the CP/M* and S-100 user's journal

$2.00
U.S.A.

MICROSYSTEMS™

JUL/AUG 1981

VOL.2/NO.4

16-BIT MICROCOMPUTER SYSTEMS

See Pages 18-53

Also in this Issue

Input Queuing on North Star by Robert Armstrong.................................. 54
Variable Speed Automatic Slow Step by Joseph Long.............................. 58
The Televideo 920-C Terminal by Glenn A. Hart.......................................60
An S-100 Clock/Calendar Circuit by Fred Deadrick.................................64

and more

Complete Table of Contents on Page 3

MORROW DESIGNS

Cost effective answers to floppy disk problems.

DMA answers. Standard, accumulator transfer floppy disk controllers can stall your microcomputer system's CPU for as long as 160 milliseconds. Just to access and transfer a sector of data to main memory. If CPU processing speed and system performance are critical, you need something better. That's where Morrow Designs' new intelligent Disk Jockey DMA™ controller comes in. This new breed of perpherial handles both 5¼" and 8" drives and can read almost any format in existence. Speed? Your CPU runs at full tilt while the DMA controller seeks and gathers a sector of data. How? Information transfers to and from main memory occur as "cycle steals" from the system bus. And the missing memory cycles are transparent to the CPU totalling only two milliseconds instead of the usual 80.

Build a buffer. Give the Disk Jockey DMA a little extra memory and your system's performance gets even better. The extra storage is used as a track buffer. So, whenever a sector on a new track is read, the track buffer is automatically filled with the other sectors on the track. The result? Additional data on the track is immediately available. Without the 80 millisecond rotational latency normally encountered. And similar efficiencies occur with disk writes. Sectors in the track buffer are conditionally written on the disk only if a new track is accessed. And only one revolution of the disk is required.

Now, prices.* All systems listed are supplied with a floppy disk drive, Disk Jockey Direct Memory Access Controller, CP/M** 2.2 operating system, power supply, fan, cables, cabinet and Microsoft 5.2 BASIC.

Even lower prices. Morrow Designs' floppy disk systems with Disk Jockey™ 2D controllers offer the same high quality at even lower prices. Now without memory mapping!

Fast answers. See Morrow Designs' family of cost effective floppy disk systems at your computer dealer. For a somewhat more leisurely answer, circle the reader service number below. For immediate answers, phone us at (415) 524-2101.

LOOK TO MORROW FOR ANSWERS.

5¼ Inch Systems (formatted data)	
400,000 Bytes (1) Dual Sided 48 TPI Drive	$1110
800,000 Bytes (2) Dual Sided 48 TPI Drive	$1899
800,000 Bytes (1) Dual Sided 96 TPI Drive	$1499
1,600,000 Bytes (2) Dual Sided 96 TPI Drive	$2599

OEM pricing available

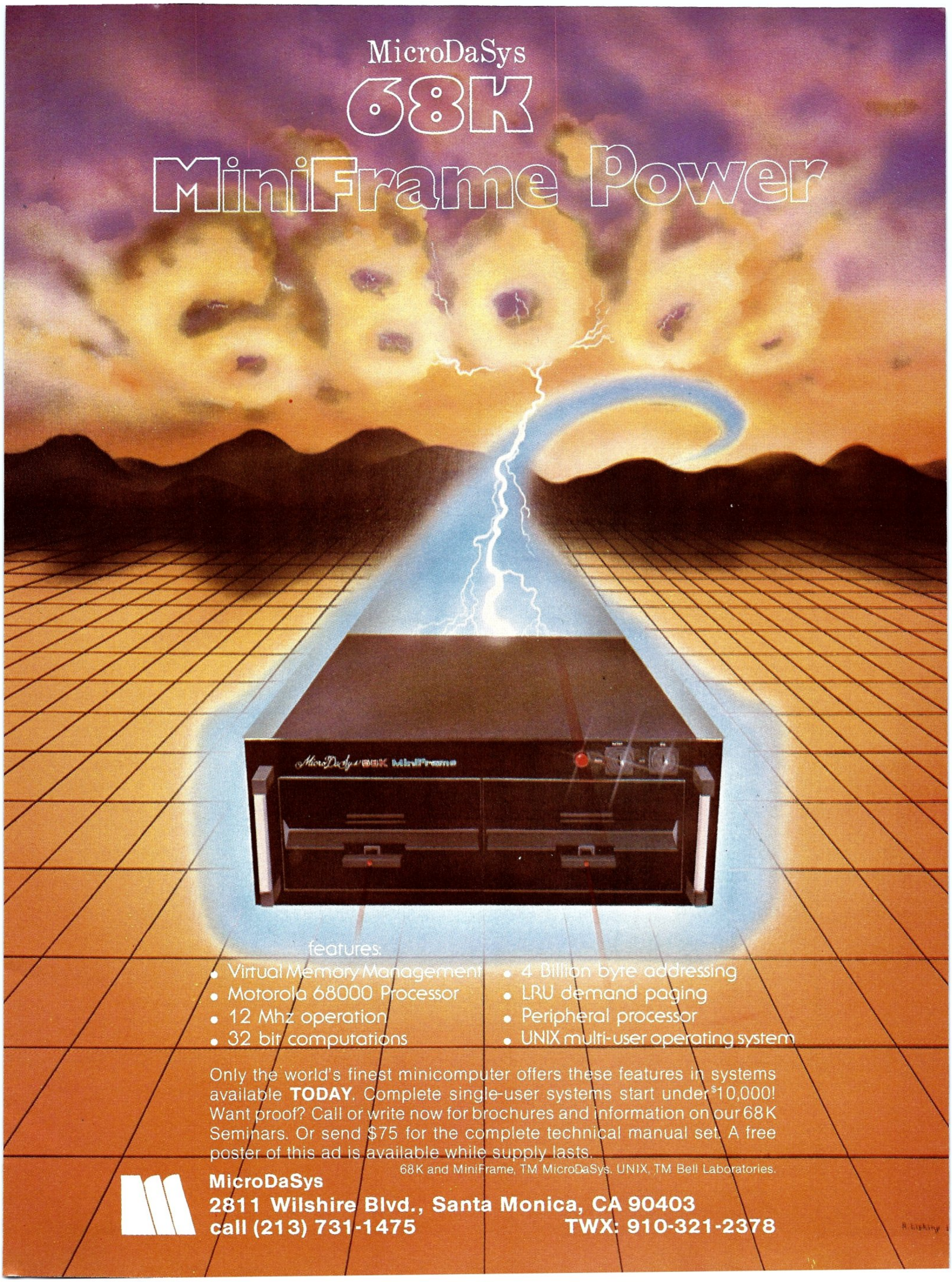
8 Inch Systems (formatted data)	
500,000 Bytes (1) Single Sided Drive	$1199
1,000,000 Bytes (2) Single Sided Drive	$1975
1,000,000 Bytes (1) Dual Sided Drive	$1595
2,200,000 Bytes (2) Dual Sided Drive	$2675

OEM pricing available

*Prices subject to change without notice.
Disk Jockey DMA and Disk Jockey 2D are trademarks of Morrow Designs
**CP/M is a trademark of Digital Research.

MORROW DESIGNS

5221 Central Avenue, Richmond, CA 94804
(415) 524-2101


MicroDaSys
68K
MiniFrame Power
MicroDaSys 68K MiniFrame
features:
• Virtual Memory Management
• Motorola 68000 Processor
• 12 Mhz operation
• 32 bit computations
• 4 Billion byte addressing
• LRU demand paging
• Peripheral processor
• UNIX multi-user operating system
Only the world's finest minicomputer offers these features in systems available TODAY. Complete single-user systems start under $10,000! Want proof? Call or write now for brochures and information on our 68K Seminars. Or send $75 for the complete technical manual set. A free poster of this ad is available while supply lasts.
68K and MiniFrame, TM MicroDaSys. UNIX, TM Bell Laboratories.
MicroDaSys
2811 Wilshire Blvd., Santa Monica, CA 90403
call (213) 731-1475
TWX: 910-321-2378

the CP/M* and S-100 user's journal

MICROSYSTEMS™

Volume 2, Number 4 July/August 1981

Staff

Editorial correspondence is welcomed and should be sent to: Sol Libes, c/o MICROSYSTEMS, Box 1192, Mountainside, NJ 07092. Phone: (201) 522-9347.

For information on commercial advertising, write to: MICROSYSTEMS, 39 East Hanover Ave., Morris Plains, NJ 07950, or call Claudette Moore at (201) 267-4558.

*CP/M is a registered trademark of Digital Research.

In This Issue

16-Bit System Reviews

The TEC-86 16-Bit Computer System....18
Chris Terry

Seattle Computer Products' 8086 System....22
Bill Machrone

Alpha Micro System Revisited....26
Henry Kee

The Godbout Dual Processor & CP/M-86....30
Bruce Ratoff

16-Bit Disk Operating Systems....50
Sol Libes

Input Queuing For North Star....54
Robert Armstrong

Variable Speed Automatic Slow Step....58
Joseph Long

The Televideo 920-C....60
Glenn Hart

An 8086/8088 Reference Book....62
Chris Terry

An S-100 Clock/Calendar Circuit....64
Fred J. Deadrick

Departments

Editors Page....4
Letters to the Editor....6
News & Views....10
The CP/M Bus....14
Software Directory....68
New Products....74
Advertiser Index....80

MICROSYSTEMS (ISSN #0199-7955) is published bi-monthly by Creative Computing, 39 East Hanover Ave., Morris Plains, NJ 07950. Second class postage paid at Morris Plains, NJ 07950 and additional mailing offices.

Subscriptions are $10 a year (6 issues USA). To subscribe, call our toll free number: (800) 631-8112, or (201) 540-0445 in New Jersey.

POSTMASTER: Send address changes to: MICROSYSTEMS, P.O. Box 789-M, Morristown, NJ 07960.

by Sol Libes

EDITOR'S PAGE

This month we are highlighting 16-bit microcomputer systems. There has been a great deal of hullaballo about 16-bit systems of late, particularly the 8086, Z8000 and 68000. But we shouldn't forget the 16-bit systems that have been operating on the S-100 bus for several years. Marinchip has had the TI-9900 and Alpha-Micro has had their LSI-11-like S-100 CPU's on the market for well over three years. Seattle Computer Products and TecMar have had their 8086 S-100 CPU's out for over two years.

These systems have met with a moderate success from systems houses. However, by comparison to 8-bit micros, their acceptance has been dismally disappointing. The lack of greater acceptance, as I see it, is due to two basic causes. First, there is a lack of software for these systems, and secondly these systems are significantly more expensive than 8-bit systems. And let's face it, 8-bit systems meet the needs of most personal computer users very nicely.

There is no doubt that the new breed of 16-bit microprocessors have a lot to offer in multi-user systems; hence we can expect the 16-bitters to dominate this market. Also, as new applications packages are introduced which capitalize on the greater power of the 16-bit designs, and prices drop, we can also expect to see some single-users switch from 8-bit to 16-bit machines.

The April 1, 1981 issue of *EDN* magazine (published by Cahners Publishing Co., 221 Columbus Ave., Boston, MA 02116, $30/yr domestic) contained the first extensive benchmark testing of what are currently the four most popular 16-bit microprocessors: the DEC LSI-11/23, the Intel 8086, the Motorola 68000 and the Zilog Z8000. I highly recommend the article to all readers interested in 16-bit micros. The article is 41 pages long and contains all the source code programs for each test, as well as some interesting insights on the comparative features of these processors. I will very briefly summarize the data presented in the article but, again, I strongly recommend reading the article (single copy is $2 domestic).

Benchmark tests are complex and difficult to carry through without prejudice. *EDN* had each manufacturer conduct seven tests from a group of tests designed by Carnegie-Mellon University, closely supervising to insure a minimum of prejudice. All the source code was published, so readers can check the results on their own systems. I feel that they've done an excellent job.

It is apparent from the test results, which I've summarized at the end of this column, that each processor has certain strong points and drawbacks, advantages and liabilities.

The tests were conducted by the manufacturers, using the maximum clock speeds available at the time of the tests (late 1980). The following are the clock speeds used (MHz):

LSI-11/23	3.33
8086	10.00
68000	10.00
Z8000	6.00

The benchmark tests use common algorithms that appear frequently in programs. *EDN* excluded the CM tests dealing with floating point math and virtual-memory handling because most of the micros didn't directly support such operations. The following are the benchmark tests conducted:

A: I/O Interrupt Kernal
B: I/O Kernal with FIFO
E: Character-string search
F: Bit set, reset test
H: Linked-list insertion
I: Quicksort
K: Bit-Matrix Transposition

The benchmark results are shown in Table 1. The number of bytes are represented on the left of the slash, the execution time in microseconds appears on the right. Results of test H and I for the 8086 and LSI-11/23 were unavailable at publication date. ■

Benchmark Tests — 16-Bit Microprocessors

	A	B	E	F	H	I	K
LSI-11/23	20/114	86/1196	76/996	70/799	138/592	-/-	152/1517
8086	55/126	85/348	70/193	46/122	94/-	347/115,669	88/820
68000	24/33	118/390	44/244	36/70	106/153	266/33,527	74/368
Z8000	18/42	106/436	66/237	44/123	96/237	386/115,500	110/646

SOFTWARE
SOFTWARE
DIGITAL MARKETING
DIGITAL MARKETING™

July 1981

DIGITAL MARKETING • 2670 CHERRY LANE • WALNUT CREEK • CALIFORNIA 94596 • (415) 938-2880

MILESTONE™ - $295. Manual alone - $30.

"Critical path" network analysis program for scheduling manpower, dollars and time to maximize productivity.

An interactive project management program that runs under CP/M and can relate together different skills, hourly pay rates and projects to maximize efficiency. MILESTONE could be used to track paper flow, build a computer, check a salesman's performance, or build a bridge. MILESTONE can be used by executives, engineers, managers, and small businessmen to:

- Find the critical tasks which can't be delayed.
- Discover which tasks are not time critical.
- See how manpower and expenses vary versus time.
- Investigate tradeoffs between manpower, dollars and time.
- Give plans to others using a printed project schedule.
- Change details and immediately see the results on screen.

Requires 54K RAM and CP/M. Specify Z80, 8080 or CDOS. Also available for Apple Pascal, UCSD Pascal or CP/M-86 operating systems.

Formats: 8, NS, MP, CDOS, SB, TRS2, APPL

PASCAL/M™. $225. Manual alone - $20.

CP/M™ compatible language for 8080/Z80 CPUs, supports full Jensen & Wirth plus 45 extensions to Standard Pascal including Random access files, 40 segment procedures & 16 bit BCD real type. **NOW INCLUDES** symbolic debugger which features trapping on stores, examining and changing varibles and tracing of program execution.

Requires CP/M 2.2 & 56K RAM. Formats: 8, NS, APPL, TRS2

PASCAL/M for 8086/88 - $270.

Manual alone - $20.

All features of Pascal/M for the 8086 and 8088 processors running under the 8086/88 version of CP/M.

Requires CP/M-86™ & 128K RAM. Format: 8

TRANS 86™ - $125. Manual alone - $20.

8086/88 Translator for existing 8080/Z80 programs. The new source code can be easily edited and assembled using ACT II to produce hex code which can be executed by 8086/88. Emphasizes the extensions and features available in the 8086/88.

Requires CP/M & 32K RAM. Formats: 8, NS, APPL

PLAN80™ - $295. Manual alone - $30.

A financial modeling system that's easy to use yet powerful enough to replace most timesharing applications.

Lets you tackle any numeric problem that can be defined in a worksheet format. It performs complex calculations quickly and precisely and lets you examine "What if?" questions so you can evaluate more planning alternatives in greater detail.

With PLAN80 you get more than your calculated results . . . you know **how** you get them, because you define rows and columns with familiar names such as UNITS, PRICE and JANUARY and express calculations in terms such as SALES=UNITS*PRICE. It's easy to review your assumptions and methods with people who have never seen PLAN80.

At any point in the PLAN80 model you may display or print results on your screen, printer or disk, save all or part of the results for use by another model, or play "What if?" by inputting new values, recalculating and displaying or printing results.

Best of all, you can incorporate PLAN80 results into any report that requires a financial model — using your word processor — to create professional results.

Requires 56K RAM and CP/M. Also available for CP/M-86. Specify Z80, 8080 or CDOS. Formats: 8, CDOS, NS, MP, SB

DATEBOOK II™ - $295. Manual alone $25.

- **NOW SCHEDULES APPOINTMENTS FOR UP TO 27 DIFFERENT PEOPLE**
- **IMPROVED FILE STRUCTURE ALLOWS APPOINTMENTS UP TO ONE YEAR**
- Replaces your office appointment calendar
- Searches for openings that fit time of day, day of week & day of year contraints
- Appointments made, modified or cancelled by a few key strokes
- Copies of day's appointments can be quickly printed

Requires 54K RAM and CP/M. Specify Z80, 8080 or CDOS. Also available for Apple Pascal, UCSD Pascal or CP/M-86 operating systems.
Formats: 8, NS, MP, CDOS, SB, APPL, TRS2

ACT I™ - $125. Manual alone - $15.

CP/M compatible macro assembler for Z80, 8080/85, 6502 & 6800.

One assembler that supports all major 8 bit micros. ACT features include full macro capabilities, comprehensive pseudo-ops, link-file structures, cross reference map, and algebraic expression processor. Requires 24K RAM & CP/M.

ACT II - $175. Manual alone - $20.

CP/M 2.x compatible cross assembler for 8086/88

ACT III - $125. Manual alone - $20.

CP/M 2.x compatible cross assembler for 6809.

ACT I and ACT II together - $225.

Formats: 8, NS, CDOS, MP/M, TRS2, APPL

SPELLGUARD™ - $295. Manual alone - $20.

20,000+ word dictionary containing commonly used words that find spelling & typographical errors in text files. Allows review of mis-matched words & speedy search routine. Proofs at 10,000 words/min.
Requires CP/M, 48K RAM & Magic Wand, WordStar™ or Spellbinder™
Formats: 8, NS. MP. SB, TRS2, CDOS

SPELLBINDER™ - $395. Manual alone - $50.

Full feature word processing system with Office Management capabilities. Its special features include ease-of-use by office personnel, flexible print formatting & output, and a powerful macro capability which allows features to be added for the unique requirements of each user. Mail list macro is included for mail merge with form letters.

Requires CP/M & 32K RAM. Formats: 8, NS, MP, CDOS, SB, APPL

dBASE II™ - $695. Manual alone - $50.

Assembly language relational data base management system. Can be used interactively with English-like commands or program it using a command file. Can read your ASCII files and add the data to its own data base. Report generator and user-definable full screen operations allows use of your existing forms.
Requires 48K RAM and CP/M. Formats: 8, NS, MP, SB, APPL, TRS2

SUPERCALC™ - $295. Manual alone - $25.

Allows a non-programmer to manipulate business data in a variety of forecasting and accounting applications. Combines the interactive nature of an electronic spreadsheet with the power and convenience of a simple simulation language. Video display can be scrolled over entire worksheet using cursor controls. Symbolic vector references eliminate repetitive low-level data manipulation commands. Menu driven "Help" command may be invoked at any time.

Requires CP/M. Formats: 8, NS, MP, SB, APPL, TRS2

MCALL™ - $85.

Communications program designed to drive an acoustic coupler. Features include:

- Time sharing Terminal emulation
- Disk file transfer between CP/M computer & Time Sharing Computer in either direction
- Disk file transfer between two CP/M computers with error detection and correction

AMCALL™ - $95.

Auto-answer, auto dial version of MCALL currently supports IDS 88-modem & Potomac Micro-Magic MM-103 boards.

Requires CP/M. Formats: 8, NS, MP, SB, CDOS

MAGIC MENU™ - $75.

With Spellguard option - $95.

Turns Magic Wand into a turnkey system. Allows move from EDIT to PRINT, backup of files or disks, system status, etc. from menu without returning to CP/M 2.x
Requires 56K RAM, CP/M and Magic Wand.

SPELL MENU™ - $95. Spellbinder version of Magic Menu.

Formats: 8, NS, MP, SB, TRS2, APPL

OEMs and SOFTWARE DEVELOPERS

We can write a custom Menu for your software packages. Call us for details.

SUPERDOS™ - $129.

Upgrade of CP/M 2.2 for Superbrain. Includes ADM/31, Hazeltine, or Superbrain Terminal emulation mode. Other new features include 132 character keyboard buffer, repeat on all keys, key click, user programmable numeric keypad, 30% disk read/write improvement, real time clock, baud rates to 19.2K on RS232 ports, printer handshake modes, 4 new utilities, and 4 fixes.
Requires Superbrain 3.0. Format: SB

MORE SOFTWARE

	PRODUCT/MANUAL ALONE
TEXTWRITER III	$125./$25.
PEARL LEVEL 1	$130./$35.
PEARL LEVEL 2	$350./$35.
PEARL LEVEL 3	$650./$50.
CBASIC2	$120./$25.
ULTRASORT-II	$175./$20.
FABS	$175./$20.
CBASIC/86	$325./$30.
SELECTOR IV	$550./$35.
GLECTOR for SELECTOR IV	$450./$35.
S-BASIC	$295./$35.
baZic	$150./$25.

DEALER INQUIRIES INVITED

MICROSTAT™ VERS. 2.0 - $295. Manual alone - $25.

Now includes stepwise and more flexible file structure. Code size smaller (= more data area). Also a data management subsystem for editing, sorting, ranking, lagging, data file transfers PLUS eleven data transformations (e.g. linear, reciprocal, exponential, etc.)

- Frequency distributions, 8 probability distributions
- Multiple regression, correction analysis
- Time series
- Crosstabs/Chi-Square, non-parametrics
- Factorials, permutations, combinations
- Scatterplots
- ANOVA (one and two-way)

Requires 48K RAM, NorthStar Basic or CP/M & CBASIC2 or Microsoft Basic 80.

Formats: 8, NS, MP, SB, TRS2, CDOS, APPL

FORMAT CODES: 8 (8″ single density IBM soft-sectored) NS (NorthStar DD) MP (Micropolis Mod II/Vector MZ) SB (Superbrain 3.0) CDOS (8″ Cromemco CDOS) TRS2 (TRS-80 Mod II) APPL (Apple II)

Pearl trademark CPU Int'l Pascal/M, ACT, Trans 86, SuperCalc trademarks Sorcim, CBASIC2, CBASIC/86 trademarks Compiler Systems, Datebook II, Milestone, Textwriter III trademarks Organic Software Spellguard trademark ISA CP/M, CP/M-86 trademarks Digital Research Superbrain trademark Intertec Data Systems Microstat trademark Ecosoft S-Basic trademark Topaz Programming Spellbinder trademark Lexisoft MCALL, AMCALL trademarks Micro-Call Services Selector IV, Selector/86, GLector trademarks Micro-Ap Prices subject to change without notice For shipping add $7. per package California residents add sales tax Terms: Prepaid check, M/C, VISA or in US COD (UPS) Prices quoted do not include dealer installation & training

DEALER INQUIRIES INVITED OUTSIDE CALIFORNIA CALL 501/442-0864 INSIDE CALIFORNIA CALL 415/938-2883

LETTERS TO THE EDITOR

Dear Editor:

My first exposure to your magazine was with the January/February issue, and the tip on the CP/M null file for returning to files in RAM was worth the price of the issue.

The evaluation article on double density controllers did your readers an injustice by neglecting the Intersystems product. The board found to be the fastest by the author would have come in second if this one had been included. Before I received my Intersystems FDC-2, I was confused by the advice people gave me about the inherent unreliability of double density and the importance of using this or that brand of premium disk. I have been running double-sided double density exclusively for over a year an a half a dozen different brands of disks, and in that time I have seen one read error message—which I provoked by ignoring the WAIT message while Wordstar was shuffling files. My worst crash required that I hit the reset button after feeding the Intersystems Pascal compiler with a corrupt ASCII file.

While this DMA controller may be a little more expensive than some other boards, the money I saved by using 64K of dynamic vs. static memory was more than the price of the controller.

Please continue with this type of comparitive evaluation whenever you can.

Aubrey Soper, III
Virginia Beach, VA

Dear Editor:

Steven Leibson was much too kind to Rodney Zaks in his review of the so-called *CP/M Handbook* (Mar/Apr 1981). A more realistic appraisal can be found in Jim Hendrix's letter to the Editor in the March 1981 issue of *Dr. Dobb's Journal*.

Oscar Goldman
Professor, Mathematics Department
University of Pennsylvania
Philadelphia, PA

Dear Editor:

Did North Star Topics get left out of the CP/M and S-100 user's journal permanently? I thought that it would provide solutions to problems I didn't know existed. I will now need to find a friendly users group. Could you perhaps rotate North Star Topics with other columns on new, improved, or compatible DOS and languages?

Yours is a great magazine with a wide variety of S-100 products covered. It complements an S-100 (maybe not IEEE compatible) computer well.

One of your new products is a "compliance H," what are other compliances? Do most CP/M programs come in some standard eight inch format?

