 and 2. Listing 1 is a Basic program that draws sine waves on my monitor—with the assistance of the machine language routine in Listing 2.

There are a couple of ways to get around this "double SAVE." The machine language routine can be imbedded in the Basic program in the form of DATA statements with the routine put into memory by POKEs or FILLs. The sine wave program of Listings 1 & 2 is shown in this form in Listing 3. It's somewhat cumbersome, if your Basic accepts only decimal POKEs and FILLs (as does North Star) and you're used to programming in Hex.

Users of the Sinclair ZX-81 computers, I understand, routinely squirrel their machine code away in REMs and strings. It took me a while to understand what they were doing, but I now see that the technique can be adapted to my North Star MDS (which runs on an IMSAI 8080). An example of the sine wave program translated to this method is shown in Listing 4. Yes, I know that it's a bit garbled-looking in line 1!

The basic premise behind this technique is that the computer *doesn't care what you put in a REM.* To put machine language in a REM, you must know where, in memory, the first line of Basic begins. And you must make the REM containing the machine language as the very first line of the program. Make it line 1 for consistency and safety.

My system runs North Star Basic version 5.2, set up to run at 0000 Hex. I find that my Basic program area begins at 341E HEX. North Star uses the first byte of a line as a "character count". The next two bytes are the line number. Next comes whatever you put on that line. North Star commands are stored as one-byte words to conserve memory (see reference). The end of a line is denoted by OD Hex and the end of a program is marked by 01 Hex.

Looking at memory, I find that if program area starts at 341E, I can start my machine code at 3424 Hex. Remember, this is with my Basic loaded at 0000 Hex.

Once you know just where your first-line REM will be, the method is very simple. You start out on line 1 of your Basic program and enter a REM, followed by sufficient spaces to accommodate your machine language routine. Next, enter your system's Monitor. My monitor is located at 9000 Hex, far enough above Basic to keep out of its way. Simply put your machine language routine into the line 1 REM area; beginning at 3424 Hex in our example.

Remember—the line containing the machine code MUST be the first line of the program. Otherwise, it will move around whenever you make any changes in the program! Listing 5 contains a program to guide you through the process of entering code in the REM.

This may not be a major breakthrough in computer science, but it helps to chip away at one of the little problems!

Dennis G. Brewer, 204 Dellwood Dr., Greenville, NC 27834

```
2 REM\ DRAWS A SINE WAVE
90 PRINT CHR$(11)
100 FOR C=1 TO24 STEP 0.25
120 A=C*5
130 B=SIN(C)*10
140 B=24-B
150 GOSUB 50000
200 NEXT C
999 END
50000 REM\ MACHINE-LANGUAGE INTERFACE SUBROUTINE
50001 REM\ ENTER WITH A (X-AXIS), B (Y-AXIS)
50020 A=INT(A)\B=INT(B)
50040 N=(256*A)+B
50060 Q9=CALL(61440,N)\ REM\ F000 (HEX)
50080 RETURN
```

-LISTING 1-

```
F000   AF 7A 1F 6F   9F 57 26 CC   01 40 00 7B   D6 03 FA 15
F010   F0 09 C3 0C   F0 5F AF 3D   17 1C FA 18   F0 14 CA 24
F020   F0 17 17 17   32 3C F0 37   3F 7E FE 80   DC 37 F0 3A
F030   3C F0 A6 F6   80 77 C9 36   BF C9 00 00   00 00 00 00
```

-LISTING 2-

```
2 REM\ DRAWS A SINE WAVE
5 REM\ THIS IS THE MACHINE-LANGUAGE CODING THAT DOES THE ACTUAL
6 REM\ "DRAWING" FOR THE VARIOUS PLOT PROGRAMS & ROUTINES.
7 REM\ IT USES MEMORY F000-F03F(HEX).
8 DATA 0,0,0,175,122,31,111,159,87,38,204,1,64,0,123,214
9 DATA 3,250,24,240,9,195,15,240,95,175,61,23,28,250,27,240
10 DATA 20,202,39,240,23,23,23,50,63,240,55,63,126,254,128,220
11 DATA 58,240,58,63,240,166,246,128,119,201,54,191,201,0,0,0,0
12 FOR P7=61440 TO 61504
13  READ P8
14  FILL P7,P8
15 NEXT P7
16 REM\ THIS ENDS THE MACHINE-LANGUAGE CODING
90 PRINT CHR$(11)
100 FOR C=1 TO24 STEP 0.25
120 A=C*5
130 B=SIN(C)*10
140 B=24-B
150 GOSUB 50000
200 NEXT C
999 END
50000 REM\ MACHINE-LANGUAGE INTERFACE SUBROUTINE
50001 REM\ ENTER WITH A (X-AXIS), B (Y-AXIS)
50020 A=INT(A)\B=INT(B)
50040 N=(256*A)+B
50060 Q9=CALL(61440,N)
50080 RETURN
```

-LISTING 3-

Introducing the powerful, multi-processing HORIZON® 8/16 from North Star.

The turbo-charged system with outstanding performance.

The new North Star HORIZON 8/16 microcomputer can handle up to eight individual users, supporting both 8-bit and 16-bit applications simultaneously.

Its advanced, multi-processor architecture makes this powerful performance possible. Unlike other multi-user systems, the HORIZON 8/16 doesn't load up its users on a single processor; instead, it provides a dedicated processor for each individual user—at a cost no greater than that of conventional multi-user systems.

The result? No degradation in processing performance, even when there are eight users on the system.

And North Star's industry standard S-100 bus gives you the flexibility to choose your options and tailor the system to meet your specific requirements.

What's more, the new North Star TurboDOS® is many times faster than standard, multi-user operating systems—and is compatible with CP/M-80,® CP/M-86® and MP/M.™

As for reliability, over 30,000 first generation HORIZONs are still in use. And each of these can be easily upgraded to the new 8/16 architecture.

The HORIZON 8/16 outperforms everything in its class. Costs no more. And is the only multi-user micro designed to meet your needs for today, and tomorrow—simply by plugging in the options you select.

You can discover North Star's HORIZON 8/16 at more than 1,000 computer stores and system houses nationwide. Call 800-722-STAR for the location nearest you. Or write North Star Computers, Inc., 14440 Catalina Street, San Leandro, CA 94577.

NorthStar™
Simply powerful solutions.

Z-80 is a registered trademark of Zilog, Inc.
TurboDOS is a registered trademark of Software 2000, Inc. CP/M-80, CP/M-86, MP/M and CP/M are either trademarks or registered trademarks of Digital Research Inc.
Systems serviced nationwide by MAI/Sorbus Service Division.

CIRCLE 11 ON READER SERVICE CARD

Hiding Machine Code continued . . .

```
1 REM\CATzoERRSETW&LEN@<9404_CAT=<4SGNH42_47?^u2LETCOS[4:_4CONT>LETw6"~GTYP

2 REM\ DRAWS A SINE WAVE
        90 PRINT CHR$(11)
        100 FOR C=1 TO24 STEP 0.25
        120 A=C*5
        130 B=SIN(C)*10
        140 B=24-B
        150 GOSUB 50000
        200 NEXT C
        999 END
        50000 REM\ MACHINE-LANGUAGE INTERFACE SUBROUTINE
        50001 REM\ ENTER WITH A (X-AXIS), B (Y-AXIS)
        50020 A=INT(A)\B=INT(B)
        50040 N=(256*A)+B
        50060 Q9=CALL(13348,N)
        50080 RETURN

              -LISTING 4-
```

```
1 REM\
2 REM\ FIRST, RUN THIS PROGRAM TO SET LINE = 165 SPACES.
3 REM\ THEN, SET UP     1 REM\   ... FOLLOWED BY ENOUGH
4 REM\ SPACES TO ACCOMMODATE THE MACHINE LANGUAGE ROUTINE.
5 REM\ NEXT, GO MONITOR AND START THE MACHINE LANGUAGE
6 REM\ AT 3424 HEX.
7 REM\ WHEN FINISHED, DELETE ALL LINES OF THIS ROUTINE
8 REM\ EXCEPT LINE 1 (WHICH CONTAINS THE MACHINE LANGUAGE.)
9 REM\ FINALLY, APPEND YOUR BASIC PROGRAM.
10 LINE 165
11 REM\ NOTE THAT 3424HEX = 13348DECIMAL

        -LISTING 5-
```

A Machine Code Loader for MBasic-80

by Larry Costa and Steve Leibson

Microsoft Basic-80 has two techniques for calling machine code routines; the USR() function and the CALL statement. Unfortunately you are left to your own devices as to how to get the routine into memory. The Apple and IBM-PC versions of Microsoft Basic have a BLOAD statement for loading machine code routines, but MBasic-80 for CP/M does not. This article will provide you with a technique for loading machine language files into memory from a running Basic program.

A frequently used method for appending machine code routines to MBasic programs is to place them into the program using DATA statements. This is essentially loading the machine code by hand, since you are typing the code into the computer. A FOR-NEXT loop is used to POKE the bytes from the DATA statements into memory.

The problem with this approach is that it is very error-prone for large routines. Generally an assembler is used to generate the machine code for the routine. The assembler produces hexadecimal values that must be typed into the Basic program by hand. Even with short machine code routines, this approach is tedious.

Since the tools to generate machine code files already exist in the form of assemblers, linkers, and loaders, it would certainly be convenient if we had a technique that could automatically load such files. For large routines this becomes a necessity. One possible approach is to place the machine code in memory before loading MBasic. The easiest way is to load an assembled .HEX file with DDT. This and other "pre-load" methods must use SUBMIT or manual operations—not very friendly.

A more practical method is to let MBasic do the loading of the machine code from a file that has been prepared from the assembled .HEX file, which we will demonstrate. One problem with this approach is getting the object code safely tucked away in high memory and protecting it from being eaten away by MBasic's strings. We will show you how to avoid this problem also.

First, we assume that the reader is either familiar with assembly language and DDT or has someone close by who is. We further assume that the MBasic program will need to exchange information with the machine code routines. In level 5.xx MBasic and above, the CALL statement is the best way to activate the machine code because it can be used to pass parameters. Revision levels below 5.xx will have to set aside memory locations for parameter passing and call the machine code routine with the USR() function, a poor second choice. The example used below assumes level 5.xx so we can use long, meaningful variable names. A brief outline of the process is:

1. Assemble and debug the routines.
2. Make a final assembly so the resulting HEX file will reflect the code's running position above the top of MBasic workspace and below the beginning of the CP/M BDOS.
3. Load the HEX file and move it to 0100H, then save the code to disk.
4. Let the MBasic program use a CLEAR statement to protect high memory.
5. Let the MBasic program treat the machine code as a random-access file, read it in sector-size chunks and POKE the individual bytes of code to memory.

Don't worry, we will go through the procedure step by step.

Larry Costa, 8600 Concord Lane, Westminster, CO, 80030 and Steve Leibson, 4040 Greenbriar Blvd., Boulder, CO 80803

Step-by-step instructions

1. Make an assembly with ORG set to zero, get the PRN file and note the following items:
   A. The length of the code module. Call this LEN for later reference.
   B. The zero-relative address of every entry point to be used in a CALL and the zero-relative addresses of any data cells that the Basic program may reference. A zero-reference address is simply the number of bytes between the beginning of the machine code routine and a particular entry point or data location. Put these in tabular form on a worksheet similar to this:

```
LEN       = 02C3
MODEMOUT  = 0024    These are example
MODEMIN   = 01AC    values only
STATUS    = 02C3
```

   C. Find the bottom of your BDOS page by looking at location 0007 with DDT. For example, if the byte in 0007 was BB (hex), then your BDOS starts at BB00 (hex). Your machine code routine must not go any higher than BB00.
   D. Round LEN up to a page boundary minus one. In this case 02C3 rounds up to 300 (hex), then subtracting one gives us 02FF. The result should always end in FF. Subtract this from the bottom of the BDOS address less one to find where the machine code routine must start and where Basic's string space must end. The DDT H command helps here. In this case, –HBB00,2FF gives the sum BDFE and the difference B800.

   The difference is the one you want. Call it MLBASE and add it to your table of values. This will be the starting address of the machine code routine. The absolute values of the entry points and data locations can now be calculated by simply adding their zero-relative addresses obtained in (B) above to MLBASE.

   All values should be in hexadecimal. The assembly at zero code origin is to simplify finding the values required. We aren't going to actually use the code in this form, so the HEX file generated by the assembler may be omitted or discarded.
2. Edit the source file to make the ORG statement agree

2500AD SOFTWARE INC

offers you a variety of Cross Assemblers for your Z-80 CP/M® System.

Z-8000 Cross Development Package $179.50

Instant Z-8000 Software! This package allows development and conversion of software for the Z8001, 8002, 8003 and 8004 based machines on a 64K Z-80 CP/M machine. This powerful package includes:

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

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

Z-80 Macroassembler $49.50

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

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

6502 Macro Cross Assembler $79.50

6502 software on your Z-80 machine! This 2500AD software allows you to develop large, powerful assembly language 6502 programs on a Z-80 CP/M System. The program uses standard mnemonics with a straightforward Zilog Z-80 type syntax. This package includes:

- 6502 Macro Cross Assembler
- the powerful 2500AD Linker
- a 50 page User Manual

Other Macro Cross Assemblers available:

- Zilog Z-8 $79.50
- Intel 8748, 8749, etc. $79.50

All 2500AD Assemblers and Cross Assemblers support the following 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.
Plan English Error Messages—

System requirements for all programs
Z-80 CP/M 2.2 System with 64K RAM and at least a 96 column printer is recommended.

Z-8000 based versions of all programs are available. **Call 2500ADSOFTWARE, 303-752-4382.**

Now available! Call for details.

- 8086 Macro Cross Assembler
- 8080 and Z-80 to 8086 Translator
- 8086 Relocating Macro-assembler for CP/M 86
- all 2500AD products running under CP/M 86

I would like to order:

☐ Z-8000 Cross Development Package	**$179.50**
☐ Z-80 Macroassembler	**$ 49.50**
☐ 6502 Macro Cross Assembler	**$ 79.50**
☐ Zilog Z-8	**$ 79.50**
☐ Intel 8748, 8749, etc.	**$ 79.50**
Check one: total	**$**______
☐ 8" Single Density shipping/handling	
☐ 5¼" Osborne ($6.50 per unit)	**$**______
total order	**$**______

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.)
____
Signature____________________

2500AD SOFTWARE INC
P.O. Box 441410 Aurora, CO 80014 303-752-4382

CIRCLE 83 ON READER SERVICE CARD

with the desired MLBASE, in this cases B800H. Reassemble the module, keep the HEX file, and get a final (we hope) listing . . . Bugs, you know!

3. Load the HEX file with DDT, then move it to 100H. Here is the DDT dialog:

```
DDT MLCODE.HEX
NEXT    PC
BA24    0100          (NEXT - 1 is the last byte of the
                      routine)
-MB800,BA23,100       (Move the code to 100H. Note
                      BA23 = NEXT - 1)
-GO                   (Go zero to leave DDT.)
```

4. Save the object code. The decimal amount to save is the value of the high-order hex byte of LEN plus 1. In this case the high byte of LEN is 02. We need to save three blocks, so . . .

```
SAVE 3 MLCODE.COD (The COD suffix could be
                  anything.)
```

Our machine code file is now on disk. Let's look at the Basic program in Listing 1, which uses this file. Lines 10 through 50 provide the program with all the information we obtained in step 1 above. The CLEAR statement on line 60 is very important. It reserves space for the machine code, protecting memory above address B7FF (hex) from use by MBasic-80.

The subroutine that starts at line 30000 actually loads the machine language file into memory. It reads the file 128 bytes (one standard CP/M sector) at a time and POKEs each byte into memory, starting at MLBASE. This is done for the proper number of sectors, and the loading is then complete. You don't need to read the file one sector at a time, but it is a convenient size to deal with.

This technique works because MBasic doesn't know that the bytes in T$ are not simply ASCII characters it got from the file MLCODE.COD. By taking the ASC() of each byte, we get the right value to POKE into memory starting with MLB%. The offset, K%, is needed so that each sector starts loading 128 bytes higher than the prior sector, then J increments to POKE each byte into the proper memory location.

Finally, if you make changes to the machine code, don't forget to adjust the relative entry points if they move from the original postions in the code! Also, if the changes you make affect the length of the code to the point where it no longer fits in the reserved area, you will need to change the start address of the routine.

Listing 1

THE MBASIC CODE REQUIRED FOR THE LOADER

```
10 MLB%     = &HB800              ' MLBASE
20 MODOUT% = &H24 + MLB%          ' ENTRY POINTS ARE
30 MODIN%   = &H1AC+ MLB%         ' RELATIVE TO CODE BASE.
40 STAT%    = &H2C3+ MLB%         ' DATA CELL REFERENCE.
50 MLLEN%   = &H2C3               ' ******** IMPORTANT *********
60 CLEAR, &HB7FF                  ' NO STRINGS BEFORE THIS CLEAR!!
70 MLFIL$ = "MLCODE.COD"          ' MACHINE CODE FILE NAME HERE
80 GOSUB 30000                    ' LOAD THE FILE
    .
    .
XX CALL MODIN% (MODIN$, STAT%)    'GET MODEM INPUT (example)
    .
    .
XX CALL MODOUT% (MOUT$)           'SEND TO MODEM    (example)
    .
    .
29999 ' ROUTINE TO LOAD ML TO HIGH MEMORY
30000 OPEN "R", #2, MLFIL$        'FILE NUM IS UNIMPORTANT
30010 FIELD #2, 128 AS T$         'SECTOR LENGTH STRING
30020 NS% = MLLEN%/128            'CALC NO. SECTORS TO GET
30030 IF (MLLEN% MOD 128) >0 THEN NS% = NS%+1 'ROUND UP IF NEEDED
30040 FOR I = 1 TO NS%:
        GET #2, I:
        K% = 128 * (I-1):   ' Sector offset for load address (I)
        FOR J = 0 TO 127:
              POKE MLB%+K%+J, ASC( MID$ (T$,J+1,1)):
        NEXT J:
      NEXT I
30050 CLOSE #2
30060 RETURN
30070 ' ----- END OF ML LOAD ---
    .
    .
32000 END
```

The subroutine that starts at line 30000 actually loads the machine language file into memory. It reads the file 128 bytes at a time and POKEs each byte into memory, starting at MLBASE. When this is done for the proper number of sectors, the loading is then complete.

Single-Density Disk Formatting

Increase storage capacity with the FD 1771 floppy disk controller

by Robert Lurie

If you are still using a disk controller based on the Western Digital FD1771 chip, you can increase the CP/M file capacity of your 8″ single-density disks from 241K to 354K by formatting each of the non-system tracks (track 2 through track 76) into two 2432-byte sectors.

Each sector contains the equivalent of 19 128-byte logical records. For this reason, the sectors cannot be deblocked using the algorithm provided by Digital Research: as listed in Appendix G of the *CP/M 2.2 System Alteration Guide*, DEBLOCK.ASM is predicated on the use of sectors whose byte size is 128 times an integer power of two. Instead, I recommend the use of a modified track-buffering procedure similar to the one that was published in my article "Track-Buffered I/O Routines for the Tarbell Single-Density Disk Controller" in the May 1983 issue of *Microsystems*. You can buffer either the entire 4.75K track or, to save RAM space at the cost of some speed, just a single sector.

The FD1771 interprets the value, *N*, of byte 4 of the sector ID field, the so-called sector-length byte, in two different ways depending on the value of bit 3 of the READ or WRITE command byte that is issued to it. If bit 3 equals zero, it evaluates the length of the sector as 16**N*. If bit 3 equals one, it evaluates the length of the sector, in accordance with the IBM 3740 standard algorithm, as 128*(2***N*). In the case of 128-byte sectors, we format the disk so that the sector-length byte equals zero, since 128*(2**0)=128, and we set bit 3 of the READ and WRITE command byte to one. In the case of the 2432-byte sectors, we format the disk so that the sector-length byte equals 152 or 98 hex, since 16*152=2432, and we reset bit 3 of the command byte to zero.

The sector-length byte can be read by means of the FD1771 READ ADDRESS command. By reading the sector-length byte of any sector on any data track, we can determine whether we are dealing with a standard IBM-formatted single-density disk or a 2432-byte/sector single-density disk. The appropriate times to read this byte are whenever a drive is selected for the first time following a warm or cold boot and whenever the BIOS's SELDSK subroutine is called with a request for a drive that is different from the last drive selected. Based on this information, we can select the proper sector translation table and disk parameter block addresses to return to BDOS, and also the proper value of bit 3 for the next READ or WRITE command byte. A BIOS that embodies these automatic format detection and dynamic system reconfiguration routines has a much friendlier user interface than one that forces the user to remain continually aware of all current disk formats and to log in drives with fictitious (i.e., "logical") names whenever disk formats are changed.

BIOS space can be increased from 0.875K to 1.625K by formatting system tracks 0 and 1 into 29 (rather than the

Robert Lurie, 8 Tingley Rd., Morristown, NJ 07960

```
A>DDT SYSGEN.COM
DDT VERS 2.2
NEXT  PC
0500 0100
-S129
0129 1A 1D
012A 01 .
-S144
0144 00 1B
0145 00 1C
0146 00 1D
0147 00 .
-^C
A>SAVE 4 SYSGEN29.COM
```

Figure 1. *SYSGEN.COM version 2.0 patch for systems tracks with 29 128-byte sectors.*

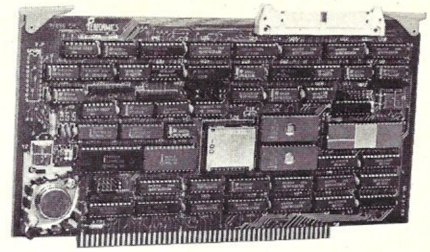
SYSTEM TRACKS 0-1 Number of bytes	Hex value	DATA TRACKS 2-76 Number of bytes	Hex value
11	FF	11	FF
6	00	6	00
1	FE	1	FE
1	00 thru 01 (track no.)	1	02 thru 4C (track no.)
1	00	1	00
1	01 thru 1D (sector no.)	1	00 thru 01 (sector no.)
1	00	1	98
1	F7	1	F7
11	FF	11	FF
6	00	6	00
1	FB	1	FB
128	E5	2432	E5
1	F7	1	F7
14 *	FF	103 **	FF
c. 122 ***	FF	c. 61 ***	FF

* Write bracketed field 29 times
** Write bracketed field 2 times
*** Write FF's until FD1771 interrupts out

Figure 2. *Formatting bytes to write using the FD1771 WRITE TRACK command.*

INTRODUCING PERFORMICS' PROBLEM SOLVERS

THE FASTEST S-100 CPU/MEMORY COMBINATION AVAILABLE!