Ron Masaoka
Gardena, CA

The editor replies:

No, North Star Topics is not out of Microsystems. *Regretfully, Randy Reitz has been extremely busy of late on his bread-winning job. He is working on another column which you can expect to see in print soon. Also, we have several other North Star articles scheduled over the next few issues.*

Regarding compliance with IEEE S-100 specs, we will have to wait for the final version of the standard. I expect this to be approved soon. Microsystems *will carry the full details as soon as they are available.*

Regarding CP/M disk format, note that the CP/M and SIG/M user group libraries are currently available in the following formats:

8" single density
5" North Star single or double density
Cromemco 5" and 8" single or double density
Micropolis 5"
DEC RSX-11M
Le Croy 8" Single density
Apple 5"
TRS-80 Model-I 5"
TRS-80 Model-II 8"

The SIG/M Group (Box 97, Iselin, NJ 08830) furnishes these disks to other clubs at $4/disk plus $2 shipping (first disk $1/disk thereafter—U.S. funds only). We hope to publish a list of all clubs who have these disks for copying.

Dear Editor:

I enjoyed Chris Terry's article, "The CP/M Connection" in the July/Aug and Sep/Oct issues of *Microsystems*, but I really must point out that Chris is mistaken on a technical point he made several times in Part 2.

In describing the allocation bitmap, he says "This map is read in when the drive is logged in, ... and is written back to disk each time a file on that disk is closed." It just isn't true! If Chris would try to show me where that map is stored, he would realize that it is not stored on the disk at all but calculated from the file allocations in the directory and kept as a bit map only in memory. When CP/M "logs in" a disk, the directory is scanned and the map is created by checking off all extents that are currently in use by a file. This is the whole purpose of the login. This map must be correct for any disk write operation that needs another block allocated, so all CP/M disk write and update operations update this map. That is why CP/M does not like you to change diskettes without "rebooting" the system.

I enjoy *Microsystems*. Please keep up the good technical articles. But maybe you need a technical wizard to proofread some of their content.

David Mitton
New England Computer Society
CP/M Users Group Chairman
Cambridge, MA

Response from Chris Terry:

Dave is absolutely right. I must have been dreaming when I wrote that—dreaming about the mapping bytes in the FCB & directory entries!

Letters, cont'd...

Dear Editor:

I became a charter subscriber in timely fashion when I bought a Cromemco Z2, to learn about the S-100 Bus. Since then I have devoured each issue, forcing *Electronics* (which comes four times more often) into a wait state whenever contention arises. The "Confuzer," as I call it, set me back a bit so I could afford only a pair of single-sided, single-density 5 1/4" minifloppy disk drives. The problem that many of us late bloomers now face is how to upgrade to 8" floppies without being left holding the bag when single/single 8" drives become defunct. The main question I would pose to you, your staff, and to the general readership is: "How long will CP/M-compatible software be available on SS/SD disks, and how quickly will the many software 'cottages' make their goodies available on DS/DD disks?" Bob Weidemann's article on double-density in the Jan/Feb 1981 issue of *Microsystems* seems to indicate that a few years will go by before such disks can be used as transfer media. But how many? And what about DS/SD? My disk controller is supposed to be able to handle either single- or double-sided SD drives using Cromemco's standard, which even they admit is different from most others. How many readers are faced with this dilemma?

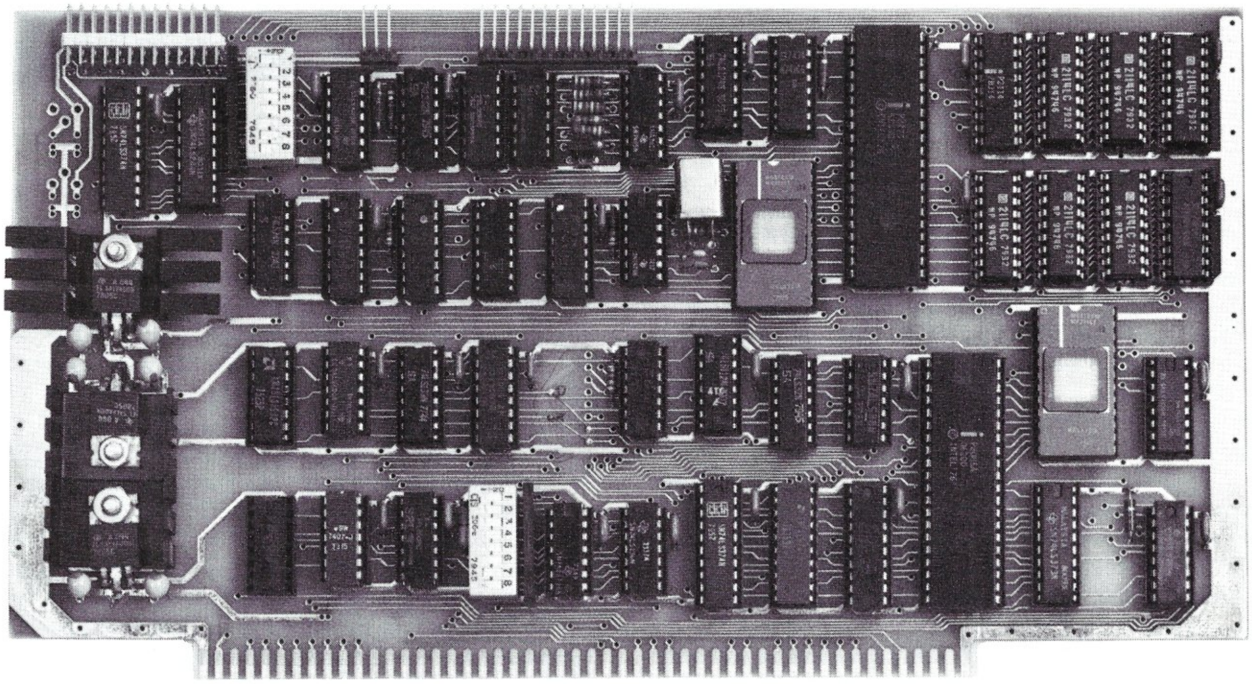
Mind you, I'm not against progress, but it is worth considering whether we S-100 junkies should bring back the tape cassette for software exchange and disk backup. The biggest plus, of course, is that Phillips won the battle over physical dimensions and recording format for this medium some time back.

In the April issue of *Interface Age* is an excellent article entitled "Proposed Cassette Data Storage Format Standard" by Lorin S. Mohler. I don't know if anything came of it, but would like to hear from readers who have, or who have knowledge of any de facto tape standard. Of critical importance in a standard is the method of encoding digital data as analog signals, the baud rate (!) and the resulting reliability of the whole package for sending a set of CP/M files from here to there. With true hindsight and a different purpose for tape in mind, I could suggest a few improvements to Mohler's proposal, which are meant to improve deliverability:

1. Rather than a CRC alone, each physical record (representing a CP/M disk sector) should use an error-correction code, such as the Hamming code.
2. Similarly, tape header information could be written redundantly using a simple 2-for-1 byte minimum-distance code I've discovered, allowing immediate error trapping and recovery.
3. Additional information, not to be included in the disk or file resulting from a transfer, could be included in the headed data written to a transmittal or archive tape.
4. Consideration should be given to those who wish to be compatible with the standard, but do not want or need to take advantage of embellishments.

Walter P. Davis
107 3rd Place
Brooklyn, NY 11231

INTELLIGENT VIDEO I/O FOR S-100 BUS



VIO-X

The VIO-X I/O Interface for the S-100 bus provides features equal to most intelligent terminals both efficiently and economically. It allows the use of standard keyboards and CRT monitors in conjunction with existing hardware and software. It will operate with no additional overhead in S-100 systems regardless of processor or system speed.

Through the use of the Intel 8275 CRT controller with an onboard 8085 processor and 4k memory, the VIO-X interface operates independently of the host system and communicates via two ports. The screen display rate is effectively 80,000 baud.

The VIO-X1 provides an 80 character by 24 line format using a 7 × 9 dot matrix to display the full upper and lower case ASCII alphanumeric 96 printable character set (including true descenders) with special characters for escape and control characters. An optional 2732 character generator is available which allows an alternate 7 × 9 contiguous graphics character set.

The VIO-X2 offers an 80 character by 25 line format using a 9 × 9 dot matrix allowing high-resolution characters to be used. This model also includes expanded firmware for block mode editing.

Both models support a full set of control characters and escape sequences, including controls for video attributes, cursor location and positioning, cursor toggle, light pen location, and scroll speed.

Video attributes provided by the 8275 in the VIO-X include:

- FLASH CHARACTER
- INVERSE CHARACTER
- UNDERLINE CHARACTER or
- ALTERNATE CHARACTER SET
- DIM CHARACTER

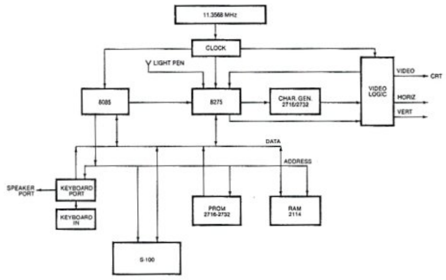
The above functions may be toggled together or separately.

The board may be addressed at any port pair in the S-100 host system. Status and data ports may be swapped if necessary. Inputs are provided for parallel keyboard and for light pen as well as an output for audio signalling. The interrupt structure is completely compatible with Digital Research's MP/M

FEATURES

- HIGH SPEED OPERATION
- PORT MAPPED S-100 INTERFACE
- FORWARD/REVERSE SCROLL or
- PROTECTED SCREEN FIELDS
- CONVERSATIONAL or BLOCK MODE
- INTERRUPT OPERATION
- CUSTOM CHARACTER SET
- CONTROL CHARACTERS
- ESCAPE CHARACTER COMMANDS
- INTELLIGENT TERMINAL EMULATION
- TWO PAGE SCREEN MEMORY

VIO-X1 80 × 24 7 × 9 A & T **$295.00**
Conversational & Limited Block Modes
VIO-X2 80 × 25 9 × 9 A & T **$345.00**
Conversational & Block Modes



VIO-X S-100 I/O INTERFACE

FULCRUM™
COMPUTER PRODUCTS

Distributed by:
WW COMPONENT SUPPLY INC. 1771 JUNCTION AVENUE • SAN JOSE, CA 95112 • (408) 295-7171

NEWS & VIEWS

Digital Research Reveals Future Plans

Gary Kildall revealed Digital Research's current projects and plans for the future at the CP/M User Group meeting held in April at the West Coast Computer Faire. Gary also reported that DR now has over 200,000 licensed CP/M users on more than 250 different types of systems.

First of all, CP/M Version 3 is in the works and may be released by the end of this year. It will add the following features: time & date, passwords, type ahead, file lockout, record lockout, test and write a record, a screen-oriented editor, much better documentation and (naturally) a smaller TPA.

Also due from DR this year are CP/M Version 2 and MP/M-86. Due in 1982 is XLT-86, an 8080-to-8086 translator, PL/I-86 (full subset-G, with 8087 math processor provisions) and CP/NET-86. DR also expects to have 32-bit software in 1983; I imagine this means that they intend to support the Intel iAPX-432 32-bit micro.

DR sees a future with CP/M, MP/M, CP/NET and MP/NET systems integrated into a sophisticated networking system that uses backplane bus, Ethernet, IEEE-488, RS-232 and high speed parallel communications links between servers, requestors and server/requestors. They see a VAX type host as the node in such a local networking system. It should be noted that DR already has a DEC-VAX machine running at their facility.

IEEE-696/S-100 Standard Status

I have been appointed secretary of the IEEE-696 Standard committee. Although the standard is essentially finalized, committee members and other interested S-100 component suppliers are being given one last opportunity to request changes before the standard is forwarded to the IEEE Standards Group for adoption. I hope to print the final addendum to the standard in the September/October issue of *Microsystems*. I also expect that the standard will be formally adopted by the IEEE early in 1982.

Most S-100 manufacturers have changed, or are in the process of changing, their products to comply with the standard. It is likely that by mid-1982 all S-100 products will be in conformance with the IEEE-696 standard. *Microsystems* will attempt, through product reviews, to insure that manufacturers comply with the standard. Although no standard ever completely guarantees compatibility, the frequent incompatibility problems that have plagued the S-100 area should soon be ancient history.

BDS-C & Amethyst User Group News

Bob Ward is the new coordinator of the BDS-C User Group (409 E. Kansas, Yates Center, KS 66783). Membership is now $10. The group has several disks of software available on 8" standard single density format. They also expect to be able to handle Heath H-89 and Micropolis 5-1/4" formats. Included in the library is Adventure in C, 6800 and 1802 assemblers and a new C complier. Disks are $8 domestic, $12 foreign.

Users of MINCE and SCRIBBLE text editor and formatter (AMETHYST) now have a user group. The main focus of the group is to provide coordination among users developing extensions to MINCE and SCRIBBLE. Membership is $6/yr. For more information write: Barry A. Dobyns, 1633 Royal Crest #1128, Austin, TX 78741, (512) 441-9466.

CP/M-UG & SIG/M Release New Disks

The CP/M-UG and SIG/M have released more volumes of public domain software. The CP/M-UG has released volume 49, containing Fortran material, and is expected to shortly release three more volumes. The disks can be obtained from CP/M-UG, 1651 Third Ave., New York, NY 10028, (212) 722-1700.

The SIG/M has released seven new disks bringing their total up to 25 volumes. The disks can be obtained from SIG/M, Box 97, Iselin, NJ 08830.

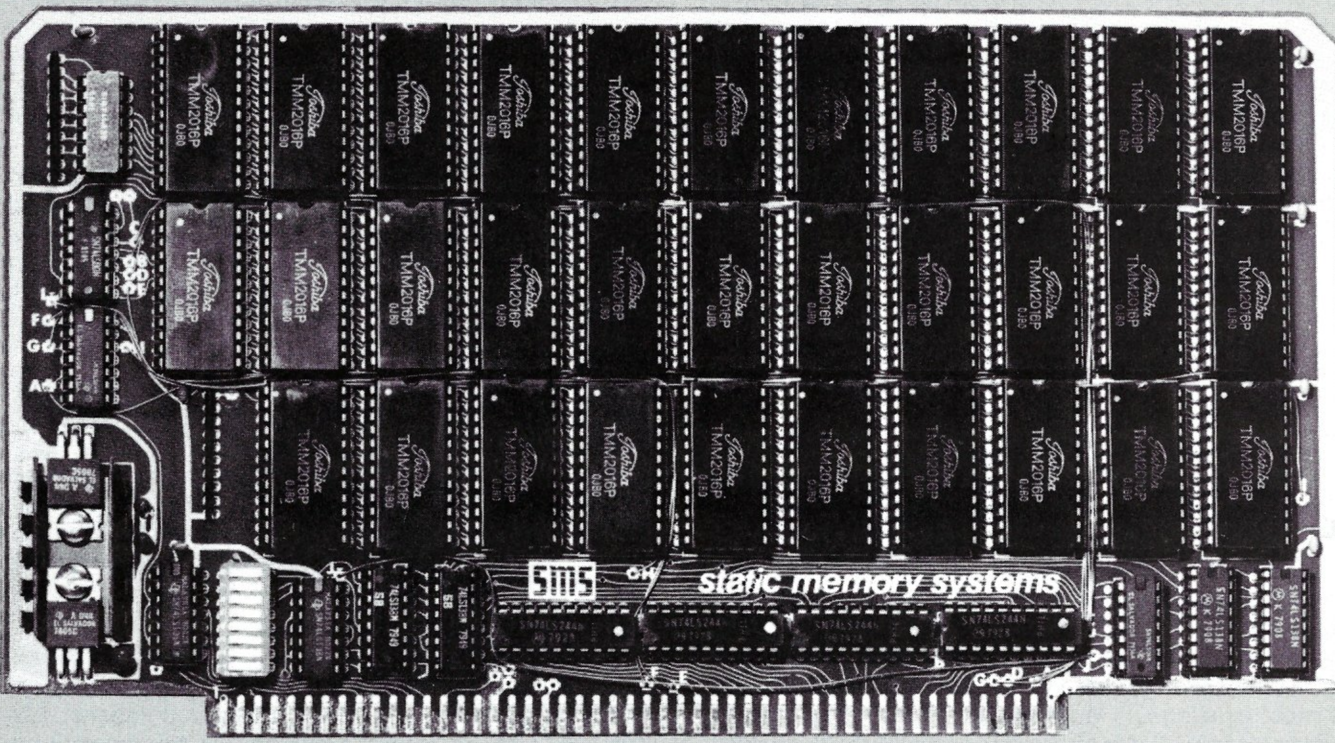
A 200 page printed catalog listing the contents of CP/M-UG volumes 1 through 49 and SIG/M volumes 1 through 18 is available for $10 domestic, $13 foreign, from NYACC (New York Amateur Computer Club), Box 106, Church Street Station, New York, NY 10008. NYACC can also furnish a listing for CP/M-UG and SIG/M local groups which furnish copies of these disks. Send a self-addressed, stamped envelope for this listing.

ADA Compiler Being Tested

Telsoftware Inc., of Sorrento Valley, CA (the company Dr. Ken Bowles, of UCSD Pascal fame, founded to develop an ADA compiler) reports that their ADA compiler is now at Beta test sites. The version released for test runs on Motorola 68000-based systems, and contains most, but not all, of the features of the DOD-ADA standard. The price for the compiler package is $2000.

According to certain reports, Western Digital, the Pascal Microengine supplier, had owned 20% of Telesoftware. However, in April WD withdrew and decided to develop its own ADA compiler. However, WD has retained a license for the Telesoftware ADA compiler.

THE LAST MEMORY



64K STATIC RAM/EPROM BOARD

At last a 64K *STATIC* memory board for S100 systems. But it's not just a 64K static RAM board, *EPROM's* can also be intermixed with RAM making it the only memory board needed for S100 systems. That's why we call it **THE LAST MEMORY.**

- *64K DENSITY*

THE LAST MEMORY uses the new 2016 byte-wide 16K static RAM to achieve a board density twice that possible with old 2114 static memories.

- *2716 EPROM COMPATIBLE*

A separate board is no longer required for EPROM's containing monitors, bootstrap loaders, etc.
2716 EPROM's can be inserted into the board without modification.

- *SIMPLE ADDRESS DECODING*

Where memory is required, just plug a RAM or EPROM in the corresponding socket. Empty memory sockets occupy no memory space, providing compatibility with memory mapped I/O devices.

- *EXTENDED ADDRESSING*

THE LAST MEMORY includes the IEEE S100 extended addresses. These are fully decoded allowing expansion to a full 16 megabyte system memory.

- *FAST*

The standard board allows 4 MHz operation.

- *LOW POWER*

Only one memory IC is ever active in byte-wide memory systems. The result is far less power consumption than older 16K static memory boards.

- *LOW COST*

Its best feature is the price:

	Kit	A&T
RAM-less Board	99.99	139.99
16K RAM	249.99	289.99
32K RAM	389.99	429.99
48K RAM	519.99	559.99
64K RAM	639.99	679.99

sms

static memory systems

15 So. Van Buren Ave. Suite 209
Freeport, Illinois 61032 (815) 235-8713

VISA

CDE DOES EXIST!

While we were at the West Coast Computer Faire a postal employee in Hollywood assumed we closed shop, since our P.O. Box wasn't emptied for a few days and began returning *all* of our mail. When we discovered this we were outraged! We always give our customers the best possible service. We are doing everything possible to correct the error. Anyone who had mail to us returned, please send it again. We will give all mail our prompt attention and are very sorry for any inconvenience that may have resulted.

Patrick Lajico
President, California Digital Engineering
P.O. Box 526
Hollywood, CA 90028

New DOS From BDS-C Author

Ed Ziemba and Leor Zolman, author of the very popular BDS-C compiler, have developed a new "UNIX-like" Disk Operating System for 8080/Z80 based systems called "MARC." It initially boots under CP/M. They claim that it includes the basic UNIX file system complete with users, groups, protections and the like, as well as much of the UNIX user interface and more. Further, they expect that the system will provide for the transparent running of most existing CP/M programs, as well as programs written for MARC. The expected price is $175, for another $75 you can have either BDS-C or the MINCE editor. We have received an advance copy of MARC and hope to publish a review shortly.

Zilog Announces New 8-Bit Micro

Zilog will soon release a new 8-bit micro that should delight the readers of this magazine. Late this year they will introduce the "Z800" (does that mean it is ten times as good as the Z80 and one-tenth as good as the Z8000?). The Z800 will be an enhanced Z80. Fully compatible with the Z80 instruction set, it will add hardware multiply and divide, and a memory-mapper circuit to access up to four Mbytes of memory. Zilog boasts that it will provide performance three times better than a four MHz Z80.

The Z800 will be offered in a non-multiplexed version like the Z80, and in a multiplexed version that can be used as a Z8000 peripheral. Zilog expects to start sampling the Z800 early this fall.

Incidentally, Zilog reported an $11 million loss on $42 million business in 1980. Zilog has yet to show a profit.

Random Rumors

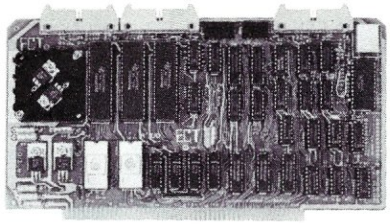
Several S-100 manufacturers are already in development on CPU cards using the new Intel iAPX-432 32-bit microprocessor. We can expect to see the first such product reach the market late next year....Xerox is rumored to be about to introduce a low-cost (to Xerox $4K-$7K is low cost) microcomputer system using CP/M. They will also furnish WordStar for it. Apparently, this is intended to compete with the Apple.

UNIX Software List Published

A comprehensive directory of UNIX and C software products is being published by InfoPro Systems, Box 33, East Hanover, NJ 07936 ($18/yr domestic, $24 foreign). The first issue I received was nine pages long and listed 29 suppliers along with very interesting comments on the suppliers and their software packages.

CUSTOM PRODUCTS

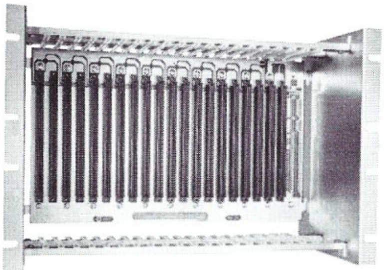
DESIGN • LAYOUT MANUFACTURING



R ²I/O

ROM/RAM & I/O

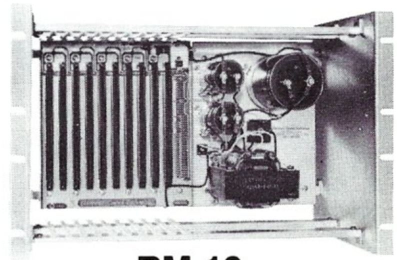
S-100 PRODUCTS



ECT-100-F

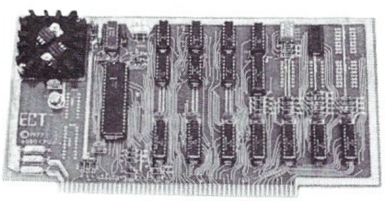
RACKMOUNT CARD CAGES

ECT™



RM-10

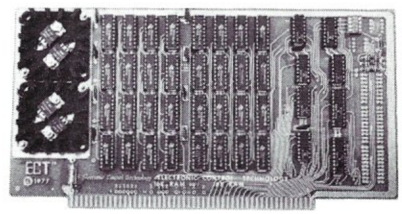
CARD CAGE & POWER SUPPLY



8080 CPU

CENTRAL PROCESSING UNITS

ECT™

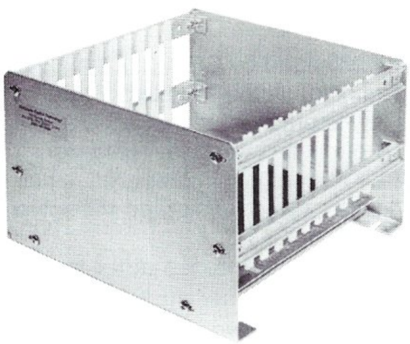


16K RAM

FULLY STATIC MEMORY

BUILDING BLOCKS FOR MICROCOMPUTER SYSTEMS, DEDICATED CONTROLLERS AND TEST EQUIPMENT

CARD CAGES, POWER SUPPLIES MAINFRAMES, CPU'S, MEMORY I/O, OEM VARIATIONS



CCMB-10-F MIN

6,10 OR 20 SLOT CARD CAGES

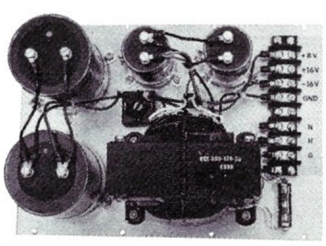
TT-10

TABLE TOP MAINFRAMES

ELECTRONIC CONTROL TECHNOLOGY, INC.

763 Ramsey Ave., Hillside, NJ 07205 (201) 686-8080

SPECIALIZING IN QUALITY MICRO COMPUTER HARDWARE



PS-30 A

POWER SUPPLIES

MULTIBUS® PRODUCTS

MULTIBUS IS A TRADEMARK OF INTEL CORP.

The CP/M Bus

by Anthony Skjellum

If you have questions about CP/M or MP/M we will attempt to answer them in this column. Send your questions to: Anthony Skjellum, 1695 Shenandoah Rd., San Marino, CA 91108.

The major topic of this column will be the continued discussion of possible enhancements for the CP/M operating system. The concept of link files will be introduced. Please refer to the material presented in the May-June "CP/M Bus."

I. More features for CP/M: Part II

It is often convenient for the same data or program to exist in more than one file on a disk. However, in some cases only one copy of the data is actually needed and it becomes a convenience to allow files to link to one another; this permits the programmer to organize data in a sophisticated manner. Link files aren't copies of files, but "point" to other files. Therefore, they only require link records and/or directory entries (depending on the type). Furthermore, when files are changed, any links to them reflect this change automatically.

Two types of link files will be defined here. They are simple and complex links, and will be treated in turn.

Link Files of the First Kind

Sixteen user areas are provided by CP/M2. Each user area requires its own copies of all the files to be used in that area. For example, transients like PIP and STAT are likely to be common to each area in use. However, it seems wasteful to place a copy in each user area, since the information is duplicated. Simple link files will solve this problem.

Simple link files consume no disk space other than a directory entry. They are identified by an attribute bit which we will call b2'. These link files will link a file in user area zero. Since a link file requires no directory map, this sixteen byte region (d0...dn is the Digital Research convention) may be used for the name of the actual file in user area zero. See page 14 of *CP/M 2.0 Users' Guide for CP/M 1.4 Owners* for more information. Link files of this type will be prohibited in user zero.

In order to make simple links useful, a new CCP (console command processor) command is proposed. This is the LN command (standing for link). LN will be used to create simple links and will obviate the need for a special initialization process of new user areas. LN will be used as follows:

LN afn user-number

or

LN ufn user-number new-name

where user-number is a valid user area number greater than zero. When user-number is omitted, the current user area is assumed (provided that we are not in user area zero). Furthermore, "afn" is the ambiguous file specification. However, if we do want to rename the link, an unambiguous specification (ufn) will be needed as will the user-number. Here are two examples of LN in use:

```
USER x                ; change to user x<>0
LN *.COM              ; link all .COM files to this user area
                      ; since user-area was omitted, x was assumed


LN PIP.COM x XFER.COM ; link PIP.COM to user area x and call it
                      ; XFER.COM
                      ; (we may be in any user area while doing this)
```

A simple link will have all attributes reset except b2'. However, they will be alterable with STAT. For example, we may want a link of a text file in user area x to be SYS even if the actual file in user zero were DIR. Also, remember that deleting a link to a file does not affect the original file in any way. The ERA command will be used to delete simple link files.

Simple link files may not be written to since they are only images of the actual file in user zero. However, reading a link file will be transparent to a transient; it will appear as though the actual file were being read, and no special BDOS commands to access this type of file are needed.

I also believe that simple link files could be included in MP/M without difficulty. Since there is no writing to these files, no problem about conflicts between multiple user access is anticipated.

The type of linking mechanism described above would be quite straightforward to implement and should be quite useful. It would definitely be advantageous in the MP/M environment also. Indeed, this is essentially the type of linking provided by operating systems like UNIX (shell command 1n). However, much more ambitious linking mechanisms are possible and complex linking is described below.

Link Files of the Second Kind

Complex links continue where simple links leave off. A complex link file may have links to several other files or portions thereof, and may also include data records. Complex link files are indicated with an attribute bit, as are simple links. We will denote this attribute bit as b3'.

Link records consist of information to tell the BDOS what file or part of a file needs to be accessed. The maximum length of a link record entry is sixteen characters, so VLR files used for complex linking will have to have record lengths of at least sixteen. A complete discussion of the internals of link records will be deferred to the next column, when we will discuss them in conjunction with the sub-directory feature.

Complex link files will use normal directory entries since they consist of a number of data and link records which may be mixed as desired. With nested linking, several files will be open at once and each will require an FCB. For example, if file A linked to B which linked to C, three files would need to be open at once. Therefore, the concept of the extended file control block (EFCB) will be introduced.

The EFCB consists of several file control blocks which will be used by CP/M. The first FCB is called the primary FCB and is used for opening the link file. It is followed by six word quantities: nx, nc, a1, a2, a3 and a4. The nx variable tells the BDOS the maximum depth of nesting supported for this file. That means that there must be nx

CP/M Bus, cont'd...

FCB areas provided besides the primary FCB. The nc variable is used by BDOS and contains the current linkage depth. Variables a1, a2, a3 and a4 contain addresses: a1 is the current FCB in use (set by BDOS relative to the address a3), and a2 is the address of the link buffer which must also be provided for the use of CP/M. The link buffer provides storage space for a link record which is being processed. (The buffer is as long as the file record length.) This buffer is necessary since multiple link entries per record are possible. The address a3 points to the start of the first extended FCB. If a3 is zero, the extended FCB's are assumed to follow a4 directly. Finally, a4 is used by CP/M to keep track of its position within the link buffer.

Note that the primary FCB is assumed to be 36 bytes long and include the r0, r1 and r2 fields added in CP/M release two. However, the extended FCB's require only 33 byte entries.

Several new BDOS commands will be needed in order to use complex link files. First of all, a create command will be needed. This will work as the standard make file command implemented in CP/M2. However, it will set the bit b3' high to indicate that the file is a complex link file. Two versions will be available, one for standard (128 byte record) files and one for VLR files. Second, a generalized open command will be needed. The DE register points to the primary FCB on entry to BDOS; all other necessary information is picked up from the FCB and words which follow it.

Several examples are provided here for clarity:

```
                        Create of Link-file
lxi      d,efcb              ; point to file control block
                             ; (probably will be the extended block
                             ; if we plan to do subsequent reads)
mvi      c,lmake             ; regular link not VLR
call     bdos                ; execute call
inr      a                   ; a is 255 on error
jz       error               ; yes...
...
```

```
                        Create of VLR Link-File
lxi      d,efcb              ; extended block
mvi      c,vlmake            ; VLR link create
lxi      h,recl              ; record length for VLR file
                             ; at least 16.
call     bdos                ; execute call
inr      a                   ; on error
jz       error               ; exit...
...
                        Open of Link-file
lxi      d,efcb              ; point to extended file control
                             ; block.
lxi      h,5                 ; there are five extended blocks
                             ; (total of six levels including
                             ; link file)
shld     nx                  ; set maximum depth
mvi      c,lopen             ; code for link-open
call     bdos                ; execute it
inr      a                   ; see if error
jz       error               ; yes...
...
                        Read of Link-file
lxi      d,efcb              ; point to extended fcb
mvi      c,lread             ; read link command
call     bdos                ; execute call
ora      a                   ; error?
jm       overfl              ; overflowed efcb's
jnz      eof                 ; end of file occurred
...
                        Typical EFCB
efcb:  ds     36             ; 36 byte primary fcb
nx:    dw     depth          ; max nesting depth (5 here)
nc:    dw     0              ; current depth
a1:    dw     0              ; current FCB in use (relative to a3)
a2:    dw     lbuffr         ; point to link buffer
a3:    dw     0              ; point to start of EFCB's.  If zero
                             ; expect directly after a4.
a4:    dw     0              ; used by CP/M to keep position in
                             ; link buffer
ebuffs:       ds     33*depth      ; extended storage
lbuffr:       ds     reclen        ; link buffer (file record lengt
                             ; 128 in this example)
depth  equ    5              ; nesting depth
reclen equ    128            ; standard file record length
```

If an overflow of the EFCB occurs on a read, the sign bit of the accumulator is set. This can be detected as in one of the examples above.

Finally, we insist that all files linked by a primary file have the same record length as that primary file.

It will also be useful to manipulate link records directly. Therefore, a read record absolute command will be provided. This command will return the next record of the file even if it is a link record. Similarly, link records can be written by making a ^Y the first character of the record written, as complex link files are always writable when the nesting depth is zero (i.e. writing to the primary file is permitted.) The a1 address word gives the program the capability to inspect the FCB's of files linked by the primary file. With this information, these files could be independently opened and modified.

Sophisticated indexing schemes are possible through the manipulation (e.g. sorting) of link records and the manipulation of record sub-ranges. Also note that the random access BDOS commands will not expand links (i.e. link records will be returned as read) so that random input-output can be used for creating an indexing method. It is left to the reader to explore these possibilities.

Sub-directories

In this and the last installment of "The CP/M Bus" we have discussed many new features and file modes which could be added to the CP/M operating system. Another extremely useful possibility is the sub-directory. This file type will provide the ability to deal with files outside the sixteen user numbers and allow a flexibility in file maintenance akin to that found on large systems. This will be a primary point of discussion in the next installment.

The TEC-86 16-Bit Computer System

by Chris Terry

The TEC-86 computer system, manufactured by TecMar Inc., is a general-purpose microcomputer system using the new Intel 8086 16-bit microprocessor. The rugged metal enclosure houses a heavy-duty power supply, an S-100 motherboard with twelve slots, and two Shugart SA800 8" floppy disk drives. The basic system is supplied with:

•CPU board equipped with an Intel 8086 microprocessor running at 5 MHz (4 or 8 MHz options available), an 8259A priority interrupt chip, and power-on jump circuitry;

•32K of 300nS static RAM on two 16K boards, expandable to 1 Megabyte; available as an option is a single 64K dynamic RAM board at the same price as four 16K boards.

•PROM I/O board equipped wih two 8251A serial ports capable of handling synchronous or asynchronous RS-232 data links at transmission speeds of up to 19,200 baud, an 8255 chip that provides 24 lines of parallel I/O, and sockets for 2K x 16 of PROM;

•Microbyte single/dual density disk controller, based on the NEC 765 LSI controller chip and capable of supporting up to four drives.

The price for the basic system is $3990; additional 16K memory boards are available at $395 each.

Hardware Documentation

The manuals supplied by TecMar for each board in the system are very good. They supply complete logic diagrams which, though reduced to half the original size, are clean and readable, as regards both lettering and layout. They are also split into convenient one page chunks, each of which contains one or more complete functions; connections that have to cross page boundaries are brought to the left or right edge of the diagram and are plainly visible. Pin connections and cabling to the outside world are clear and have text clarifications where necessary. On-board jumpers to select options are similar to those found on disk drives—contact pins which are connected together by jumper connectors in plastic covers. The placement of jumpers is both described and illustrated for each option, and the user should have no difficulty in setting up or changing the jumpers correctly. Switch settings are defined as "Open" or "Closed" according to the marking on the switches, and there are clear statements as to whether a switch closure represents a 1 or a 0 on the associated line.

The theory sections contain enough detail to clue in a person who already has a fair amount of hardware experience, and are enhanced by simplified logic diagrams of functions that might otherwise be difficult to understand. This is a most welcome change from so many other manuals where highly detailed and dense descriptions refer to equally dense fold-outs, with no clue as to where in the drawing to look.

TecMar is to be congratulated on these manuals. They have obviously hired professional writers and given them reasonable time and budget to do a first class job. The language is just informal enough to be readable without losing exactness, and clarity has been made a prime goal.

I found only one typographic error (the notorious "intergrated" chips, which conjures up visions of elves diligently grating cheese into the inter-chip spaces). And only one factual error—which in any case is not calamitous—the I/O board manual calls out RS232 signal levels as +5 to +15 volts for a Mark (1) and -5 to -15 volts for a Space (0). In fact, the RS232-C spec defines the signal level limits as 3 volts to 25 volts in either direction relative to signal ground; the positive level is a SPACE (0) for a data line and ON for a control line, whereas the negative level is a MARK (1) for a data line and OFF for a control line.

The Software

Software to support the TEC-86 consists of CP/M-86 from Digital Research, Inc., and Basic-86 from Microsoft, Inc. TecMar also has Pascal/M-86 from Sorcim available as an option. Mention is made in the PROM I/O board manual of a system monitor for which the PROM sockets are intended, but this does not appear on the current price list. The PROM in the evaluation system contains the CP/M-86 bootstrap and disk primitives, but no monitor accessible to the programmer. It would not be necessary,

Chris Terry, 324 East 35th St., New York, NY 10016.

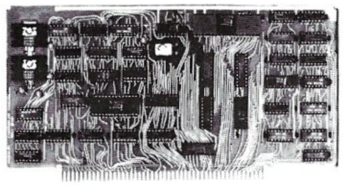
16 BIT 8086 MICRO SYSTEM

TEC	-86W	$9990
TEC	-86	$3990
CP/M	-86™	$ 250
BASIC	-86™	$ 350
PASCAL / M86™		$ 270

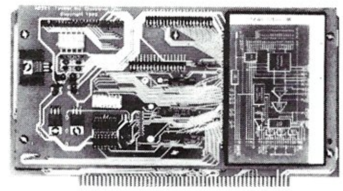
- *Tec-86W includes a 31 Mbyte Winchester plus memory increase to 256K plus all the Tec-86 features below.*

- *IEEE 696 S-100 • CPU w/Vectored Interrupts*
- *Two RS-232 Ports • 24 Parallel I/O lines (3 ports)*
- *64K RAM (256K RAM with Tec-86W)*
- *ROM boot for CP/M-86™*
- *Dual 8-inch Shugart floppies with controller*
- *Attractive, all metal desk top enclosure*

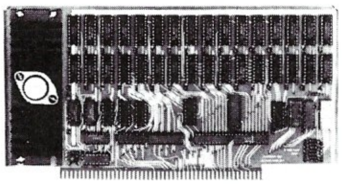
S-100 Boards



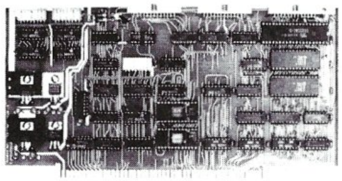
8086 CPU with Vectored Interrupts



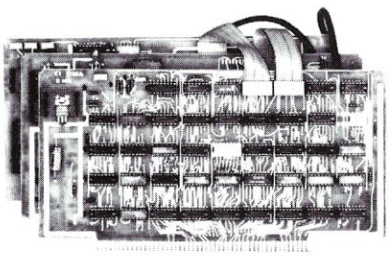
Analog to Digital Converter and Timer/Counter
16 channels expandable to 256
12, 14, or 16 bit accuracy
30,40,100,or 125 KHz
Programmable Gain



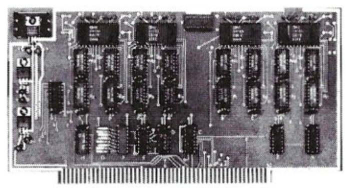
16 Kilobyte Static RAM
8 and 16 bit transfer



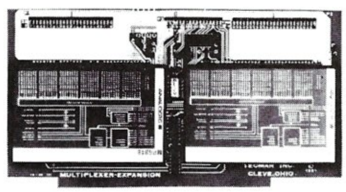
PROM and I/O
2 RS 232 - PIO
CP/M-86 ROM Boot



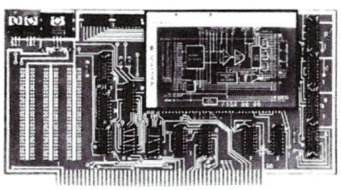
Real Time Video Digitizer and Display



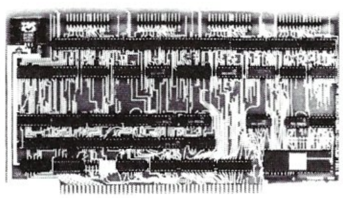
Digital to Analog Converter
4 channels - 12 bit accuracy
3 microsecond conversion rate



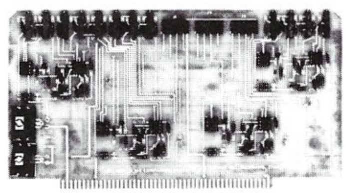
Expansion Multiplexer



Analog to Digital Converter
16 channels - 12 bit accuracy
30 KHz Conversion rate

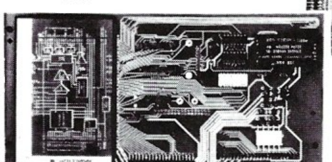


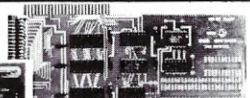
4 PIO and Timer/Counter



Digital to Analog Companion
4-20 mA output - Filters

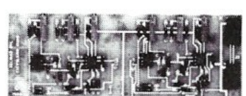
Apple Boards

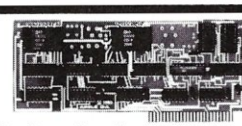




Analog to Digital Converter
16 channels
12, 14, or 16 bit accuracy
30, 40, 100, or 125 KHz

Digital to Analog Companion
4-20 mA output - Filters





Digital to Analog Converter
2 channels - 12 bit accuracy
3 microsecond conversion rate
1 parallel output port

Data Acquisition Systems and
Video Digitization Systems Available
23600 Mercantile Rd. • Cleveland, OH 44122

TECMAR, INC.
(216) 464-7410

since the CP/M-86 DDT is perfectly adequate for this purpose.

CP/M-86

This operating system is functionally equivalent to CP/M Version 2.X for the 8080/Z80 systems. The differences are due mainly to the use of separate memory segments for code, data, and stack, and the addition of function calls—CP/M-86 has 59 function codes, compared to the 36 of CP/M-80 Version 2.X. Page 0 is used for the same purposes as in CP/M-80, but the operating system is usually loaded at 400H, directly above the interrupt locations. You can, however, change this location. Relocatable transient programs load above the operating system, starting at 2A00H. Unlike CP/M-80, CP/M-86 does not use absolute locations for system entry or default variables; instead, entry to BDOS takes place through a software interrupt, and entry to BIOS is by a new function call. Most of the new function calls are related to the allocation or releasing of memory.

Because of the additional BDOS functions and a larger BIOS, CP/M-86 is too large to fit on two single-density tracks, though it fits comfortably on two double-density tracks. If single-density is used, the bootstrap loads only the cold-start loader; this in turn loads CP/M-86 from the file area (not the system tracks). A warm start is somewhat simpler than in CP/M-80, since you are not required to reload the CCP and BDOS. Further, relocation of the system is somewhat simpler because relocatable code is used. Thus, there is no MOVCPM utility; the only change is to the cold boot, telling it where to start loading the operating system.

The standard system supplied by TecMar is configured to run in a 64Kbyte memory; however, the distribution disk also contains systems to run in 32K or 96K.

CP/M-86 Documentation

As the Duke of Gloucester remarked when presented with Volume 4 of *The Decline & Fall of the Roman Empire*: "Another damned thick, square, book! Always scribble, scribble, scribble! Eh! Mr. Gibbon?" The TecMar system documentation consists of a six page leaflet describing how to boot up the system (simplicity itself—turn on power, hit RESET, put the disk in the A drive, and close the door!), how to format disks for single or double density, and how to copy the system tracks, for which TecMar has provided utilities to suit the Microbyte controller and formats.

Digital Research has been (necessarily) more lavish. In addition to the *Introduction to CP/M Features and Facilities, The CP/M 2.2 User's Guide,* and *The Ed User's Manual*, which are standard for all versions, there is a huge amount of completely new material. *The CP/M-86 Reference Guide* has 138 pages, *The ASM-86 User's Guide* has 75 pages, and *The DDT-86 User's Manual* has 19 pages. *The CP/M-86 Reference Guide* is, like most Digital Research manuals, a tough nut to crack. All the required information is there, but it's not always easy to find. The definitions of BIOS routines and BDOS function calls are easy—they are presented in order, concisely, and reasonably clearly. It's the mass of other information that causes me trouble. I wish I knew why. I cannot complain that the manuals are badly written or disorganized. Individual sentences are perfectly clear, and there is organization. But it always takes me more time than I like to find what I am looking for. What is frustrating is that I cannot think of just how the manual could be better organized. I suppose you just have to read and read and read until you know it almost by heart, and then your brain goes "Click!" and the pieces drop into the places in your brain from which you can most easily retrieve them. Perhaps an index would help?

Performance

For me, the TecMar system has behaved in an exemplary way. I unpacked it, spent three or four hours with the manuals, plugged it in, connected a Lear-Siegler ADM-3A terminal set for 19,200 baud (as instructed), booted up, and away we went. Operationally, the instructions were clear and simple. Except for copying single-density Basic-86 to a double-density working disk, which gave me a little trouble at first, it's just like running CP/M 2.2 and Basic-80.

I have not yet found a huge increase in speed, but that is because I have not yet gotten to any real number-crunching in A86. Basic-80, as I understand, is a simple translation of the interpreter from 8080 language to 8086 language, without optimization to make use of the special features of the 8086 CPU and architecture. Thus, when I loaded my Basic program for testing sorting routines, the interpreter (which runs on a 5-MHz clock) executed Bubble, Heap, Shell-Metzner, and Quick sorts in a shade less than half the time it takes on my 2 MHz 8080 machine using Basic-80. For 200 random numbers, the Bubble sort took 148 seconds instead of 310, Heap took 32 instead of 67, Shell-Metzner took 34 instead of 71, and Quick took 17 instead of 34 (average of three runs each). But I suspect that a Z80 running at 4 MHz would have done nearly as well.

However, I am sure that the speed advantages will be seen when there is more software around that is optimized for the 8086. A nice screen editor like Wordmaster, for example. ED is for the birds unless you still have a Teletype, and I am thankful to hear that impending CP/M-80 Version 3.X will have a screen editor. If an 8086 version also appears that uses the magnificent string handling capability of the 8086, it will probably be a joy to use.

Conclusions

The TEC-86 is rugged, easy to get going, has given me no hardware problems and only minor software puzzlement (I didn't read the manual carefully enough to start with). A price tag of $4600 (which includes 64K of RAM, CP/M-86 and Basic-86) is probably too much for the average hobbyist. But for a small business or a professional user it will be extremely good value, once the software starts being available. And don't forget that there is much more available right now than you might think—you can run any existing Basic-80 program on the 8086, provided that you save it on a single-density disk as ASCII source code. As you may have gathered, I like TecMar's product and their hardware manuals. I wish I could afford it for myself! ■

Available from: TecMar, Inc., 23600 Mercantile Rd., Cleveland, OH 44122, (216)382-7599.

Seattle Computer Products' 8086 System

by Bill Machrone

Considerable attention has been generated by some of the recent entries into the 16-bit arena for the S-100 bus. Some manufacturers are still talking about it, others are doing something about it and a few are already old hands at it. Seattle Computer Products (SCP) has been manufacturing IEEE-696 compatible 8086 processors and 16-bit wide memories for more than two years. Additionally, they offer a system support board and a serial I/O board, all compatible with the 8086 or any other processor that follows the IEEE-696 Standard.

All the well-intentioned hardware in the world is worthless without software to make it go, and Seattle has pioneered here as well. Long before CP/M-86 was released, Seattle's 86-DOS was a reality. Below, we'll take a look at the available products and give an evaluation of just how fast it is and how useful it could be in your system.

Hardware

The processor board itself contains an 8 MHz 8086 which can be switch-selected to run at 4 MHz. The board produces or responds to all the standard S-100 signals, including SXTRQ* and SIXTN*. This means that the board can address memories that are either eight or sixteen bits wide and, in accordance with the IEEE-696 Standard, permits intermixing them in the same system. The memory cards must support 24-bit extended addressing. The processor handles memory and I/O references as either eight bit transfers, sixteen bit transfers or "double eight bit transfers" where memory is incapable of a sixteen bit transfer. There is a provision for an Imsai-style front panel, but a small modification is necessary to make it work. Examine and deposit functions are inoperative with the 8086.

The CPU Support Board has all the goodies necessary to make the system functional, including a monitor/bootstrap EPROM, two 8259A interrupt controllers, two 16-bit counter/timers, a 24 hour clock (more timers, actually) with provision for battery backup, a serial port, a parallel port and a sense switch input port. Strangely enough, the parallel port is configured as a separate parallel input port and a parallel output port, each with its own cable header on top of the board. This may be advantegeous for some applications, but doesn't permit a full handshaking bidirectional configuration. The bootstrap EPROM has a full 8086 monitor program which allows memory inspection, tracing, debugging and booting the disk controller. At this writing, Seattle does not manufacture a floppy disk controller, so you can request an EPROM which boots one of several popular disk controllers, such as the Tarbell double density controller or the Cromemco 4FDC.

The timers are implemented with the versatile AMD 9513, which provides five timers—one intended as a baud rate generator, two general purpose and two which can be configured as a time of day clock with 0.01 second resolution, or which can also be used as general purpose timers. It has settable alarm registers, which can generate interrupts. Much has already been written about the 8259A interrupt controllers, and their power and versatility is well known. They are configured in a master/slave relationship on the CPU Support Board. Further slave controllers or interrupt sources can be added via the S-100 vectored interrupt lines. Most of the board's options can be selected by dipswitch, and there are several pin jumpers for other options.

The boards were subjected to all the normal abuses, such as fast clock rates and high ambient temperatures, and performed flawlessly.