MAKE NO MISTAKE! Before PERFORMICS developed the *PRIVATE ACCESS BUS*, to run an INTEL 80286 on the S-100 bus you had to slow it down by 50%. Hard to believe? Consider the following:

The IEEE-696 spec defines a bus cycle to be three clock ticks in duration	+	INTEL defines a bus cycle for the 80286 to be two clock ticks	=	Poor performance unless you choose PERFORMICS' problem solvers with the *PRIVATE ACCESS BUS*

PRIVATE ACCESS BUS

P-286

FEATURES:
- 80286 CPU
- Programmable Interrupt Controller
- Three 16 Bit Timers Cascadable
- Dual Ported Architecture (S-100 + *PRIVATE ACCESS BUSSES*)
- Socket for 80287 Math Processor Running Independent Clock (Option)
- Two Jedec 28 pin sockets for up to 64K EPROM/ROM

P-128

FEATURES:
- 128K CMOS Static RAM
- Addressable on 64K Boundaries
- Battery Backup (Option)
- Dual Ported Architecture (S-100 Bus + *PRIVATE ACCESS BUSSES*)
- Cascadable
- No Wait State Operation with P-286

S-100 BUS

"Private Access Bus + S-100 = THE WINNING COMBINATION"

If you are going to pay for performance, insist on getting it. Call for pricing and details now!

PERFORMICS INC.

P.O. BOX 3207 • NASHUA, NEW HAMPSHIRE 03061 • (603) 881-8334
TELEX 294088 PERF UR

CIRCLE 34 ON READER SERVICE CARD

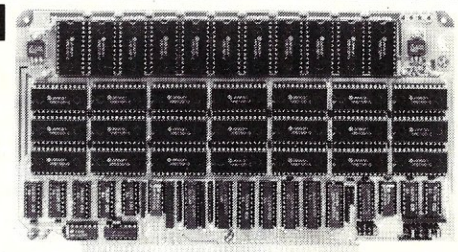
CHECK SUNTRONICS NEW LOW PRICES

S-100 Products

S-100
64K MEMORY BOARD

Assembled and Tested without RAM

$155.00



64KSM Assembled and Tested with 64k RAM (32-6116's) **339.00**

The 64KSM is an outstanding value in S-100 Memory. Using the CMOS 6116P-3 (150nS) RAM chips, it runs at excess of 6MHz using less than ½ amp. Flexible bank select circuitry works with Cromemco, Cromix and CDOS. The 64KSM allows four (4) 16K banks to be used independently or connected as 16K and 48K, 32K and 32K or one 64K. 2716 system PROM's can replace one or all 6116 RAM chips. IEEE-696 interface works with Northstar, Morrow, IMS and many others. Works great with 8088 boards. 24 bit extended addressing with bank selecting allows more than 16 MB memory operating system.

UFDC-1 Floppy Controller, Assembled and Tested **245.00**
UFDC-1 Floppy Controller, Kit **225.00**

The UFDC-1 Floppy Controller uses a WD1795 which runs either and/or 8"/5¼" Disk Drives. For detailed review see August 83 issue of *Microsystems*, page 76.

SBC-880 Z80A CPU, A&T **$169.00**
SBC-880 Z80A CPU, Kit **149.00**
4MHz Z80A CPU boards with Serial/Parallel Ports. For detailed review see August 83 issue of *Microsystems*, page 76.

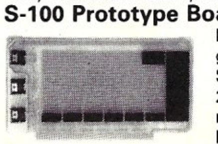
Mother Boards & Card Cages

SLOTS	BARE BD	KIT	A & T	w/CAGE
6	$12.00	$37.00	$52.00	$77.00
8	16.00	48.00	73.00	108.00
12	22.00	68.00	103.00	143.00

10MHz, No termination. Includes power indicator and wiring for muffin fan. Uses OKI connector for solderless installation and removal of power & reset lines.

CLOCK CALENDAR A&T **115.00**
CLOCK CALENDAR Kit **95.00**
This S-100 Clock Calendar Board has 4 interrupts, Time, Day of Week and Battery Backup.

S-100 Prototype Board



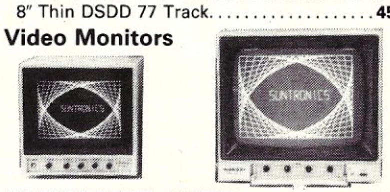
Double Sided glass with gold plated, numbered S-100 terminals. Matrix of 25 x 78 solder plated donuts on .15" x .1" spacing. Locations for headers and regulators. Great for 14, 16, and 24 pin IC's.

SUN-721 .... **12.95**

General Products

Mitsubishi Disk Drives, 5¼" and 8" Slim
5¼" Thin DSDD 40 Track **310.00**
8" Thin DSDD 77 Track **450.00**

Video Monitors



SAMWOO GREEN 9" 18MHz **$115.00**
SAMWOO AMBER 9" 18MHz **119.00**
SAMWOO GREEN 12" 12MHz **124.00**
SAMWOO AMBER 12" 12MHz **127.00**
Composite video I/O. 750 lines resolution. 75/10K ohm impedance. Note: Please add $7.50 shipping and handling for the video monitors.

Assembled Connectors and Cables

Centronics Type Cable Assemblies 36 pin flat cable with male to male or male to female connectors.
4 ft.—**$24.95** 6 ft.—**$27.95**

RS232 Cable Assemblies 25 pin flat cable with male to male or male to female connectors.
5 ft.—**$29.95** 10 ft.—**$35.95**

Assemble Your Own Cables

Ribbon Cable		Card Edge Connector	IDS Socket	Header (w/w)
Cond.	Price/ft.	Price	Price	Price
20	$0.50	$2.46	$3.06	$4.24
26	0.65	4.80	3.87	4.68
34	0.83	5.93	6.30	5.25
40	1.00	6.90	7.20	5.95
50	1.30	7.58	7.50	6.20

IDS DB-25P Connector **$5.95**
IDS DB-25S Connector **$6.25**

Super Sale

P/N	8-24	25up	P/N	8-24	25up
2716 (450nS)	**3.80**	**3.55**	**6116P-3** (150nS)	**6.10**	**5.75**
2732 (450nS)	**4.15**	**3.98**	**2114L-2** (200nS)	**1.62**	**1.62**
2532 (450nS)	**5.95**	**5.95**	**4164**	**Call**	**Call**
2764 (28 pin)	**5.95**	**5.95**	**Z80A CPU**	**4.99**	**4.99**

SUNTRONICS CO., INC.
12621 Crenshaw Blvd., Hawthorne, CA 90250

STORE HOURS: MON.-FRI. 9:00am to 6:00pm
SATURDAY 10:00am to 5:00pm

CALIFORNIA **213-644-1149** (for Tech Info and Calif. orders)

OUTSIDE CALIFORNIA TOLL FREE **1-800-421-5775** (Order Desk Only)

Mail Order—Minimum Order: $10. Send Check or Money Order to: **P.O. BOX 1957 Dept B, HAWTHORNE, CA 90250.** Visa or MasterCard (please include expiration date). Add $2.00 postage and handling for first 3 pounds plus .50 for each additional pound to your order. California residents add 6% sales tax.

CIRCLE 78 ON READER SERVICE CARD

usual 26) 128-byte sectors. Compatibility is still maintained with any existing ROM-based routine for loading the coldstart boot program on sector 1, track zero. Of course the coldstart boot program itself must be modified to load into memory whatever extra sectors you decide to use for your BIOS. A system-generation program also must be written for writing to and copying the enlarged system tracks. Fortunately, SYSGEN.COM can easily be modified to perform these last two functions. Figure 1 shows how to do it.

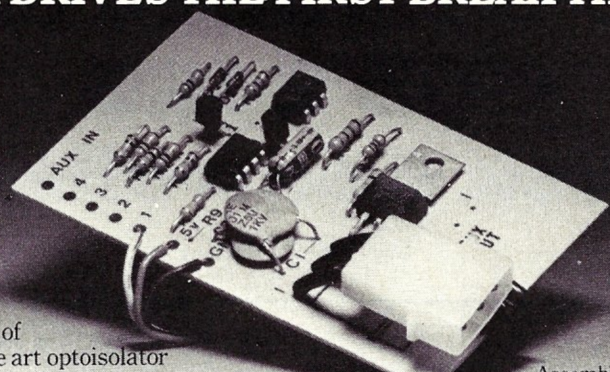
Figure 2 shows the byte sequences used to write to a disk in order to create the formats we just described. The bytes are written using the FD1771 WRITE TRACK command. Although it is not essential, we nevertheless strongly recommend that each WRITE TRACK command be followed by a nonsynchronized READ TRACK command, during execution of which a count is made of the number of bytes that were written to the track. The byte count should fall within plus-or-minus 20 bytes or so of the 5208.333 bytes per track that is the nominal capacity of an 8″ single-density disk. This serves to verify that the disk was up to speed at the time it was formatted, and thus provides assurance that the physical length of the gaps between sectors is large enough so that writing data to one sector will not result in overwriting the ID field of the succeeding sector, thereby destroying it. Recurring failure to meet the byte-count test is an indication that the disk hole or drive hub may be worn, or that the drive's spindle motor belt needs to be replaced, or its pulleys require cleaning. Following this test, a sequential data read of all the sectors on the track should be made to complete the verification process.

I was prompted to write this note after reading the article entitled "Triple-Density Floppy Disk Storage" that appeared in the February 1983 issue of *Microsystems*. The fact of the matter is that not only can no existing double-density disk controller yield three times the capacity of a standard IBM-formatted single-density disk, but no existing double-density disk can yield even two times the capacity of a single-density disk that has been optimally formatted for use with the FD1771. (The optimum 8″ double-density format, incidentally, contains nine 1024-byte sectors per track. I mention it here because at no point was it explicitly mentioned in that article.)

Boards based on the FD1771, such as the once-popular Tarbell single-density disk controller, the Versafloppy I, and the Cromemco 4FDC, are quite a bit more powerful than their current usage would make them appear. I hope this article will help you better appreciate their potential before you discard them as obsolete.

Bob Lurie designs and manufacturers precision optical components. He is the inventor of a type of telescope known as the catadioptric conic-mirror anastigmat. Bob's computer interests include the development of arithmetic software, systems programming, and the design of sailboats.

WE GAVE YOUR DRIVES THE FIRST BREAK THEY EVER HAD...



Our DCU is the original Drive Control Unit that turns floppy drives off during periods of inactivity by using a state of the art optoisolator with zero crossover control and built in activity monitor.

We've continued to improve the design (it's the size of a business card to fit within the drive), ease installation time (about 15 minutes) and models are now available for virtually all popular 8 inch drives (including a foreign version).

So for those of you, who are still grinding down your drives, wearing out media and exposing yourself to unnecessary noise...isn't it time to give them a break?

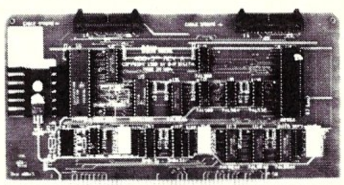
Assembled and tested $49.95
Kit with Documentation $29.95
Type of drive MUST be stated with order.
NY residents add local tax. Include $1.50 for postage and handling.

OPTRONICS TECHNOLOGY
P.O. Box 81, Pittsford, N.Y. 14534, (716) 377-0369

CIRCLE 188 ON READER SERVICE CARD

THE 488+3 IEEE 488 TO S-100 INTERFACE

IEEE-488



S-100

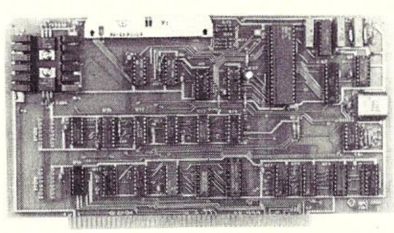
- Handles all IEEE-488 1975/78 functions
- IEEE 696 (S-100) compatible
- MBASIC subroutines supplied; no BIOS mods required
- 3 parallel ports (8255A-5)
- Industrial quality; burned in and tested
- $375

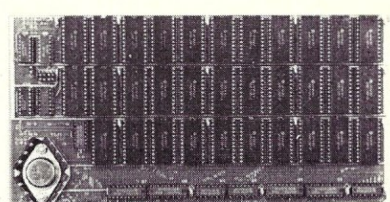
[Dealer inquiries invited]

D&W DIGITAL
20655 Hathaway Ave.
Hayward, CA 94541 (415) 887-5711

CIRCLE 66 ON READER SERVICE CARD

MCP MAKES HIGH PERFORMANCE S-100 BOARDS AFFORDABLE





MCP/SCC-6 Six MHz S-100 CPU

- Six MHz Z80B, operates at 2, 4 or 6 MHz.
- Two full RS232 serial ports.
- Bauds rates from 50 to 19.2k.
- Three 8 bit programmable parallel ports.
- Monitor eprom (2716) and 1k 2114 ram onboard.
- Eprom and ram can be disabled on system boot. **PRICE $200.00**

MCP/FDC DOUBLE DENSITY FLOPPY DISK CONTROLLER

- Single or double density operation.
- Single or double sided drives.
- Allows mixed density & mixed drives.
- Supports Shugart, Qume, Siemens, Mitsubishi or compatable 8" disk drives.
- CP/M 2.2 bios & support utilities are supplied on single density diskette. CP/M 86 & MS-DOS support available.

PRICE $200.00

MCP 64KS STATIC MEMORY BOARD

- High performance 150 ns Cmos ram.
- Extended addressing.
- Perfect for high performance Z80 and 8088.
- Available as 64K x 8 or 32K x 8 module.

PRICE 64K $238.00 32K $180.00

MCP COMPUTER PRODUCTS • 6992 El Camino Real, Suite 104-444 • Carlsbad, CA 92008 • (619) 438-3270

CIRCLE 44 ON READER SERVICE CARD

8080 Operation of the CCS 2422 Disk Controller

by Bill Kibler

As a test S-100 chassis, the IMSAI 8080 still cannot be beaten—especially when a new disk controller is added. At least that was my original position. After many months of work, I am still impressed with the IMSAI, but not with my choice of disk controller, a CCS 2422. The sales literature indicated it would work with the 8080 CPU chip; however, the manuals clearly state that the boot loader and monitor must be rewritten in 8080 code. This article covers rewriting the monitor prom, boot loader, and other important facts on 8080 operation with the CCS 2422.

CCS 2422

The CCS 2422 disk controller uses a WD 1793 controller chip and has an 2716 EPROM with Z80 monitor. The unit is intended to be used with the CCS 2810 Z80 CPU. The monitor has a modified jump table for entries, as well as standard IOBYTE-handling routines. The 2810 CPU is a 2 or 4MHz Z80 with serial port on board. The software and monitor routines use the serial port as the TTY function, requiring changes for other serial or terminal operation. Two versions are currently out: V.001 and V.002. The V.002 should work better; however, I have worked only with V.001. Several options are possible through jumpers on the controller: AUTO boot; ROM, BANK, PR and WAIT enables. All but the AUTO are solder jumpers and, except for the wait, will not need changing.

There are two wait circuits on the controller, a prom and chip wait, and an auto delay until controller data is ready. The prom wait adds an extra cycle whenever the board is accessed; with a 2MHz 8080 the wait in not needed. Adding a jumper to the WAIT jumper pads will disable the wait circuit (bending out pin 10 or U41 for a temporary fix also works). The auto wait is a software-enabled circuit which waits until the 1793 chip signals that it is ready to receive or send data (the manual has a good explanation of this feature). The 1793 timing requires that data either be supplied or removed as fast as the chip operates, otherwise the operation will be aborted. At 2MHz, this requires memory that has total cycle times 200 to 250 ns, or 4MHz memory even when running at 2MHz. For larger bus systems, you may have to use a 74 S244 output bus driver if the 74 LS244 does not have sufficient power.

For use with PHANTOM operation, a 1K pullup resistor may be needed on S-100 pin 67, if there is not one elsewhere. U21 is the memory-mapped option and should not have a chip in the socket when shipped from the factory. The board has several other disk jumpers but, depending on the type of drives used, it generally does not need changing. Consult the manual first for drive types. If the IMSAI VI0 board is intended for use, jumpers from the CCS phantom to the VI0 board will be needed to make sure any underlying memory is disabled. This can be done using diodes and the A16 to A19 decoder (see Figure 2). The CCS uses I/O ports for all data and status operations; however, port 04H cannot be used as the IMSAI serial port (CCS status port). CCS uses ports 30H to 34H and 04H for control or data, with 40H being used as bank select.

Software bugs

The monitor and boot prom are both in Z80 code and will need changing. The STDBIOS and CCCBIOS also will need changes, not only for different I/O but for some missed 2MHz problems. The BIOS has the I/O setup for the CCS 2810, which would allow the ports to be matched, eliminating these changes. Two BIOSes are used: one is part of the boot and only handles one format, while the other is longer and is loaded through a relocator program after boot. In testing the system it was found that some changes for 2MHz operation were made to the relocated BIOS, but the same changes were not done to the STDBIOS by CCS. This prevents write operations when under the STDBIOS, but swapping the STA and OUT operations in the WRDAT removes the delay and allows write operations under STDBIOS. STDBIOS will not respond to control S unless the ADI FE and ADI BF in the TTY routines are set to ADI FF.

Timing problems were encountered in the CCBIOS, causing lockup of the system. It was found that NOPs were needed between two consecutive calls to the controller. RDAT needs the OUT and STA swapped for proper read. Remove the EI (if no interrupts are to be used) in EOJ and use a NOP after swapping the STA and OUT commands in EOJC. In SEEK1 put a NOP between LDA and OUT and a NOP on both sides of IN DTRCK in RDWRT1, as my system locked up several times at this IN instruction. A NOP should also be put between the IN and OUT instructions at IDRD2. Not all 8080 systems may need all the NOPs, but mine wouldn't run the CCCBIOS until the NOPs were added (which is why the STDBIOS needs to be fully usable).

The Z80 code is mainly the shorter JUMP RELATIVE (a two-byte jump instruction) and can be replaced by the longer, but same function 8080 equivalent. After replacing the Z80 codes, the boot will be 5 bytes too long, and will require the dropping of a disk table. Two separate boat routines will be made, a mini and maxi. The monitor changes supplied with this article list the new code routines for those who wish to burn their own version of the monitor. I have written a modified version that provides a different approach, mainly to allow for startup without a running CP/M system. Several monitor functions, as well as the IOBYTE test routines, have been removed and replaced with a console finder program and the 8080 boot program, which gets relocated for boot operation. Auto boot, however, uses the boot from the disc and

Bill Kibler, Kibler Electronics, P.O. Box 487, Cedarville, 96104

can be used once the software is changed. The jump table is also the same as CP/M 2.2, which will allow the prom to work as a BIOS. This approach is achieved by locating several routines, which can be changed by the user, outside of the monitor. An example is the DPB table. Currently my prom will return a 00 in H (which indicates an error to CP/M) after being returned from address 02CH where a C9 was put during INIT. Should a DPB table be available, the RET (C9) gets changed to a JUMP to the location of the routine that handles the disk select and DPBs (see the CP/M Alteration Guide for more information on DPBs).

CCS application note 7 covers the rebuilding of MOVCPM, but does not cover the addition of the bit map for the new 8080 CBOOT. It will be necessary to figure the bit map by hand and patch it into the new MOVCPM (see listing 5). The application notes should provide all needed information to properly modify the system to run on an 8080. Users with older and slower disk drives should check STEP5 and STEP8, the step rate variable. You may find a longer step time is needed. The step values are in both STDBIOS and CCCBIOS.

CCSYGEN

For users with mini (5 ¼″) disk systems, an extra problem has been created by CCS—namely, system generation. A separate program for generating systems on mini disks, CCSYSGEN.COM, is supplied and is supposed to work. Unfortunately, this program uses the same disk routines as CCCBIOS, and if your CCCBIOS does not work without the NOPs, CCSYSGEN also will not work. To load new systems on mini disks, use the monitor's READ and WRITE commands. The first step will be to get a usable system. This is done by reading the system tracks into memory, then using the S command to patch the IN/OUT routines, followed by writing it back on the disk. After a running STDBIOS system is working, the new system can be generated by loading the program into memory, resetting to monitor, and then writing it to disk from memory (see listing 4 for sample system generation).

F800 Use

In 64K systems, the current address of the prom is F000, which leaves the area above the prom unused. The BIOS table is located at F200H, which prevents using the prom as a BIOS with the standard CP/M. Burning a prom for F800 would help if the board could be made to work on that address. Chip U41 has an extra inverter; thus by cutting the line from U37 to U23 (address buffer to address decoder) and inserting an inversion (see Figure 1) the board will see F800 (U37 pin 18 to U41 pin 1, and U41 pin 2 to U23 pin 4). This will get the prom at F800, the STDBIOS can now be modified with jumps (at the CCS JUMP TABLE at F200H) into the F800 prom.

In addition, don't forget to remove the OUT to port 40H, in order to turn off the bank in the BOOT loader (or to turn it back on in the STDBIOS). The new STDBIOS could then be changed to have plenty of room to handle IOBYTE tests, list device routines, and disk parameter blocks, which would eliminate the need of a RELOCATED BIOS routine.

Conclusion

After the changes, I found that the CCS 2422 worked fine in single density, and is sensitive to memory speed and double density at 2MHz. To date, CCS has not responded to my pleas for help with this project; several calls to their customer support people only showed that I know more about their system than they do. The above statements are not intended to either endorse or reflect negatively on CCS, but to point out the difficulties I had and help others with similar problems. The price of the controller makes it attractive, especially when CP/M is included. The current interest in dual processing (8085/8088) also makes the need for 8080 operation of the CCS 2422.

Note

Kibler Electronics can supply burned proms for 8080 operation. These proms are set to check for some common serial I/O devices and have an 8080 boot loader, relocated from the prom, to boot with. The ports for serial data are 1, 2, 20, 22 and check bit 1 and 2 of ports 0, 3, 21, 23 for data OUT ok and data ready IN respectively. The boot relocator program asks for memory size and then changes the boot loader as required. This prom should allow for initial operation and changing of programs without a second system. An F800H version is also available for $35 and can be obtained by contacting Kibler Electronics, 2918 33rd St., Sacramento, CA 95817.