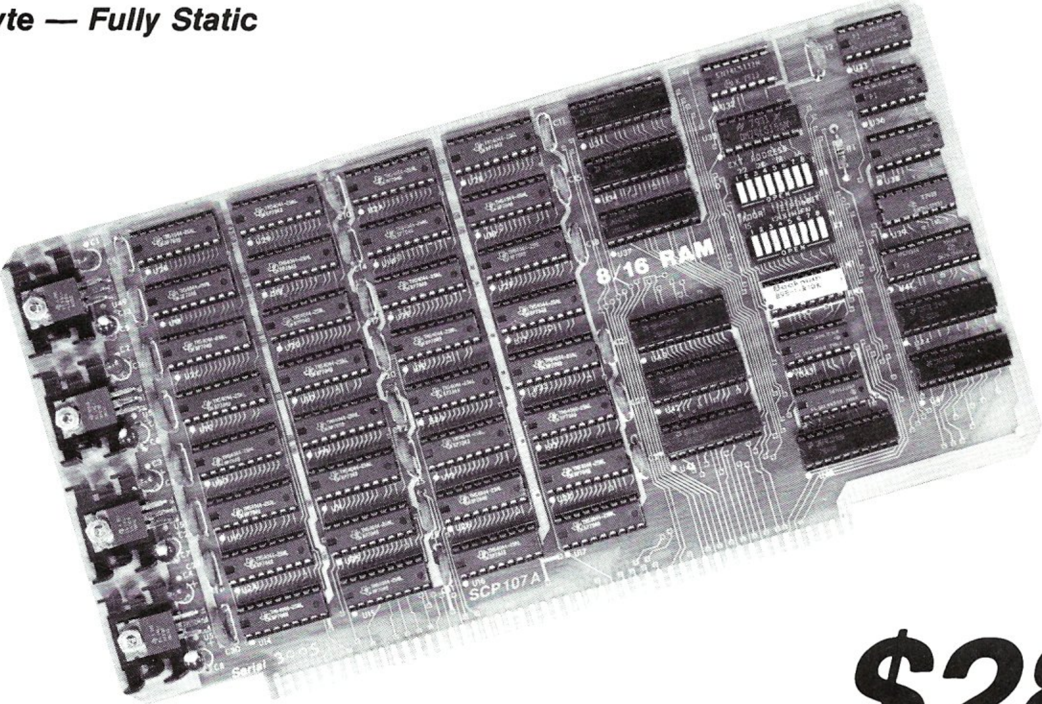
The decision to spread the CPU and system management functions over two boards is a sound one. Both are uncluttered, easy to configure and run cool. The two-board approach also gives the user some flexibility in upgrading existing systems. The CPU Support Board could be used by any processor, although it might duplicate one or more of the functions found on the popular 8-bit CPU cards. It's also possible to use someone else's support card with the 8086 card, such as Godbout's new System Support 1.

Bill Machrone, 595 West Hartsdale Ave., White Plains, NY 10607.

8/16 RAM

FOR YOUR 8-BIT SYSTEM TODAY
FOR YOUR 16-BIT SYSTEM TOMORROW

16K Byte — Fully Static



Quantity 1-9 **$280**

- Can act as either an 8-bit or 16-bit wide memory. Dynamic data bus switching per IEEE Standard.
- Fully static design eliminates system timing problems. Promotes reliable operation with a wider range of CPU cards and DMA devices.
- Easy to Integrate into your system. Addressable on 4K boundaries. PHANTOM, extended addressing, 16-bit operation may be switched off if desired.
- Able to use the full 24-bit address bus of the IEEE S-100 Standard for a 16-megabyte address range. A single switch allows 16-bit addressing for your 8-bit CPU.
- Fast 200 nanosecond memory chips help you keep up with the ever-rising clock speeds of newer CPUs.
- The 8/16 RAM is designed to run at full speed with our 16-bit 8Mhz. 8086 CPU card. And, it has plenty of speed to spare.

OTHER HIGH-QUALITY 8-BIT RAMS

16K PLUS RAM — this fully static RAM has become the standard of the industry. It features 200 nsec. chips and Cromemco style bank select using port 40H. Addressable to any continuous 16K on 4K boundaries. Any 4K block may be disabled. High reliability, low noise design. Prices: 1-9, $280; 10-19, $260.

16K STANDARD RAM — this fully static RAM is frequently used by OEMs in systems which do not require bank select. High reliability, low noise design. Uses 200 nsec. chips. Addressable to any continuous 16K on 4K boundaries. Any 4K block may be disabled. Prices: 1-9, $265; 10-19, $245.

TO ORDER: May be ordered through your local computer store or factory direct. Personal checks, CODs, VISA and MC accepted from within USA. 10-day return privilege on factory orders. Shipping paid by SCP on prepaid USA and Canadian orders. Add $15 per board extra for shipping overseas. All boards fully assembled, guaranteed one year.

Seattle Computer Products, Inc.
1114 Industry Drive, Seattle, WA. 98188
(206) 575-1830

SCP Review, cont'd...

Seattle has been producing rock-solid memory boards for as long as they have been in business. The 8/16 RAM follows in that tradition, providing a sixteen bit data path for fastest performance in an 8086 environment. As the name implies, it can be used as an 8-bit memory as well. It appears as either 16K of 8-bit memory or 8K of 16-bit memory. Each card can be addressed anywhere in the 16 Megabyte S-100 address space and can be set to respond to PHANTOM*. They are fully static and use the standard 4044 memory chip.

As with the boards mentioned previously, these boards are models of spacious layout and clean design. The boards provided for the review were subjected to all the normal abuses, such as fast clock rates and high ambient temperatures, and performed flawlessly. The 6 MHz Z-80B actually places more demands on them during instruction fetch cycles than the 8086 does at 8 MHz in any operation mode. They proved to be a match for the worst conditions I could provide in several system environments.

Software

Over the months that I've had the Seattle system for review, the software has been a living, growing thing. I received an early copy of 86-DOS and have received several updates. Then there was a long delay while we waited for Microsoft to modify stand-alone Basic-86 to run under 86-DOS. The conversion was finally done by Seattle Computer Products, with help from Microsoft.

86-DOS is similar enough to CP/M to make you feel at home, but different enough to get you into trouble if you assume that it's really the same. It is conceptually similar, but the differences could be considered departures or enhancements, depending on your point of view. The fact that there are so many good ideas within a framework familiar to the user shows that SCP has some good software people with minicomputer exposure, as well as talented hardware designers.

Typical of the enhancements is the line editor built into the command line interpreter. It uses the DEC VT52 function key escape sequences to permit the last line entered to be edited and resubmitted—just the thing when you make dumb typographical errors and you really don't feel like re-entering the entire line. It's also handy when the next command you are going to enter differs by only a few characters from the last command entered. Also, the file copy utility is memory-resident, which saves you the time required to load PIP. The control characters have essentially the same effect except that a control-N is required to un-toggle the printer after a control-P has started it.

The utility software provided includes a resident 8086 assembler, a line-oriented editor, a CP/M to 86-DOS file converter, a Z-80 to 8086 source code converter and a breakpointing debugger. I did not spend much time with the assembler or line editor, but the editor is just as bad as any other line editor I have attempted to use. The Z-80 to 8086 source code converter is interesting. It does a fairly good job until it gets to special Z-80 instructions like block I/O and some of the extra register functions. At this point you have to code by hand. I cannot attest to the relative efficiency of the 8086 code generated because I'm not sufficiently conversant with its instruction set.

The debugger is as good as any of the general-purpose debuggers to be found in the 8-bit world. It loads only object files (no HEX) and, as the manual points out, it will even trace ROM. Every instruction is traced correctly, unlike most 8080 and Z80 debuggers. It doesn't do anything fancy like using a symbol table, but what good is a program like ZSID when the thing misinterprets the object code? The debugger also includes a disk read and write capability.

All of the development tools are important, but the real thing that makes a new processor go is the availability of high level languages. The 8086 languages are coming on strong and Microsoft was there first with Stand-alone Disk Basic-86. With the conversion chronicled above, SCP became the first manufacturer to offer the full hardware, operating system and high level support of an 8086 on the S-100 bus. Virtually anything that is written for the 8080 Microsoft Basic interpreter will run on the 8086 interpreter, but it will go faster because of the higher clock rate and throughput of the 16-bit machine.

The manuals are oriented toward the experienced micro computerist, particularly one who is graduating from an 8-bit processor in the S-100 world.

Documentation

Before we go on to a comparison of execution times between the 8-bit and 16-bit worlds, a few words are in order about documentation. The folks at SCP have been conscientious in keeping current owners updated with new manuals and releases. Most of the material I received from them had a "Dear User" flavor, giving no indication of preferential status as a reviewer. The manuals are complete, clear and well written, but they are definitely oriented toward the experienced microcomputerist, particularly one who is graduating from an 8-bit processor in the S-100 world. They convey enough information for an experienced person to get the system configured and running, but I think that a relative newcomer or an Apple-wizard would be somewhat bewildered. More examples and pictorials of option switch settings would be helpful.

The one manual in which pictorials are used is the 8/16 RAM manual. Unfortunately, they are a total failure. The artist selected strange trapeziodial directional indicators for the dipswitches which have confused everyone to whom I have shown them.

Comparisons

Aside from those who always have to have the best, newest or fastest computer equipment, there are a limited number of reasons why a user would select a high performance 16-bit system over a high performance 8-bit system. There is no doubt in anyone's mind that the 8086 can move data around faster than even a 6 MHz Z-80, especially when the data path is 16 bits wide. All the standard benchmarks peg the 8 MHz 8086 as having five times the throughput of a 4 MHz Z-80, so all your programs will run five times faster, right?

SCP Review, cont'd...

I wish it was that simple. The real stumbling block is software, not CPU speed. I ran "known quantity" programs that I had written in Microsoft Basic on both machines and found some interesting results. I should point out that I use Basic-80 strictly as a development tool for the Basic Compiler, which represents a plateau of efficiency for 8-bit high level languages, since only PL/I-80 (to the best of my knowledge) produces faster object code. Now, you may object and say that it's unfair to compare a Compiler and an interpreter even when the CPU is five times faster, but we're talking about reasons to buy the 16-bit machine. The software state-of-the-art is a major factor.

First, let's state the facts: Basic-80 is definitely slower than Basic-86. If throughput in executing interpreted Basic programs was the sole criterion, there would be no contest. The second fact is that the Basic Compiler does everything faster than Basic-80, and here again, there is definitely no contest. Its slowest functions, such as string concatenation, are still three or four times faster than the interpreter. Its fastest operations, such as integer arithmetic, are up to twenty times faster than the interpreter.

So when we benchmark the 8086-based interpreter against the compiler, what do we find? We find the compiler still faster in most instances. One exception is string concatenation, which was actually faster than the compiler in the tests I made. This should not be a surprise, because large portions of Basic-86 appear to be translated 8080 code. By no means does this suggest that you shouldn't consider buying an 8086-based system. Can you imagine how fast a Basic-86 compiler or PL/I-86 will be? Or how much less contention will be experienced in a multi-user environment? Once software becomes available that is well optimized for the 8086, the performance will be remarkable.

By the way, for my fellow hardware freaks, there is a switch on the CPU board which limits it to 8-bit data transfers. It allows you to demonstrate the degree of throughput gain you get with the 16-bit data transfers. What with slogging through all the code in the Basic interpreters, I noticed very little difference between the 8086 in 8-bit mode and a 4 MHz Z-80. For that matter, there was no discernible difference in the operation of Basic-80 with a 4 MHz Z-80 and a 6 MHz Z-80. The 8086 in 16-bit mode was sufficiently faster than 8-bit mode to be noticeable, but the difference was not breathtaking. Again, the quality of the software being executed has a major effect on how efficient the processor will appear.

Conclusions

The conclusion I have drawn from living with the 8086 for a number of months is that the SCP hardware is an excellent foundation upon which to build your entry into the 16-bit world. It is solid, reliable stuff and their software works. (This cannot be said of all manufacturers who create their own operating and utility software.) The availability of Basic-86 is a tremendous convenience, one that bodes well for the future. As an OEM/systems integrator, I'm sure that I will use the 8086 in a commercial system in the not-to-distant future.

First, however, I'll need WordStar-86, MDBS-86 and all the other "spoilers" which make life in the 8-bit world so enjoyable. The advent of 16-bit high level language compilers will be the crowning touch. Then, look out DEC, Hewlett Packard, et al. ■

System Product Review

Alpha Micro System Revisited

by Hank Kee

Background

The Alpha Micro system was originally introduced in December of 1976. It has been around for so long that many of us have tended to overlook the system as the first 16-bit system available on the S-100 bus. This system is often used as the benchmark for all other microcomputer systems. It was originally advertised and promoted to the hobbyist in various microcomputer journals. However, they now are no longer selling "direct" to the general public but prefer to sell through dealers. The main thrust of their dealers' selling efforts today is to the "small" commercial business user.

There are well over 5,000 Alpha Micro systems running; last year the company reported sales of over 21 million. There is also a very active Alpha Micro users group called AMUS (c/o Steve Elliot, Front Range Computer, 1966 13th St., Boulder CO 80306).

The system is based on the conceptual architecture of the LSI series designed by Digital Equipment Corporation. The smallest basic configuration (eight systems are available) consists of a two-board CPU (AM-100), a six port serial I/O board (AM-300), and a floppy disk controller (AM-210) interfacing to CDC drives. A hard disk cartridge system (CDC Hawk or Phoenix) could be added for greater disk storage capacities (360 Mbytes maximum). Additional available equipment includes 8.5 Mbyte Winchester and 9 track 1/2" tape peripherals. There are now variations of these boards with different options. Further, both serial and parallel pointers (300, 600 or 900 LPM) with two spoolers are supported. This review is necessarily confined to their original product offerings only because of limited access to their hardware.

Overall Architecture

The AM-100 CPU consists of two boards populated by a five chip set micro-encoded processor manufactured by Western Digital. Western Digital was the original manufacturer of the LSI series for DEC. The AM-100 CPU contains hardware floating-point math. The mnemonic code of the AM-100 is essentially the same as the LSI series, but they differ at the object code level. The Alpha Micro has an improved instruction set compared to the LSI. Assembler source code from the LSI can be easily converted onto the Alpha Micro. A separate 8 to 10 VAC is required to generate the real-time clock pulse. This could be easily tapped off the power supply of the main frame transformer prior to it being rectified into DC.

The AM-210 is the floppy disk controller which has the addition of a Z80 processor. This allows for an interrupt driven operating system. Unlike many other micro based disk systems, interrupts on the Alpha Micro need not be disabled during disk operations. The user can key-in ahead instead of waiting for the system to poll for character input. The original system I worked on interfaced to either Persci 277 or Wangco 76 8" drives. The floppy disk system currently offered by Alpha Micro uses CDC drives and the AM-210 controller. The CDC's are dual-sided double density units. The current floppy disk systems offered by Alpha Micro now has over 2MBytes available to the user.

The functions of I/O are handled by the AM-300 six serial port board. An AM-310 is also sold to those who wish to interface synchronous as well as asynchronous devices to the Alpha Micro.

To complete the basic system, a Piiceon 64K dynamic memory board is available. Up to eight memory boards may be installed for a maximum of 512K of memory. This board has optional parity checking features. I have tried numerous other bank select dynamic memories. Only the Piiceon, which was recommended by Alpha Micro, works. An alternative is to use static memories with bank select. Memory boards need not be rated any faster than 450 nanoseconds. Each additional concurrent user requires about 32K of memory. As the number of concurrent users increase, so will the number of memory boards. The use of static memories tend to cause system heat build-up.

Alpha Micro also offers the CDC Hawk cartridge hard disk system on the AM-500 hard disk controller. The drive comes with 5Mbytes of fixed and 5Mbytes of removable storage. Winchester technology without removable media is high risk on small business systems. Alpha Micro does not sell their AM-500 hard disk controller card separate from the CDC Hawk drive. Many owners contract with CDC for monthly service maintenance of

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

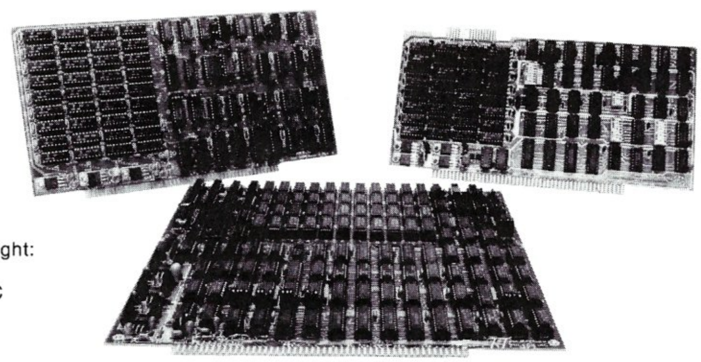
HOW'S YOUR MEMORY?

Our MEMORIES Have Made US Famous!

WHY?

Reliability!

That's the secret of our success. We are dedicated to producing the most reliable memories possible for the S-100 bus and the new IEEE S-100 bus.



Shown Left to Right:
SUPERAM 4
SUPERMEM-ECC
SUPERAM 2

Flexibility!

No matter what CPU you use, we have the memory for you.

SUPERAM™ 2—We've delivered over 5,000 of these 64K dynamic RAM boards to OEMs worldwide. This board sets the standard for reliable operation.

SUPERAM™ 4—Now available for Z80A and 8085A users, with bank select and optional parity bit.

SUPERAM™ 5—Our forthcoming IEEE-696 (S-100) compatible RAM will provide 64 Kbytes of dynamic RAM as either 32Kx16 or 64Kx8.

With ECC, too!

Announcing the SUPERMEM™ ECC! This is the ultimate in memory data protection: built-in automatic error correction. The SUPERMEM transparently corrects single bit errors and detects double bit errors.

We'll Design One for You!

If you have a special situation that calls for a custom design involving memory, processors, controllers, or even complete systems, give PIICEON a call. We might have the answer to your product need.

And That's Not All!

We also supply the following Microcomputer Boards:

- C-86 — 16-bit 8086 processor board with multi-processor arbitration logic, on-board PROM and RS-232 port.
- V-100 — 80x24 high-speed video output board.
- D-100 — Coming soon, the answer to 5¼ and 8-inch Winchester control.

WRITE FOR OUR NEW BROCHURE

PIICEON™

OEM Computer Products
2350 Bering Drive
San Jose, CA 95131
(408) 946-8030

SCP Review, cont'd...

the Hawk drive, but would prefer owning an extra disk controller card for the purposes of backup. Konan now sells a compatible disk controller (KNX-500). Instead of using the Z80 for data transfer functions, the Konan board uses the 8085. I have found the Konan controller to be an acceptable substitute.

All Alpha Micro manufactured boards, with the exception of the AM-100, can be used with existing S-100 boards for CP/M operation. (Alpha-Micro does not support CP/M on their systems.) BIOS Coding is generally available. The Konan KNX-500 is also supplied with BIOS coding.

Other "foreign" S-100 boards can be successfully included as part of the overall configuration. You will need an Alpha Micro system, though, to initialize other devices. Alpha Micro has included as part of their software, drivers for boards of many other S-100 manufacturers. Non-Alpha Micro boards, however, tend to be non-interrupt driven. The easiest and most efficient configuration is the basic system as offered by Alpha Micro. Originally the Alpha Micro was designed to work with the Tarbell disk controller along with the Imsai SIO or Processor Tech VDM boards.

Alpha Micro has since introduced the AM-100T (about a year and a half ago). This CPU uses a 16-bit address structure, as opposed to 8-bit address architecture. It run substantially faster than the AM-100.

It is possible to interface as many as 22 concurrent users onto the system using their hard disk systems. But I have found degradation of response becomes significant when there are more than a half-dozen or so users on the system. The size of available user storage decreases as the operating system increases to reflect the greater number of concurrent users.

Software Availability

The greatest asset of the Alpha Micro system is that the software is bundled with the purchase of hardware. Many of their software systems are excellent. The AMOS (Alpha Micro Operating System) includes a superb multi-user AlphaBasic in either interpretive or compiler mode, AlphaPascal, AlphaLisp, and a screen oriented editor (VUE). Rather than dwell on how the operating system works, it suffices to say that it is equivalent to a DEC system running RT-11. For those of you who are familiar with CP/M, it works very much like CP/M. Since CP/M is a variant of RT-11, it might be more equitable to say that AMOS and CP/M are similar to DEC's RT-11.

AMOS assigns disk space by project ID's. The operating system ID contains all the system level commands. The user may elect to add customized calls or eliminate others from this space if they are not referenced. This allows for a very small kernel operating system to be resident in memory. Commonly accessed system modules such as the Basic runtime package can be made resident in memory and available to all users. The Basic compiler and runtime modules are reentrant. The typical operating system would use about 32K bytes of memory. Only 64K of memory can be referenced at any one time by the user, including space required by AMOS. AMOS, in its design, permits shared reentrant code. Most of Alpha Micro's software can be made resident and reentrant. Basic generates reentrant user code.

There is password control on the Alpha Micro for each user of the system. A master account is available for unrestricted access. Instead of comparing this to CP/M, it is really a superset of MP/M.

Basic for the Alpha Micro is very powerful. The compiler can generate reentrant code for access by multi-users. It even tells the user how much time it took to compile a program. Available on this system is the capability to "MAP" variables, very much like COBOL. This allows the programmer to reference overlay areas of the same field with ease. Basic can also be used in interpretive mode. Variable names are not limited to two or three characters. They can be defined to be much more meaningful since up to 31 characters may be used.

Some Alpha Micro dealers have since added the capability to accept data from CP/M or IBM floppy disk formats. Utility programs are provided to perform these functions. Similar routines have been included to transform AMOS oriented formats into CP/M or IBM compatible data structures.

The Alpha Micro system has an excellent method for systems generation. A SYSTEM.INI file is created by the user defining configuration to be generated at dynamic boot-time. It is very easy to modify the operating system to include additional equipment. Having worked with CP/M, AMOS is superior to implement. AMOS also has facilities for running a modified system initialization without affecting the original SYSTEM.INI file.

The Alpha Micro System is the Rolls Royce of S-100 systems.

Alpha-Pascal is an enhanced UCSD Pascal with multi-user, multi-tasking features. Alpha Micro also offers LISP. FORTRAN, COBAL and APL are available from other sources and Alpha Micro dealers. I have not had an opportunity to explore these systems.

If you are interested in running packaged business software, your choices are limited. Alpha Micro offers an Accounting Package which includes the functions of accounts payable/receivable, general ledger, payroll, and inventory and order control.

Almost everyone I know who has implemented this "system" indicates that modifications are very extensive. It is not what one would call an easily adaptable turnkey business application system.

With the exception of the Alpha Micro Accounting System, the abovementioned software comes with the purchase of hardware. A variety of legal, medical, and other type packages are available from Alpha Micro dealers.

The software documentation supplied with the system is very good and quite complete. It is relatively easy to understand. The program reference materials are not tutorial in nature. These were meant to be of assistance to users who have a first-hand working knowledge of programming systems.

Reliability

The system has certain quirks. When running the Alpha Micro in multi-user mode, it is possible for one user to bring down all other users due to addressing of out-of-bounds memory or hardware "bus" failure. There is no form of hardware protection. In general, running production programs in multi-user mode will be of no problem. But it is advised that application development should not be running concurrent with production processing. During the past two years, I have experienced various board problems with the system. These are typical of past experiences I have had with other microcomputer boards. The only difference is that the repairs require returning the malfunctioning board(s) to an Alpha Micro dealer for servicing. Schematics are not available to the end-user. This arrangement is not always practical in terms of turn-around time. For business environments, it is almost mandatory to configure an overall system with sufficient back-up boards. This is expensive and sometimes not possible.

Conclusion:

The Alpha Micro system is the Rolls Royce of S-100 systems. The manufacturer's selling and maintenance policies however, are restrictive. Small business systems have been successfully designed around the Alpha Micro, but one must almost duplicate a total system to insure continual processing of business. For high performance, it will compete on its own against typical "minicomputer" suppliers. For the general hobbyist, the Alpha Micro may tend to get a little too rich.

Prices for the system are set by dealers and vary depending the configuration and value added by the dealers. Prices range typically from $10,000 to $15,000.

The Godbout Dual Processor Board and CP/M-86

by Bruce Ratoff

Well, by now it seems like you've always had that Z80A running at "4 Meg," and the full 64K of high-speed RAM you got to go with it has collected a nice layer of dust since you haven't changed a board in months. Your bank account is finally recuperating from the purchase of that double sided double density disk system you bought a few months back. Right about now, you're congratulating yourself on finally putting together a state-of-the-art system. Guess again! The 16-bit micros have finally come alive, with enough off-the-shelf hardware and software available to make assembling a 16-bit S-100 system a reasonable project for an experienced microcomputerist.

For the past few months I have had the opportunity to install and use Godbout's 8085/8088 Dual Processor Board with Digital Research's new 8086 implementation of the CP/M operating system. The hardware and software were received in their standard, unconfigured form. I was thus able to experience the installation of this new processor and operating system on an existing system. Through this report, I hope to convey to you my impression of these two powerful and exciting new tools.

A Quick Look

The Godbout Dual processor, as the name implies, contains an 8085 microprocessor for the execution of existing 8080-family software, along with an 8088 microprocessor for the execution of the newer 8086-family software. The system powers up with the 8085 active. By means of a software command, the user may then switch back and forth between it and the 8088. This is accomplished by an input command to an I/O port, whose address is switch selectable on the card. An output to the same I/O port sets the value of extended address lines A16 through A23, allowing the 8085 to overcome its normal 64 kilobyte addressing limits and access all 16 megabytes defined by the IEEE-696 standard. Only the upper four bits of this port are used when the 8088 is active, since this processor has built-in addressing for 1 megabyte.

Bruce R. Ratoff, 26 Broad Street, Cranford, NJ 07016.