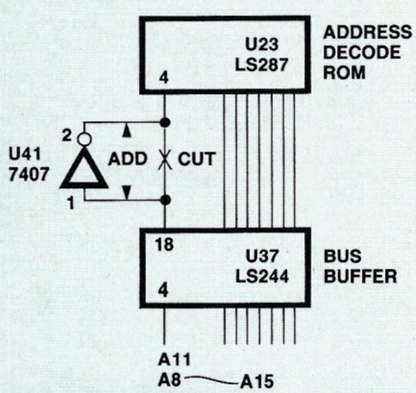


Figure 1. F800 modification to CCS 2422 uses spare inverter of U41.

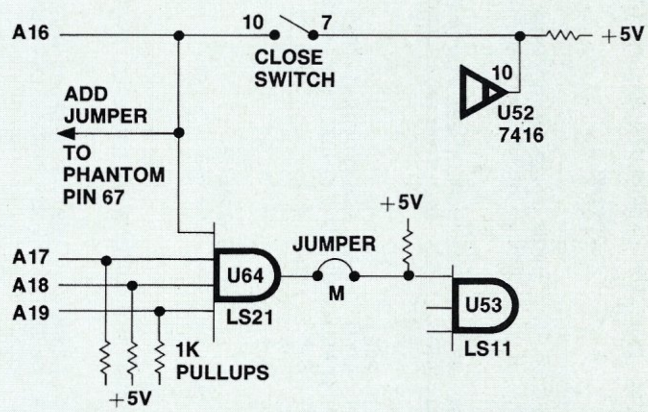


Figure 2. VIO modifications for operation with CCS 2422: jumper M, close switch 7, add pullups to A17 through A19.

The following are the needed changes to the STDBIOS for 8080 operation:

LISTING 1

```
:
:
STEP5:     EQU  3     ; CHECK THESE VALUES FOR PROPER STEP RATE
STEP8:     EQU  1     ; SOME OLDER DRIVES REQUIRE SLOWER VALUES
:
:
TTST:      ADI  0FFH ; MAKE SURE THESE ROUTINES USE FF OR THEY
TTOST:     ADI  0FFH ; WILL NOT STOP ON ^S UNDER CP/M
:
:
WRDAT:     ORI  20H
           STA  CMND  ; THIS COMAND FIRST THEN OUT
           OUT DCMMD ; OR CONTROLLER WILL ERROR AND STOP
WRT1:      MOV  A,M
:
```

LISTING 2

The following are the needed changes in CCBIOS for 8080 operation:

```
:
STEP5 AND STEP8: SET TO SAME VALUES USED IN STDBIOS
:
:
RDAT:      OUT  DCMMD      ;SET UP A DELAY BEFORE READING
           STA  CMND       ;FROM CONTROLLER AGAIN
READ1:     IN   DDATA
:
:
EOJC:      OUT  DCMMD      ;AGAIN SWAP FOR DELAY BETWEEN
           STA  CMND       ;CONSECUTIVE CONTROLLER OPERATIONS
EOJ:       NOP             ;USE NOP IN PLACE OF EI UNLESS USING
           IN   DFLAG      ;INTERRUPT SYSTEM
:
:
SEEK1:     LDA  SECTOR
           OUT  DSCTR      ;AGAIN DELAY BETWEEN CONSECUTIVE
           NOP             ;OPERATIONS
           IN   DTRCK
:
:
RDWRT1:    ADI  RDSEC
           MOV  C,A
           NOP             ;DONOT KNOW WHY BUT NEEDS NOP TO
           IN DTRCK        ;KEEP FROM CRASHING HERE
           NOP
           ORA  A
:
:
IDRD2:     IN   DSCTR
           NOP
           OUT  DTRCK
```

LISTING 3

The following are changes in the MOSS monitor for 8080 operation.

```
:
USE THESE CODES FOR 8080 EQIVALENT OF THEIR Z80 CONTERPART:
:
DJNZ=      DCR       B
           JNZ
:
JRNZ=      JNZ
:
JRNC=      JNC
:
LDIR=      NAME:     MOV       A,M
                     STAX      D
                     INX       D
                     INX       H
                     DCX       B
                     JNZ       NAME
:
JR=        JMP
:
JRC=       JC
:
OMIT THE FOLLOWING ROUTINES:
:
BYE, XMNE, IOBYTE, I8250,
:
PUT QPRT IN PLACE OF ABOVE ROUTINE NAMES IN TBL:
:
CHANGE LISTED ROUTINES TO NEW CODE LISTED BELOW:
:
; I/O OUTPUT ROUTINE
;
INPT:      CALL      EXPR1     ;GET I/O PORT
           POP       B         ;PUT IN BC
           MOV       A,C       ;GET PORT # IN A
           STA       INBUF+1   ;STORE IN BUFFER
           CALL      INBUF     ;DO INPUT USING BUFFER
           MOV       E,A       ;PUT DATA IN E
           JMP       BITS2     ;JMP TO OUTPUT IT
;
OUPT:      CALL      EXPR      ;GET I/O PORT
           POP       D         ;GET DATA
           POP       B         ;GET PORT #
           MOV       A,C       ;SET UP PORT .
           STA       OUBUF+1   ;PUT IN BUFFER
           MOV       A,E       ;GET DATA IN A
           CALL      OUBUF     ;OUTPUT IT
           RET
;
; I/O BUFFER INITIALIZATION ROUTINES TO BE CALLED AT INIT TIME
;
INBUF:     EQU       0CH       ;UNUSED INTERRUPT
OUBUF:     EQU       01CH      ;VECTOR AREAS
;
INTBUF:    MVI       A,0DBH    ;LOAD IN COMAND
           STA       INBUF     ;PUT IN BUFFER
           MVI       A,0D3H    ;LOAD OUT COMAND
           STA       OUBUF     ;PUT IN BUFFER
           MVI       A,0C9H    ;LOAD RETURN COMAND
           STA       INBUF+2   ;PUT IN BUFFERS
           STA       OUBUF+2
           RET
;
:
REMOVE ALL LINES FROM  EXAF IN      RS8:      (F42F+08)
                 TO    PUSH B                 (F43F C5)
AND:             FROM  POP  B       EXIT:     (F445 C1)
                 TO    EXX  IN                (F455+D9)
```

```
:
INSTALL INITIALIZATION FOR I/O DEVICES IN I8250 (F49F)
:
REPLACED HEXN WITH NEW ROUTINES LISTED BELOW
:
; ADD AND SUBTRACT ROUTINES
HEXN:    CALL    EXLF     ;GET VALUES
         PUSH    H        ;SAVE H VALUES
         DAD     D        ;ADD THEM
         CALL    LADRB    ;OUTPUT ADD
         POP     H        ;GET VALUE AGAIN
         MOV     A,H      ;GET PART OF VALUE
         CMA              ;TWOS COMPLIMENT
         MOV     H,A
         MOV     A,L      ;GET LOWER BITS
         CMA
         ADI     01H      ;ADD ONE
         MOV     L,A      ;MAKE COMPLIMENT OF VALUE
         ORA     A        ;CLEAR A
         DAD     D        ;CAUSE SUB ON ADD
         JMP     LADRB    ;OUTPUT IT AND RET
;
:
CHANGE ROUTINES IN TTST: THROUGH TTYOUT: TO REFLECT NEW I/O
DEVICES.
:
CHANGE THE LAST ROUTINES IN READ AND WRITE FOR USE OF DIRECT I/O
AND NOT Z80 I/O.
:
;
RDAT:    OUT     DCMMD    ;CHANGED TO REFLECT SLOWER CPU
         STA     CMND
READ1:   IN      DDATA    ;IN DISK DATA
         MOV     M,A      ;MOVE TO BUFFER
         INX     H
         DCR     B
         JNZ     READ1    ;GO DO AGAIN IF STILL DATA
         DCR     D
         JNZ     READ1
         ANI     09CH     ;ISOLATE ERROR BITS
         RET
;
;
WRDAT:   ORI     20H      ;ADD WRITE COMMAND
         STA     CMND
         OUT     DCMMD    ;DISK COMMAND PORT
WRT1:    MOV     A,M      ;GET DATA
         OUT     DDATA
         INX     H
         DCR     B
         JNZ     WRT1
         DCR     D        ;IN CASE >256 BYTES
         JNZ     WRT1
         JMP     EOJ
;
; The GOTO routine can be shortened considerably by dropping use
; of the breakpoint option ( saves over 70 bytes ).
; Breakpoints can be set manually by using 'S' function
; and putting RST 1 (0CFh) which causes jump to location 09h.
GOTO:    LXI     H,REST
         SHLD    9        ;SET UP POSSIBLE RETURNS
         LXI     H,RS9
         PUSH    H        ;PUSH A RET ONTO STACK
         CALL    EXPR1    ;GET GOTO ADDRESS IN H
         POP     H
         PCHL
;
; Without breakpoints, REST can be shortened and again save
; over 60 bytes. This routines provides reseting system after
; RST or error condition. Use D command to inspect pushed
; registers after restart.
;
REST:    PUSH    H
         PUSH    D
         PUSH    B
         PUSH    PSW
         CALL    MEMSIZ
         XCHG
         LXI     H,10
         DAD     SP
         JMP     WINITA
RS9:     PUSH    H
         RST     1
;
EXIT:    POP     D
         POP     B
         POP     PSW
         POP     H
         SPHL
         LXI     H,0
         JMP     0
ENDX:    EQU     $
;
;
:
The above listing should provide enough information to burn a new
EPROM and get the system up and running.
:
```

LISTING 4

```
;
; CCS routines for generating a system under monitor operation
; starting with existing systems on disk.
;
MOSS MONITOR 2.2          ;system sign on
-P0 12 0                  ;set for disk 0, mini 12h sectors
                          ;and one sided
-Q0 0 1                   ;track 0,side 0, sector 1
-R0100 1AFF               ;read needed sectors to fill
                          ;memory from 0100 to 1AFF
-DXXXX XXXX               ;display memory to find changes
-SXXXX                    ;change memory at XXXX, use space
                          ;to continue, cr to end command
-Q0 0 1                   ;reset to beginning TK & SCTR
-W0100 1AFF               ;write changes back to disk
-B                        ;now see if system will boot...
;
; If a modified MOVCPM.COM is used for generating a MINI
; system under a MAXI the following dialog would work.
; assume drives 0, 1 are 8" and 3 is 5 1/4".
;
A>MOVCPM 60 *
```

```
GENERATING 60K CPM SYSTEM
READY FOR SYSGEN OR SAVE 37 CPM60.COM

A>                          ;reset system
MOSS MONITOR 2.2
-P2 12 0                    ;disk c: 12h sectors, 0 sides
-Q0 0 1                     ;tk 0,side 0,sector 1
-W0900 22FF                 ;write sytem from memory
-B                          ;go back to system or change
                            ;drive jumpers and do MINI boot
;
;
; BIT MAP information needed for generating new MOVCPM.COM
; when installing new BOOT program. Shown is 8080 code for
; loader and by changing value at 0989 from F4 to F8 the
; loader will work on MINI disks ( F8 is address of MINI
; sector table). Three locations are effected by bit map
; 09A0, 09A3, and 09DC. Please note that after system size
; generation the BOOT will be at address 0900. The MOVCPM
; program moves everything down 080h as it is changing
; addresses based on bit map table. To stop CPM from trying
; to load the RLOCBIOS program put 00 in location 0A07.
;
```

```
                     Listing 5:
                   BIT HEX VALUE                              BITS

        8  4  2  1  8  4  2  1  8  4  2  1  8  4  2  1
       !     A     !     B     !     C     !     D     ! AB CD

0980: 3E 01 D3 40 3A 4A 00 47 21 F4 00 4E 78 E6 40 CA  00 00
0990: 93 00 2B A8 47 3A 51 00 85 6F 7E 32 44 00 21 FF  00 00
09A0: FF 11 02 1C 78 D3 34 3E 0D D3 30 DB 34 1F D2 AB  90 00
09B0: 00 DB 30 E6 98 C2 80 00 7B D3 32 DB 33 78 F6 80  00 00
09C0: D3 34 3E 88 D3 30 23 DB 33 77 DB 34 1F D2 C6 00  00 00
09D0: 2B DB 30 E6 9C C2 80 00 7C BA D2 00 16 7B 1C B9  00 08
09E0: C2 B8 00 3A 44 00 4F 3A 4A 00 47 D3 34 3E 5D 1E  00 00
09F0: 01 C3 A9 00 1A 0F 08 04 12 0A 05 02 00 00 00 00  00 00
0A00: C3 5C 03 C3 58 03 7F 08 52 4C 4F 43 42 49 4F 53  24 00


;
; BIT map starts at location 2500h in MOVCPM with 2500 to 250F
; representing 0900 to 097Fh. The BOOT program is at 2510
; to 251F and represents 0980 to 09FF. To preserve the BIT
; map use SAVE 40, after generating a new system size
; SAVE 37 is used (bit map is no longer needed).
;

2500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2510: 00 00 00 00 90 00 00 00 00 00 00 08 00 00 00 00
2520: 24 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

THE KIBLER ELECTRONICS 8080 MONITOR IS A VERSION OF THE Z80 BASED MONITOR SUPPLIED BY CCS WITH THEIR CCS 2422 DISK CONTROLLER. THIS VERSION HAS BEEN PROGRAMMED FOR 8080 DEVICES WITH A BOOT LOADER PROGRAM IN THE MONITOR. MOST MONITOR OPTIONS STILL REMAIN AND THE MONITOR IS INTENDED TO USE SERIAL TERMINALS. CONFILICTS WITH VIO MEMORY MAPPED DISPLAYS PREVENTS ANY OTHER FORM OF TERMINALS OPERATION (2422 USES F000 TO F800 FOR PROM). AUTO BOOT FUNCTION WILL WORK AS ORIGINALLY INTENDED BUT WITH 8080 CODE, ALSO ANY VIO WILL HAVE TO BE DISABLED DURING BOOT BY HARDWIRE JUMPER USING THE CCS 2422 PHANTOM. THE BOOT FUNCTION CAN LOAD DIFFERENT SIZE SYSTEMS BY ADDING A VALUE TO THE "MEM SIZE ?" THAT REPRESENTS SYSTEM SIZE WHEN USING THE 'B' COMMAND. THROUGH THE USE OF A SERIAL TERMINAL IT SHOULD BE ABLE TO MODIFY SECTORS OF THE BOOT DISK FOR PROPER OPERATION. THIS PROM CAN BE USED AS A BIOS BUT CPM WILL GIVE A DISK ERROR IF THE RETURN (C9) IS NOT CHANGED TO A JUMP TO A DISK SELECT ROUTINE (MUST RETURN DBP TABLE ADDRESS IN H REGISTER)(SEE CPM ALTERATION GUIDE FOR INFORMATION ON DPB TABLES AND CHECK CCCBIOS FOR THEIR ROUTINES).

```
SERIAL PORTS USED:
IN/OUT      PORTS:01,02,20H,22H
STATUS      PORTS:00,03,21H,23H
DATA TEST BITS ALL PORTS: DATA IN READY 'ANI 02H'
                          DATA OUT READY TEST BIT 01
                          ALL TEST BIT SET TO ONE FOR READY
```

CONSOLE FINDER ROUTINE LOOKS FOR SPACE CHARACTER TO SET TERMINAL PORT VALUE, AND THEN USES CORRESPONDING STATUS PORT TO CHECK FOR DATA READY (I.E. PORT 02 FOR TERMINAL,PORT 03 FOR STAUS,BITS 01 AND 02 USED (NORMAL 8251 DEVICES)). ALL STATUS PORTS ARE GIVEN FOLLOWING VALUES FOR INITIALIZATION 40H,AEH,37H,. IF 8251,INITIALIZATION WILL GIVE 16X, 8 DATA BITS, NO PARITY, 2 STOP BITS, WITH DTR AND RTS ON.

```
BOOT COMMANDS: B
"MEM SIZE?"
       A          AUTO BOOT
       7          64K SYSTEM
       6          60K SYSTEM
       5          56K SYSTEM
       4          48K SYSTEM
       3          32K SYSTEM
       2          20K SYSTEM
```

AVAILABLE MONITOR COMANDS (SEE CCS 2422 MANUAL FOR PROPER OPERATION).

```
B         BOOT                       P      DISK PARM
D         DISPLAY                    Q      Q DISK PARM
F         FILL                       R      READ DISK
G         GOTO (NO BRK PNTS)         W      WRITE DISK
H         HEX NUM                    S      SUBSTITUTE MEM
I         IN FROM PORT               O      OUT TO PORT
T         MEM TEST                   V      VERIFY
```

MONITOR PROM ENTRY POINTS:

```
F000       COLD BOOT ENTRY POINT
F003       WARM BOOT ENTRY WILL JUMP TO AUTO BOOT ROUTINE
F006       CONSUL STATUS OF INPUT DEVICE
F009       CONSUL INPUT WITH VALUE IN "C"
F00C       CONSUL OUTPUT FROM "C" REGISTER
F00F       LIST DATA SENT TO CONSUL OUT
F012       PUNCH DATA SENT TO CONSUL OUT
F015       READER DATA FROM CONSUL IN
F018       WILL SET TRACK MEMORY LOCATION TO "0"
F01B       STORES DISK NUMBER AND CALLS 2CH TO SEE IF JUMP TO
           DISK PARAMENTER TABLE, SET H TO "0" (CPM ERROR)
F01E       PUTS TRACK VALUE IN MEMORY LOCATION
F021       PUTS SECTOR VALUE IN MEMORY LOCATION
```

CCS 2422 Disk Controller continued . . .

```
F024    PUTS DMA ADDRESS IN MEMORY
F027    DOES READ OPERATION BASED ON MEMORY VALUES
F02A    DOES WRITE OPERATION BASED ON MEMORY VALUES
F02D    RETURNS CONSUL STATUS
F030    TRANSLATE SECTOR TABLE BASED ON DISK SIZE

FOR USE AS CPM BIOS OR WITH F800 FROM THE FOLLOWING ROUTINES WILL
BE NEEDED:

SELDSK:    ROUTINE TO DETERMINE DISK NUMBER AND RETURN ADDRESS OF
DISK  PARAMETER  TABLE  IN "H".  CURRENT  ROUTINE  CHECKS  MEMORY
LOCATION  2CH FOR RETURN IF NOT CHANGED WILL RETURN WITH "H"  SET
TO "0" AND WILL ERROR CPM. FOR PROPER CPM OPERATION THIS LOCATION
SHOULD HAVE JUMP TO NEW DISK SELECTION ROUTINE, THAT WILL PUT THE
LOCATION OF THE DISK PARAMENTER TABLE IN "H", BEFORE RETURNING TO
CPM. CHECK CPM ALTERATION GUIDE OR LISTING OF STDBIOS FOR EXAMPLE
OF ROUTINES NEEDED.

SECTRAN:   SECTOR  TRANSLATION  TABLES BEYOUND THE  TWO  STANDARD
TABLES  INCLUDED IN THE PROM.   THE PROM CHECKS TO SEE IF MINI  OR
MAXI  AND THEN TRANSLATES ACCORDINGLY. SHOULD OTHER THAN THE  TWO
SECTOR  TABLES SHOWN BELOW BE NEEDED SOME ROUTINE MUST BE LOCATED
ELSEWHERE. NO JUMP ROUTINE IS PROVIDED FOR THIS OPERATION.

MINI:      DB    1,5,9,13,17,3,7,11,15,2,6,10,14,18,4,8,12,16
MAXI:      DB    1,7,13,19,25,5,11,17,23,3,9,15,21,2,8,14,20,
                 26,6,12,18,24,4,10,16,22
```

If your thirst for technology is combined with an interest in the IBM Personal

A N N O U N C I N G

PC TECH JOURNAL

FOR IBM PERSONAL COMPUTER USERS

To The Limits... And Beyond

As you well know, the IBM Personal Computer is proving to be the success story of the 1980s. Only 18 months after its introduction, the IBM Personal Computer commands almost a fifth of all personal computer sales—a remarkable achievement. At present the number of IBM Personal Computers in users' hands exceeds a quarter-million, with perhaps as many as half a million on the way in the next 12 months.

But despite its widespread public acceptance and the wealth of products now available to support it, we are convinced that *vast areas of its potential remain unexplored,* not to mention those products waiting to be introduced for which the IBM Personal Computer is the precursor.

For those serious enough about microcomputers to accept the challenge, the IBM Personal Computer represents a host of untapped possibilities.

For those with a genuine appetite for technology, there are ways of stretching the IBM Personal Computer to the very limits of its specifications. Ways to willfully expand those specifications...to throw light into the dark recesses of its complex software systems. Ways to enlarge our understanding of the equipment and to *increase its usefulness.*

If this sounds appealing, then *PC Tech Journal* is for you. For it will be the definitive source of authoritative, technically-satisfying information about your IBM Personal Computer, its future transformations and the host of products developed to support it.

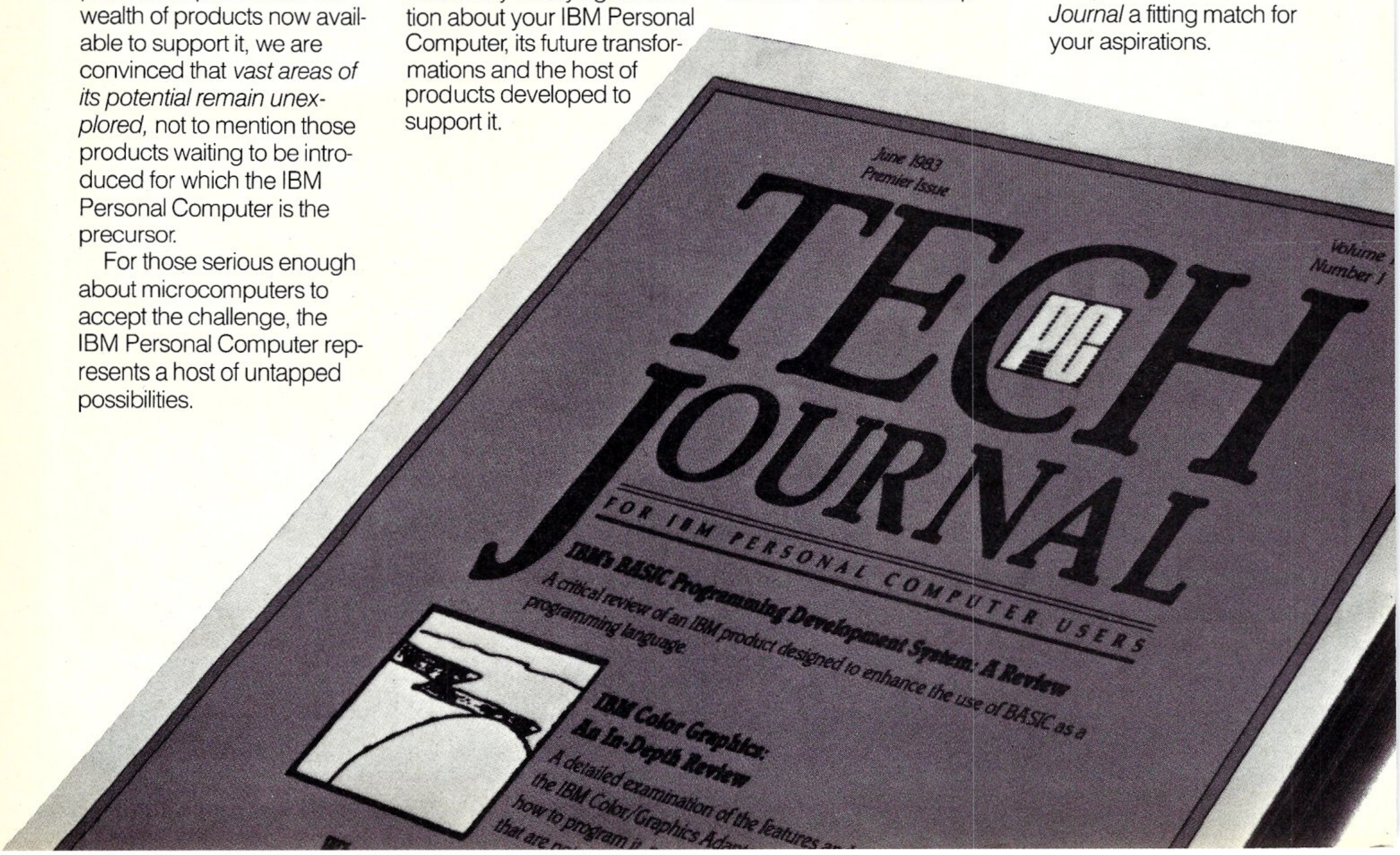
Why a Technical Journal?

The readership of *PC* Magazine has, in the last year, grown at a rate to match the success of the IBM Personal Computer itself. But as the number of committed readers has expanded, so has that portion of users who are not only interested in *output*—that is, using the IBM Personal Computer to better solve problems—but who care deeply about the *process* of the computer itself.

We consider you to be among that select group of readers. Your relationship with microcomputers, vocational or otherwise, has led you to levels of technical competence and curiosity which go far beyond problem-solving. In effect, you have a compelling interest not only in *what* your IBM Personal Computer does, but also *how* it does it and how it can do more.

For you, technological sophistication is a welcome challenge, perhaps even something of an adventure.

In both its depth and breadth of applicable sophistication, we are confident that you will find *PC Tech Journal* a fitting match for your aspirations.



Computer you will want to take part in exploring its vast technological landscape.

Enroll now as a Charter Subscriber at a special Charter rate ...and get your own copy of our Premiere Issue.

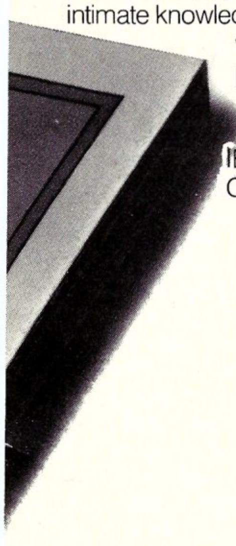
A Select Audience

Needless to say, *PC Tech Journal* is not for everyone. But among *PC* Magazine's current audience is a sizable segment where *PC Tech Journal* will find an active, committed readership:

Practicing professional developers who will want to read the *Tech Journal* for information they can't find anywhere else.

Serious programmers—both full-time professionals and non-professionals alike—with a clear interest in expanding the range and utility of software compatible with the IBM Personal Computer.

Management and staff serving in MIS and DP roles with a requirement for intimate knowledge of the working processes of their companies' IBM Personal Computers.



Independent businessmen and professionals who understand the increased benefits of tailoring their IBM Personal Computers to their own specific demands.

Members of the scientific and academic community whose ability and training allow them to enhance their use of their IBM Personal Computer.

And, of course, anyone with a genuine appetite for state-of-the-art information about the emerging possibilities presented by the IBM Personal Computer.

A Menu for Technical Gourmets

Under the leadership of Editor-in-Chief Will Fastie, author of the widely respected "IBM Images" column in Creative Computing Magazine, we've assembled a staff and a stable of regular contributors all with just the sort of demonstrated expertise you'd expect from *PC Tech Journal.*

Every bimonthly issue will be filled with useful articles on subjects such as...

- *How to fully exploit—or, if necessary, avoid—the complexities of BIOS, the IBM Personal Computer's installed firmware.*
- *Detailed examinations of component hardware available for the IBM Personal Computer, including multi-function cards, display devices, laboratory data acquisition equipment and voice recognition and synthesis devices.*
- *Elegant programming methodologies which exploit the IBM Personal Computer's various operating system environments, plus reviews of new operating environments.*
- *Evaluation of the IBM Personal Computer's communications protocols and standards, including a detailed discussion of computer-to-computer interconnection, terminal emulation and distributed data processing.*
- *Plus a range of articles on networks, mass storage and important user development tools.*

Moreover, as *PC Tech Journal* evolves, it is our explicit intent that you, our readers, play an important part in shaping the magazine. For we are convinced that it will be you who shape the future usefulness of the IBM Personal Computer itself.

To that end, we will make a special effort to involve you in the focus of the magazine, listening to your needs and together forging workable technical solutions to fully take advantage of IBM's revolutionary Personal Computer.

Charter Subscription Offer

Just fill in and mail the coupon or the attached card. We'll enroll you as a Charter Subscriber, send your Premiere Issue and bill you at the special introductory price—a discount of up to 33% (based on the full one-year subscription price.)

PC TECH JOURNAL
P.O. Box 598 Morris Plains, New Jersey 07950

8H120

☐ Yes. Please enroll me as a Charter Subscriber to *PC Tech Journal.*

☐ One year (6 issues) only $19.97—20% off!
☐ Two years only $36.97—26% off!
☐ Three years only $49.97—33% off!

Savings based on full one-year subscription price of $24.97.
CHECK ONE: ☐ Payment enclosed. ☐ Bill me later.

Mr./Mrs./Ms. ______________________
(please print full name)

Company ______________________

Address ______________________

City ______________________

State ______________________ Zip ________

A MICROSYSTEMS TUTORIAL

Relocating Assemblers and Linkage Editors: Part 2

by Andrew L. Bender

Part one of this article discussed the basic concept of relocation, how binary relocatable code is generated, and the requirement that all programs be assembled relative to memory location zero. This part of the article will discuss how programs and subprograms communicate with each other, and how areas of memory can be designated as being in common with each other.

Once a programming problem has been defined and broken up into functional components, the relocating assembler allows for these pieces to be developed independently of each other by different programmers. There must, of course, be a set of rules defining the common data structures and the protocol for transferring information between the various parts of the program. When all of the components of the program have been developed, the linkage editor is used to collect these pieces and to link them together.

Because the relocatable binary output of the assembler contains not only the machine instructions actually coded by the programmer, but also information as to the relocatability of certain addresses and even the names of certain types of symbols, such a module cannot be directly executed. Even when using the CP/M assembler ASM, one cannot directly load the resulting ".HEX" file into memory. It must be formatted into a ".COM" file first. The same is true in the case of a relocatable program assembled by M80 or RMAC. The linkage editor generates a ".COM" file from all of the relocatables collected under commands from the programmer.

Relocatable programs and subprograms, being assembled independently of each other, have no means of knowing their final locations in memory. Thus, if a programmer wishes to call a subroutine that is outside of his own module, he must declare its name as "external."

During program assembly, the assembler notes the names of all external symbols, and they are placed in the relocatable binary output module. Depending on the implementation, there are various ways in which these names may be associated with specific memory locations, each of which will eventually be filled with the actual address represented by an external symbol. The task of filling in the proper address is given to the linkage editor because, as we will see, the absolute address cannot be known until the linkage editor has loaded all of the modules into memory in the desired order and computed the length of each one. The order in which the relocatable modules are linked depends on the order in which the operator specifies files in the command invoking the linkage process.

During preparation of a program, the programmer might desire that one module be referenced by a certain symbolic name which would be known to all other modules being collected by the linkage editor. This name is declared as an "entry" point or "public" symbol, and may be referenced in another module as an external symbol. An entry point may be defined only once during linkage editing, just as a label is allowed only one definition during a given assembly. While all of the other symbols in an assembly are discarded at the conclusion of the assembly, the entry points and external symbols are retained in the relocatable binary output module. The linkage editor determines the address of each entry point as the program is being collected and stores the address in a table of entry points. In addition, each time an entry point is defined, the linkage editor makes a search of all of its external symbols, for which there is no current definition. If the entry point matches one of those external symbols, the external symbol references are replaced by the address of the entry point, and the matching entry in the external symbol table is marked as defined. This is the basic process performed by the linkage editor. Let's look at a program prepared to use several external symbols that also has one entry point (Listing 1).

Note that each reference to an external symbol in the listing of the assembly contains the value zero followed by an asterisk. The asterisk is the flag printed by this assembler to indicate that the value is external. The symbol table contains an address that corresponds to the location of the first appearance of the external symbol in the assembly for each external symbol. This is only for the assembler that we are using to demonstrate this facility. Other assemblers may use other designations. The entry point is flagged by the letter "I" and the relative address is printed in the symbol table. There is no practical limit to the number of entry points or external symbols that may appear in any given program. Usually the main program has no entry point, and the symbol on the END statement receives control after the absolute file is loaded. On the other hand, the main program may contain many entry points if desired.

The usual means of identifying the main program in a collection of relocatable programs is that it will be the one program assembled with a "named" END statement. A "named" END statement is an END statement that contains a label as its operand. By convention only the main program may contain such a named END statement. Under normal circumstances, the linkage editor compiles a jump instruction to the label specified on the named END statement, storing this jump instruction at 0100H. When the absolute loader in CP/M turns control over to 0100H, the jump instruction there will turn over to the proper location in the program. Many linkage editors will accept a directive from the programmer to turn control over to any entry point in any of the loaded modules, or even to an absolute memory address. In the case of this directive being given, the default address of the named END statement is superceded by the programmer-supplied address. Specification of an actual memory address is not a good idea because it puts constraints on the placement of programs.

Andrew L. Bender, M.D., Neurological Services, 336 Center Ave., Westwood, NJ 07675

When the linkage editor encounters a named END statement in the binary stream it is reading, it sets the address of the symbol aside so that it will have it handy when the collection and linkage of programs has concluded.

Let us follow the linkage process for two relocatable binary modules. The main program is contained in a file "MAINPROG.REL"; the file "SUBROUT.REL" contains a single module with two entry points needed as subroutines in the main program. Directions to the linkage editor as to how to find these programs, which programs to load, and loading options, are described in the third installment of this series.

The order in which these two routines are loaded into memory by the linkage editor is not important. Take the case in which the subroutines are loaded first.

Since programs running under CP/M start at 0100H, the relocatable binary code will be loaded into memory, and each CSEG flagged address will have 0100H added to it (if the linkage editor places a jump instruction at 0100H as described above, each CSEG flagged address will have 0103H added to it). Each DSEG flagged address will have the overall length of the CSEG section plus 0100H (or 103H) added to it. The entry point names of each routine will be recorded in the entry point table together with their relocated addresses. When the end of module indicator is read in the relocatable binary output module, the linkage editor tries to continue to read that file.

If there is another program in the file, that program will be loaded and processed, and so on until an end-of-file indicator is read. In our example there is no other program, so the file is closed and the next file (the main program) is read. Here all of the CSEG addresses will be biased by the length of the subroutine module read previously. This would be the sum of the initial program bias (0100H or 0103H), the length of the CSEG section and the length of the DSEG section. These values are added to each CSEG flagged address and the DSEG flagged addresses are assigned to follow the CSEG section. To calculate the bias for the DSEG, the linkage editor uses the initial base value of this module and adds the length of the CSEG to it.

As the external symbol names are encountered in the relocatable module, the table of entry points is consulted and the appropriate relocated addresses are substituted in the proper place in the program being loaded. When the end-of-module indicator is encountered, there is an associated address, generated by the END statement, which is appropriately relocated also. This address is recorded. Now there is no more input, so relocation is complete.

If desired, the absolute module resulting from the collection of these two programs can be executed and/or written out as a .COM file by instructing the linkage editor to do so. Any external symbols still undefined after the collection may be assumed to be in a "library." External symbols which remain undefined after collection of library routines are known as "unresolved external references," and they are usually indicative of an error, due either to the omission of a needed subroutine or to the misspelling of a subroutine entry point or an external symbol. Depending on the implementation of the linkage editor, there are specific rules about the design and searching of these libraries.

We haven't considered some important things that we cannot ignore in a practical discussion of just how all of this happens. First, we are bound by the rules for writing the source program that are imposed by the assembler we use. Second, we are constrained as to what kind of arithmetic and symbol manipulation the linkage editor can do. Can an external symbol reference be 8 bits wide, or must it be 16 bits wide? Can the linkage editor add or subtract a bias to an external symbol? Can it add or subtract two external symbols? How many characters can an external symbol and entry point contain? These are very important things which you will have to know about your development software. To find them out, you must read the manuals carefully.

Experiments sometimes provide more knowledge than the manuals. Your linkage editor or assembler may not permit any address arithmetic on external symbols. You will be safe if you assume that only 16-bit arithmetic can be done on entry point and external symbol locations. You'll probably be wrong if you assume that the linkage editor can do arithmetic on external symbols, such as adding or subtracting them. Even very large mainframe software doesn't usually permit this type of arithmetic. Some linkage editors can add or subtract numerical constants to or from external sysmbols. These features are usually quite dependent on the effort put into writing the linkage editor and assembler software. Many features would require too much analysis of the relocatable code and perhaps more "passes" over the relocatable modules. This is time-consuming, and contributes to slowing the overall program development cycle.

Because the linkage editor usually must reside in memory with the program it is linking, available memory is reduced by the amount of space which the linkage editor occupies during linkage. Some linkage editors can only link programs that are as big as the memory remaining after the linkage editor software has been loaded. Certain linkage editors allow for two passes over the relocatable programs to effect linkage, and therefore allow programs which are larger than the available memory space to be linked. These two passes cause a moderate reduction in the speed of linkage but are sometimes necessary in cases where large programs are being constructed. Some linkage editors indicate how much memory space was used during linkage editing as a percentage of the total space available.

The CSEG and DSEG directives allow for the segregation of code and data into separate areas of memory within a module. There is yet another segregation technique called COMMON banking or COMMON blocks. To establish a common block storage segment, the programmer gives a symbolic name to an area of storage and declares it

Relocatable programs, assembled independently of each other, have no means of knowing their final locations in memory. If a programmer wishes to call a subroutine outside his own module, he must declare its name as "external."

PROFIT MAKER

Microsystems not only sells itself at a nice profit, it creates awareness of the hardware and software in your store. It's like having another salesperson on your staff—one who *pays you*. One who goes home with customers and continues to influence buying decisions long after you're closed for the day. If you're interested in a low-ticket item with top-of-the-line sales power, let's talk now.

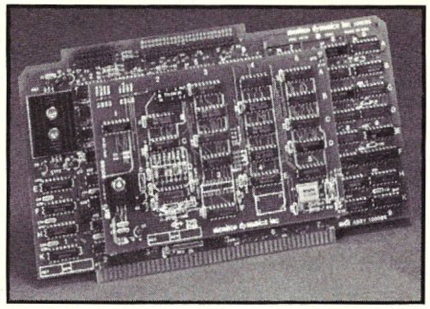
CALL COLLECT: (212) 725-7679

Or write: Ziff-Davis Publishing Company

ZD One Park Avenue
New York, NY 10016

Minimum order, 10 copies.
We pay all shipping costs to your store.

WINCHESTER CONTROLLERS



- S100 / IEEE 696, Single Board or Z-80 Intercept board
- Supports 5¼", or 8" Fixed or Removable Cartridge drives
- Supports Two Drives — mix or match
- Drivers for TurboDOS, CP/M 2.2, MP/M II
- Single Quantity Price — **$495**
- Currently Installed in Teletek, ICM, CCS, SD Systems, IMS Dynabyte, JADE, KAYPRO, Z-100, NORTH STAR, DELTA
- Intelligent controller, supports high level commands

MONITOR DYNAMICS SINGLE & DUAL SUBSYSTEMS

INCLUDES: CONTROLLER, DRIVE(S), CABINET & CABLES
(Assembled, Formatted, Tested, and with Driver)

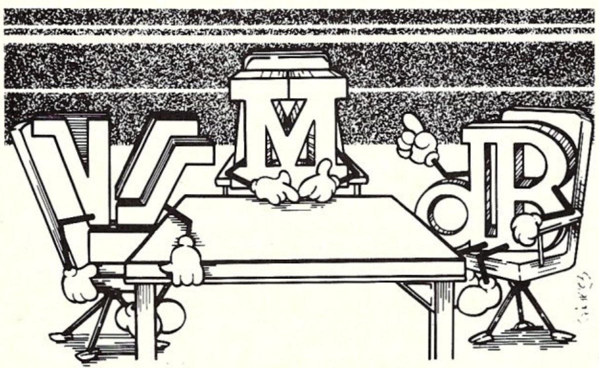
	SINGLE	DUAL
13 Megabyte Seagate Subsystem	$1495	$2195
27 Megabyte Rodime Subsystem	$2095	$3495

VISA
MASTERCARD

LIMITED OFFERING
SEAGATE ST-412 13 Megabytes $750
Dealers & Distributors Invited

mdi 1121 W. 9TH • UPLAND, CA 91786 • (714) 985-7214

CIRCLE 74 ON READER SERVICE CARD



"Microstat talks to dBase II, Wordstar, etc. . ."

Microstat® Rel. 3.0 can now access external ASCII data files for statistical analysis. Virtually any comma-deliniated ASCII file may be used.

Release 3.0 requires Micro Mikes baZic, a Z80 CPU and CP/M. The price is $325.00 or the manual may be purchased for $25.00 (credited towards purchase). If you need baZic, add $100.00. For further information, call or write:

VISA — ECOSOFT INC. — MasterCard

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

Trademarks: Digital Research (CP/M), Aston-Tate (dBase II), Micro Mike (baZic) and Micropro (Wordstar)

CIRCLE 154 ON READER SERVICE CARD

CP/M® Software

A>**DBPACK:** Data base management; indexing, sort/search, tabulation, address labels ...
A>**DBPACK-II:** Advanced data management; payroll, inventory, large & complex data bases ...
A>**COMCOM:** Communication program; powerful, yet easy to use.
A>**CPMCPM:** Transfer files (any type) between CP/M computers.
A>**FILER:** Compresses, archives, catalogs & organizes files.
A>**UNERA:** Recovers erased files.
A>**MULTED:** Multi-file text editor.

CP/M is a registered trademark of Digital Research

Configured for a wide variety of systems.
Disk formats include 8-inch, Osborne, Xerox ...

Call or write for information

COMPU-DRAW
1227 Goler House
Rochester, NY 14620
Phone: (716)-454-3188

Dealer inquiries invited

MasterCard, Visa & American Express cards welcome.
Separately ordered documentation may be returned for full refund within 10 days!

It's the writing on the wall

CIRCLE 207 ON READER SERVICE CARD

as common. A common block can be referenced by its name in any program linked with the program containing the common block. The same name can be defined by several different programs, though there are some restrictions. First, if common blocks with the same name have different lengths in programs to be linked together, then the linkage editor must encounter the *longest* definition *first*. Second, common blocks are sensitive to spelling; unlike external symbols, no error will be indicated if the name of a common block is misspelled in one of the modules—it will merely be treated as a separate block. This can lead to errors that are difficult to diagnose. Thus no warning is given if a common block name appears in only one program; this makes spelling errors very hard to find.

The linkage editor assigns the same storage area to all common blocks with the same name. This is the reason for the first restriction; it is not possible to move around the memory image of a partially linked program to accommodate a change in common block length. Only the name of the common block is retained—not the names of the symbols (unless they happen to be entry points or external symbols) defined within the common block. This feature makes it possible for different people to work on different subroutines referencing the same data in the same order, though perhaps using different names for areas within the common block. There is no means of specifying the length, position, or type of data item within a common block. The bookkeeping attached to these attributes is completely the responsibility of the programmer. An example consisting of two subroutines referencing the same common block is given in Listing 2.

Notice that the common block name is "LABEL". The convention in this assembler is that the name of the common block is set off by a pair of virgules (slash marks). If no name appears, then the name of the common block is the default name called "blank common." In some linkage editors, blank common is loaded in a different place in memory than in other linkage editors, hence it has a special meaning to these linkage editors. Some linkage editors load blank common at the highest possible location of memory, while others may place it beginning at 0103H. Other linkage editors may not treat blank common differently than any other common block. Common block references are flagged by an exclamation point in the program listing of this assembler to differentiate them from other relocation bases. In the symbol table of module, TINY, the common block name "LABEL" appears with the value 3 followed by the letter "C." This indicates that "LABEL" is a common block name with a block length of three. Now, look at the listing of the second subroutine, given in Listing 3.

In Listing 3, the first subroutine contained the same common block name as the second subroutine. However, in the first subroutine only the first three memory locations of the common block are defined; in the second subroutine, 16 memory locations are defined. The linkage editor would require that the second subroutine be loaded before the first subroutine so that the proper amount of common block memory could be allocated. The first three memory locations in the second example will be assigned to the SAME three memory locations as the first three memory locations in the first example, without regard to the fact that the symbolic names for these locations are different in each subroutine. Thus SS, KK, and SDT are the same as locations AA, BB, and CC, respectively.

Common blocks provide a means to transfer information from one relocatable subprogram to another in a convenient manner. A completely general subroutine usually takes and returns its parametric information from the registers of the computer or from a data structure in memory; an example would be the CP/M subroutines accessed by calling BDOS. Subroutines written with common blocks, on the other hand, usually take and return most if not all of the information they require from one or more of these common blocks.

In certain cases it is necessary to assign certain areas of memory to fixed locations in the address space of the target computer (the computer that is going to run the final program). An example of a situation in which a fixed memory assignment is necessary is a computer system in which there is a memory-mapped video screen. There are techniques that can be used to force some linkage editors to assign fixed memory locations to areas with certain attributes such as common blocks or a data region. In a few cases these techniques may not be available, or they may be cumbersome.

An assembler directive called ASEG may be used to dictate absolute placement of code or data within an otherwise relocatable program. While the symbols defined within the scope of an ASEG directive are not relocatable symbols, the programmer is free to use any relocatable symbol or external symbol within an ASEG. An ASEG may contain entry points. The positioning of code or data within the ASEG is by means of the ORG directive. In all cases the ASEG begins at absolute memory location zero, but this origin can be changed by use of the ORG directive. The example shown in Listing 4 helps in understanding an ASEG.

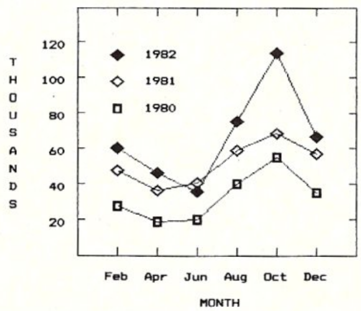
The sample program shows the use of the ASEG, CSEG, and DSEG, directives. The program listing indicates an ASEG by the absence of any relocation flags next to the items which refer to the ASEG segment. The ASEG contains a jump table at 0EA00H and a data region at 0BA00H. Note that the ASEG can be manipulated by that portion of the program which is relocatable and that it can contain symbols defined outside of the ASEG. There is but one rule for using an ASEG: An ASEG cannot overlay the BDOS, the linkage editor, the tables created by the linkage editor, or the first page of memory (00 to 0FFH). If this happens, the linkage editor may not be able to complete

In any module, a symbolic name that is to be accessible to all other modules is declared as an "entry point" or "public" symbol, which may be referenced in another module as an "external" symbol.

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 15 ON READER SERVICE CARD

BRIDGE GRAPHICS

PLOTPAK is a complete plotting library that runs under FORTRAN-80 and performs a variety of functions: ***windowing, linear print arrays, automatic polygon drawing, annotations, plotting symbol/line selection, labeling, coordinate conversions.***

PLOTPAK can drive a screen and plotter simultaneously while maintaining a window which allows you to communicate back to your FORTRAN program. PLOTPAK includes your choice of the following drivers:

SCREENS

- MicroAngelo MA 512, 520
- ADM + Retrographics
- TEK 4010 Compatible Terminals
- Otrona ATTACHE Portable Computer

PLOTTERS

- H. P. Plotters 7225B & 7470
- Houston Instruments DMP-4

PLOTPAK (.REL file) + two drivers.......... **$190**
PLOTAK (source code) + two drivers........ **$240**

BRIDGE
Computer Company
DIVISION OF SEA DATA CORPORATION
ONE BRIDGE ST., NEWTON, MA 02158
PHONE (617) 244-8190

MAGNOLIA MICROSYSTEMS

Makes the ZENITH 89/90 a REAL Business Computer

The solid **Dependability** of the Z89/90 with the **Flexibility** to configure it the way **you** need.
- Memory Boards 16-128 KBytes
- Hard Disk Interfaces and Subsystems 5-50 MBytes
- Floppy Disk Controller Mix 8" and 5" drives
- MMS-Net™ Networking
- High-Speed/High-Capacity Auxiliary Processor
- The most powerful, flexible CP/M™ available

Ask your Zenith dealer about Magnolia products

MAGNOLIA MICROSYSTEMS, INC.
2264 15th Ave. West • Seattle, WA 98119
(206) 285 7266 (800) 426 2841

CP/M is a trademark of Digital Research

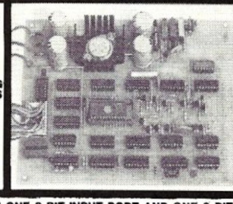
CIRCLE 244 ON READER SERVICE CARD

BURN EPROMS USING CP/M*

EPROM PROGRAMMING SYSTEM RUNS UNDER CP/M

PROGRAMS: 2708, 2758, 2716, 2732, 2732A, 2764 and 27128.

– FEATURES –
- STAND ALONE BOARD
- ELECTRONIC SWITCHING OF EPROM TYPES
- ON BOARD SUPPLIES
- 28 PAGE MANUAL
- DESIGNED WITH EASY-TO-GET PARTS
- WORKS WITH ANY CP/M SYSTEM
- USES 24 VOLT XFMR FOR POWER



– INTERFACES THROUGH ONE 8 BIT INPUT PORT AND ONE 8 BIT OUTPUT PORT. 16 WIRES – NO SPECIAL HANDSHAKE LINES.

– CONTROL SOFTWARE COMMANDS –
- PROGRAM EPROM(S) FROM DISK FILE
- PROGRAM EPROM(S) FROM RAM
- READ DISK FILE INTO RAM
- COMPARE EPROM W/RAM
- VERIFY EPROM IS ERASED
- COPY EPROM
- DISPLAY/MODIFY RAM (MONITOR) (MONITOR HAS 11 SUB COMMANDS FILL-DUMP-XFER-MODIFY ETC.)

ALSO AVAILABLE FOR OSBORNE 1 (TM)

BARE P.C. BOARD WITH COMPLETE DOCUMENTATION AND SOFTWARE ON AN 8" SINGLE DENSITY DISKETTE **$69.**

TO ORDER SEND CHECK, MONEY ORDER OR CALL

WRITE OR CALL FOR MORE INFO. 513/752-7218

ADD $3.00 SHIPPING
Ohio res. add 5.5% tax
$3.00 for C.O.D.

VISA OR M.C. ACCEPTED

AndraTech
P.O. BOX 222
MILFORD, OHIO 45150

* CP/M is a trademark of Digital Research

CIRCLE 199 ON READER SERVICE CARD

FREE CATALOG!

Just let us know and we'll mail you a FREE *Creative Computing* Catalog—16 pages filled with books, buyer's guides, magazines, and more!

To get your FREE catalog, write to: *Creative Computing* Catalog, Dept NA1X 39 East Hanover Ave., Morris Plains, NJ 07950.

Function Keys—I/O Controller
12K Print Buffer—Clock/Calendar
For $525.00
THE INTERPRETER

A SINGLE BOARD COMPUTER programmed for use as a MENU (programmable function keys) and as a versatile I/O buffer-controller and clock. Other controlling programs downloaded via MONITOR. Used in series with any peripheral (serial protocol, parallel option). Easily labeled keys (via overlay), eliminate the need to memorize control codes or awkward strings. Multistroke commands and repetitive tasks become single stroke functions.

BIG TIME SAVERS
Word processing
Data base management
Non technical users
Supports any software

STANDARD $399
DELUXE $525

DELUXE UNIT FEATURES
> All std. features
> 8k Keyset Space
> 12K PRINT BUFFER
> Parallel Port
> Clock/Calendar (Batt. backed)

STD UNIT FEATURES
> Over 300 Keys (4 x 75)
> Units are cascadeable
> Type Ahead Buffer
> 2K Keyset (Bat Back)
> Blank Overlays
> User's Manual
> Up/Down Load Keysets

OPTIONS
> Engineering Manual
> Modem
> Extension Keyboard
> Bus Interface
> Custom Overlays
> Custom Programs
> OEM designs

MOUNTAIN MICRO SYSTEMS INC.
2161 Bowen St; Longmont, Co. 80501 (303)776-2505

CIRCLE 32 ON READER SERVICE CARD

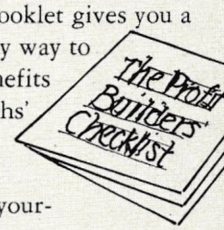
Applications Software Developers

How Whitesmiths' better system software can help you build a better bottom line.

The portability of Whitesmiths' C Compilers and Idris operating system enables you to offer your applications to a wide range of end-users without reworking your code. Just one development effort works on many architectures, because Whitesmiths' C Compilers span over 30 systems, and Idris, the most complete and portable UNIX-like system for micros, is UNIX-compatible in both directions. You can also lower costs and simplify distribution by eliminating end-users licenses using Whitesmiths' new "licensing under copyright" Authorization Seal.

SEND FOR OUR FREE BOOKLET
"THE PROFIT-BUILDERS' CHECKLIST"

Our new booklet gives you a quick and easy way to check the benefits of Whitesmiths' products for your operation. See for yourself how we can cut your costs, widen your markets, and build your bottom line.



Idris is a trademark of Whitesmiths, Ltd.; UNIX is a trademark of Bell Laboratories.

Whitesmiths, Ltd.
97 Lowell Road
Concord, Mass. 01742
(617) 369-8499

NAME ______
TITLE ______
COMPANY ______
ADDRESS ______
CITY ______
STATE ______ ZIP ______
TELEPHONE ______

CIRCLE 88 ON READER SERVICE CARD

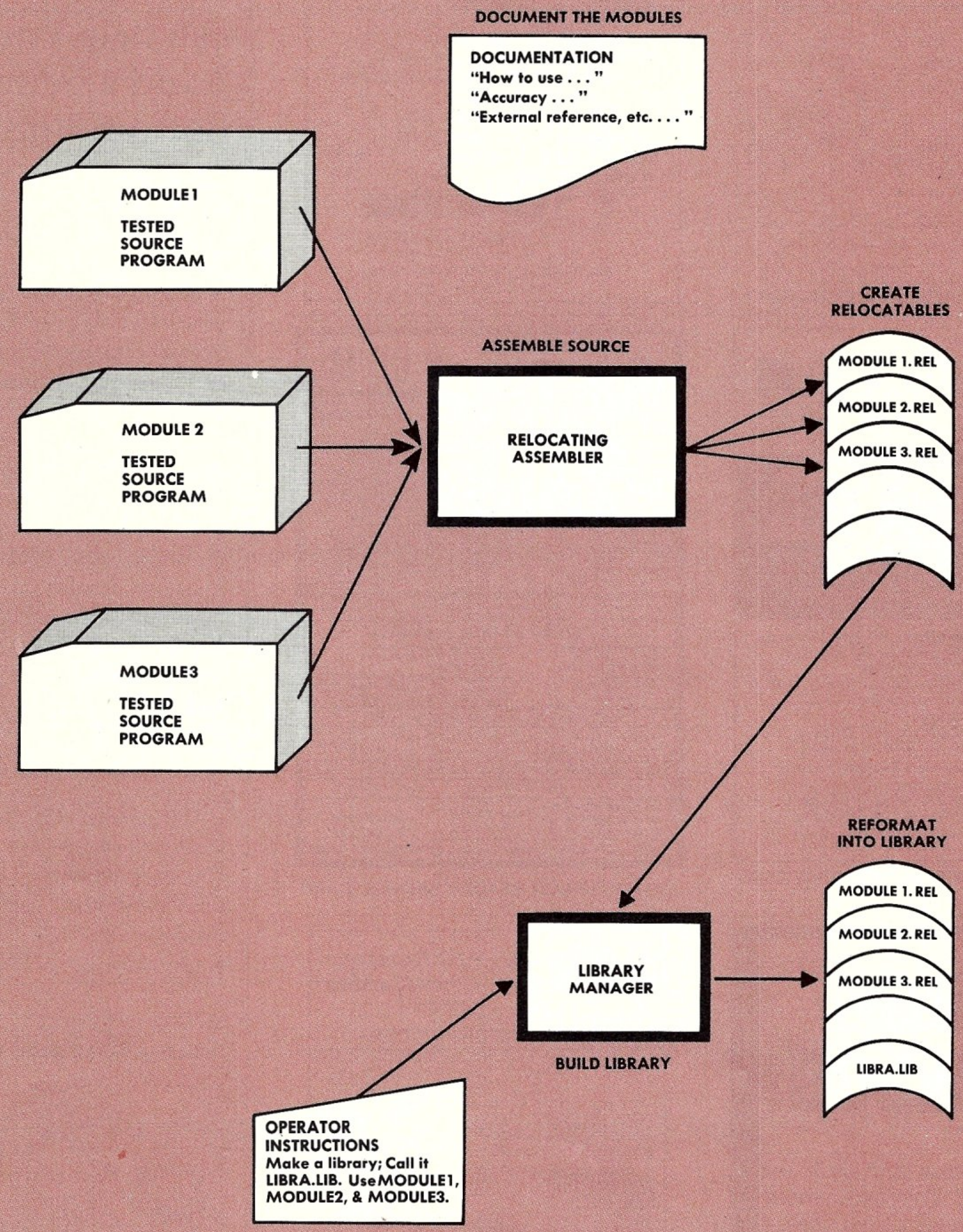


Figure 1. Creating a relocatable library.

the linking and collection of programs.

Sometimes you really don't want an ASEG. What you want is to assemble a program to eventually reside at an absolute address inside a program that is relocatable. The main reasons for doing this are in preparing programs to be burned into PROM or programs to be moved by software to another part of the memory. Let us look at the task of preparing a program that is to reside in PROM.

You have just written a super monitor that you want to test, then burn, into PROM. You spend six weeks testing it and correcting all of the known bugs. After 10 weeks it's ready to burn into PROM, but your PROMblaster requires that the program to be burned must reside someplace in memory where the PROMblaster can find it. If you assemble the program to reside at the proper address, it will be in ROM memory, so that won't work because when the linkage editor tries to load it there it won't load. If you relocate the program to other addresses, then all the addresses inside the monitor program will be wrong because the program was relocated away from the memory area it belongs in. This can be a real problem unless your assembler allows you to assemble a program as if it resides at one address but actually assembles it at another memory address. These directives are called "PHASE" and "DEPHASE". The example in Listing 5 shows the monitor assembled with PHASE and DEPHASE instructions with a program for blasting the monitor into PROM.

Notice that the monitor has been assembled at 0F00H, but it resides entirely inside the program which calls the burn subroutine. All you need to do is call your burn subroutine which, naturally, you wrote as a relocatable subroutine, give it the addresses of where to find the stuff to burn, put your PROM in the programmer, turn the programmer on and execute this program.

The PHASE directive operand is the address at which you want the program to look as if it is being assembled. DEPHASE turns off the action of PHASE. This technique can be extended to many other useful areas of systems programming.

Compare the addresses in the ASEG program with the addresses in the .PHASE/.DEPHASE program. Notice that in the ASEG program the addresses in the ASEG are the actual addresses set by the ORG statement. This is not the case in the .PHASE/.DEPHASE program. Within the code delimited by the .PHASE and .DEPHASE statements, the *addresses* of the instructions have been relocated to the address of the .PHASE statements, *but* the entire portion of the .PHASE/.DEPHASE program resides in the relocatable shell program surrounding it just as if it were data assembled by a "DB" statement. You should verify to your own satisfaction that inside the .PHASE/.DEPHASE section the instructions are actually assembled as if they do reside the following the starting address specified on the .PHASE statement.

Listing 1