The 8085 chip is basically an enhanced 8080, which eliminates the clock generator chip and negative power supply required for an 8080 system. It also practically eliminates the need for an interrupt controller chip in systems requiring interrupts, since input pins and vectoring hardware are provided on the processor for four new interrupts, in addition to the non-vectored interrupt carried over from the original 8080. One of these, the Non-Maskable Interrupt, is brought out to the newly-defined NMI pin of the S-100 bus. The remaining three new interrupts, which are maskable in software, may be jumpered to any of the eight S-100 vectored interrupt pins. These three new interrupts are referred to as RST 5.5, RST 6.5 and RST 7.5, since they generate calls to addresses 4 bytes above the original 8080's RST 5, RST 6 and RST 7 instructions. The 8085 instruction set is identical to that of the 8080, with the addition of two instructions to enable and disable the three new maskable interrupts. It is important to note that the additional Z80 instruction set is *not* implemented. A premium version of the 8085 is used on the Godbout board, allowing operation with a 5 MHz clock rate. A switch is provided to drop the 8085's speed to 2 MHz, to accommodate older (and slower) memory boards.

The 8088 contains pipeline logic which will fetch up to the next four memory bytes while the current instruction is being decoded and executed.

The 8088 microprocessor chip represents Intel's recognition of the large number of microprocessor users who would like to upgrade to a 16-bit microprocessor without having to convert all their 8-bit bus hardware and peripherals. The result is an 8086 processor which has been internally modified to convert each 16-bit memory or port access into two sequential 8-bit accesses.

The 8088 contains pipeline logic which will fetch up to the next four memory bytes while the current instruction is being decoded and executed. Internal operations may therefore proceed at full 16-bit speed, resulting in an overall execution speed almost equivalent to that obtainable on a true 16-bit bus. The bus timing for memory accesses was also made somewhat more liberal, with the result that an 8088 operating a 5 MHz (as on Godbout's board) will work with most memory designed for 2 or 3 MHz 8-bit systems, without the need to add wait states. Godbout apparently found this to be true, since no means is provided to slow the 8088's 5 MHz clock.

CP/M-86 is Digital Research's first venture into the 16-bit micro software market. It implements the same basic file structure, utilities and commands as the current version (2.2) of 8-bit CP/M. Disks written by the two systems are fully interchangeable, as long as the same disk definitions are used in the 8- and 16-bit BIOSes. 8086 equivalents of all the standard CP/M utilities such as ASM, PIP, ED and DDT are provided. Those programs necessary to configure the system (such as the 8086 assembler) are also provided in 8080-executable form. This should allow the use of an existing CP/M-80 system to develop and install a CP/M-86 BIOS. All the CP/M-80 version 2.2 BDOS calls are present and use the same function numbers, easing the task of converting existing programs. New BDOS functions have been added to provide controlled access to the 8086 memory management features.

Testing

Two system configurations were used to test the hardware and software. The main one consisted of a non-front panel enclosure, containing a Vector motherboard, an Imsai SIO2-2 serial interface, an iCom 3712 8-inch single density diskette subsystem, and 64K of various brands and speeds of static RAM. It should be pointed out that some of the memory was already known not to operate with a 4 MHz Z80A. The iCom disk system seemed like a good choice for a first attempt at bringing up CP/M-86, since it used a buffered controller and simple parallel interface with no wait state insertion or special timing requirements. The second test system was an Imsai I8080 front-panel type system, containing the original Imsai motherboard, two SSM I04 I/O boards for serial I/O, 64K of fast static RAM and an Industrial Micro Systems 400 diskette controller. This configuration allowed me to test the Godbout board's operation in the potentially troublesome areas of DMA (on the IMS controller) and front panel operation. Time did not permit installing CP/M-86 on the second system, so the software part of this review is based on operation with the iCom disks only.

Hardware Evaluation

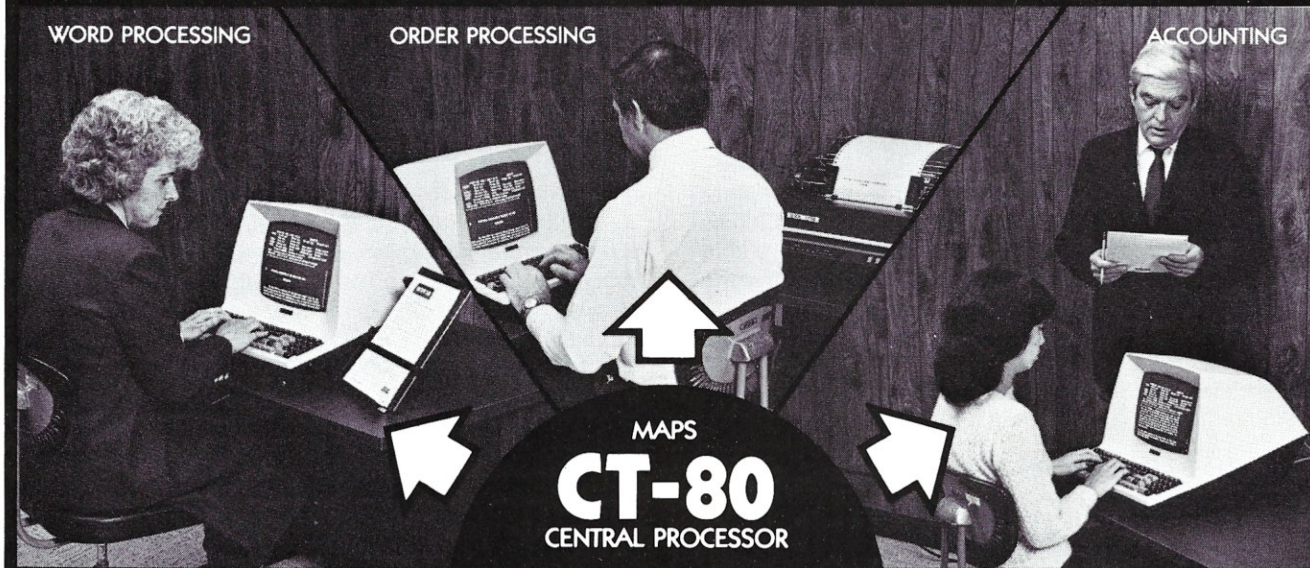
The Godbout board gives a very good first impression as it comes out of its shipping carton. The layout appears clean and open, in spite of the fact that the board contains over 40 IC's. The two five volt regulators sit on the left side (where the vents are on most S-100 cabinets), balanced by the two 40-pin microprocessor IC's on the right. In the upper right corner is a 16-pin DIP socket for the optional connection to a front panel. Card ejectors are provided in the upper corners of the board (I wish more manufacturers would provide these, as they prevent skinned knuckles when changing cards in a tight motherboard). The board is solder-masked on both sides, and appears to have been wave-soldered. The silkscreened legends on the component side of the board identify each IC by both its sequential number in the schematic, as well as its generic type number (7400, 8085, etc.). Each option switch (and there are many) has its function clearly marked. One minor annoyance is the absence of metal "fingers" on the unused S-100 connector pins. The high cost of gold plating has caused a lot of manufacturers to omit these, but the result is that the motherboard sockets become dirty sooner, and the user is prevented from making any hardware modifications that might have required the additional pins.

While there are a great many option switches to be set on this board, most are more or less self-explanatory. In either case, the manual explains them in detail and shows the most common initial setup. A large red toggle switch near the upper right corner of the board selects between 2 MHz and 5 MHz operation of the 8085 processor (the 8088 is fixed at 5 MHz). There are three sets of 8 DIP switches. The one in the bottom row selects the I/O port number used to control the processor. An output to this port sets the extended address lines. An input returns meaningless data, but causes control to switch from the current processor to the other one. I set this to the recommended value of OFD hex. The middle set of switches sets the address for the power-on-jump logic to any 256-byte boundary. I used the address of the disk boot PROM in each of my systems. The last set of switches, located near the top of the board, control miscellaneous options. These include: whether to disable the extended address lines during DMA, whether to clear the extended address lines (to all 0's) at each reset, whether to insert wait states in all I/O operations, whether to reset each processor every time it becomes active or let it continue from where it was, whether to do a jump on reset, whether to do a power-on-jump, and whether to generate the S-100 MWRITE signal. I selected power-on-jump and jump-on-reset in both systems. MWRITE generation was required only in the non-front panel system, since the front panel of my Imsai does its own generation of this signal. I selected the "continue" mode of operation for both processors. However, I did install an additional jumper, described in an addendum to the manual, which allowed the bus reset button to affect both processors, rather than just the 8085. I discovered through experimentation that the I/O wait option was only necessary when operating the 8085 at 5 MHz. All my I/O devices seemed to work fine without wait states when the 8088 was in control.

I was quite pleased with the operation of the board in both systems. Once the correct options were set up, the board performed flawlessly. I have run just about every popular CP/M-based language and package on the 8085 section of the board without any problems. Once potential "catch" concerns operation of the board with DMA devices: due to the manner in which the processor changeover is accomplished, one cannot use the "reset or changeover" option when DMA devices are present, since the DMA is seen as a processor changeover and causes a reset to occur. This should pose no problem in running CP/M-86, since the reset feature is not required.

MAPS®

Multi-Application Processing System



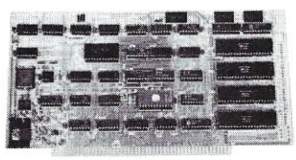
WHY BUY 3 COMPUTER SYSTEMS

When You Can Share Time On 1?

- Three business systems can run concurrently - that's three times the overall system productivity!
- Time share word processing, accounting, order processing, inventory, forms processing, billing and more!
- Three workstations can share data base - preparations can be done by several operators concurrently!
- All workstations can share common peripherals.
- Uses DIGIAC MAPS-80 operating system. (Digital Research MP/M)
- High level language processors including Fortran - Basic - Pascal - Cobol.
- Complete turn-key system for ease of operation and learning.

DIGIAC MAPS® 1000

S-100 COMPATIBLE MP/M SUPPORT MODULE MAKES IT ALL POSSIBLE!



The Digiac MAPS-1000 MP/M Universal Support Module has been designed to meet the total demands required by Digital Research's MP/M multi-user, multi-tasking operating system. All input/output, interrupt generation for task switching and disk bootstrapping are resident functions on the MAPS-1000.

The MAPS-1000 has been designed with all the following powerful features:

● Four (4) independent RS-232C Serial Communication channels ● One 8 bit TTL parallel port ● On-board phantom controlled disk boot prom/monitor ● Power on jump capability ● Crystal controlled MP/M interrupt generation Logic ● On-board extended memory bank switching Logic.

MAPS-1000 Board - Available separately, fully assembled & tested
Price: $425.00

BE SURE TO INQUIRE ABOUT OUR **MAPS**®

MP/M Users Group

MAPS is the only information network providing timely applications support for MP/M users. Members receive the **MAPS Digest**, a quarterly newsletter highlighting practical information exchange between MP/M users. No matter what your application, business or home hobbyist, the **MAPS Digest** provides information at your fingertips. Ask for your subscription!

DIGIAC® CORPORATION

For Additional Information Contact:
MAPS
Commercial Products
Division Of:
DIGIAC CORPORATION
175 Engineers Road
Smithtown, New York
11787
Phone (516) 273-8600

MP/M is a trademark of
Digital Research Corporation

Godbout & CP/M-86 Review, cont'd...

By now I'm sure some of you are saying "but why couldn't they have used a Z80 instead of the 8085?" The reason is simple—there is a great similarity between the timing of the 8085 and 8088 processors. Intel did this to make it easy for their industrial users to adapt existing 8085 designs to the 8088. In the case of the Godbout board, it allows the two processors to share most of the S-100 bus interface logic. Since the timing of the Z80 is vastly different, it probably would have necessitated two totally separate interface circuits, which would not have fit on a single S-100 card. There may be some hope, however, National Semiconductor makes a processor called the NSC800, which they claim has the Z80 instruction set, but timing similar to an 8085. Unfortunately, the NSC800 and the 8085 are not pin-compatible, so some wiring changes would be necessary. Also, the chip seems to be in relatively short supply. Maybe someone at Godbout should be looking into the use of this chip in some future revision to the board (are you listening, Mr. G?).

The first thing that struck me about CP/M-86 was the remarkable degree of similarity to CP/M-80 in both the user and system levels of interface.

There is really only one feature of this board that in my opinion does not live up to expectations. That is the "powerful memory management" alluded to in the company's advertising. What is actually provided on the board would be more accurately called "centralized bank switching." There is a single parallel port with its outputs connected to S-100 address lines 16 through 23 (when the 8085 is in control) or 20 through 23 (when the 8088 is in control). The trouble with this simple scheme is that the output instruction which sets the extended address lines must be executed from a memory card that doesn't recognize the extended address. Otherwise, the program would be knocking its own memory out from under itself! This is not much of a problem when running 8-bit software such as MP/M, which requires some non-banked memory for parts of the operating system anyway. It is also not a serious problem for the 8088, since the CPU directly addresses a megabyte before bank switching is required. The hassle comes when the two processors are used together, if the 8085 needs to access memory above the first 64K to perform some task for the 8088. An example would be the setting up of the 8088's reset vector (at address OFFFFO hex) prior to switching control from 8085 to 8088. The non-extended memory required to perform this operation would require a gap the size of the non-extended card to be left in each 64K of the 8088's one megabyte space, reducing the maximum size of each 8088 memory segment by the size of the non-extended card. A possible solution to the specific problem of starting up the 8088 is to use a PROM monitor in the extended address space. Alternatively, the extended PROM could simply contain a jump instruction to somewhere in the first 64K, making extended references by the 8085 unnecessary. In any event, I would hope that future processors adopt some true form of address translation or mapping so that practical use may be made of the full addressing capabilities of the S-100 bus.

Software Evaluation

The first thing that struck me about CP/M-86 was the remarkable degree of similarity to CP/M-80 in both the user and system level of interface. This consistency helped me to immediately feel at home, in spite of the fact that I was on a brand new processor and operating system. The software comes on two 8 inch single density floppies. A looseleaf binder contains copies of the *CP/M 2.2 Users Guide*, the *ED Users Manual* and *An Introduction to CP/M Features and Facilities*, all of which are the same manuals supplied with the CP/M-80. Three new manuals provided are the *CP/M-86 System Reference Guide*, the *CP/M-86 Assembler Users Guide*, and the *DDT-86 Users Guide*. *The System Reference* appears to be the equivalent of both the "Interface Guide" and "Alteration Guide" found in the CP/M-80 documentation package. These manuals seem to be best organized for looking things up rather than reading straight through. All the necessary information is presented in a well organized manner, with several example programs provided both in the appendices and on the release diskettes. There is a great deal of information presented, but it does all fall into place quickly.

CP/M-86 is larger than CP/M-80, and therefore does not fit on the two system tracks of a standard diskette. Instead, it sits in a file called CPM.SYS. An abbreviated version of the system occupies the system tracks, and is used to load the system file during boot-up. Unlike CP/M-80, the system is not reloaded every time a program exits. Control-C issued to a running program simply causes a return to the CCP prompt. Control-C to the CCP causes the disks to be re-logged in. CP/M-86 takes advantage of the inherent relocatability of 8086 object code. The system may be loaded anywhere in memory without the need for a MOVCPM-like program. The normal procedure is to boot the system into address 00400 hex, just above the 8086 interrupt vector area. This leaves memory from about 02A00 and up free for loading programs.

In CP/M-86, the familiar .COM file type for executable code has been replaced by a new .CMD file type. Besides denoting the presence of 8086 object code rather than 8080, the .CMD file has a header record that describes the program's space requirements for code, data and stack space. This results in much more compact program storage on disk. A new utility called GENCMD is used to create .CMD files from the extended hex (.H86) files produced by the assembler. This replaces the LOAD program found in a CP/M-80 system. The executable files thus produced may use one of three memory configurations: the "8080 model," in which code and data are given a single memory area of up to 64K, the "small model," where two separate areas of up to 64K each are allotted for code and data, or the "compact model," in which up to eight separate memory areas of up to 64K each may be allocated for code and data. The necessary configuration is determined automatically by the system from the information contained in the .CMD header record.

"Magic Typewriter -- the most useful program I ever bought . . ."

MAGIC TYPEWRITER is the reason why I bought a computer.

When Pat Lajko installed my North Star, I told him what features I wanted to use it for. He couldn't find software that would suit my needs, so he had to write it himself. After two years of development, testing, and redevelopment, MAGIC TYPEWRITER is finally being brought to the market.

I wanted more than just another word processing program. I wanted to use my computer to catalog my files, to organize the notes for a trilogy, to keep track of phone numbers and addresses, to catalog my library of books and records, and a thousand other uses. And I wanted all of these functions to be easy to access and easy to use.

MAGIC TYPEWRITER provides all that for me -- and more. It is a system that does not limit my capabilities, but expands them because it provides word-processing and data-base management all in one program. And its command structure is powerful and easy to learn.

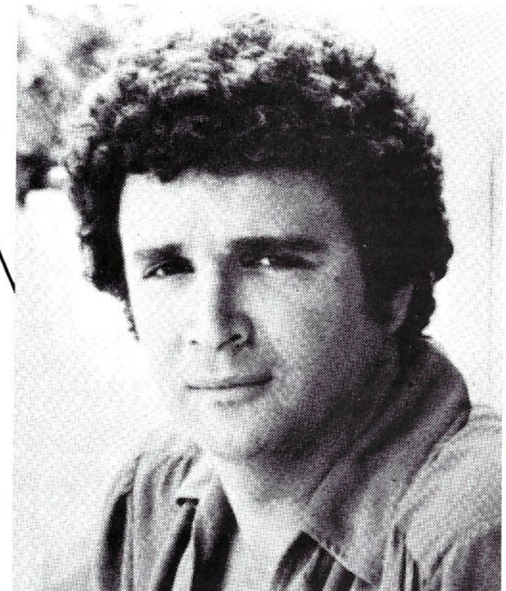
As a word processor, it is a pleasure to use because the system works with me instead of against. I particularly like its formatting functions. I can print out a whole novel with chapter and page numbers, page headings, full right and left justification and customized formatting where necessary. Because MAGIC TYPEWRITER allows for imbedded commands in the text, each file can contain its own instructions for output.

As a data-base, it is equally powerful. MAGIC TYPEWRITER can sort by record, by field, or by characters within a record; it can sort and reformat data on output so the same file can produce customized mailing labels or a sorted phone list. One of its nicest features is that it can retrieve data directly from a disk file in a matter of seconds without disturbing a file in memory. All of your data files are always accessible, even when you are working on something else.

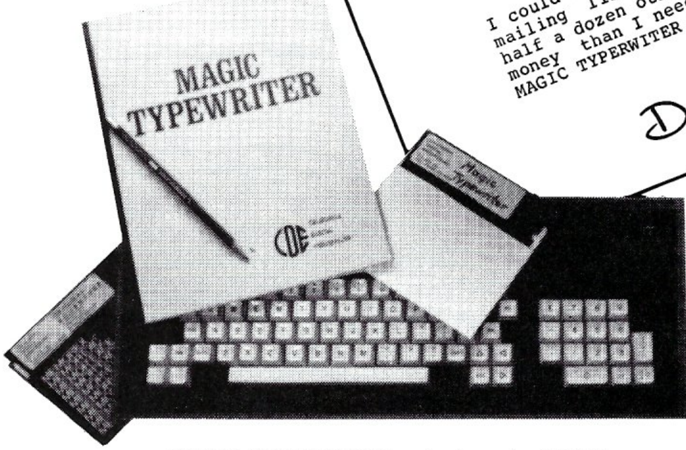
The only difficult part of the system is deciding which of its features I like best. MAGIC TYPEWRITER has too many virtues, but if I had to pick its strongest selling point, it is the fact that it is only $175. Some of the other popular word processing programs can cost up to twice as much. Some of the other popular data-base systems can also cost up to twice as much. MAGIC TYPEWRITER is for the home user who wants all the features of a powerful program without the powerful price. The amazing part of it is that MAGIC TYPEWRITER is extremely compact, requiring only 11K of program space!

I could have bought a lot of other software -- a fast-sort, a mailing list program, a text-formatter, a data-base manager, and half a dozen other special purpose programs -- and paid a lot more money than I needed to. Instead, I chose MAGIC TYPEWRITER -- and MAGIC TYPEWRITER does it all!

David Gerrold



DAVID GERROLD is the author of "The Trouble With Tribbles," an episode of *Star Trek*. He has written almost a dozen novels, including *When Harlie Was One* and *The Man Who Folded Himself*. He has been nominated for the Hugo and Nebula Awards a total of seven times.



MAGIC TYPEWRITER is a trademark of C.D.E.

CALIFORNIA DIGITAL ENGINEERING
P.O. BOX 526 ★ HOLLYWOOD, CA 90028

EXPAND my system's capabilities with ***MAGIC TYPEWRITER.***

Please send ______ copies of MAGIC TYPEWRITER at $175 each. (Include $1.50 shipping. California orders include 6% tax.)

I have enclosed a payment of $__________

☐ Check payable to C.D.E.

☐ Visa ☐ MasterCard MC bank code__________

Exp. Date__________ Card No.__________

Signature__________

NAME__________

ADDRESS__________

CITY__________

STATE__________ ZIP__________

PHONE (______)__________ Ext.______

8" CP/M______ 5" CP/M______ Northstar DOS______

Single Density __________ Double Density __________

Godbout & CP/M-86 Review, cont'd...

The interface between a program and the system has been modified slightly. The page 0 BIOS and BDOS vectors of CP/M-80 have been done away with. Instead, the 8086 software interrupt instruction is used to perform BDOS calls. Since there is no more "warm boot vector" at location 0 for performing direct BIOS calls, a new BDOS function has been added for direct access to all the BIOS routines. The IOBYTE has been moved from location 0003 into the BIOS, with two new calls added to read and set it. Instead of an absolute page 0, the first page of the program's data segment is used by the system to pass the amount of available memory, the default FCB's, and the default I/O buffer. When the "8080 model" configuration is used, this will result in a setup nearly identical to CP/M-80. Due to the absence of a warm boot vector, program termination via "jmp 0" is no longer possible. The program must do a BDOS function 0, or an 8086 "return far" instruction to exit back to the operating system.

CP/M-86 contains added BDOS functions to handle the 8086's memory segmentation features. An added BIOS function allows you specify a table of up to eight non-contiguous areas of memory for programs and data. This allows you to bypass any ROM or other dead blocks in your system. CP/M-86 will then further divide the areas you specify if necessary to provide a total of up to eight separate memory segments. New BDOS calls are provided to allow a program to request additional memory, and to request another program to be loaded. This means that programs may call each other in nested fashion up to eight levels deep.

The CP/M built-in commands remain just about the same as before. DIR, ERA, REN, TYPE and USER operate identically to CP/M-80. The SAVE command has been done away with, however, due to the confusion that it would cause in a segmented memory environment (how would you know which area to save?). Instead of SAVE, a Write command has been added to DDT for saving patched object files. The other noticable difference at the keyboard is that control-P is no longer canceled when a program terminates or control-C is typed. It will remain in effect indefinitely, until another control-P is typed. This greatly improves your ability to get hardcopy of your console output.

I found installing my first CP/M-86 to be much easier than what I recall of my first few attempts with CP/M-80 back in the days of version 1.3. I simply took a listing of my current CP/M-80 BIOS, hand-translated the disk and console portions into 8086 mnemonics, and edited them into the CP/M-86 BIOS skeleton provided on one of the release diskettes. I then used the thoughtfully-provided ASM86.COM to assemble the new BIOS on the 8085 and CP/M-80. Because of the relocatability of 8086 code, there are no equates in the BIOS for memory size (although there is the aforementioned table of available memory areas), and the whole mess of calculating load offsets for DDT has been eliminated. One simply used PIP to concatenate the provided CPM.H86, which contains the CCP and BDOS, with your just-assembled CBIOS.H86. GENCMD.COM, an 8080-executable version of the CP/M-86 program loader, is used to turn the combined hex file into an 8086 object file. At this point came the big question: "Now that I've got it, how do I boot this thing?" This is where having both processors on one board really paid off. I simply wrote a short preamble for CPM.SYS in 8085 code, which set the 8086 reset vector to jump to the 8086 BIOS and then switched processors. Voila! A CP/M-86 system that executes as a CP/M-80 .COM file. As a finishing touch, I would later make this the embedded command in my CP/M-80 system, so that I could appear to boot straight into CP/M-86.

With the details of starting up the system worked out, it was time to begin testing. I keyed in the command "CPM86" (I had saved the 8086 system with the 8085 preamble as CPM86.COM) and waited. In a few seconds, I was quite tickled to see the message:

CP/M-86 Version 1.0
System Generated 03/15/81

and then...