```
                          ENTRY   WHURP    ;PUBLIC DEFINITION
                          EXTRN   MUNCH,HUNCH,LUNCH
0000'   21 0000   WHURP:  LXI     H,0      ;ENTRY POINT
0003'   39                DAD     SP
0004'   22 0000"          SHLD    HOLD
0007'   CD 0000*          CALL    LUNCH    ;AN EXTERNAL SUBROO
000A'   32 0000*          STA     HUNCH    ;AN EXTERNAL MEM ADDR
000D'   2A 0000"          LHLD    HOLD
0010'   11 0006           LXI     D,6
0013'   19                DAD     D
0014'   CD 0000*          CALL    MUNCH    ;ANOTHER EXTERNAL SR
0017'   C9                RET
0018'                     DSEG
0000"   0000      HOLD:   DW      0
                          END

Symbols:
HOLD     0000"    HUNCH    000B*    LUNCH    0008*    MUNCH    0015*
WHURP    0000I'
```

Listing 2

```
                  ;COMMON BLOCK SAMPLE #1 - SMALL COMMON BLOCK
                          ENTRY   TINY
0000'                     CSEG
0000'   3A 0001!  TINY:   LDA     BB
0003'   21 0002!          LXI     H,CC
0006'   86                ADD     M
0007'   32 0000!          STA     AA
000A'   C9                RET
                          COMMON  /LABEL/
0000!             AA:     DS      1
0001!             BB:     DS      1
0002!             CC:     DS      1
                          END

Symbols:
AA       0000!    BB       0001!    CC       0002!    LABEL    0003C
TINY     0000I'
```

Listing 3

```
                  ;SAMPLE PROGRAM #2 BIG COMMON BLOCK
                          ENTRY   BIG
0000'                     CSEG
0000'   3A 0001!  BIG:    LDA     KK
0003'   21 0003!          LXI     H,GLORP
0006'   96                SUB     M
0007'   32 000E!          STA     JJ
000A'   11 0008!          LXI     D,WHURP
000D'   1A                LDAX    D
000E'   86                ADD     M
000F'   32 0007!          STA     PV
0012'   C9                RET
                          COMMON  /LABEL/
```

ARE YOU DROWNING IN A SEA OF SOLUTIONS?

Recognize the symptoms?

Applications backlog? Technology blur? Mounting user pressures?

You suspect that suppliers may have answers to your pressing problems, but you don't have enough time to sort through fragmented information to determine what will really work for you.

Like the shoemaker's kid who goes barefoot, you (of all people) need a system. An information system that displays all your current software, hardware and communications options for instant reference.

That system is DATA SOURCES, and you're invited to make it work for you. Simply fill in the coupon and we'll send you DATA SOURCES at 50% off its regular cover price.

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

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

READ WHAT USERS SAY...

"*...Current and comprehensive listings are becoming indispensable.*"

Gary Yost
Marketing Services Director
ASK Computer Systems, Inc.

"*Literally use it daily...dramatically shortens research time...provides more alternatives than would have been known or considered.*"

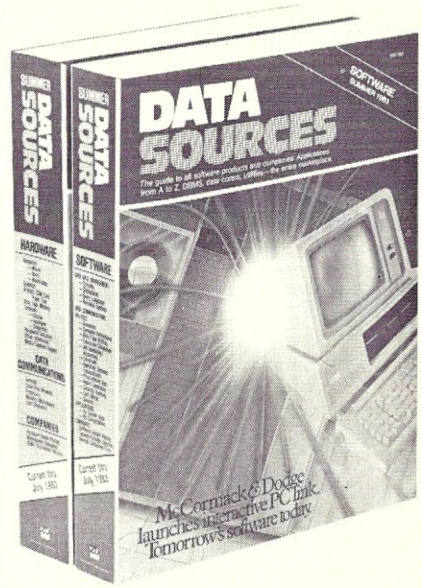
Curt Flatness, V.P.
Logic, Inc.

"*Answer to a Systems Integrator's prayer...very effective for locating the equipment we need ...using it for everything from single-user systems to systems interfacing with mainframes.*"

Richard Fletcher, President
Micro Computer Adaptation Procedure

"*Dependable source of information otherwise unobtainable or costly to obtain...*"

Barry Kukes, President
RSI Repair Service, Inc.



NEW FEATURES...

- Quick-reference product comparison charts.
- Market-tracking trends of new DP products.
- Before-you-buy selection criteria and checklist.
- Geographical listing of vendors offering on-site and depot maintenance.

YOU GET...

Hardware—Over 12,000 products from Micros to Mainframes including peripherals—terminals, printers, memory devices, etc...organized by systems compatibility.

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

Communications Equipment—from modems and multiplexors to local networks, carriers and services. Quick reference comparison charts, plus all diagnostic and test equipment.

Services, Suppliers, Support—Most complete organized listing of suppliers from installation design to maintenance and data center operations.

ORDER NOW AND SAVE 50%

Yes! Rush me the latest two-volume edition of DATA SOURCES and enter my subscription at $60.00 for 4 quarterly editions. I save a full 50% off the regular $120.00 cover price.

Name ______
Title ______
Company ______
Address ______
City ______ State ______ Zip ______
Telephone ______
Nature of Company's Business ______

Please check appropriate box: ☐ END USER ☐ VAR/OEM ☐ DISTRIBUTOR ☐ DP MFG ☐ OTHER ______

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

Return coupon to:

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

SAVE TIME...CALL DIRECT
1-800-227-1617 Ext. 251
(In Ca., 1-800-772-3545 Ext. 251)

Residents of Ca., Co., Ct., D.C., Fl., Il., Ma., Mi., Mo., N.J., N.Y., Vt., please add applicable state taxes. T118

CIRCLE 21 ON READER SERVICE CARD

```
0000!               SS:     DS      1
0001!               KK:     DS      1
0002!               SDT:    DS      1
0003!               GLORP:  DS      4
0007!               PV:     DS      1
0008!               WHURP:  DS      6
000E!               JJ:     DS      2
                            END
Symbols:
BIG     0000I'  GLORP   0003!   JJ      000E!   KK      0001!
LABEL   0010C   PV      0007!   SDT     0002!   SS      0000!
WHURP   0008!
```

Listing 4

```
                        ;SAMPLE ASEG PROGRAM
                                EXTRN   FUN,GAMES,REFRSH,PRINT
                                EXTRN   MAIN,INIT
0000'                           CSEG
0000'   31 0028"        START:  LXI     SP,STACK
0003'   21 001F'                LXI     H,HI
0006'   CD 0000*                CALL    PRINT
0009'   AF                      XRA     A
000A'   01 0200                 LXI     B,512
000D'   21 BA00                 LXI     H,ABSMEM
0010'   36 00           CLRABS: MVI     M,0
0012'   23                      INX     H
0013'   0B                      DCX     B
0014'   78                      MOV     A,B
0015'   B1                      ORA     C
0016'   C2 0010'                JNZ     CLRABS
0019'   CD 0000*                CALL    INIT
001C'   C3 0000*                JMP     MAIN
001F'                           ASEG
                                ORG     0BA00H
BA00                    ABSMEM: DS      512
                                ORG     0EA00H
EA00    C3 0000*                JMP     FUN
EA03    C3 0000*                JMP     GAMES
EA06    C3 0000*                JMP     REFRSH
EA09                            CSEG
001F'   0D 0A 48 49     HI:     DB      0DH,0AH,'HI THERE$'
0023'   20 54 48 45
0027'   52 45 24
002A'                           DSEG
0000"                           DS      40
0028"                   STACK   EQU     $
                                END     START
Symbols:
ABSMEM  BA00    CLRABS  0010'   FUN     EA01*   GAMES   EA04*
HI      001F'   INIT    001A*   MAIN    001D*   PRINT   0007*
REFRSH  EA07*   STACK   0028"   START   0000'
```

Relocating Assemblers continued . . .

Listing 5

```
                          EXTRN  BURNIT
0000'  11 000C'  BURN:    LXI    D,MONST  ;START OF MONITOR
0003'  21 001D'           LXI    H,MONEND;END OF MONITOR
0006'  CD 0000*           CALL   BURNIT   ;BURN PROM
0009'  C3 0000            JMP    0        ;DONE
000C'            MONST    EQU    $
                          .PHASE  0F000H  ;START OF MONITOR
                 ;CODE IS ASSEMBLED IN THIS PROGRAM AS IF IT
                 ;RESIDED AT 0F00H
F000   C3 F003   MON:     JMP INTOIT      ;FIRST INSTR OF MON
                 ;
                 ;
F003   31 0100   INTOIT:  LXI    SP,0100H,STACK
F006   3E 23              MVI    A,'#'
F008   CD F00B            CALL   CONOT
                 ;
                 ;
                 ;ETC....
F00B   DB 00     CONOT:   IN     00H
F00D   17                 RAL
F00E   DA F00B            JC     CONOT
                 ;ETC
                          .DEPHASE
001D'            MONEND   EQU    $
                          END    BURN

Symbols:
BURN   0000'   BURNIT  0007*   CONOT  F00B   INTOIT  F003
MON    F000    MONEND  001D'   MONST  000C'
```

Run the Parallel MX-80 with North Star 5.2DQ

by Oliver C. Stokes, Jr.

I recently purchased an EPSON MX-80 for use with my North Star Horizon 2 computer. To keep my initial cash outlay at a minimum, I purchased the parallel output version. The North Star system manual provided the information required to construct a cable to interface the MX-80 to the Horizon parallel port, and everything seemed to be going fine until my computer sent some ASCII text to the MX-80.

The symptoms

The Horizon sent ASCII text; the MX-80 printed graphics characters. My first thought was that the MX-80 had a problem. Then I noticed that the difference in binary code between the ASCII character sent and the graphics character printed was a one in the most significant bit. Swapping my MX-80 with that of a friend proved the printer was OK, and I looked to my computer for the problem.

Oliver C. Stokes, Jr., P.O. Box 1683, Sierra Vista, AZ 85635

The problem

Some prowling around in the I/O routines of NS DOS 5.2 revealed the difficulty. For some reason, in the printer parallel output routine, the character to be sent is "OR'd" with 80H (10000000 binary) to set the MSB high prior to outputting the character. Before the two following successive outputs, the character is "EXCLUSIVE OR'd" with 80H to set the MSB low. Apparently the MSB is being utilized as a strobe. Since a separate strobe is available at the parallel output port, the rationale for this is not readily apparent to me. Unfortunately, setting the MSB high turns on the MX-80 graphics. Fortunately the fix is simple.

The fix

In location OACA of NS DOS 5.2DQ is the machine instruction F680 to "OR" the character with 80H before output. All that is required is to change this to F600 so that the character is "OR'd" with zero, thus leaving the MSB unchanged.

To do this permanently, what must be done is to load an

SUPER FAST!

Z80 DISASSEMBLER

$69.95

- Two pass operation - generates labels at referenced locations
- Generates Zilog mnemonics
- Allows user defined labels
- Allows define byte, define word and define space directives
- COMPLETE cross-reference
- 28 page manual
- Output to console, list or disk device(s) in any combination
- Generates mnemonics for CP/M system calls
- Illegal instructions generate define byte sequence
- Start and stop at any location in file
- Source or complete listing type output

SPEED - disassembles a typical 17K .COM file, generating a 110K .Z80 file (over 10,000 lines of source) and a 52K .XRF file in less than 1 minute 45 seconds using standard bios and 8" SS/SD!

Available for Z80 CP/M and TRS-80 III

SLR Systems

1622 North Main Street, Butler, PA 16001

(412) 282-0864

Terms: add $2 shipping US, others $5. PA add 6% sales tax. Specify format required. Check, MO, Visa, M/C, COD accepted

Z80, CP/M, TRS-80 TM's of Zilog, Digital Research, Tandy Corp resp.

CIRCLE 53 ON READER SERVICE CARD

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

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

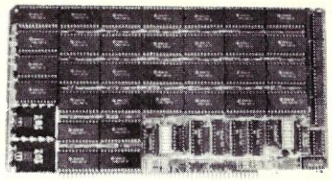
64K S100 STATIC RAM

$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)

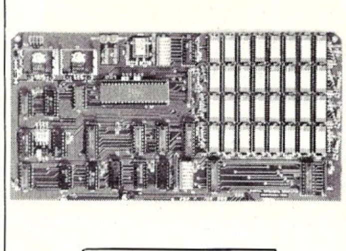
FOR 56K KIT $185

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 ON DISK) $69.95

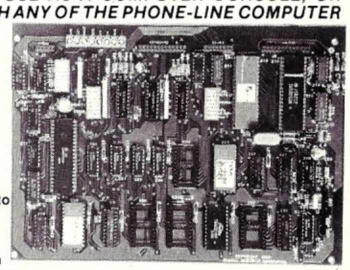
$399.00 #LS-100 (FULL 256K KIT)

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.



BLANK PCB WITH 2716 CHAR. ROM, 2732 MON. ROM $59.95

SOURCE DISKETTE - ADD $10

SET OF 2 CRYSTALS - ADD $7.50

 # ZRT-80

WITH 8 IN. SOURCE DISK!

$129.95 (COMPLETE KIT, 2K VIDEO RAM)

Digital Research Computers

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

TERMS: Add $3.00 postage. We pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Texas Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50 add 85¢ for insurance.

CIRCLE 63 ON READER SERVICE CARD

MX-80 continued . . .

image of DOS into RAM, modify it, and read it back onto the disk. Since DOS 5.2DQ loads in two parts (sectors 4-8 into locations 100-AFFH and sectors 8-9 into A00-DFFH), data in sectors 8 and 9 will load 100 Hex bytes lower than would be expected from their addresses. What we want to do is change the contents of OACB, in the DOS, from 80H to 00. One sequence that does this is as follows:

a. Boot the system up from a copy (not the original provided by North Star) of your system diskette, without write protect.

b. Follow this sequence