Nothing! The system had printed the signon and then hung up somewhere. Well, let's see. Since the signon printed, the console routines must be working, so the problem must be somewhere in the disk logic, when it goes to log in drive A. The code looks OK, so what am I missing? Wait a minute! Let's have a look at that iCom schematic. Just as I suspected, it's decoding the port number from the upper address bus. This is a common problem on older S-100 boards, where the layout designer took advantage of the fact that the 8080 duplicates the I/O port number on address lines 8 through 15. Most S-100 Z80 cards have extra logic to perform this function, so there's no problem there, but what do you do on a processor like the 8088 that allows port numbers greater than 255? (In fact, the 8088 uses 16 address bits for port numbers, allowing 64K of I/O ports.) Well, back into CP/M-80, and find a way to make it work. Aha! I can write the 8086 code using 16-bit port numbers that have the same lower and upper byte. That should keep all the old boards happy. The only drawback is that to get the 16-bit port numbers requires loading the CX register with the port number before each I/O instruction, since that's the only means provided on the 8088 for accessing the higher port numbers. Anyway, a few quick edits, reassemble and try it again. This time, the system signs on, and I get the familiar "A " prompt. Fantastic! I type "DIR", and the system responds (a bit more rapidly than CP/M-80, I believe) with a proper directory listing. TYPE also seems to be doing its thing. OK, I know the disk read logic must be working, so the next step is to try to write a file. In this case, I tried to PIP something into another file. No go. After I reboot the system, I can see the new name that was created in the directory, so it must be almost working. Examination of the disk write code showed that I had forgotten to pop a register, so I fixed that and tried again. Still just as bad! At this point, I got an object lesson on the effect of the segment registers. I had changed the data segment register in order to obtain the data to be written from the calling program's data segment. Since I forgot to set it back to my own data segment, all further references to my BIOS variables were coming from somewhere south of Lower Slobbovia! Another well-placed push/pop pair and disk writes started behaving themselves. ***There I finally was, with a real live and working CP/M-86 system!*** I then used the working CP/M-86 system to further enhance the CBIOS with a handshaking list driver for my Diablo printer, and various other minor bells and whistles. Once I had set up

CPM86.COM for auto-execute from CP/M-80, I was ready to log some program development time.

The difference in speed between the .COM and .CMD versions of ASM86 was immediately noticeable, although not quite as great as I would have expected. GENCMD was drastically improved, with the .COM version seeming to take forever, while the .CMD was about as fast as the CP/M-80 LOAD program. ED, PIP and STAT all seemed slightly faster, while SUBMIT seemed about the same. One can reasonably assume that compute-bound programs will benefit the most, especially if they are partially rewritten to take advantage of the 8088's added instruction set. Disk-bound programs are of course limited by the disk transfer rate and won't show much improvement.

As a final example, I converted my Super Directory program from the SIG/M library into 8086 code. This program contained many opportunities to take advantage of the 8086, since it contains a character-string sort routine and a large number of 16-bit computations. I recoded the sort routine to use the 8086 string-compare routine, thereby eliminating about twenty lines of code. I changed the decimal output routine to use the hardware divide instruction, shortening that code. The ability to store constants directly into memory, as well as the ability to increment and decrement memory directly, without the use of a pointer register, were very useful throughout the program. The index registers and multiple bit shifts were also put to good use. The end result of my work on CBIOS and SD appears at the end of this article, and will be available on a SIG/M library diskette at some later date as part of a collection of 8086 programs.

The one program which requires a bit of getting used to is the 8086 assembler, ASM86. As was stated in the manuals, this assembler is mostly faithful to the Intel standard in mnemonics and basic design. The main area of deviation is that inter-segment jumps, calls and returns have unique mnemonics rather than being detected automatically. The tricky part of the Intel standard is that the code generated when a particular identifier is used depends on how that identifier was defined. If it was defined by an EQU, for example, it is treated as a numeric literal and generates an immediate-mode instruction. The label of a DB instruction causes an 8-bit instruction to be generated wherever it is referenced, while DW's cause 16-bit instructions to be generated. Code labels cannot be used in data-reference instructions, and will produce an error message from the assembler. One "feature" which does not seem to be mentioned in the manuals is that code labels must be followed by a colon (:), while data labels *must not* be, and will cause error messages at every reference to that label. While this is no problem when writing new code, it caused a bit of head-scratching at first when converting existing programs. Also, for some reason the "jump carry" (jc) and "jump not carry" (jnc) opcodes seem to be missing from the assembler. Once again, this is only a problem with existing code, since the synonyms "jump below" (jb) and "jump above or equal" (jae) are present and work properly.

The dual processor board makes it possible to step up to 16 bits without sacrificing any existing hardware, or having to swap CPU cards to run 8 bit software.

Conclusions

In spite of some of the minor problems mentioned here, both the hardware and software tested appear to be solid, reliable tools which may be had at a very reasonable cost. The dual board makes it possible to step up to 16 bits without sacrificing any existing hardware, or having to swap CPU cards to run 8-bit software. Likewise, CP/M-86 allows a smooth upgrade to 16-bit programming without the need to learn a totally new operating environment. Given the similarity between the 8086/88 and 8080/Z80 architectures, combined with the familiarity of CP/M, most programmers and their software should make the transition with ease. Digital Research is to be congratulated for once again providing a standard-setting product that will provide a consolidated market for the software of the 1980's.

With these products and the others which will now surely follow, 16-bit computing has finally arrived!

Programs begin on page 40.

```
                      title   'Customized Basic I/O System'

                ;*********************************************
                ;*                                           *
                ;* This Customized BIOS adapts CP/M-86 to    *
                ;* the following hardware configuration      *
                ;*     Processor:  8085/8088 Dual Processor  *
                ;*     Brand:  CompuPro (Godbout)            *
                ;*     Controller:  iCom 3712                *
                ;*                                           *
                ;*                                           *
                ;*     Programmer:  Bruce R. Ratoff          *
                ;*     Revisions :  04/30/81 20:40           *
                ;*                                           *
                ;*********************************************

FFFF            true            equ -1
0000            false           equ not true
000D            cr              equ 0dh ;carriage return
000A            lf              equ 0ah ;line feed

                ;*********************************************
                ;*                                           *
                ;* Loader bios is true if assembling the     *
                ;* LOADER BIOS, otherwise BIOS is for the    *
                ;* CPM.SYS file.                             *
                ;*                                           *
                ;*********************************************

0000            loader bios     equ false
00E0            bdos int        equ 224 ;reserved BDOS interrupt

                ;*********************************************
                ;*                                           *
                ;*          I/O Port Assignments             *
                ;*                                           *
                ;*********************************************
                ;
                ;Diskette interface (iCom 3712)
                ;Note: Port numbers are "doubled up" because iCom card
                ;       counts on 8080 "address mirror" effect.
C0C0            datai   equ     0c0c0h  ;data/status input port
C1C1            datao   equ     0c1c1h  ;data output port
C0C0            cntrl   equ     0c0c0h  ;command output port
                ;
                ;Console interface (IMSAI SIO2-2 port 1)
0003            cstat   equ     3       ;status
0002            cdata   equ     2       ;data
0002            cimsk   equ     2       ;input ready mask
0001            comsk   equ     1       ;output ready mask
                ;
                ;Printer interface (IMSAI SIO2-2 port 2)
0005            lstat   equ     5       ;status
0004            ldata   equ     4       ;data
0001            lomsk   equ     1       ;output ready mask
0002            limsk   equ     2       ;input ready mask

                        IF      not loader bios
                ;---------------------------------------------
                ;|                                           |
2500            bios code       equ 2500h
0000            ccp offset      equ 0000h
0B06            bdos ofst       equ 0B06h ;BDOS entry point
                ;|                                           |
                ;---------------------------------------------
                        ENDIF   ;not loader bios

                        IF      loader bios
                ;---------------------------------------------
                ;|                                           |
                bios code       equ 1200h ;start of LDBIOS
                ccp offset      equ 0003h ;base of CPMLOADER
                bdos ofst       equ 0406h ;stripped BDOS entry
                ;|                                           |
                ;---------------------------------------------
                        ENDIF   ;loader bios

                        cseg
```

```
                                org     ccpoffset
                        ccp:
                                org     bios code

                        ;*********************************************
                        ;*                                           *
                        ;* BIOS Jump Vector for Individual Routines  *
                        ;*                                           *
                        ;*********************************************

2500 E9 3C 00            jmp INIT        ;Enter from BOOT ROM or LOADER
2503 E9 85 00            jmp WBOOT       ;Arrive here from BDOS call 0
2506 E9 C8 00            jmp CONST       ;return console keyboard status
2509 E9 CE 00            jmp CONIN       ;return console keyboard char
250C E9 D5 00            jmp CONOUT      ;write char to console device
250F E9 DD 00            jmp LISTOUT     ;write character to list device
2512 E9 20 01            jmp PUNCH       ;write character to punch device
2515 E9 1E 01            jmp READER      ;return char from reader device
2518 E9 54 01            jmp HOME        ;move to trk 00 on cur sel drive
251B E9 32 01            jmp SELDSK      ;select disk for next rd/write
251E E9 51 01            jmp SETTRK      ;set track for next rd/write
2521 E9 58 01            jmp SETSEC      ;set sector for next rd/write
2524 E9 61 01            jmp SETDMA      ;set offset for user buff (DMA)
2527 E9 6C 01            jmp READ        ;read a 128 byte sector
252A E9 AD 01            jmp WRITE       ;write a 128 byte sector
252D E9 DC 00            jmp LISTST      ;return list status
2530 E9 4E 01            jmp SECTRAN     ;xlate logical->physical sector
2533 E9 57 01            jmp SETDMAB     ;set seg base for buff (DMA)
2536 E9 59 01            jmp GETSEGT     ;return offset of Mem Desc Table
2539 E9 FD 00            jmp GETIOBF     ;return I/O map byte (IOBYTE)
253C E9 FE 00            jmp SETIOBF     ;set I/O map byte (IOBYTE)

                        ;*********************************************
                        ;*                                           *
                        ;* INIT Entry Point, Differs for LDBIOS and  *
                        ;* BIOS, according to "Loader Bios" value    *
                        ;*                                           *
                        ;*********************************************

                        INIT:   ;print signon message and initialize hardware
253F 8C C8                      mov ax,cs       ;we entered with a JMPF so use
2541 8E D0                      mov ss,ax       ;CS: as the initial value of SS:,
2543 8E D8                      mov ds,ax       ;DS:,
2545 8E C0                      mov es,ax       ;and ES:
                                ;use local stack during initialization
2547 BC BE 29                   mov sp,offset stkbase
254A FC                         cld             ;set forward direction

                                IF      not loader bios
                        ;---------------------------------------------
                        ;|                                           |
                                ; This is a BIOS for the CPM.SYS file.
                                ; Setup all interrupt vectors in low
                                ; memory to address trap

254B 1E                         push ds         ;save the DS register
254C C6 06 8C 27 00             mov IOBYTE,0    ;clear IOBYTE
2551 B8 00 00                   mov ax,0
2554 8E D8                      mov ds,ax
2556 8E C0                      mov es,ax       ;set ES and DS to zero
                                ;setup interrupt 0 to address trap routine
2558 C7 06 00 00 9A 25          mov int0 offset,offset int trap
255E 8C 0E 02 00                mov int0 segment,CS
2562 BF 04 00                   mov di,4
2565 BE 00 00                   mov si,0        ;then propagate
2568 B9 FE 01                   mov cx,510      ;trap vector to
256B F3 A5                      rep movs ax,ax  ;all 256 interrupts
                                ;BDOS offset to proper interrupt
256D C7 06 80 03 06 0B          mov bdos offset,bdos ofst
2573 C7 06 00 00 8E 25          mov int0 offset,offset int0 trap
2579 C7 06 10 00 94 25          mov int4 offset,offset int4 trap
257F 1F                         pop ds          ;restore the DS register

                        ;       (additional CP/M-86 initialization)
                        ;|                                           |
                        ;---------------------------------------------
                                ENDIF   ;not loader bios
```

```
                    IF      loader bios
               ;--------------------------------------------
               ;|                                          |
                    ;This is a BIOS for the LOADER
                    push ds         ;save data segment
                    mov ax,0
                    mov ds,ax       ;point to segment zero
                    ;BDOS interrupt offset
                    mov bdos offset,bdos ofst
                    mov bdos segment,CS ;bdos interrupt segment
               ;    (additional LOADER initialization)
                    pop ds          ;restore data segment
               ;|                                          |
               ;--------------------------------------------
                    ENDIF   ;loader bios

2580 BB 97 27       mov bx,offset signon
2583 E8 BC 00       call pmsg       ;print signon message
2586 B1 00          mov cl,0        ;default to dr A: on coldstart
2588 E9 75 DA       jmp ccp         ;jump to cold start entry of CCP

258B E9 78 DA  WBOOT: jmp ccp+6     ;direct entry to CCP at command level

                    IF      not loader bios
               ;--------------------------------------------
               ;|                                          |
               int0 trap:
258E FA             cli
258F BB F4 27       mov bx,offset int0 trp
2592 EB 0A          jmps int halt
               int4 trap:
2594 FA             cli
2595 BB 0B 28       mov bx,offset int4 trp
2598 EB 04          jmps int halt
               int trap:
259A FA             cli             ;block interrupts
259B BB DA 27       mov bx,offset int trp
               int halt:
259E 8C C8          mov ax,cs
25A0 8E D8          mov ds,ax       ;get our data segment
25A2 E8 9D 00       call pmsg
25A5 5B             pop bx          ;get offset
25A6 58             pop ax          ;print segment
25A7 53             push bx         ;save offset
25A8 E8 0A 00       call PHEX
25AB B1 3A          mov cl,':'      ;colon
25AD E8 34 00       call CONOUT
25B0 58             pop ax          ;print offset
25B1 E8 01 00       call PHEX
25B4 F4             hlt             ;hardstop

               PHEX:
25B5 50             push ax
25B6 8A C4          mov al,ah
25B8 E8 01 00       call PHXB       ;print upper byte
25BB 58             pop ax          ;restore to print lower byte
               PHXB:
25BC 50             push ax         ;save byte
25BD B1 04          mov cl,4        ;get high nibble
25BF D2 E8          shr al,cl       ;into low bits
25C1 E8 03 00       call PHXD       ;print digit
25C4 58             pop ax          ;restore byte
25C5 24 0F          and al,0fh      ;isolate low nibble
               PHXD:
25C7 04 90          add al,90h      ;first half of conversion trick
25C9 27             daa
25CA 14 40          adc al,40h      ;second half of same
25CC 27             daa
25CD 8A C8          mov cl,al       ;now print digit
25CF EB 13          jmps CONOUT
               ;|                                          |
               ;--------------------------------------------
                    ENDIF   ;not loader bios

               ;********************************************
               ;*                                          *
               ;*   CP/M Character I/O Interface Routines  *
               ;*                                          *
               ;********************************************
```

```
                  CONST:          ;console status
25D1 E4 03             in      al,cstat        ;get status byte
25D3 24 02             and     al,cimsk        ;check input mask
25D5 74 02             jz      const1          ;not ready yet...return al=0, ZF=1
25D7 0C FF             or      al,0ffh         ;ready...return al=0FFh, ZF=0
                  CONST1:
25D9 C3                ret

                  CONIN:                  ;console input
25DA E8 F4 FF          call CONST
25DD 74 FB             jz CONIN        ;wait for RDA
25DF E4 02             in      al,cdata;get byte
25E1 24 7F             and     al,7fh  ;strip parity
25E3 C3                ret

                  CONOUT:         ;console output
25E4 E4 03             in      al,cstat        ;get status
25E6 A8 01             test    al,comsk        ;check output bits
25E8 74 FA             jz      conout          ;loop till ready
25EA 8A C1             mov     al,cl           ;setup
25EC E6 02             out     cdata,al        ;send character
25EE C3                ret             ;then return data

                  LISTOUT:                ;list device output
25EF E8 1A 00          call    LISTST  ;get output status
25F2 74 FB             jz      LISTOUT ;wait for TBE
25F4 8A C1             mov     al,cl   ;setup
25F6 E6 04             out     ldata,al        ;send char
25F8 E4 05             in      al,lstat        ;check for handshake received
25FA 24 02             and     al,limsk
25FC 74 0D             jz      LISTOUT2        ;no handshake...exit
25FE E4 04             in      al,ldata        ;get handshake char
2600 24 7F             and     al,7fh          ;strip parity
2602 3C 13             cmp     al,'S'-40h      ;XOFF?
2604 75 05             jnz     LISTOUT2        ;nope
2606 C6 06 8B 27 FF    mov     lstactive,0ffh  ;set list active flag
                  LISTOUT2:
260B C3                ret

                  LISTST:                 ;poll list status
260C E4 05             in      al,lstat        ;get status byte
260E 24 01             and     al,lomsk        ;test output bits
2610 74 20             jz      LISTST1         ;not ready...exit with al=0, zf=1
2612 A0 8B 27          mov     al,lstactive    ;line ready...waiting for XON?
2615 F6 D0             not     al
2617 84 C0             test    al,al
2619 75 17             jnz     LISTST1         ;not waiting...say ready
261B E4 05             in      al,lstat        ;check for handshake
261D 24 02             and     al,limsk
261F 74 11             jz      LISTST1         ;not yet...say still busy
2621 E4 04             in      al,ldata        ;got something...
2623 24 7F             and     al,7fh          ;strip parity
2625 3C 11             cmp     al,'Q'-40h      ;is it XON?
2627 B0 00             mov     al,0
2629 75 07             jnz     LISTST1         ;no, return false
262B F6 D0             not     al              ;ready...exit with al=0ffh, zf=0
262D C6 06 8B 27 00    mov     lstactive,0     ;clear list active flag
                  LISTST1:
2632 84 C0             test    al,al           ;make sure flags are set
2634 C3                ret

                  PUNCH:          ;write punch device
2635 C3                ret     ;is a "bit bucket"

                  READER:
2636 B0 1A             mov     al,1ah  ;is an EOF source
2638 C3                ret

                  GETIOBF:
2639 A0 8C 27          mov al,IOBYTE
263C C3                ret

                  SETIOBF:
263D 88 0E 8C 27       mov IOBYTE,cl   ;set iobyte
2641 C3                ret             ;iobyte not implemented

                  pmsg:
2642 8A 07             mov al,[BX]     ;get next char from message
2644 84 C0             test al,al
```

```
2646 74 26              jz return       ;if zero return
2648 8A C8              mov CL,AL
264A E8 97 FF           call CONOUT     ;print it
264D 43                 inc BX
264E EB F2              jmps pmsg       ;next character and loop

                ;*********************************************
                ;*                                           *
                ;*        Disk Input/Output Routines         *
                ;*                                           *
                ;*********************************************

                SELDSK:         ;select disk given by register CL
 0002           ndisks  equ     2 ;number of disks (up to 16)
2650 C6 06 8D 27 FF     mov seekfg,0ffh ;set seek flag
2655 88 0E 8E 27        mov disk,cl     ;save disk number
2659 BB 00 00           mov bx,0000h    ;ready for error return
265C 80 F9 02           cmp cl,ndisks   ;n beyond max disks?
265F 73 0D              jnb return      ;return if so
2661 B5 00              mov ch,0        ;double(n)
2663 8B D9              mov bx,cx       ;bx = n
2665 B1 04              mov cl,4        ;ready for *16
2667 D3 E3              shl bx,cl       ;n = n * 16
2669 B9 56 28           mov cx,offset dpbase
266C 03 D9              add bx,cx       ;dpbase + n * 16
266E C3         return: ret             ;bx = .dph

                HOME:   ;move selected disk to home position (Track 0)
266F B9 00 00           mov cx,0        ;set disk i/o to track zero
                                        ;**** fall through ****
                SETTRK: ;set track address given by CX
2672 89 0E 8F 27        mov trk,CX
2676 C6 06 8D 27 FF     mov seekfg,0ffh ;set seek flag
267B C3                 ret

                SETSEC: ;set sector number given by cx
267C 89 0E 91 27        mov sect,CX
2680 C3                 ret

                SECTRAN: ;translate sector CX using table at [DX]
2681 8B D9              mov bx,cx
2683 03 DA              add bx,dx       ;add sector to tran table address
2685 8A 1F              mov bl,[bx]     ;get logical sector
2687 C3                 ret

                SETDMA: ;set DMA offset given by CX
2688 89 0E 93 27        mov dma adr,CX
268C C3                 ret

                SETDMAB: ;set DMA segment given by CX
268D 89 0E 95 27        mov dma seg,CX
2691 C3                 ret
                ;
                GETSEGT:  ;return address of physical memory table
2692 BB 51 28           mov bx,offset seg table
2695 C3                 ret

                ;*********************************************
                ;*                                           *
                ;*  All disk I/O parameters are setup:       *
                ;*     DISK     is disk number     (SELDSK)  *
                ;*     TRK      is track number    (SETTRK)  *
                ;*     SECT     is sector number   (SETSEC)  *
                ;*     DMA ADR  is the DMA offset  (SETDMA)  *
                ;*     DMA SEG  is the DMA segment (SETDMAB) *
                ;*  READ reads the selected sector to the DMA*
                ;*  address, and WRITE writes the data from  *
                ;*  the DMA address to the selected sector   *
                ;*  (return 00 if successful,  01 if perm err)*
                ;*                                           *
                ;*********************************************

                READ:
2696 B1 0A              mov cl,10       ;set retry count
                READ1:
2698 E8 89 00           call STUP       ;set up unit/track/sector
269B B0 03              mov al,3        ;send read command
269D E8 D5 00           call DLOOP
26A0 BA C0 C0           mov dx,datai    ;set port number
26A3 EC                 in al,dx        ;get back status
26A4 A8 08              test al,8       ;check CRC flag
26A6 74 16              jz RDOK         ;no error...go get data
26A8 FE C9              dec cl          ;got an error...count retrys
26AA 75 03              jnz READ2       ;some retrys left...continue
26AC B0 01              mov al,1        ;bad news....return error
26AE C3                 ret
                READ2:
26AF F6 C1 03           test cl,3       ;time for a re-seek?
26B2 7B E4              jpo READ1       ;no, just reread
26B4 C6 06 8D 27 FF     mov seekfg,0ffh ;yes, set seek flag
26B9 E8 C6 00           call RESET      ;clear errors, home drive
26BC EB DA              jmps READ1      ;try read again
                RDOK:
26BE B9 80 00           mov cx,128      ;set byte counter
26C1 FC                 cld             ;set forward direction
26C2 06                 push es         ;save extra segment
26C3 C4 3E 93 27        les di,dword ptr dma adr        ;set dest index and segment
26C7 BA C0 C0           mov dx,cntrl
                RDLUP:
26CA B8 40 00           mov ax,40h      ;send "examine read buffer" command
26CD EE                 out dx,al       ;to disk control port
26CE EC                 in al,dx        ;get data byte
26CF AA                 stos al         ;store it, bump pointer and count
26D0 B0 41              mov al,41h      ;send "step read buffer" command
26D2 EE                 out dx,al       ;to controller
26D3 E2 F5              loop RDLUP      ;repeat 128 times
26D5 07                 pop es          ;restore extra segment
26D6 B0 00              mov al,0        ;return good status
26D8 EE                 out dx,al       ;also put controller in status mode
26D9 C3                 ret

                WRITE:
26DA B9 80 00           mov cx,128      ;set 128 byte counter
26DD FC                 cld             ;set forward direction
26DE 1E                 push ds         ;save current data segment
26DF C5 36 93 27        lds si,dword ptr dma adr        ;set source index
                WRLUP:
26E3 AC                 lods al         ;get next byte
26E4 BA C1 C1           mov dx,datao
26E7 EE                 out dx,al       ;send to controller
26E8 B0 31              mov al,31h      ;send "shift write buffer" command
26EA BA C0 C0           mov dx,cntrl
26ED EE                 out dx,al       ;to controller
26EE B0 00              mov al,0        ;remove command
26F0 EE                 out dx,al       ;(bit 0 must toggle to be seen)
26F1 E2 F0              loop WRLUP      ;repeat for sector length times
26F3 1F                 pop ds
                RTRYP:
26F4 E8 2D 00           call STUP       ;setup for write
26F7 EC                 in al,dx        ;check controller status
26F8 A8 10              test al,10h     ;write protected?
26FA 74 08              jz TRYWR        ;no, continue
26FC BB 36 28           mov bx,offset prtmsg    ;say "protected"
26FF E8 0A 00           call ERROR      ;and wait for user action
2702 EB F0              jmps RTRYP      ;retry if user hits return key
                TRYWR:
2704 B0 05              mov al,5        ;send write command
2706 E8 6C 00           call DLOOP      ;to controller with wait
                WROK:
2709 B0 00              mov al,0        ;return good status
270B C3                 ret

                ;*********************************************
                ;*                                           *
                ;*          Disk Utility Routines            *
                ;*                                           *
                ;*********************************************
                ;
                ;print an error message and wait for user response
                ;if control-c, then abort to cp/m, else return
                ;to caller and (usually) retry operation
                ERROR:
270C E8 33 FF           call PMSG       ;print an error message
270F E8 C8 FE           call CONIN      ;wait for user response
2712 50                 push ax         ;save character
2713 BB 4E 28           mov bx,offset crlf      ;echo cr, lf
2716 E8 29 FF           call PMSG
2719 58                 pop ax          ;now look at char
```

```
271A 3C 03                  cmp al,3        ;control-c?
271C 74 01                  jz ERR1         ;yes, return to cp/m
271E C3                     ret             ;else retry error'd operation
                    ERR1:
271F B1 00                  mov cl,0        ;tell cp/m user 0, drive A
2721 E9 DC D8               jmp ccp         ;bye-bye

                    ;Perform select and possibly seek logic for either a
                    ;read or write operation.
                    STUP:
2724 B0 0B                  mov al,0bh      ;issue "reset errors" command
2726 E8 4C 00               call DLOOP      ;to controller with wait
2729 A0 8E 27               mov al,disk     ;get drive number
272C B1 06                  mov cl,6        ;prepare to shift into
272E D3 E0                  shl ax,cl       ;high 2 bits of cmd byte
2730 0B 06 91 27            or ax,sect      ;put sector number in low bits
2734 BA C1 C1               mov dx,datao
2737 EE                     out dx,al       ;send to controller
2738 B0 21                  mov al,21h      ;issue "set unit/sector" command
273A E8 38 00               call DLOOP
273D BA C0 C0               mov dx,datai
2740 B9 64 00               mov cx,100      ;set up delay loop
                    STUP0:
2743 BB 40 1F               mov bx,8000     ;inner delay loop
                    STUP1:
2746 EC                     in al,dx        ;get controller status
2747 A8 20                  test al,20h     ;check "drive fail" (ready) flag
2749 74 0D                  jz STUP2        ;no problem...continue
274B 4B                     dec bx          ;count down inner delay loop
274C 75 F8                  jnz STUP1
274E E2 F3                  loop STUP0      ;count down outer delay loop
2750 BB 24 28               mov bx,offset rdymsg    ;timed out...complain
2753 E8 B6 FF               call error      ;and wait for response
2756 EB CC                  jmps STUP       ;retry the whole mess
                    STUP2:
2758 B0 00                  mov al,0        ;clear seek flag
275A 86 06 8D 27            xchg al,seekfg  ;and fetch previous value
275E 84 C0                  test al,al      ;was it set?
2760 75 01                  jnz stup3       ;yes, go do seek or home
2762 C3                     ret             ;no seek needed...exit
                    STUP3:
2763 A1 8F 27               mov ax,trk      ;look at track number
2766 84 C0                  test al,al      ;is it 0?
2768 74 18                  jz RESET        ;yes, do a home
276A BA C1 C1               mov dx,datao
276D EE                     out dx,al       ;otherwise, set new track
276E B0 11                  mov al,11h      ;give "set track" command
2770 E8 02 00               call DLOOP
2773 B0 09                  mov al,9        ;then give "seek" command
                                            ;**** fall through ****
                    ;
                    ;This routine issues a controller command and waits for
                    completion
                    DLOOP:
2775 BA C0 C0               mov dx,cntrl
2778 EE                     out dx,al       ;send command
2779 B0 00                  mov al,0        ;strobe it off
277B EE                     out dx,al
                    LOOP1:
277C EC                     in al,dx        ;get controller status
277D A8 01                  test al,1       ;check ready bit
277F 75 FB                  jnz LOOP1       ;loop till ready
2781 C3                     ret             ;then exit
                    ;
                    ;This routine issues a "clear" command followed by a "home"
                    command
                    RESET:
2782 B0 81                  mov al,81h      ;send "clear"
2784 E8 EE FF               call DLOOP
2787 B0 0D                  mov al,0dh      ;send "home"
2789 EB EA                  jmps DLOOP


                    ;*********************************************
                    ;*                                           *
                    ;*               Data Areas                  *
                    ;*                                           *
                    ;*********************************************
```

```
278B                    data offset     equ offset $

                                dseg
                                org     data offset     ;contiguous with code segment
278B 00                 lstactive       db      0       ;set if list handshake active
278C 00                 IOBYTE  db      0       ;i/o assignments (unused at present)
278D 00                 seekfg  db      0       ;set to 0ffh if next access requires seek
278E 00                 disk    db      0       ;disk number
278F 00 00              trk     dw      0       ;track number
2791 00 00              sect    dw      0       ;sector number
2793 00 00              dma adr dw      0       ;DMA offset from DS
2795 00 00              dma seg dw      0       ;DMA Base Segment

2797 0D 0A 0D 0A        signon  db      cr,lf,cr,lf
279B 43 50 2F 4D 2D 38          db      'CP/M-86 Version 1.0 for iCom 3712',cr,lf
     36 20 56 65 72 73
     69 6F 6E 20 31 2E
     30 20 66 6F 72 20
     69 43 6F 6D 20 33
     37 31 32 0D 0A
27BE 53 79 73 74 65 6D          db      'System Generated 04/30/81'
     20 47 65 6E 65 72
     61 74 65 64 20 30
     34 2F 33 30 2F 38
     31
27D7 0D 0A 00                   db      cr,lf,0

27DA 0D 0A              int trp db      cr,lf
27DC 49 6E 74 65 72 72          db      'Interrupt Trap Halt at ',0
     75 70 74 20 54 72
     61 70 20 48 61 6C
     74 20 61 74 20 00

27F4 0D 0A              int0 trp db     cr,lf
27F6 44 69 76 69 64 65          db      'Divide Trap Halt at ',0
     20 54 72 61 70 20
     48 61 6C 74 20 61
     74 20 00

280B 0D 0A              int4 trp db     cr,lf
280D 4F 76 65 72 66 6C          db      'Overflow Trap Halt at ',0
     6F 77 20 54 72 61
     70 20 48 61 6C 74
     20 61 74 20 00

2824 0D 0A              rdymsg  db      cr,lf
2826 44 72 69 76 65 20          db      'Drive not ready',0
     6E 6F 74 20 72 65
     61 64 79 00

2836 0D 0A              prtmsg  db      cr,lf
2838 44 72 69 76 65 20          db      'Drive write protected',0
     77 72 69 74 65 20
     70 72 6F 74 65 63
     74 65 64 00

284E 0D 0A 00           crlf    db      cr,lf,0

                        ;       System Memory Segment Table

2851 01                 segtable db 1   ;1 segment
2852 DC 02                      dw tpa seg      ;1st seg starts after BIOS
2854 23 0D                      dw tpa len      ;and extends to 0ffff

=                               include singles.lib ;read in disk definitions
=                       ;               DISKS 2
=   2856                dpbase  equ     $               ;Base of Disk Parameter Blocks
= 2856 85 28 00 00      dpe0    dw      xlt0,0000h      ;Translate Table
= 285A 00 00 00 00              dw      0000h,0000h     ;Scratch Area
= 285E 9F 28 76 28              dw      dirbuf,dpb0     ;Dir Buff, Parm Block
= 2862 3E 29 1F 29              dw      csv0,alv0       ;Check, Alloc Vectors
= 2866 85 28 00 00      dpe1    dw      xlt1,0000h      ;Translate Table
= 286A 00 00 00 00              dw      0000h,0000h     ;Scratch Area
= 286E 9F 28 76 28              dw      dirbuf,dpb1     ;Dir Buff, Parm Block
= 2872 6D 29 4E 29              dw      csv1,alv1       ;Check, Alloc Vectors
=                       ;               DISKDEF 0,1,26,6,1024,243,64,64,2
=                       ;
=                       ;        1944:  128 Byte Record Capacity
=                       ;         243:  Kilobyte Drive  Capacity
```

```
=                       ;           64:   32 Byte Directory Entries
=                       ;           64:   Checked Directory Entries
=                       ;          128:   Records / Extent
=                       ;            8:   Records / Block
=                       ;           26:   Sectors / Track
=                       ;            2:   Reserved  Tracks
=                       ;            6:   Sector Skew Factor
=                       ;
=   2876                dpb0    equ       offset $          ;Disk Parameter Block
= 2876 1A 00                    dw        26                ;Sectors Per Track
= 2878 03                       db        3                 ;Block Shift
= 2879 07                       db        7                 ;Block Mask
= 287A 00                       db        0                 ;Extnt Mask
= 287B F2 00                    dw        242               ;Disk Size - 1
= 287D 3F 00                    dw        63                ;Directory Max
= 287F C0                       db        192               ;Alloc0
= 2880 00                       db        0                 ;Alloc1
= 2881 10 00                    dw        16                ;Check Size
= 2883 02 00                    dw        2                 ;Offset
=   2885                xlt0    equ       offset $          ;Translate Table
= 2885 01 07 0D 13              db        1,7,13,19
= 2889 19 05 0B 11              db        25,5,11,17
= 288D 17 03 09 0F              db        23,3,9,15
= 2891 15 02 08 0E              db        21,2,8,14
= 2895 14 1A 06 0C              db        20,26,6,12
= 2899 12 18 04 0A              db        18,24,4,10
= 289D 10 16                    db        16,22
=   001F                als0    equ       31                ;Allocation Vector Size
=   0010                css0    equ       16                ;Check Vector Size
=                       ;                 DISKDEF 1,0
=                       ;
=                       ;       Disk 1 is the same as Disk 0
=                       ;
=   2876                dpb1    equ       dpb0              ;Equivalent Parameters
=   001F                als1    equ       als0              ;Same Allocation Vector Size
=   0010                css1    equ       css0              ;Same Checksum Vector Size
=   2885                xlt1    equ       xlt0              ;Same Translate Table
=                       ;                 ENDEF
=                       ;
=                       ;       Uninitialized Scratch Memory Follows:
=                       ;
=   289F                begdat  equ       offset $          ;Start of Scratch Area
= 289F                  dirbuf  rs        128               ;Directory Buffer
= 291F                  alv0    rs        als0              ;Alloc Vector
= 293E                  csv0    rs        css0              ;Check Vector
= 294E                  alv1    rs        als1              ;Alloc Vector
= 296D                  csv1    rs        css1              ;Check Vector
=   297D                enddat  equ       offset $          ;End of Scratch Area
=   00DE                datsiz  equ       offset $-begdat ;Size of Scratch Area
= 297D 00                       db        0                 ;Marks End of Module
  297E                  loc stk rw  32  ;local stack for initialization
  29BE                  stkbase equ offset $

  29BE                  lastoff equ offset $
  02DC                  tpa seg equ (lastoff+0400h+15) / 16
  0D23                  tpa len equ 0fffh - tpa seg      ; 64K less 16 byte reset
                                                           vector less cp/m size

29BE 00                         db 0    ;fill last address for GENCMD

                        ;*********************************************
                        ;*                                           *
                        ;*           Dummy Data Section              *
                        ;*                                           *
                        ;*********************************************
  0000                          dseg    0         ;absolute low memory
                                org     0         ;(interrupt vectors)
0000                    int0 offset     rw        1
0002                    int0 segment    rw        1
                        ;       pad to overflow trap vector
0004                                    rw        6
0010                    int4 offset     rw        1
0012                    int4 segment    rw        1
                        ;       pad to system call vector
0014                            rw      2*(bdos int-5)

0380                    bdos offset     rw        1
0382                    bdos segment    rw        1
                                END
```

```
                        ;
                        ;                        SD.A86
                        ;                  (revised 05/05/81)
                        ;
                        ;                 SUPER DIRECTORY PROGRAM
                        ;                   by Bruce R. Ratoff
                        ;
                        ;Displays the directory of a CP/M disk, sorted alphabetically,
                        ;with the file size in K, rounded to the nearest CP/M block size.
                        ;
                        ;This latest variation on a common theme will automatically adjust
                        ;itself for any block size and directory length.  If the screen fills,
                        ;the program will pause until a key is struck (see NPL and LPS equates
                        ;below).  Total space used and number of files are printed at end.
                        ;
                        ;Command: SD FILENAME.FILETYPE or just SD
                        ;
                        ;Allows '*' or '?' type specifications.  Drive name may also be
                        ;specified.  Ignores "SYS" files unless SOPT is TRUE and 'S' option
                        ;is given (i.e., SD *.* S will print all files).
                        ;
                        ;05/05/81 Fixed division overflow problem in decimal output routine.
                         (BRR)
                        ;
                        ;05/03/81 First 8086 version. (Bruce R. Ratoff)
                        ;
                        ;Based on 'DIRS' by Keith Petersen, W8SDZ
                        ;
                        ;
0000                    FALSE   EQU     0               ;DEFINE LOGICAL FALSE
FFFF                    TRUE    EQU     NOT FALSE       ;DEFINE LOGICAL TRUE
                        ;
FFFF                    SOPT    EQU     TRUE    ;PUT TRUE TO ALLOW 'DIR *.* S' FORM
FFFF                    WIDE    EQU     true    ;PUT TRUE TO ALLOW 4 NAMES ACROSS
FFFF                    user    equ     true    ;print user numbers for cp/m 2.x also?
                        ;
                        ;
                        ;
0000                    BASE    EQU     0
0100                    TPA     EQU     100H
                        ;
005C                    FCB     EQU     5CH
                        ;
                                IF      WIDE
0004                    NPL     EQU     4       ;NUMBER OF NAMES PER LINE
                                ENDIF
                        ;
                                IF      NOT WIDE
                        NPL     EQU     3       ;NUMBER OF NAMES PER LINE
                                ENDIF
                        ;
0017                    LPS     EQU     23      ;NUMBER OF LINES PER SCREEN
003A                    DELIM   EQU     ':'     ;FENCE (DELIMITER) CHARACTER
                        ;
                                org     TPA
                        ;
                        START:
0100 FC                         cld
                        ;
                                IF      SOPT
0101 A0 6D 00                   mov     al,byte ptr .FCB+17     ;SAVE S OPTION FLAG
0104 2E A2 4A 05                mov     SOPFLG,al       ;(BLANK OR LETTER S)
                                ENDIF
                        ;
0108 2E C6 06 4B 05 00          mov     USERNO,0        ;DEFAULT TO USER 0
010E 2E C6 06 3D 05 00          mov     LINCNT,0        ;CLEAR COUNT OF LINES ON SCREEN
0114 B1 0C                      mov     cl,12
0116 E8 15 04                   CALL    BDOS    ;CHECK CP/M VERSION
0119 2E 89 1E 4E 05             mov     word ptr VERFLG,bx  ;LO ORD >0 IF 2.X, HI ORD>0 IF MP/M
011E B2 FF                      mov     dl,0FFH
0120 B1 20                      mov     cl,CURUSR ;INTERROGATE USER NUMBER
0122 E8 09 04                   CALL    BDOS
0125 2E A2 4B 05                mov     USERNO,al
                        ;
                                if      not user
                                mov     al,MPMFLG       ;MP/M?
                                test    al,al   ;IF SO, TYPC HEADING LINE
                                JZ      CHKDRV  : ELSE SKIP IT
```

```
                                        endif
                             ;
0129 BA 0F 02                           mov     dx,offset USRMSG ;DISPLAY IT
012C B1 09                              mov     cl,PRINT
012E E8 FD 03                           CALL    BDOS      ;FIRST PART OF MESSAGE
0131 2E A0 4B 05                        mov     al,USERNO
0135 3C 0A                              cmp     al,10     ;IF USER NO. > 9 PRINT LEADING 1
0137 72 0B                              JB      DUX
0139 B0 31                              mov     al,'1'
013B E8 67 03                           CALL    TYPC
013E 2E A0 4B 05                        mov     al,USERNO          ;PRINT LOW DIGIT OF USER NO.
0142 2C 0A                              sub     al,10
                             ;
0144 04 30                   DUX:       add     al,'0'
0146 E8 5C 03                           CALL    TYPC
0149 BA 23 02                           mov     dx,offset USRMS2 ;PRINT TAIL OF MESSAGE
014C B1 09                              mov     cl,PRINT
014E E8 DD 03                           CALL    BDOS
0151 2E C6 06 3D 05 01                  mov     LINCNT,1           ;WE USED A LINE
                             ;
0157 BE 5C 00                CHKDRV:    mov     si,offset FCB
015A AC                                 lods    al        ;get drive name
015B 84 C0                              test    al,al     ;ANY SPECIFIED?
015D 75 0A                              JNZ     START2    ;YES SKIP NEXT ROUTINE
015F B1 19                              mov     cl,CURDSK
0161 E8 CA 03                           CALL    BDOS      ;GET CURRENT DISK NR
0164 FE C0                              inc     al        ;MAKE A:=1
0166 A2 5C 00                           mov     byte ptr .FCB,al
                             ;
0169 04 40                   START2:    add     al,'A'-1           ;MAKE IT PRINTABLE
016B 2E A2 68 04                        mov     DRNAM,al           ;SAVE FOR LATER
016F BF 5D 00                           mov     di,offset FCB+1 ;POINT TO NAME
0172 8A 05                              mov     al,[di] ;ANY SPECIFIED?
0174 3C 20                              cmp     al,' '
0176 75 07                              JNZ     GOTFCB
                             ;No FCB - make FCB all '?'
0178 B9 0B 00                           mov     cx,11     ;FN+FT COUNT
017B B0 3F                              mov     al,'?'
                             ;
017D F3 AA                              rep stos al       ;fill fcb with '?'
                             ;
                             GOTFCB:
017F C6 06 68 00 3F                     mov     byte ptr .FCB+12,'?'    ;FORCE WILD EXTENT
0184 A0 5C 00                           mov     al,byte ptr .FCB  ;CHECK FOR EXPLICIT DRIVE
0187 FE C8                              dec     al
0189 8A D0                              mov     dl,al     ;SELECT SPECIFIED DRIVE
018B B1 0E                              mov     cl,SELDSK
018D E8 9E 03                           CALL    BDOS
0190 C6 06 5C 00 00                     mov     byte ptr .FCB,0
                             ;
0195 B1 1F                              mov     cl,CURDPB;IT'S 2.X OR MP/M...REQUEST DPB
0197 06                                 push    es        ;save current extra segment
0198 CD E0                              int     224       ;return bx=offset dpb, es=segment dpb
019A 83 C3 02                           add     bx,2
019D 26 8A 07                           mov     al,es: [bx]
01A0 2E A2 33 05                        mov     BLKSHF,al          ;GET BLOCK SHIFT
01A4 43                                 inc     bx        ;BUMP TO BLOCK MASK
01A5 26 8A 07                           mov     al,es: [bx]
01A8 2E A2 34 05                        mov     BLKMSK,al
01AC 83 C3 02                           add     bx,2
01AF 26 8B 07                           mov     ax,es: [bx]
01B2 2E A3 35 05                        mov     BLKMAX,ax
01B6 83 C3 02                           add     bx,2
01B9 26 8B 07                           mov     ax,es: [bx]
01BC 2E A3 37 05                        mov     DIRMAX,ax          ;SAVE IT
01C0 07                                 pop     es                 ;restore our extra segment
                             ;
01C1 40                      SETTBL:    inc     ax        ;DIRECTORY SIZE IS DIRMAX+1
01C2 D1 E0                              shl     ax,1      ;DOUBLE DIRECTORY SIZE
01C4 05 51 05                           add     ax,offset ORDER ;TO GET SIZE OF ORDER TABLE
01C7 2E A3 3E 05                        mov     TBLOC,ax ;NAME TABLE BEGINS WHERE ORDER
                                                TABLE ENDS
01CB 2E A3 40 05                        mov     NEXTT,ax
01CF 8B 1E 06 00                        mov     bx,word ptr .BASE+6     ;MAKE SURE WE HAVE
                                                 ROOM TO CONTINUE
01D3 3B C3                              cmp     ax,bx
01D5 72 03                              jb      SFIRST
01D7 E9 A5 00                           JMP     OUTMEM
                             ;
```

```
                             ;Look up the FCB in the directory
                             ;
01DA B1 11                   SFIRST: mov     cl,FSRCHF ;GET 'SEARCH FIRST' FNC
01DC BA 5C 00                        mov     dx,offset FCB
01DF E8 4C 03                        CALL    BDOS      ;READ FIRST
01E2 FE C0                           inc     al        ;WERE THERE ANY?
01E4 75 4D                           JNZ     SOME      ;GOT SOME
                             ;
01E6 BA FF 01                NONE:   mov     dx,offset FNF   ;PREPARE MP/M ERROR MESSAGE
01E9 2E A0 4F 05                     mov     al,MPMFLG
01ED 84 C0                           test    al,al     ;USE IT IF REALLY MP/M
01EF 74 03                           jz      NOFILE
01F1 E9 33 03                        JMP     ERXIT1
01F4 E8 2F 03                NOFILE: CALL    ERXIT     ;ELSE USE CP/M ERROR MESSAGE
01F7 4E 4F 20 46 49 4C               DB      'NO FILE$'
     45 24
01FF 46 69 6C 65 20 6E       FNF     DB      'File not found.$'
     6F 74 20 66 6F 75
     6E 64 2E 24
                             ;
020F 44 69 72 65 63 74       USRMSG  DB      'Directory for user $'
     6F 72 79 20 66 6F
     72 20 75 73 65 72
     20 24
0223 3A 0D 0A 24             USRMS2  DB      ':',13,10,'$'
                             ;
                             ;Read more directory entries
                             ;
0227 B1 12                   MORDIR: mov     cl,FSRCHN ;SEARCH NEXT
0229 BA 5C 00                        mov     dx,offset FCB
022C E8 FF 02                        CALL    BDOS      ;READ DIR ENTRY
022F FE C0                           inc     al        ;CHECK FOR END (0FFH)
0231 74 60                           JZ      SPRINT    ;NO MORE - SORT & PRINT
                             ;
                             ;Point to directory entry
                             ;
0233 FE C8                   SOME:   dec     al        ;UNDO PREV 'INR A'
0235 B1 05                           mov     cl,5
0237 D2 E0                           shl     al,cl     ;entry no. times 32
0239 B4 00                           mov     ah,0
023B 04 80                           add     al,80h
023D 8B D8                           mov     bx,ax ;POINT TO BUFFER
                                                   ;(SKIP TO FN/FT)
                             ;
                                     IF      SOPT
023F 2E A0 4A 05                     mov     al,SOPFLG         ;DID USER REQUEST SYS FILES?
0243 3C 53                           cmp     al,'S'
0245 74 06                           JZ      SYSFOK
                                     ENDIF
                             ;
0247 F6 47 0A 80                     test    byte ptr 10[bx],80H     ;check bit 7 of SYS byte
024B 75 DA                           JNZ     MORDIR  ;SKIP THAT FILE
                             ;
024D 2E A0 4B 05             SYSFOK: mov     al,USERNO         ;GET CURRENT USER
0251 3A 07                           cmp     al,[bx]
0253 75 D2                           JNZ     MORDIR  ;IGNORE IF DIFFERENT
0255 43                              inc     bx
                             ;
                             ;Move entry to table
                             ;
0256 8B F3                           mov     si,bx   ;si points to name
0258 2E 8B 3E 40 05                  mov     di,NEXTT          ;NEXT TABLE ENTRY TO di
025D B9 0C 00                        mov     cx,12   ;ENTRY LENGTH (NAME, TYPC, EXTENT)
                             ;
0260 AC                      TMOVE:  lods    al        ;GET ENTRY CHAR
0261 24 7F                           and     al,7FH  ;REMOVE ATTRIBUTES
0263 AA                              stos    al        ;store in table
0264 E2 FA                           loop    TMOVE
0266 8A 44 02                        mov     al,2[si]          ;get sector count
0269 88 05                           MOV     [di],al ;STORE IN TABLE
026B 47                              inc     di
026C 2E 89 3E 40 05                  mov     NEXTT,di          ;SAVE UPDATED TABLE ADDR
0271 2E FF 06 42 05                  inc     COUNT
0276 83 C7 0D                        add     di,13   ;SIZE OF NEXT ENTRY
0279 2B 3E 06 00                     sub     di,word ptr .BASE+6     ;PICK UP TPA END
027D 72 A8                           JB      MORDIR  ;IF TPA END>NEXTT THEN LOOP BACK FOR MORE
                             ;
027F E8 A4 02                OUTMEM: CALL    ERXIT
0282 4F 75 74 20 6F 66               DB      'Out of memory.',13,10,'$'
```

```
        20 6D 65 6D 6F 72
        79 2E 0D 0A 24
                                ;
                                ;Sort and print
                                ;
0293 2E 8B 0E 42 05     SPRINT: mov     cx,COUNT        ;GET FILE NAME COUNT
0298 85 C9                      test    cx,cx
029A 75 03                      jnz     SPRINI
029C E9 47 FF                   jmp     NONE    ;NONE, EXIT
                        ;Init the order table
029F 2E A1 3E 05        SPRINI: mov     ax,TBLOC        ;GET START OF NAME TABLE
02A3 BF 51 05                   mov     di,offset ORDER ;POINT TO ORDER TABLE
                        ;
02A6 AB                 BLDORD: stos    ax
02A7 05 0D 00                   add     ax,13
02AA E2 FA                      loop    BLDORD
02AC 2E 8B 1E 42 05             mov     bx,COUNT        ;GET COUNT
02B1 2E 89 1E 44 05             mov     SCOUNT,bx       ;SAVE AS # TO SORT
02B6 4B                         dec     bx      ;only 1 entry?
02B7 74 38                      JZ      DONE    ;..YES, SO SKIP SORT
                        ;
02B9 2E C6 06 46 05 00 SORT:    mov     SWITCH,0        ;SHOW NONE SWITCHED
02BF 2E 8B 1E 44 05             mov     bx,SCOUNT       ;GET COUNT
02C4 4B                         dec     bx              ;use 1 less
02C5 2E 89 1E 4C 05             mov     word ptr TEMP,bx        ;SAVE # TO COMPARE
02CA 2E 89 1E 44 05             mov     SCOUNT,bx       ;SAVE HIGHEST ENTRY
02CF 74 20                      JZ      DONE    ;EXIT IF NO MORE
02D1 BB 51 05                   mov     bx,offset ORDER ;POINT TO ORDER TABLE
                        ;
02D4 B9 0C 00           SORTLP: mov     cx,12   ;# BYTES TO COMPARE
02D7 E8 36 02                   CALL    COMPR   ;COMPARE 2 ENTRIES
02DA 76 03                      jbe     NOSWAP
02DC E8 39 02                   CALL    SWAP    ;SWAP IF NOT IN ORDER
02DF 83 C3 02           NOSWAP: add     bx,2    ;bump order table ptr
02E2 2E FF 0E 4C 05             dec     TEMP    ;BUMP COUNT
02E7 75 EB                      JNZ     SORTLP  ;CONTINUE
                        ;One pass of sort done
02E9 2E A0 46 05                mov     al,SWITCH       ;ANY SWAPS DONE?
02ED 84 C0                      test    al,al
02EF 75 C8                      JNZ     SORT
                        ;
                        ;Sort is all done - print entries
                        ;
02F1 BB 51 05           DONE:   mov     bx,offset ORDER
02F4 2E 89 1E 40 05             mov     NEXTT,bx
                        ;
                        ;Print an entry
                        ;
                                IF      NOT WIDE
                                CALL    DRPRNT  ;PRINT DRIVE NAME
                                ENDIF
02F9 B9 04 00                   mov     cx,NPL  ;NR. OF NAMES PER LINE
02FC 2E C7 06 39 05 00          mov     TOTSIZ,0        ; TOTAL K USED
     00
0303 2E C7 06 3B 05 00          mov     TOTFIL,0        ; AND TOTAL FILES
     00
                        ;
030A 2E 8B 1E 42 05     ENTRY:  mov     bx,COUNT        ; CHECK COUNT OF
                                                          REMAINING FILES
030F 4B                         dec     bx      ; skip compare if only 1 left
0310 74 2D                      JZ      OKPRNT
0312 51                         PUSH    cx
                        ;
0313 B1 06                      mov     cl,dconio       ;get console status
0315 B2 FF                      mov     dl,0ffh
0317 E8 14 02                   call    bdos
031A 84 C0                      test    al,al   ;char?
031C 74 03                      jz      nobrk   ;no char, bypass the other stuff
031E E9 0B 02                   jmp     exit    ;abort
                        ;
0321 2E 8B 1E 40 05     NOBRK:  mov     bx,NEXTT
0326 B9 0B 00                   mov     cx,11
0329 E8 E4 01                   CALL    COMPR   ;DOES THIS ENTRY MATCH NEXT ONE?
032C 59                         pop     cx
032D 75 10                      JNE     OKPRNT  ;NO, PRINT IT
032F 83 C3 02                   add     bx,2    ;SKIP, SINCE HIGHEST EXTENT COMES
                                                 LAST IN LIST
0332 2E 89 1E 40 05             mov     NEXTT,bx
0337 2E FF 0E 42 05             dec     COUNT   ;COUNT DOWN
033C E9 CB FF                   JMP     ENTRY   ;GO GET NEXT
                        ;
                        OKPRNT:
033F 51                         push    cx
                        ;
                                IF      NOT WIDE
                                CALL    FENCE   ;PRINT FENCE CHAR AND SPACE
                                ENDIF
                        ;
0340 2E 8B 1E 40 05             mov     bx,NEXTT        ;GET ORDER TABLE POINTER
0345 8B 37                      mov     si,[bx]
0347 83 C3 02                   add     bx,2
034A 2E 89 1E 40 05             mov     NEXTT,bx        ;SAVE UPDATED TABLE POINTER
034F B9 08 00                   mov     cx,8    ;FILE NAME LENGTH
0352 E8 60 01                   CALL    TYPCIT  ;TYPC FILENAME
0355 B0 2E                      mov     al,'.'  ;PERIOD AFTER FN
0357 E8 4B 01                   CALL    TYPC
035A B9 03 00                   mov     cx,3    ;GET THE FILETYPC
035D E8 55 01                   CALL    TYPCIT
0360 8A 14                      mov     dl,[si]
0362 B6 00                      mov     dh,0
0364 46                         inc     si
0365 8A 04                      mov     al,[si] ;GET SECTOR COUNT OF LAST EXTENT
0367 B1 04                      mov     cl,4    ;# OF EXTENTS TIMES 16K
0369 D3 E2                      shl     dx,cl
036B 2E 02 06 34 05             ADD     al,BLKMSK       ;ROUND LAST EXTENT TO BLOCK SIZE
0370 B1 03                      mov     cl,3
0372 D2 E8                      shr     al,cl           ;CONVERT FROM SECTORS TO K
0374 B4 00                      mov     ah,0
0376 03 D0                      add     dx,ax   ;add to total K
0378 2E A0 34 05                mov     al,BLKMSK       ;GET SECTORS/BLK-1
037C B1 03                      mov     cl,3
037E D3 E8                      shr     ax,cl           ;CONVERT TO K/BLK
0380 F7 D0                      not     ax              ;USE TO FINISH ROUNDING
0382 23 D0                      and     dx,ax
0384 2E 01 16 39 05             add     TOTSIZ,dx       ;add to total used
0389 2E FF 06 3B 05             inc     TOTFIL  ;INCREMENT FILE COUNT
038E 8B C2                      mov     ax,dx           ;GET BACK FILE SIZE
0390 E8 1B 00                   CALL    DECPRT  ; AND PRINT IT
0393 B0 6B                      mov     al,'k'  ;FOLLOW WITH K
0395 E8 0D 01                   CALL    TYPC
                        ;
                                IF      NOT WIDE
                                CALL    SPACE
                                ENDIF
                        ;
                        ;See if more entries
                        ;
0398 2E FF 0E 42 05             dec     COUNT   ;COUNT DOWN ENTRIES
039D 59                         pop     cx
039E 74 58                      JZ      PRTOTL  ;IF OUT OF FILES, PRINT TOTALS
03A0 49                         DEC     CX      ;ONE LESS ON THIS LINE
03A1 74 05                      jz      DOCRLF
                        ;
                                IF      WIDE
03A3 E8 F5 00                   CALL    FENCE   ;NO CR-LF NEEDED, DO FENCE
                                ENDIF
                        ;
03A6 EB 03                      jmps    NOCRLF
                        ;
03A8 E8 1D 01           DOCRLF: CALL    CRLF    ;CR-LF NEEDED
03AB E9 5C FF           NOCRLF: JMP     ENTRY
                        ;
                        ;Print HL in decimal with leading zero suppression
                        ;
                        DECPRT:                 ;CLEAR LEADING ZERO FLAG
03AE 2E C6 06 50 05 00          mov     LZFLG,0
03B4 BB E8 03                   mov     bx,1000 ;PRINT 1000'S DIGIT
03B7 E8 11 00                   CALL    DIGIT
03BA BB 64 00                   mov     bx,100  ;ETC
03BD E8 0B 00                   CALL    DIGIT
03C0 BB 0A 00                   mov     bx,10
03C3 E8 05 00                   CALL    DIGIT
03C6 04 30                      add     al,'0'  ;GET 1'S DIGIT
03C8 E9 DA 00                   JMP     TYPC
                        ;
03CB BA 00 00           DIGIT:  mov     dx,0    ;init hi order dividend
03CE F7 F3                      div     bx      ;divide ax by digit value (dx gets rmdr)
03D0 04 30                      add     al,'0'  ;convert to ASCII digit
```

```
                         ;
03D2 3C 30                        cmp      al,'0'  ;ZERO DIGIT?
03D4 75 16                        JNZ      DIGNZ   ;NO, TYPC IT
03D6 2E A0 50 05                  mov      al,LZFLG        ;LEADING ZERO?
03DA 84 C0                        test     al,al
03DC B0 30                        mov      al,'0'
03DE 75 12                        JNZ      DIGPR   ;PRINT DIGIT
03E0 2E A0 47 05                  mov      al,SUPSPC       ;GET SPACE SUPPRESSION FLAG
03E4 84 C0                        test     al,al   ;SEE IF PRINTING FILE TOTALS
03E6 74 0D                        jz       DIGNP           ;YES, DON'T GIVE LEADING SPACES
03E8 B0 20                        mov      al,' '
03EA EB 06                        JMPS     DIGPR   ;LEADING ZERO...PRINT SPACE
                         ;
03EC 2E C6 06 50 05 FF DIGNZ:     mov      LZFLG,0ffh ;SET LEADING ZERO FLAG SO NEXT
                                                        ZERO PRINTS
03F2 E8 B0 00            DIGPR:   call     TYPC    ;AND PRINT DIGIT
03F5 8B C2               DIGNP:   mov      ax,dx   ;set up remainder for next digit
03F7 C3                           ret
                         ;
                         ;Show total space and files used
                         ;
03F8 2E C6 06 47 05 00 PRTOTL: mov      SUPSPC,0        ;SUPPRESS LEADING SPACES
                                                         IN TOTALS
03FE E8 C7 00                     CALL     CRLF    ;NEW LINE (WITH PAUSE IF NECESSARY)
                         ;
                                  IF       WIDE
0401 BA 68 04                     mov      dx,offset TOTMS1 ;PRINT FIRST PART OF
                                                             TOTAL MESSAGE
                                  ENDIF
                         ;
                                  IF       NOT WIDE
                                  mov      dx,offset TOTMS1+1 ;PRINT FIRST PART OF
                                                               TOTAL MESSAGE
                                  ENDIF
                         ;
0404 B1 09                        mov      cl,PRINT
0406 E8 25 01                     CALL     BDOS
0409 2E A1 39 05                  mov      ax,TOTSIZ       ;PRINT TOTAL K USED
040D E8 9E FF                     CALL     DECPRT
0410 BA 75 04                     mov      dx,offset TOTMS2;NEXT PART OF MESSAGE
0413 B1 09                        mov      cl,PRINT
0415 E8 16 01                     CALL     BDOS
0418 2E A1 3B 05                  mov      ax,TOTFIL       ;PRINT COUNT OF FILES
041C E8 8F FF                     CALL     DECPRT
041F BA 7B 04                     mov      dx,offset TOTMS3;TAIL OF MESSAGE
0422 B1 09                        mov      cl,PRINT
0424 E8 07 01                     CALL     BDOS
0427 B1 1B                        mov      cl,GALLOC       ;GET ADDRESS OF
                                                            ALLOCATION VECTOR
0429 06                           push     es      ;save our ES
042A CD E0                        int      224     ;return bx=offset ALV, es=segment ALV
042C 2E 8B 16 35 05               mov      dx,BLKMAX       ;GET ITS LENGTH
0431 42                           inc      dx
0432 B9 00 00                     mov      cx,0    ;INIT BLOCK COUNT TO 0
                         ;
0435 53                  GSPBYT:  PUSH     bx      ;SAVE ALLOC ADDRESS
0436 26 8A 07                     mov      al,es: [bx]
0439 B3 08                        mov      bl,8    ;SET TO PROCESS 8 BLOCKS
                         ;
043B D0 E0               GSPLUP:  shl      al,1            ;TEST BIT
043D 72 01                        JB       NOTFRE
043F 41                           inc      cx
                         ;
0440 4A                  NOTFRE:  dec      dx      ;COUNT DOWN BLOCKS
0441 74 08                        JZ       ENDALC  ;QUIT IF OUT OF BLOCKS
0443 FE CB                        dec      bl      ;COUNT DOWN 8 BITS
0445 75 F4                        JNZ      GSPLUP  ;DO ANOTHER BIT
0447 5B                           POP      bx      ;BUMP TO NEXT BYTE
0448 43                           INC      bx      ;OF ALLOC. VECTOR
0449 EB EA                        JMPS     GSPBYT  ;PROCESS IT
                         ;
044B 07                  ENDALC:  pop      es      ;restore our es
044C 8B C1                        mov      ax,cx
044E 2E 8A 0E 33 05               mov      cl,BLKSHF       ;GET BLOCK SHIFT FACTOR
0453 80 E9 03                     sub      cl,3    ;CONVERT FROM SECTORS TO K
0456 74 02                        JZ       PRTFRE  ;SKIP SHIFTS IF 1K BLOCKS
                         ;
0458 D3 E0                        shl      ax,cl   ;mult blks by k/blk
                         ;
```

```
045A E8 51 FF            PRTFRE:  CALL     DECPRT  ;PRINT K FREE
045D BA 88 04                     mov      dx,offset TOTMS4
0460 B1 09                        mov      cl,PRINT
0462 E8 C9 00                     CALL     BDOS
0465 E9 C4 00                     JMP      EXIT    ;ALL DONE...RETURN TO CP/M
                         ;
0468 20 3A 20 54 6F 74 TOTMS1     DB       ' : Total of $'
     61 6C 20 6F 66 20
     24
 0468                    DRNAM    equ      TOTMS1
0475 6B 20 69 6E 20 24 TOTMS2     DB       'k in $'
047B 20 66 69 6C 65 73 TOTMS3     DB       ' files with $'
     20 77 69 74 68 20
     24
0488 6B 20 73 70 61 63 TOTMS4     DB       'k space remaining.$'
     65 20 72 65 6D 61
     69 6E 69 6E 67 2E
     24
                         ;
                         FENCE:
                                  IF       WIDE
049B E8 05 00                     CALL     SPACE
                                  ENDIF
049E B0 3A                        mov      al,DELIM        ;FENCE CHARACTER
04A0 E8 02 00                     CALL     TYPC    ;PRINT IT, FALL INTO SPACE
                         ;
04A3 B0 20               SPACE:   mov      al,' '
                         ;
                         ;Type character in A
                         ;
04A5 51                  TYPC:    PUSH     cx
04A6 52                           PUSH     dx
04A7 53                           push     bx
04A8 56                           push     si
04A9 8A D0                        mov      dl,al   ;use bdos calls, that's what they're there for
04AB B1 06                        mov      cl,dconio
04AD E8 7E 00                     call     bdos
04B0 5E                           pop      si
04B1 5B                           POP      bx
04B2 5A                           POP      dx
04B3 59                           POP      cx
04B4 C3                           RET
                         ;
04B5 AC                  TYPCIT:  lods     al
04B6 E8 EC FF                     CALL     TYPC
04B9 E2 FA                        loop     TYPCIT
04BB C3                           RET
                         ;
                         ;Fetch character from console (without echo)
                         ;
04BC B1 06               CINPUT:  mov      cl,dconio
04BE B0 FF                        mov      al,0ffh
04C0 E8 6B 00                     call     BDOS
04C3 24 7F                        and      al,7FH
04C5 74 F5                        jz       CINPUT
04C7 C3                           RET
                         ;
04C8 2E A0 3D 05         CRLF:    mov      al,LINCNT       ;CHECK FOR END OF SCREEN
04CC FE C0                        inc      al
04CE 3C 17                        cmp      al,LPS
04D0 72 0D                        JB       NOTEOS  ;SKIP MESSAGE IF MORE LINES LEFT ON SCREEN
04D2 BA F1 04                     mov      dx,offset EOSMSG;SAY WE'RE PAUSING FOR INPUT
04D5 B1 09                        mov      cl,PRINT
04D7 E8 54 00                     CALL     BDOS
04DA E8 DF FF                     CALL     CINPUT  ;WAIT FOR CHAR.
04DD B0 00                        mov      al,0    ;SET UP TO ZERO LINE COUNT
                         ;
04DF 2E A2 3D 05         NOTEOS:  mov      LINCNT,al       ;SAVE NEW LINE COUNT
04E3 B0 0D                        mov      al,13   ;print cr
04E5 E8 BD FF                     call     TYPC
04E8 B0 0A                        mov      al,10   ;lf
04EA E8 B8 FF                     call     TYPC
                         ;
                                  IF       NOT WIDE
                                  CALL     DRPRNT  ;DRIVE NAME
                                  ENDIF
                         ;
04ED B9 04 00                     mov      cx,NPL  ;RESET NUMBER OF NAMES PER LINE
04F0 C3                           RET
```

```
                            ;
04F1 0D 0A 28 53 74 72 EOSMSG  DB      13,10,'(Strike any key to continue)$'
     69 6B 65 20 61 6E
     79 20 6B 65 79 20
     74 6F 20 63 6F 6E
     74 69 6E 75 65 29
     24
                            ;
                                    IF      NOT WIDE
                            DRPRNT: mov     al,DRNAM
                                    JMP     TYPC
                                    ENDIF
                            ;
                            ;Compare routine for sort
                            ;
0510 8B 37                  COMPR:  mov     si,[bx]
0512 8B 7F 02                       mov     di,2[bx]
0515 F3 A6                          repe cmps al,al
0517 C3                             ret
                            ;
                            ;
                            ;Swap entries in the order table
                            ;
0518 2E C6 06 46 05 01      SWAP:   mov     SWITCH,1        ;SHOW A SWAP WAS MADE
051E 8B 17                          mov     dx,[bx]
0520 87 57 02                       xchg    dx,2[bx]
0523 89 17                          mov     [bx],dx
0525 C3                             ret
                            ;
                            ;Error exit
                            ;
0526 5A                     ERXIT:  POP     dx      ;GET MSG
                            ;
0527 B1 09                  ERXIT1: mov     cl,PRINT
                            ;
0529 E8 02 00               CALLB:  CALL    BDOS    ;PERFORM REQUESTED FUNCTION
                            ;
                            ;(fall into exit)

052C B1 00                  ;Exit - all done, restore stack
                            ;
                            EXIT:   mov     cl,0    ;exit is via BDOS call 0
                            ;
052E 06                     BDOS:   push    es      ;preserve es thru bdos call
052F CD E0                          int     224     ;call bdos 8086 style
0531 07                             pop     es
0532 C3                             ret
                            ;
                            ;Temporary storage area
                            ;
0533 00                     BLKSHF  DB      0       ;# SHIFTS TO MULT BY SEC/BLK
0534 00                     BLKMSK  DB      0       ;SEC/BLK - 1
0535 00 00                  BLKMAX  DW      0       ;HIGHEST BLOCK # ON DRIVE
0537 00 00                  DIRMAX  DW      0       ;HIGHEST FILE # IN DIRECTORY
0539 00 00                  TOTSIZ  DW      0       ;TOTAL SIZE OF ALL FILES
053B 00 00                  TOTFIL  DW      0       ;TOTAL NUMBER OF FILES
053D 00                     LINCNT  DB      0       ;COUNT OF LINES ON SCREEN
053E 00 00                  TBLOC   DW      0       ;POINTER TO START OF NAME TABLE
0540 00 00                  NEXTT   DW      0       ;NEXT TABLE ENTRY
0542 00 00                  COUNT   DW      0       ;ENTRY COUNT
0544 00 00                  SCOUNT  DW      0       ;# TO SORT
0546 00                     SWITCH  DB      0       ;SWAP SWITCH FOR SORT
0547 FF                     SUPSPC  DB      0FFH,   ;LEADING SPACE FLAG FOR DECIMAL RTN.
0548 80 00                  BUFAD   DW      BASE+80H ;OUTPUT ADDR
054A 00                     SOPFLG  db      0       ;SET TO 'S' TO ALLOW SYS FILES TO PRINT
054B 00                     USERNO  db      0       ;CONTAINS CURRENT USER NUMBER
054C 00 00                  TEMP    dw      0       ;SAVE DIR ENTRY
054E 00                     VERFLG  db      0       ;VERSION FLAG
054F 00                     MPMFLG  db      0       ;MP/M FLAG
0550 00                     LZFLG   db      0       ;0 WHEN PRINTING LEADING ZEROS
  0551                      ORDER   EQU     $       ;ORDER TABLE STARTS HERE
                            ;
                            ;BDOS equates
                            ;
  0001                      RDCHR   EQU     1       ;READ CHAR FROM CONSOLE
  0002                      WRCHR   EQU     2       ;WRITE CHR TO CONSOLE
  0006                      DCONIO  EQU     6       ;direct console i/o
  0009                      PRINT   EQU     9       ;PRINT CONSOLE BUFF
  000B                      CONST   EQU     11      ;CHECK CONS STAT
  000E                      SELDSK  EQU     14      ;SELECT DISK
  000F                      FOPEN   EQU     15      ;0FFH=NOT FOUND
  0010                      FCLOSE  EQU     16      ;   "   "
  0011                      FSRCHF  EQU     17      ;   "   "
  0012                      FSRCHN  EQU     18      ;   "   "
  0019                      CURDSK  EQU     25      ;GET CURRENTLY LOGGED DISK NAME
  001B                      GALLOC  EQU     27      ;GET ADDRESS OF ALLOCATION VECTOR
  001F                      CURDPB  EQU     31      ;GET CURRENT DISK PARAMETERS
  0020                      CURUSR  EQU     32      ;GET CURRENTLY LOGGED USER NUMBER (2.x ONLY)
                            ;
                                    END
```