```
+LF DOS 4100    Load DOS into RAM
+GO MOE00       Load and run monitor
>FM 49CB 00     Change 80H to 00 at what will
                be working location OACB
>OS             Return to DOS
+SFDOS4100      Save modified DOS back on disk
```

c. Remove disk, write protect, and label as "System Disk Modified for MX-80 Parallel Interface"

Now, when running North Star Basic programs, all that should be required is a "PRINT #2" or "#2" statement to get text output on your MX-80.

Oliver C. Stokes, Jr., is a Lieutenant Colonel in the U.S. Army Signal Corps, assigned to the Communication Electronics Engineering Installation Agency of the U.S. Army Communications Command at Fort Huachuca, AZ. He has a B.S.E.E. from Johns Hopkins.

A Directory of User Groups

by Don Libes

The following directory does not claim to be all inclusive. It consists merely of the groups that have notified us of their existence, along with others mentioned in various publicatons. If we have left your group out, we are sorry! Please let us have the details.

CP/M USER GROUPS

SIG/M
Subgroup of the Amateur Computer Group of New Jersey. Has over 100 volumes of software and publishes a printed catalog ($2; $2.50 foreign).
Contact: *SIG/M*
Box 97
Iselin, NJ 08830

CP/M USERS GROUP (CPMUG)
Publishes a monthly magazine, has over 90 volumes of software, and has a printed software catalog.
Magazine: $24/yr, $50/yr foreign
Contact: *CPMUG*
1651 Third Ave.
NY, NY 10028
tel: *(212) 860-0300*, ext. 343

The following organizations have CP/M user groups that generally meet monthly:

North Orange Computer Club, *Box 3616, Orange, CA 92665*

Sacramento Micro Users Group, *Box 16153, Sacramento, CA 95816*

Valley Computer Club, *Burbank, CA 91510*

Denver Amateur Computer Society, *Box 1235, Englewood, CO 80150*

Connecticut CP/M Users Group, *110 Day Hill Rd., Windsor, CT 06095*

Washington, D.C. CP/M Users Group, *7315 Wisconsin Ave., Washington, D.C. 20014*

Chicago Area Computer Hobbyists Exchange, *824 Jordan Place, Rockford, IL 61108*

CP/M Users Group Northwest, *1346 NE 28th St., Portland, OR 98323*

Boston Computer Society, *Three Center Plaza, Boston, MA 02108*

Amateur Computer Group/NJ, *Box 319, So. Bound Brook, NJ 08880*

NY Amateur Computer Club, *Box 106, Church St. Station, NY, NY 10008*

RAMS, *Box 908008, Rochester, NY 14609*

Rhode Island Computer Hobbyists, *Box 599, Bristol, RI 02809*

CP/M Users, *c/o M. von Schneidermesser, Dept. of Ag. Economics, 427 Lorch St., B3, Univ. of Wisconsin, Madison, WI 53715*

Long Island Computer Association, *P.O. Box 71, Hicksville, NY 11802*

Don Libes, 4012 Adams Drive, Wheaton, MD 20902

CP/M.IRL, THE CP/M USER GROUP OF IRELAND
Meets monthly in Dublin area and publishes a monthly newsletter.
Membership: IR 5 pounds
Contact: *Doug Notley*
Gardner House
Ballsbridge
Dublin 4, Ireland
tel: *01-686411*

CP/M USER GROUP—U.K.
Contact: *David Powys-Lybbe*
11 Sun St., Finsbury Square
London EC2M 2QD England
tel: *247-0691*

CP/M GEBRUIKERS GROEP NEDERLAND
Subgroup of the Hobby Computer Club; distributes CPMUG and SIG/M software. Publishes a printed catalog (Hfl. 20.00)
HCC Membership: HFl 40.00/yr
Contact: *CP/M Gebruikers Groep Nederland*
Trajanusplein 1
4041 AK Kesteren
Netherlands
tel: *08886-1667*

C USER GROUPS

C USERS GROUP
Publishes a bimonthly newsletter, has over 30 disks of software, maintains a BDS C Compiler update service, a C-node network, and has software development products.
Annual membership is $10; $20 foreign
Contact: *Sheila Henson*
C Users' Group
Box 287, 112 N. Main
Yates Center, KS 66783
tel: *(316) 625-3554*, mornings only

PASCAL USER GROUPS

PASCAL/Z USERS GROUP
Publishes a bimonthly newsletter, has 24 volumes of software, and serves as a clearing house for members. Has a catalog of software ($3).
Annual membership: none
Newsletter: $9/year
Contact: *Charlie Foster*
7962 Center Parkway
Sacramento CA 95823
tel: *(916) 392-2789*

PASCAL/Z USERS GROUP OF EUROPE
Affiliated with U.S. group. Has 12 disks available and publishes a monthly newsletter.
Contact: *George Brooke*
Sebastian Bauerstrasse 20c
8000 Munich 83m
West Germany

UCSD SYSTEM USER SOCIETY OF UK
Contact: *John Ash*
Dicoll Data Systems, Ltd.
Bond Close, Kingsland Estate
Basingstoke, Hants RG24 0QB
England

UCSD PASCAL U.K. USERS GROUP
Contact: *Malcolm Harper*
Oxford University Computing Laboratory
Programming Research Group
45 Banbury Rd
Oxford OX2 6PE England

PASCAL USER GROUP OF U.K.
Contact: *Nick Hughes*
P.O. Box 52, Pinner
Middx HA5 3FE England
tel: *01-866-3816*

PASCAL/MT USER GROUP
Publishes a quarterly newsletter and has seven volumes of software.
Annual membership: $7 U.S., $13 foreign; Europe 30DM
Contacts: *PASCAL/MT User Group of U.S.*
c/o Henry Lucas
Box 192
Westmont, IL 60559
tel: *(312) 986-1550*

PASCAL/MT User Group of Europe
c/o Guenter Musstopf
Schimmelmannstr, 37a
D-2070 Ahrensburg, West Germany
tel: *04102/56629*

FORTH USER GROUPS

FORTH INTEREST GROUP
Publishes bimonthly newsletter, has 25 volumes of software, and provides information and products for Forth. They have a free catalog of their software and products.
Annual membership: $15, foreign $27
Contact: *Roy Martens*
Forth Interest Group
Box 1105
San Carlos, CA 94070
tel: *(415) 962-8653*

FORTH INTEREST GROUP OF U.K.
Contact: *K.C. Goldie-Morrison*
15 St. Albans Mansion
Kensington Court Place
London W8 5 QH England
tel: *01-937-3231*

UNIX USER GROUPS

UNI-OPS
Oriented toward new users. Has newsletter, mailing lists, conferences, local meetings, tutorials.
Membership: $24/yr
Contact: *UNI-OPS*
P.O. Box 5182
Walnut Creek, CA 94596-1182

USENIX
Oriented toward academic users. Has newsletter, conferences; distributes software.
Membership: $30/yr
Contact: *USENIX*
321 Mystic St.
Arlington, CA 02174

UNIFORUM (FORMERLY/USR/GRP)
Oriented toward UNIX vendors. Has newsletter, conferences; software and hardware catalog.
Contact: *Uniforum*
P.O. Box 8570
Stanford, CA 94305-0221

OTHER USER GROUPS

CBASIC USERS GROUP
This group is just starting up.
Contact: *Al Dallas*
11669 Balerio St., #213
North Hollywood, CA 91605
tel: *(213) 765-3957*

CROMEMCO USER GROUP
Publishes a bimonthly magazine, offers group insurance rates, and serves as a resource center for Cromemco users.
Annual membership: $35, $41 Canada/Mexico, $48 International
Contact: *Richard Kaye or Kathleen Heckman*
International Association of Cromemco Users
Box 17658
Irvine, CA 92713

DYNABYTE USERS GROUP
Publishes monthly newsletter, has 84 volumes of software, and serves as a clearing house for members.
Membership: $15/yr
Contact: *Kelly or Patti Borsum*
Random Factors Ltd.
Box 2875
Durango, CO 81301
tel: *(303) 247-9306*

EXIDY SORCERER USER GROUP
Contact: *Andy Marshall*
44 Arthurs Bridge Rd.
Woking, Surrey, GU21 4NT
England
tel: *04862/66084*

HX-20 USERS GROUP
Contact: *Terence L. Ronson*
25 Sawyers Lawn
Drayton Bridge Road
Ealing, London W13
England

ITHACA INTERSYSTEMS & S-100 BUS USERS GROUP

Contact: *George Brooke*
Sebastian Bauerstrasse 20c
8000 Munich 83
West Germany

ITHACA AUDIO S-100 USERS GROUP OF U.K.

Contact: *Dave Weater*
North Kykeham
Lincoln, LN6 8LN, England

JANUS/ADA USERS GROUP

Has two volumes of software, bulletin board system, and publishes a quarterly newsletter.

Annual membership: $4
Contact: *Randall C. Brukardt*
Box 1512
Madison, WI 53701
tel: *(608) 244-6436*

MICROPOLIS/VECTOR GRAPHIC USERS GROUP

Publishes monthly newsletter, has 55 volumes of software, and offers discounts on commercial software, parts and manuals for Micropolis drives; free software catalogue.

Annual membership: $18, $25 foreign
Contact: *Buzz Rudow*
604 Springwood Circle
Huntsville, AL 35803
tel: *(205) 881-1697*

MSDOS/SEATTLE COMPUTER PRODUCTS USERS GROUP

Publishes newsletter five-six times a year, has three volumes of software, an on-line bulletin board system and member referral service.

Annual membership: $18
Contact: *Joseph Boykin*
47-4 Sheridan Dr.
Shrewsbury, MA 01545

NEVADA COBOL USERS GROUP

Publishes quarterly newsletter, has one volume of software; offers discounts on computer products.

Annual membership: $10; $15 foreign
Contact: *Bob Blum*
5536 Colbert Trail
Norcross, GA 30092
tel: *(404) 449-8948*

NORTH STAR COMPUTER SOCIETY (NSCS)

Oriented toward professionals and hobbyists; supports exchange of informationon N* computers, software, and peripherals. Publishes monthly newsletter (*Polaris*).

Annual membership: $24
Contact: *NSCS*
P.O. Box 311
Seattle, WA 98111

OASIS USERS GROUP

Has 20 volumes of copyrighted software, holds meetings, and maintains an on-line bulletin board system. Has a printed software catalog ($1).

Membership: $35 (one-time charge)
Contact: *Fred Bellomy*
Box 2400
Santa Barbara, CA 93120

OSBORNE/MCGRAW-HILL BUSINESS SOFTWARE USERS GROUP

Publishes infrequent newsletter, has 14 volumes of copyrighted and public domain software (printed catalogue is $2), sells software and supplies.

Membership: $15/yr; $25/yr foreign
Contact: *Jack Ellis*
OSBUG
2252 Main St., #15
Otay, CA 92011
tel: *(619) 423-0538*

OSBORNE NATIONAL USERS GROUP

Assists in exchange of information and sale/exchange of equipment; publishes bimonthly newsletter.

Membership: $10/yr (includes subscription to bimonthly newsletter)
Contact: *Oborne National Users Group*
P.O. Box 424
Northridge, CA 91328

PROCESSOR TECHNOLOGY USERS GROUP

Publishes quarterly newsletter, has eight volumes of software, and comprehensive documentation on PTCo products.

Membership: $30/yr; $38/yr foreign
Contact: *PROTEUS*
1690 Woodside Rd. #219
Redwood City, CA 94061
tel: *(415) 368-2300*

SOFTWARE TOOLS USERS GROUP

Has newsletter, conferences, and software distributions: C and Ratfor (Fortran).

Membership: $15/yr
Contact: *Software Tools Users Group*
1259 El Camino Real, #242
Menlo Park, CA 94025

SORCERER'S APPRENTICE

Publishes a newsletter eight times a year, has four volumes of software, and a bulletin-board system: (313) 535-9186.

Annual membership: $18; $32 foreign
Contact: *Don Gottwald*
Sorcerer's Apprentice
Box 33
Madison Heights, MI 48071
tel: (313) 286-9265

TURBODOS USERS GROUP

Contact: *Steve Elias*
663 Joost Ave.
San Francisco, CA 94127

Z80 STARTER KIT USERS GROUP

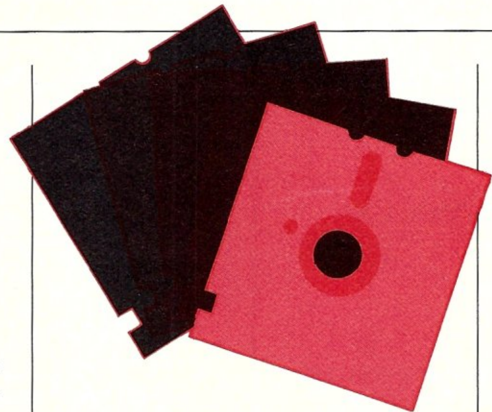
Publishes monthly newsletter and offers software for sale.

Membership: $10/yr; $15 foreign
Contact: *Cary Davids*
6000 Puffer Rd.
Downers Grove, IL 60516
tel: *(312) 969-9417*, evenings & weekends

Software Directory

Program name: BAKUP
Hardware system: Z80 running CP/M 2.2; 8″ SD or $5^1/_4$″ Osborne formats.
Minimum memory size: 48K
Language: Object Code
Description: BAKUP is a set of machine language programs for file backup operations in systems using hard disk and floppy disk storage. The ARCHIVE utility backs up hard disk files onto floppy disks; files that are too large to fit onto one floppy are automatically segmented into two or more parts on separate floppies. The files on the hard disks are marked as having been archived and will not normally be recopied during later archival operations. The archive copy is verified on a byte-by-byte basis. The archive operation may be selective or total, with standard wild card inputs. If global, files in all user areas are copied to the same user areas on the floppy disk(s). The RESTORE utility restores files from floppy to hard disk; multipart files on separate floppies are automatically restored to a single large file on the hard disk. Other utilities provided determine the archive status of files, automatically increment the version number of a program under development, compare files of the same name in different user areas or on different drives, and perform a number of other functions.
When released: 1983
Price: $49.95
Included with package: Additional programs from the UTILITY II package.
Available from:
Ficomp, Inc.
3017 Talking Rock Drive
Fairfax, VA 22031
(703) 280-1394
CIRCLE 309 ON READER SERVICE CARD

Program name: InfoShare
Hardware system: CP/M or MP/M system
Description: InfoShare is a communications package that allows microcomputers running either CP/M-80 or MP/M II to exchange information with each other or with a central MP/M system. Data of common interest may be stored centrally, while data of local interest only is kept out of the host system. The package also avoids duplication of expensive



hardware—particularly hard disk drives and printers. The package has a Host component and a Remote component; the Remote component can be operated independently as a terminal emulator for communication with other computers, timesharing systems, and information systems. A security utility included with the package can make entry to the host system dependent upon the correct entry of a password. Interconnection of the Host and Remotes is via an EIA RS-232C interface; the connection may be hard wired for distances of a few hundred feet, or through modems for longer distances. Reliable information exchange is obtained at speeds up to 9600 baud.
When released: 1983
Price: One-time system license (regardless of the number of Remotes) $250.
Included with price: Host component, remote component, security package, documentation.
Available from:
The Information People
443 Hudson Avenue
Newark, OH 43055
(614) 349-8644
CIRCLE 310 ON READER SERVICE CARD

Program name: Z80ASM
Hardware system: Z80 CPU and CP/M 2.2
Minimum memory size: 24K
Language: Z80 or 8080 assembly language
Description: Z80 assembler including full source code and tutorial on assembler theory. Assembler accepts standard Zilog mnemonics plus 19 pseudo ops, including XLIST, TITLE, and nested conditionals with ELSE. It can read source from multiple input files, and prints a sorted symbol table as part of its standard listing output. Modular construction makes it easily adaptable as a cross assembler, and symbolic definition of important parameters such as the number of characters allowed in a symbol makes it simple to hand-tailor details of language or syntax as desired. The accompanying tutorial manual contains a complete description of how assemblers work, with explanations of such advanced professional techniques as expression processing by recursive descent, radix 40, op code analysis, binary search, recursive processing of nested conditionals, etc.
When released: January 1983
Price: Manual alone, $25; manual and disk, $50
Included with price: Manual contains tutorial and complete source code listing in both Z80 and 8080 assembly language. Standard CP/M 8″ SSSD disk contains above source code plus Z80 ASM.COM.
Available from:
King Software
PO Box 208
Red Bank, N.J. 07701
(201) 530-7245
CIRCLE 311 ON READER SERVICE CARD

Program name: MCS-3 manufacturing control system
Hardware system: Runs on any R/M Cobol supported hardware
Minimum memory size: 64K
Language: Cobol
Description: MCS-3 Manufacturing Control System was designed for small to medium-sized manufacturing firms. The system is available in modules, but fully integrated within each module. Modules available include: Inventory control, MRP, order entry, A/P, A/R, G/L, and payroll. Fourth-quarter releases include: Financial resource planning, forecasting, job costing, capacity planning, and shop floor control.
When released: December 1979
Price: $750-$2500, depending on the module
Included with price: 6 weeks free telephone support, manual and documentaton. Support available for yearly fee. Installation by dealer.
Available from: Dealer network, or if uncommitted territory, then direct:
Micro Manufacturing Systems

2550 Corporate Exchange Drive
Columbus, OH 43229
(614) 895-0738
CIRCLE 312 ON READER SERVICE CARD

Program name: TLX-A-SYST
Hardware system: CP/M, IBM PC, Apple, Osborne, Attache, Kaypro, Molecular, Altos, Victor 9000, TRS-80
Minimum memory size: 64K
Language: CBasic
Description: Interfaces computers to domestic and international telex network, TWX, telegram, and mailgram, using ordinary phone lines. Menu-driven; very easy to use. No special coding or protocol required. Archiving and traffic log provided. Directory of telex numbers; friendly, comprehensive manual provided. Designed for nontechnical users: help menus simplify training and increase productivity.
When released: October 1982
Price: $250
Included with price: Manual and disk
Available from:
XYZZY or dealers
PO Box 9002-116
Boulder, CO 80301
(303) 444-6675
CIRCLE 313 ON READER SERVICE CARD

Program name: ICT 1.0
Hardware system: CP/M 2.0
Minimum memory size: 24K
Language: Object code
Description: ICT is a software package that runs under CP/M and transfers ISIS II files to CP/M format. ISIS II directories can be displayed and files can be viewed. Redirection of drive assignments permits flexibility in transferring files.

The ASCII or object file transfer mode is selected by the operator. A file transferred in the ASCII mode does not have to be edited to remove extraneous characters from the end of the file. Double density ISIS II disks can be read with ICT if the host controller reads 128-byte sectors from the double-density disks.

ICT is menu driven, with many descriptive messages displayed for ease of use.
When released: March 1983
Price: $95
Included with price: 8″ SSSD disk; manual

Available from:
Dantek Software, Inc.
4550 Schoolhouse Rd.
Batavia, OH 45103
(513) 752-1921
CIRCLE 315 ON READER SERVICE CARD

Program name: GRAM-A-SYST
Hardware system: CP/M, IBM PC, Apple, Osborne, Kaypro, Victor 9000, TRS-80
Minimum memory size: 64K
Language: CBasic
Description: Interfaces computers to domestic and international telex network, TWX, telegram, and mailgram using ordinary phone line. Menu-driven; very easy to use. Archiving and traffic log provided. Directory of telex numbers and names and addresses. Comprehensive, friendly manual is provided. Designed for nontechnical users: help menus simplify training and increase productivity.
When released: January 1983
Price: $250

Included with price: Manual and disk
Available from:
XYZZY or dealers
PO Box 9002-116
Boulder, CO 80301
(303) 444-6675
CIRCLE 314 ON READER SERVICE CARD

Program name: QUESTEXT III
Hardware system: 2.x 56K CP/M, or IBM PC and a 24 x 80 ASCII encoded CRT. A version is also available to run under VAX or RSTS.
Description: Information Reduction Research, an international firm established to publish research-oriented databases, has announced the introduction of QUESTEXT III for mini and microcomputers. QUESTEXT III is intended for all computer users, from system developers to novices, and it can be learned in one session.

QUESTEXT III is a carefully developed general-purpose system for organizing, storing, and retrieving textual information. It is not a simple word processor, since it imposes structure on entered text, and it is not a DBMS, since its structure is invisible to the user. QUESTEXT III organizes text into tree-like menu structures without programming and debugging overhead. Prompts and other aids are designed to accommodate all user levels comfortably.

QUESTEXT III is economical because users can avoid buying many special-purpose programs due to its broad functionality. It also allows users to develop their own applicatons without the time and cost of programming, debugging, and testing. QUESTEXT III is easily updatable at any point.
Price: $299.95. Electronic tutorial and five applications, $29.95. A miniversion is also available as part of a money-back trial offer. Miniverson, $49.95. Free money-back offer on miniversion if returned in good condition within 14 days.
Available from:
Information Reduction Research
1538 Main St.
Concord, MA 01742
CIRCLE 316 ON READER SERVICE CARD

Program name: CP/M Recovery
Hardware system: CP/M
Description: CP/M Recovery completely eliminates data and text loss in computer memory due to system crashes, program errors, operator error, failure to backup, disk failures, unexpected full disk conditions, or for virtually any other reason. Unlike other products that recover disk data files, CP/M Recovery allows the computer user to recover memory, conduct editing on data within memory, including control characters, and to save that data to any disk file. It functions for both single- and multiuser systems, and is extra user-friendly.
Price: $99; complete instructions included.
Available from:
Lion Micro Systems/In-Sync Systems, Inc.
1900 Pacific Ave., Suite 501
Dallas, TX 75201
(214) 760-9120
In U.K. write to **Lion House,** *227 Tottenham Court Rd., London WIP OHX, England; phone 01 637 1601.*
CIRCLE 317 ON READER SERVICE CARD

Software Directory
continued . . .

Program name: EXPENSE TRAC
Hardware system: IBM PC, TRS-80 Model III; other CP/M-based microcomputers
Language: R/M Cobol
Description: EXPENSE TRAC automates fund accounting procedures for school administration, small- profit and nonprofit organizations, and departmentalized budgeting for divisions of larger companies. It is structured to provide administrators and management with control information on expenditures in relation to budgeted targets.

EXPENSE TRAC allows users to define values for accounting structure such as funds, cost centers, and account numbers. It maintains a master file of current balances for budgeted, expended, and encumbered funds. It provides a detailed audit trail printout summarizing all transactions entered into the system. It allows the user to see on-screen displays of account balances, account details, requisition details, and vendor code details. It provides up to 15 summary and detailed reports in a variety of sequences and totalling schemes. It also provides increased file space through a unique data compaction process.
When released: May 1983
Price: $395
Included with price: complete documentation
Available from:
Output Inc.
2401 East Washington St.
Bloomington, IL 61701
(309) 663-9396

CIRCLE 318 ON READER SERVICE CARD

Program name: TECHTYPE
Hardware system: CP/M
Minimum memory size: 32-38K
Language: written in Fortran, but can be used without a compiler.
Description: TECHTYPE is a text formatting system designed especially for scientific, engineering, mathematical, and multilingual document production. It was developed by Green Mountain Radio Research Company for their own use, and has been used in their office for over three years. TECHTYPE is adaptable to most hardware.

TECHTYPE is particularly useful to the engineer or scientist who must prepare documents that contain equations. It allows unlimited sub-

Software Directory continued . . .

and superscripting and has the ability to mix up to 10 fonts of the user's choice (Greek, math symbols, italic, Russian, Hebrew, etc.). In addition, TECHTYPE provides control of format, pitch, emphasis (underscore, double underscore, and boldface, and can even address envelopes and mark classified materials. With TECHTYPE, you can mix Gothic, italic, and Greek; use boldface to designate vectors and matrices; and set up multilevel ratios.

The three principal programs that make up TECHTYPE are DISPLAY, DRAFT, and PRINT. DISPLAY allows the user to preview the document on the screen, with emphasis features displayed. On bit-mapped CRTs, such as the North Star Advantage, all 10 fonts as well as true sub and superscripts are displayed. For conventional CRTs, DISPLAY doubles the vertical spacing (so you can see sub/superscripts) and uses alternate fonts, reverse video, and other available attributes to represent printed text.

The second program, DRAFT, is used with a multifont dot-matrix printer to produce high-speed drafts and working papers. PRINT is used with a daisywheel printer to produce camera-ready copy and final reports. Multipass printing allows the printwheel to be changed only *once* per page per font. This program can be used on the more common single-font printers, and two-font printers such as the Diablo 630ECS or the Qume Twin Track.

When released: July 1983
Price: $300
Included with price: User manual (which includes tutorial), installation manual software and reference card. The program can also be specially tailored to equipment.
Available from:
Green Mountain Radio Research Co.
240 Staniford Rd.
Burlington, VT 05401
(802) 862-0997

CIRCLE 324 ON READER SERVICE CARD

Program name: UNE/CON version 3.0
Hardware system: CP/M 2.2
Minimum memory size: 32K
Language: 8080 Assembler
Description: The new UNE/CON file recovery program from Elliam Associates combines the features of the earlier UNERA and CONFLICT programs into one easy-to-use program. The command ERA*.BAS instead of ERA*.BAK is no longer a fatal mistake. Just enter UNE/CON and reactivate any or all ERAsed files.

The UNE/CON program reads all the filenames in the directory, both active and ERAsed. The program then checks the disk space assignments for each filename. If the disk space of an erased filename is *not* being used by an active filename, the program allows the user to recover that file. If the disk space of an erased filename *is* being used by one or more active filenames, the program allows the user to recover that file. If the disk space of an erased filename is being used by one or more active filenames, the program reports that the disk space is assigned to both active and erased filenames. The manual offers suggestions for file recovery when there is a conflict for disk space.

In addition to the new UNE/CON program, improved versions of both the original UNERA and CONFLICT are included on the distribution disk. The UNERA program may be used to recover a single filename with the command UNERA <filename>. CONFLICT displays or prints the status of the disk space.

These programs are designed to work on all CP/M 2.2 systems using a standard CP/M directory format with single, double-, and quad-density floppy disk drives as well as hard disks. Soft-sectored minidisk formats are available for most popular CP/M computers such as Osborne, TeleVideo, Superbrain, Xerox, Otrona, Kaypro, IBM PC (CP/M), as well as hard-sectored disks for the North Star computer.
When released: 1983
Prices: distribution disk and instruction manual, $75. Manual alone, $10 (applicable to purchase). Specify disk format when ordering.
Available from:
Elliam Associates
24000 Bessemer St.
Woodland Hills, CA 91367
(213) 348-4278
CIRCLE 325 ON READER SERVICE CARD

Program name: TaxCalc
Hardware systems: IBM PC, Apple II, TRS-80, North Star, Osborne, Superbrain, Xerox 820
Minimum memory size: 48K
Description: TaxCalc, a tax planning program designed by CPA Harry S. Chud, allows the computation of income tax variables and actually selects the lowest tax alternative.

Using VisiCalc, SuperCalc, 1, 2, 3, or Multiplan templates, TaxCalc is compatible with most personal computers now on the market. This easy-to-use spreadsheet template was designed for the rapid calculation of income taxes, to perform tax-planning functions, and to verify existing tax returns.

TaxCalc input lines follow IRS forms. Computations include capital gains deductions, capital loss limitations, contribution limitations, and the two-earner married couple deduction. TaxCalc also calculates the income averaging tax, minimum tax, alternative minimum tax (including tax credit limitations), and then selects the lowest alternative.

CIRCLE 155 ON READER SERVICE CARD

CIRCLE 271 ON READER SERVICE CARD

CIRCLE 23 ON READER SERVICE CARD

Using the VisiCalc, SuperCalc 1, 2, 3, or Multiplan window on a computer monitor, the user can actually see the results of tax variables and "what if" scenarios. Upon completion of the program, a printer can provide hard copies as a permanent record.

TaxCalc is designed to give the latitude necessary for a complete tax-planning software package that would be most beneficial to the taxpayer. It is also being used in several of the "Big Eight" accounting firms, such as Arthur Young & Co., Coopers & Lybrand, and Touche Ross & Co.

When released: November 1982
Price: $125 the first year; yearly updates, $50
Included with price: TaxCalc template; instructions
Available from:
TaxCalc
4210 W. Vickery
Fort Worth, TX 76107
(817) 738-3122
CIRCLE 326 ON READER SERVICE CARD

Program name: MIS (Medical Information Service)
Hardware system: CP/M
Description: A new medical information management product from Mohawk Data Sciences allows multihospital and multiclinic organizations to access large volumes of patient and financial information, improve medical services, and control costs. The new product, called the MDS Medical Information System (MIS), is a complete patient billing and accounting system that is ideally suited for hospitals and clinics of up to 200 beds. MIS utilizes proven, specialized medical data management software designed specifically for use on the MDS Series 21 family of data processing systems.

MIS consists of several components: patient admission/discharge, billing, accounts payable/receivable, insurance claims processing, payroll, general ledger, and monthly revenue reporting. MIS software can be easily customized for specific applications. And, with MDS' Personal Computing 21 product, which incorporates CP/M, any of the more than 2000 applications packages available for personal computers can be used with MIS.

Of particular interest in the administration of a group of associated hospitals or clinics is the ability to process data locally and communicate that information among member groups and central headquarters as well as with insurance companies. The Series 21 may be configured for dial-up or leased lines.

Filing of insurance claims is made easy with the Series 21, and MIS allows insurance data to be transmitted via telephone lines directly to the processing centers of the major insurance companies. Automatic filing of claims results in faster payment processing, and, since all editing and validating of data is done on the Series 21 at the healthcare facility, the data received at the insuror is error free, which further reduces processing time.

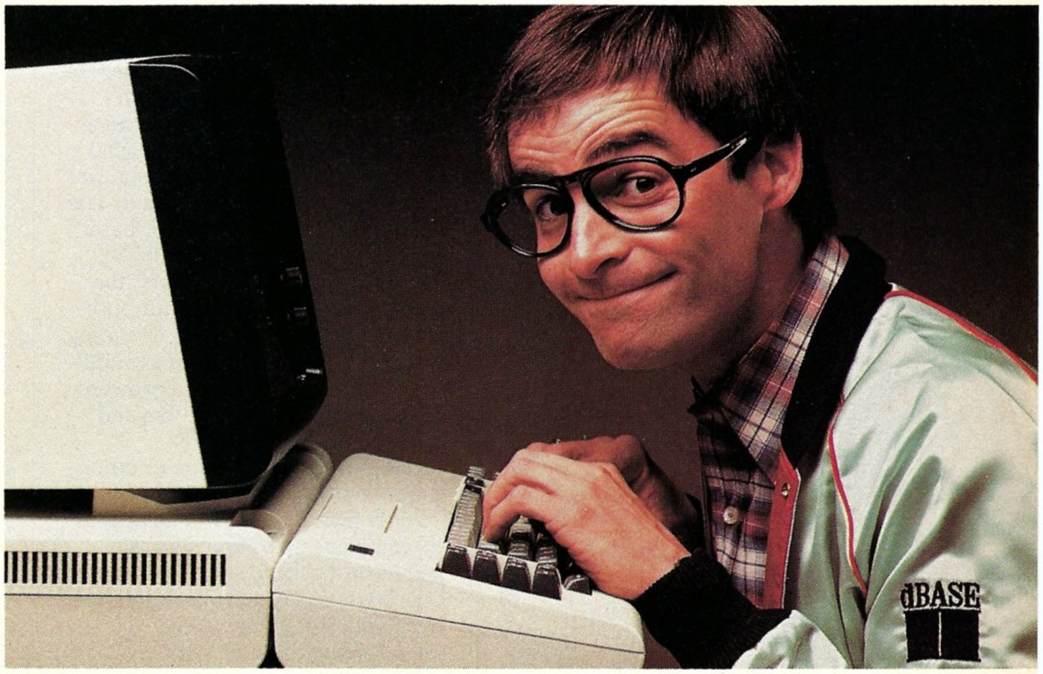
Through menu-driven screens on the Series 21, users without data processing experience can automatically perform routine applications and generate detailed reports.

Series 21 has earned an excellent reputation for reliability and ease of use. As a distributed processing system, it provides more accurate data capture, faster error correction, reduced mainframe load, lower communication costs, and faster turnaround time. A modular system, Series 21 was designed to offer a chain of upward growth for future company computing needs. These building blocks, designed to support systems with up to 16 stations, can be configured for current customer demands and are easily upgraded for changing requirements.

Available from:
Mohawk Data Sciences
Seven Century Drive
Parsippany, NJ 07054
(210) 540-9080
CIRCLE 327 ON READER SERVICE CARD

Program name: DR Graph
Hardware system: CP/M, both 8-bit and 16-bit

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



You've written all those slick dBASE II programs.

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

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

But that's just not good enough.

Uh-uh.

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

Here's how.

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

And it can turn you into an instant software publisher.

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

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

We'll tell the world.

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

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

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

Go for it.

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

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

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

dBASE™ II

ASHTON·TATE

CIRCLE 13 ON READER SERVICE CARD

Minimum memory size: 192K
Language: (menu driven)
Description: Digital Research, Inc., has announced DR Graph, a presentation-quality graphing application from the CP/M Graphics™ family.

Business professionals can use this simple, interactive graphics and editing tool to develop slide presentations, trend charts, and financial analysis reports. DR Graph requires no programming or training and uses GSX™ peripheral libraries for hard copy and output device flexibility.

DR Graph is built on GSX, the graphics system extension for both the 8-bit and 16-bit CP/M operating systems. GSX gives CP/M the ability to interface with many hardware devices ranging from plotters and printers to CRTs.

DR Graph provides presentation-quality output through very user-friendly menus. With DR Graph, users can place up to four separate graphs on a page to simplify complex page layouts. In addition, graphs can be created using data from spreadsheet programs such as VisiCalc and SuperCalc. DR Graph also features different type styles, line styles, and full color support:
When released: September 1983
Price: $295
Included with price: documentation
Available from:

Digital Research, Inc.
160 Central Avenue
Pacific Grove, CA 93950
(408) 649-3896

CIRCLE 328 ON READER SERVICE CARD

Program name: DECISION-ANALYST
Hardware system: CP/M, CP/M-86, or MS-DOS
Minimum memory size: 52K of RAM (96K RAM with CP/M-86 and MS-DOS); a 24 x 88 column screen and an 80-column printer.
Description: DECISION-ANALYST assists professional managers and businessmen in analyzing complex business problems where there are many alternatives and/or criteria. It structures the decision-making process into logical and easy-to-follow steps. The program is designed so that it can be learned by using only the "help" screens.

DECISION-ANALYST takes the user through eight menu-selected sections including problem definition, statement of decision purpose, establishing and valuing "must" and "want" criteria, calculation of criteria values, defining alternatives, weighting and scoring alternatives against criteria, assessing possible adverse consequences, and final conclusions and choice. The final reports are printed in polished format.
Price: $139
Included with price: thoroughly indexed and detailed manual
Available from:

Executive Software, Inc.
2 North State St.
Dover, DE 19901
(705) 722-3373

CIRCLE 329 ON READER SERVICE CARD

Program name: BRAINSTORMER
Hardware system: TRS-80, CP/M 80-column monitor machines including Apple II, Osborne I, and Kaypro II. All systems require MBasic, 2 drives (5¼" SS or SD only).
Minimum memory size: 48K
Description: BRAINSTORMER is a powerful software tool for generating potential solutions to complex problems. It works by building a description of a problem in terms of the themes and variations that affect its solution. The description of the problem is "probed" by BRAINSTORMER to generate ideas about potential solutions to the problem. The user refines the process by controlling the occurrence of particular themes and variations until a sufficient quantity of potential solution strategies is produced. Up to 10 billion "idea probes" can be generated for any user-specified problem.

BRAINSTORMER was designed by Dr. Shawn Boles, an experimental psychologist with a background in both Creativity Theory and microcomputer software development. Implementing a proven problem-solving technique, the Morphological Box, BRAINSTORMER provides both a structure for describing problems and an effective process for finding solutions to them.

With BRAINSTORMER, the user is led through a series of steps to produce a structured representation of the problem that he is interested in solving. Then BRAINSTORMER guides the user through a process of examination and reconsideration of the structure by generating new ways

Software Directory
continued . . .

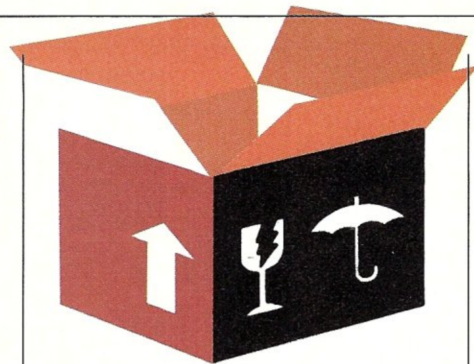
of looking at the problem.

Since the program can be used to generate new ideas about any topic, it offers virtually unlimited avenues of use. Potential applications include: increasing flexibility of your thinking, discovering new products, targeting new markets, and exploring organizational problems.

The program is easy to learn. BRAINSTORMER is supplied with a set of files covering example applications. These files are used with self-paced demonstration/tutorial exercises from the User's Guide to achieve rapid mastery of the program. The User's Guide also contains selections covering the detailed use of each command, and instructions for using BRAINSTORMER effectively.

When released: July 1983
Price: $50 for a single machine.
A program for concurrent group license for 2 to 10 machines used by a single organization is $100.
Included with price: User's Guide.
Available from:
Soft Path Systems
c/o Cheshire House
105 North Adams
Eugene, OR 97402
(503) 342-3439
CIRCLE 330 ON READER SERVICE CARD

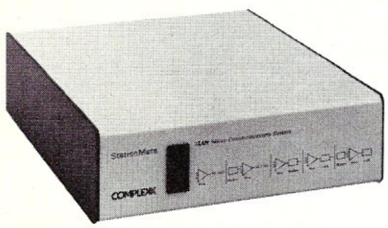
New Products



Microcommunications system

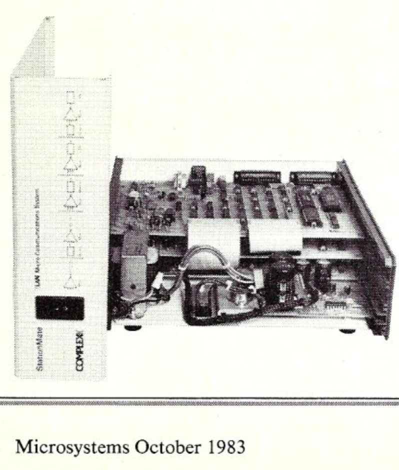
StationMate is a unique data communications system that combines a statistical multiplexer, an intelligent modem with an automatic dialer, and a local area network (LAN) interface into a single device. The fully integrated design of the hardware and firmware results in a system that offers independent use of the three basic functions, while allowing all three to be used together in various configurations to create sophisticated networks for micro and mini computers.

StationMate is the first of Complexx's XLAN local area network devices. XLAN provides the first truly independent, low-cost LAN for mini and micro computers. It uses a twisted, shielded wire communications medium for fast, easy installation. XLAN can support up to 64 devices over a 5000-ft bus.



StationMate provides three data ports with standard RS-232C connectors. The model gives a fourth (remote) user access to the system. This means that the user can establish a remote workstation at home or at any other location and have full access to the network.

StationMate and all Complexx products use an adaptation of the American Bell X.25 level three communication protocol. This protocol provides automatic switching, port selection, and port class selection. Other StationMate features include a large (16K) segmented buffer, menu-driven set-up and configuration with two levels of password security, auto baud to permit easy interface to user devices, full diagnostic capability, and error-free data transmission with detection and retransmission in case of errors.



Price: $1,450.

Complex Systems, Inc., *P.O. Box 12597, Huntsville, AL 35802; (205) 830-4310.*

CIRCLE 301 ON READER SERVICE CARD

Desktop computer serves as master station in HiNet LAN

Digital Microsystems has unveiled an enhanced version of its popular CP/M-based portable computer (the Fox) with 15 MB of built-in formatted Winchester disk storage. This stand-alone system can also serve as the master station in the company's HiNet local area network.

Called the DMS-15, this system brings together in one compact unit the DSC-3 (Z80A) processor with 64K RAM, a 9″ CRT, one 5¼″ DSDD 614K floppy, 15 MB of formatted Winchester disk storage, the network interface, and four RS-232C serial ports.

The DMS-15 makes it possible for users with large data bases to take advantage of low-cost single-user microcomputer technology and a broad range of CP/M software available from many sources. Typical applications include extensive inventory tracking, large legal and library data bases, and comprehensive mailing lists. In addition, the system can serve as a master station in a HiNet LAN. HiNet can support up to 32 users and address as many as 255; it has been installed at more than 1,000 locations worldwide.

The DMS-15 lists for $7495, including CP/M 2. The software needed to run the system as a HiNet master station is an additional $500.

Digital Microsystems, *1755 Embarcadero, Oakland, CA 74696; (415) 532-3686.*

CIRCLE 302 ON READER SERVICE CARD

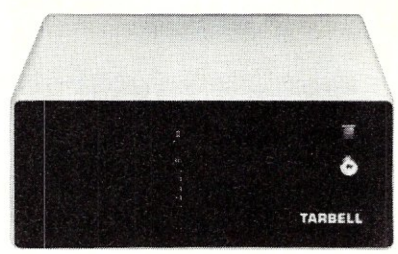
New computer from Tarbell

Tarbell Electronics has introduced the new Rebel computer, which combines more memory with increased speed and capability. Hardware includes Z80B CPU operating at 6MHz, 19 MB of hard disk memory, 372K of floppy disk memory, and 64K of main (semiconductor) memory expandable to 1 MB. Included are two RS-232 serial input/output ports for CRT and printer.

The software consists of MicroPro's WordStar word processing module, Tarbell Database System, Tarbell's Basic, and Digital Research's CP/M 2.2. The Rebel can be used for word processing, data bases, planning, accounting, inventory control, mailing lists, etc.

Price: $4,995.

Tarbell Electronics, *950 Dovlen Place, Suite B, Carson, CA 90746; (213) 538-4251.*



CIRCLE 303 ON READER SERVICE CARD

Enhancements to HiNet LAN

Digital Microsystems has added two enhancements to its HiNet local area network: a low-cost integrated Z80A-based workstation with a 12″ CRT, and a 46 MB Winchester master station.

The DMS-1280 workstation may be the lowest-cost LAN work station currently on the market, and the 46 MB HiNet master station effectively doubles the company's previously available formatted Winchester disk capacity for only a 20 percent increase in price. Together with the DMS-15, a 15 MB stand-alone system that can also serve as a HiNet master station, these products give users a broad range of cost-effective Z80 and 8086-based alternatives for building and expanding their LANs.

The new DMS-1280 work station, with an integrated 12″ video monitor and detachable keyboard, offers a variety of features including a 4 MHz Z80Z processor, 64K RAM, 2K PROM, a 500K baud RS-422 network port, and a 9600 baud RS-232 printer part. The DMS-1280 operates at 115 and 230 volts AC at 50/60 Hz.

The integrated video monitor can emulate five CRT terminals: ADM-3A, ADM-5, TeleVideo 910, Hazeltine 1420, and ADDS Regent 25. The unit also provides a 24-row by 80-character display area, a 25th row for displaying status or user information, and dipswitch-selectable character sets for English, Dutch, Japanese, Danish/Norwegian, Swedish/Finnish, Spanish, French, and German/Swiss.

HiNet is a CP/M-based packet-switched network providing 500K baud serial data transmission with SDLC protocol. The network, which supports up to 32 users, uses a master/slave polling scheme with RS-422 electrical specifications using twisted-pair or flat-ribbon cable.

Price: DMS-1280 work station, $1,695; DMS-3/103 46 MB master station, $11,990.

Digital Microsystems, *1755 Embarcadero, Oakland, CA 94606; (415) 532-3686.*

CIRCLE 304 ON READER SERVICE CARD

UNIX-based supermini computer

Pyramid Technology Corp. has announced a UNIX-based 32-bit virtual memory high-performance computer. The Pyramid 90x, the first supermini designed for a UNIX operating system environment, features a proprietary Central Processing Unit and instruction set, Pyramid's high-performance XTEND™ bus, enhanced UNIX operating system, a sophisticated memory hierarchy, and intelligent Input/Output processors.

The Pyramid 90x's proprietary 32-bit CPU has a 125-nanosecond cycle time and fits on three boards. Most instructions are executed in two machine cycles and are pipelined for further performance. The CPU also features a 4K high-speed instruction cache.

Pyramid incorporated recent advances in Reduced Instruction Set Computer (RISC) theory to design the system's unique architecture, which includes a register-intensive instruction set. The processor's 32-bit addressing ability, virtual memory operations, interrupt handling, memory management, instruction cache, symmetrical I/O instructions, and large number of registers provide a fast, efficient environment for UNIX and high-level languages.

The Pyramid 90x's synchronous XTEND bus has a flexible open-ended design that will allow future multiprocessor configurations and the integration of new technology. The bus has a 32 MB/second bandwidth and accepts the CPU, one to four memory modules (1 to 2 MB each), the System Support Processor, as well as multiple I/O processors which allow for interfacing to peripherals, networks, and adapters to other buses such as the Multibus™.

Pyramid's port of Bell Laboratories' UNIX System V (under license from Western Electric) incorporates enhancements made by the Univ. of California at Berkeley and proprietary Pyramid 90x features. Pyramid's additions include streamlined system calls and trap recovery, 2K block size for the file system, I/O logic off-loading, and simplified UNIX configuration. This enhanced virtual memory UNIX automatically handles programs and data arrays that are larger than physical memory, eliminating the need for programmed "overlays." Pyramid's C, Pascal, and Fortran 77 compilers generate code optimized for this register-intensive architecture.

The Pyramid 90x's memory space can range from 1 to 8 MB. Its memory hierarchy provides each UNIX process with 4 gigabytes virtual address space utilizing a 2K page size for demand-paging. The system supports 16 to 128 users.

Price: ranges from under $100,000 to over $300,000, depending on system configuration.

Pyramid Technology Corp., *1295 Charleston Rd., Mountain View, CA 94043; (415) 965-7200.*

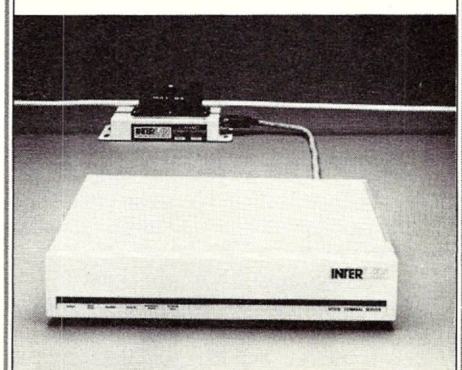
CIRCLE 305 ON READER SERVICE CARD

Ethernet terminal server

Interlan, Inc., has introduced the NTS10 terminal server, a device that interfaces any asynchronous EIA RS-232C serial I/O device onto the industry standard Ethernet/IEEE-802.3 local area network. The NTS10 terminal server provides some of the most advanced networking features available in the industry at a cost-per-device connection of less than $400. Data processing devices that can be connected to the network through the NTS10 include terminals, mini- and mainframe computer ports, personal computers, printers, and modems.

The NTS10 provides a "virtual circuit" communication service for electronically interconnecting user equipment. These virtual circuits appear as direct physical connections between user devices, but are electronically created, maintained, and terminated by the protocol procedures working within the NTS10 unit. These protocols resolve EIA RS-232C device incompatibilities in a manner completely transparent to either device. This means, for example, that a hard-copy terminal set to operate at 1200 baud can be logically connected to a computer port set to 9600 baud.

The NTS10 can be used in operations that include port switching, port contention, resource sharing, personal computer networking, and simplified wiring for dispersed terminals and printers. Advanced networking features include Ethernet compatibility, shared



transceiver connection, switched and permanent virtual circuits, network security, and network and system diagnostics.

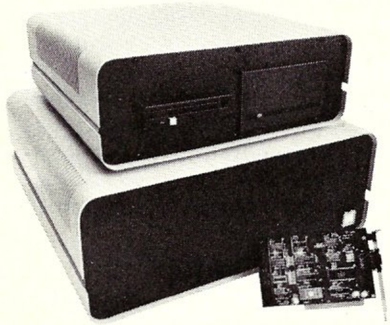
Prices: Interlan NTS10 terminal server: $3200 per 8-port unit; $2500 per 4-port unit.

Interlan Inc., *3 Lyberty Way, Westford, MA 01886; (617) 692-3900.*

CIRCLE 306 ON READER SERVICE CARD

NET/PLUS product line

NET/PLUS™ is a network systems product line from Interlan, Inc., that provides data communications and information sharing between both homogeneous and heterogeneous systems on the Ethernet/IEEE-802.3 local area network.



NET/PLUS incorporates Interlan's existing and recently introduced hardware, software, and system products into an integrated network architecture that addresses a principal concern of network users: that of multivendor equipment compatibility.

In contrast to proprietary local area networks that allow only one manufacturer's system to intercommunicate, NET/PLUS lets users tie together data-processing equipment built by different manufacturers. This provides them with the freedom to choose vendor equipment on the basis of price, performance, and functionality rather than communications compatiblity.

Interlan, Inc., designs, manufacturers, and sells a variety of hardware, software, and system products that provide data communications and information sharing between heterogenenous systems in a local area network.

Interlan, Inc., *3 Lyberty Way, Westford, MA 01886; (617) 692-3900.*

CIRCLE 307 ON READER SERVICE CARD

Closed-loop 5¼″ Winchester Disk Drive

A closed-loop 5¼″ Winchester disk drive that provides 50 MB of fast-access storage capacity on plated media has been introduced by Tandon Corporation. The Model TM705 is the second in a family of high-capacity drives offering high-performance, low-cost random-access memory for multiuser microcomputer and minicomputer business systems. The new 50 MB drive is designed to meet the growing demand for even higher storage capacity in the standard 5¼″ form factor. The first model, the TM703, featured a 30 MB capacity.

The TM705 features an unformatted storage capacity of 50.1 MB using three 5¼″ plated disks. High-density recording is performed on five data surfaces, with the sixth dedicated to servo control. The drive has a track density of 1000 tracks per inch, and a lineal density of 10,416 bytes per track. The configuration of 962 recording cylinders is nearly three times that of most open-loop

5¼″ Winchesters.

Higher performance and increased ruggedness are provided by use of plated media rather than the conventional industry-standard oxide-coated disks. The disks are plated with a thin film of nickel and cobalt over an aluminum substrate at Tandon's plated-media facility in Santa Clara, CA. The plating provides a more durable surface than oxide coating. It also provides better performance at the same price. Window margin performance, which is important for determining data integrity, is twice as good.

The TM705's closed-loop servo system, featuring a rotary voice-coil positioning arm, provides extremely fast operating times. Track-to-track access time is 5 ms. Average access time is 39 ms, with a maximum of 85 ms. Head settling time is 2 ms. An on-board microprocessor allows the TM705 to buffer positioning information and compute the fastest, most efficient positioning path from one track position to another.

Other features include a brushless DC motor, a data transfer rate of 5.0 MB/s, and an industry-standard interface and power supply. The TM705 measures 3¼″ × 5¾″ × 8″.

Price: Approximately $1,000.

Tandon Corporation, *20320 Prairie St., Chatsworth, CA 91311; (213) 993-6644.*

CIRCLE 320 ON READER SERVICE CARD

New SKS 8/16-bit portables

SKS Computers, Inc., has introduced two leather-briefcase styled, fully modularized 8/16-bit portable microcomputer lines, one of which is fully compatible with the firm's complete line of desktop computers, and both of which feature the new, powerful Intel 80186 microprocessor in the 16-bit configuration. Both are IBM compatible.

The desktop-compatible unit, called the SKS 2502 Nano™, in its standard 8-bit configuration contains dual 5¼″ 400K minifloppy disk drives, a Z80A CPU with 80K or RAM memory, a built-in CRT and separate keyboard controller, two RS-232C serial ports for printer/modem connections, and the CP/M operating system.

The CRT in the Nano series is a 5″ × 9″ green rectangular screen displaying 16 or 24 lines by 80 characters, with reverse video/magnified character attributes. The software includes the CP/M operating system, Modified CBasic, Perfect Writer, Perfect Filer, Perfect Speller, Perfect Calc, and Menu Runtime.

With the addition of the 16-bit 80186 microprocessor option, a parallel port, or a 5¼ MB Winchester hard disk, the Nano becomes a very sophisticated dual processor 8/16 portable with 128K or RAM, able to run the growing number of popular 16-bit software programs with the MS-DOS operating system. Oasis and CP/M-86 are also available.

Another new SKS entry in the portable microcomputer field, the SKS 252 Pico, packs the power of the Nano in a smaller (also leather-encased) package. It has the same type of keyboard, 8/16-bit CPU, 80 to 128 K of RAM, plus other options (except the Winchester) and the same CRT as the Nano series, but incorporates two $3\frac{1}{2}$″ 200K microfloppies instead of $5\frac{1}{4}$″ minifloppy drives. The Pico is designed for those who want maximum power in the smallest, lightest-weight package possible. The under-22-pound weight and carry-on luggage dimensions make the Pico portable extremely attractive to anyone needing maximum portability in a full-featured microcomputer.

Prices: 2502 Nano, $2,495; dual 8/16-bit Nano portable with the 80186 processor, $3,295.

SKS Computers, Inc., *4091 Leap Rd., Hilliard, OH 43026; (614) 876-8668.*

CIRCLE 322 ON READER SERVICE CARD

Local area network package

NET 8-16, a local area network strategy that allows users of CompuPro's multiuser microcomputer system to substantially increase both the maximum number of active workstations and their effective storage capacity has been introduced by Gifford Computer Systems. Prior to this development, only single-user workstations could support CP/M applications programs within a local area network.

NET 8-16 provides each user with features such as record locking, password protection, time stamping, automatic backup, queues, and multitasking commands. In addition, any combination of 8-bit and 16-bit programs can be supported simultaneously in a network with up to 64 terminals.

The NET 8-16 networking package incorporates a network controller board and proprietary software from Gifford Computer Systems that permits one or more "requester" multiuser systems to be linked by coaxial cables to "server" systems. Every terminal on each requester system has transparent access to the mass storage devices of the server system and to its own disks, and can be linked by modems to other networks or to external computers.

Networks can be in the form of a star (many requesters linked to a single server in a star configuration), a chain, or, in more complex patterns involving several servers, depending on the required applications. Every link has a controller board at its end and is connected by coaxial cable over distances of up to a few thousand feet.

Base price for the NET 8-16 local area network package for CompuPro systems is $1,995, which includes two controller boards and software.

Gifford Computer Systems, *1922 Republic Ave., San Leandro, CA 94577; (415) 895-0798.*

CIRCLE 323 ON READER SERVICE CARD

$5\frac{1}{4}$″ Winchester disk controller uses "burst mode" DMA interface

An intelligent $5\frac{1}{4}$″ Winchester disk controller capable of accessing a full 16 MB address space by using "burst mode" DMA technology has been introduced by CompuPro.

The new Disk 3 controller, compatible with the IEEE 696/S-100 bus standard, incorporates a local processor that enables CompuPro System 8/16 users to request transfers of

Microsystems Mart

New Products
continued . . .

large data blocks between disk and memory with a single command specifying source, destination, and length. The channel selector can then perform all required seeking, reading, and writing without external intervention.

Use of the high-speed DMA approach minimizes host processor overhead associated with disk operations while providing a table-oriented command structure.

The Disk 3 interfaces to as many as four ST-506 series-compatible drives or four Shugart SA1100 series drives, any of which may have different capacities, formats, and seeking characteristics. Full compatibility is offered for all CompuPro 8/16-based multiuser, multitasking systems, with support for multiple DMA devices conforming to the IEEE 696/S-100 bus standard.

Price: $795. Certified System Component (CSC) version, $895. OEM discounts provided.

CompuPro, *3506 Breakwater Court, Hayward, CA 94545.*

CIRCLE 331 ON READER SERVICE CARD

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

ULTIMENU FOR DOS simplifies IBM's, PC-DOS to meet your needs. $24.95. DB/DC Software Associates, P.O. Box 4695, Manchester, NH 03108.

COMPUTER EQUIPMENT/SUPPLIES

COMPUTER PAPER—Save $$$: Lowest single carton prices. Free samples. Bob's charts, 405 East Third, Long Beach, CA 90802.

FREE! Computer a Supplies Catalog—low prices—satisfaction guaranteed—DATA SYSTEMS, Box 99, Fern Park, FL 32730. (305) 788-2145.

PUBLICATIONS

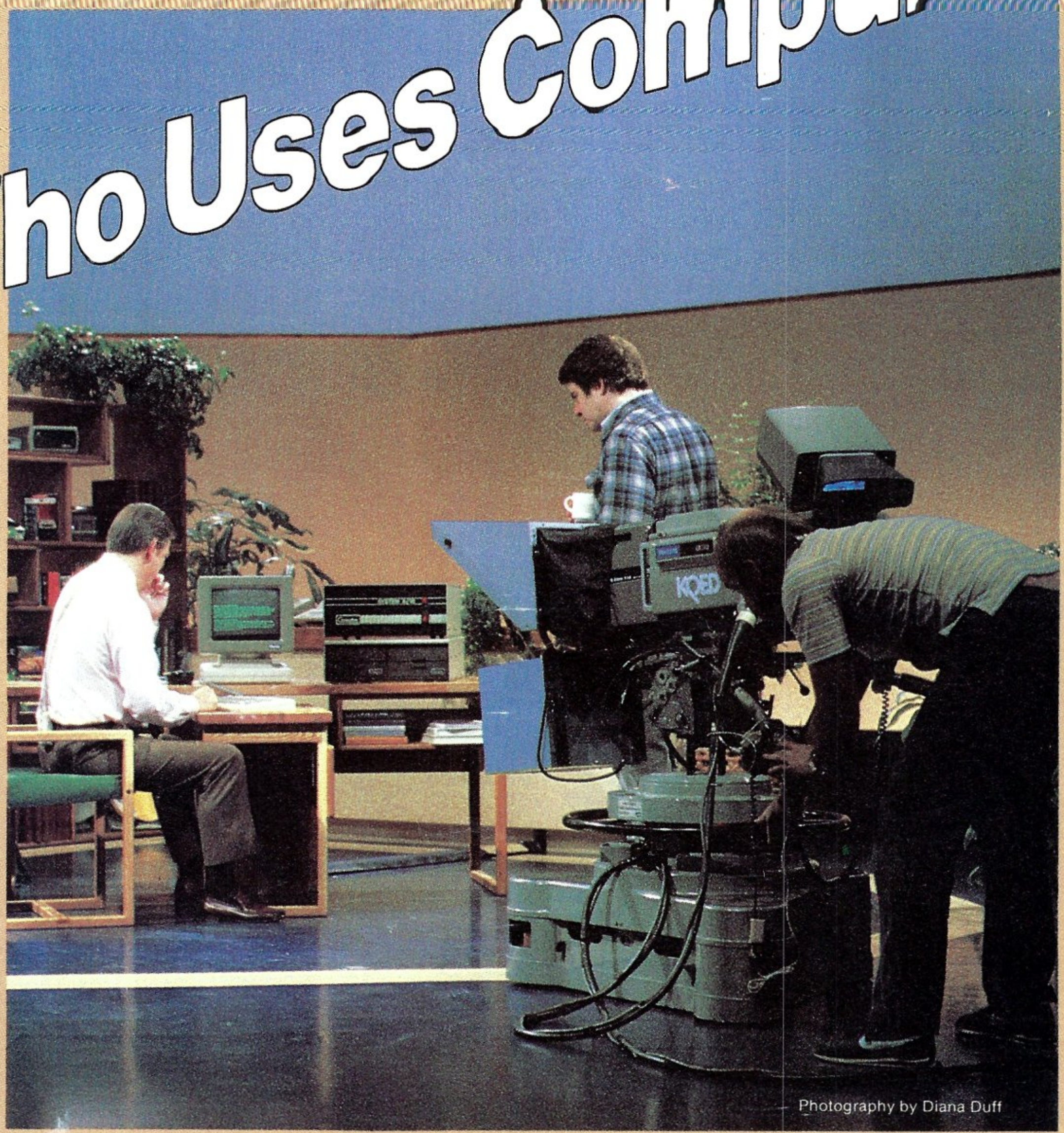
WHOLESALE Prices to Computer Club Members & Industry! Southern Calif. customers only. Our customers include authors of articles in this magazine. Computers, Disk Drives, Floppy Disks, Modems, Monitors, Paper, Printers, Software, Terminals. Visit our demo rooms! PATIO COMPUTER SALES CO., N. Hollywood. (213) 762-0020.

ALARMS

PROTECT MICROSYSTEMS AND ELECTRONICS! Free catalog of systems for SCHOOLS, OFFICES, HOMES. SGM Corporation, 6 West Main, Bound Brook, NJ 08805. (201) 469-8585.

ADVERTISERS INDEX

Who Uses CompuPro?



Photography by Diana Duff

San Francisco's public TV station, KQED, employs a **CompuPro** system on the set of a computer information series called **"Bits, Bytes and Buzzwords."** But it's much more than a prop.

Behind the scenes, a multi-user **CompuPro** system supplies 25 kinds of information on more than 4,000 TV programs, and formats this information for weekly schedules, second-by-second engineers' logs, newspaper listings, publicity releases, and more. Soon the hard-disk **CompuPro** system will inventory more than 2,000 videotapes — all of this on top of office management functions.

Despite continual and varied activity by five users, the **CompuPro** system has never failed. Janis Blackschleger, TV division administrator, said, "It's been so reliable that we have taken it for granted."

Organizing air time or your time, **CompuPro** delivers performance, quality and reliability. For business, scientific and industrial computing solutions, contact your **Full Service CompuPro System Center** today; call **(415) 786-0909**, ext. 206 for location.

KQED's **CompuPro** system was integrated by Gifford Computers of San Leandro.