NEW! TPM* for TRS-80 Model II
NEW! System/6 Package
Computer Design Labs

Z80* Disk Software

We have acquired the rights to all TDL software (& hardware). TDL software has long had the reputation of being the best in the industry. Computer Design Labs will continue to maintain, evolve and add to this superior line of quality software.
— Carl Galletti and Roger Amidon, owners.

Software with Manual/Manual Alone

All of the software below is available on any of the following media for operation with a Z80 CPU using the CP/M* or similar type disk operating system (such as our own TPM*).

for TRS-80* CP/M (Model I or II)
for 8" CP/M (soft sectored single density)
for 5¼" CP/M (soft sectored single density)
for 5¼" North Star CP/M (single density)
for 5¼" North Star CP/M (double density)

BASIC I
A powerful and fast Z80 Basic interpreter with EDIT, RENUMBER, TRACE, PRINT USING, assembly language subroutine CALL, LOADGO for "chaining", COPY to move text, EXCHANGE, KILL, LINE INPUT, error intercept, sequential file handling in both ASCII and binary formats, and much, much more. It runs in a little over 12 K. An excellent choice for games since the precision was limited to 7 digits in order to make it one of the fastest around. $49.95/$15.

BASIC II
Basic I but with 12 digit precision to make its power available to the business world with only a slight sacrifice in speed. Still runs faster than most other Basics (even those with much less precision). $99.95/$15.

BUSINESS BASIC
The most powerful Basic for business applications. It adds to Basic II with random or sequential disk files in either fixed or variable record lengths, simultaneous access to multiple disk files, PRIVACY command to prohibit user access to source code, global editing, added math functions, and disk file maintenance capability without leaving Basic (list, rename, or delete). $179.95/$25.

ZEDIT
A character oriented text editor with 26 commands and "macro" capability for stringing multiple commands together. Included are a complete array of character move, add, delete, and display function. $49.95./$15.

ZTEL
Z80 Text Editing Language - Not just a text editor. Actually a language which allows you to edit text and also write, save, and recall programs which manipulate text. Commands include conditional branching, subroutine calls, iteration, block move, expression evaluation, and much more. Contains 36 value registers and 10 text registers. Be creative! Manipulate text with commands you write using Ztel. $79.95/$25.

TOP
A Z80 Text Output Processor which will do text formatting for manuals, documents, and other word processing jobs. Works with any text editor. Does justification, page numbering and headings, spacing, centering, and much more! $79.95/$25.

MACRO I
A macro assembler which will generate relocateable or absolute code for the 8080 or Z80 using standard Intel mnemonics plus TDL/Z80 extensions. Functions include 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical operations, local and global symbols, chaining files, linking capability with optional linker, and recursive/reiterative macros. This assembler is so powerful you'll think it is doing all the work for you. It actually makes assembly language programming much less of an effort and more creative. $79.95/$20.

MACRO II
Expands upon Macro I's linking capability (which is useful but somewhat limited) thereby being able to take full advantage of the optional Linker. Also a time and date function has been added and the listing capability improved. $99.95/$25.

LINKER
How many times have you written the same subroutine in each new program? Top notch professional programmers compile a library of these subroutines and use a Linker to tie them together at assembly time. Development time is thus drastically reduced and becomes comparable to writing in a high level language but with all the speed of assembly language. So, get the new CDL Linker and start writing programs in a fraction of the time it took before. Linker is compatible with Macro I & II as well as TDL/Xitan assemblers version 2.0 or later. $79.95/$20.

DEBUG I
Many programmers give up on writing in assembly language even though they know their programs would be faster and more powerful. To them assembly language seems difficult to understand and follow, as well as being a nightmare to debug. Well, not with proper tools like Debug I. With Debug I you can easily follow the flow of any Z80 or 8080 program. Trace the program one step at a time or 10 steps or whatever you like. At each step you will be able to see the instruction executed and what it did. If desired, modifications can then be made before continuing. It's all under your control. You can even skip displaying a subroutine call and up to seven breakpoints can be set during execution. Use of Debug I can pay for itself many times over by saving you valuable debugging time. $79.95/$20.

DEBUG II
This is an expanded debugger which has all of the features of Debug I plus many more. You can "trap" (i.e. trace a program until a set of register, flag, and/or memory conditions occur). Also, instructions may be entered and executed immediately. This makes it easy to learn new instructions by examining registers/memory before and after. And a RADIX function allows changing between ASCII, binary, decimal, hex, octal, signed decimal, or split octal. All these features and more add up to give you a very powerful development tool. Both Debug I and II must run on a Z80 but will debug both Z80 and 8080 code. $99.95/$20.

ZAPPLE
A Z80 executive and debug monitor. Capable of search, ASCII put and display, read and write to I/O ports, hex math, breakpoint, execute, move, fill, display, read and write in Intel or binary format tape, and more! on disk $34.95/$15.

APPLE
8080 version of Zapple $34.95/$15.

NEW! TPM now available for TRS-80 Model II!

TPM*
A NEW Z80 disk operation system! This is not CP/M*. It's better! You can still run any program which runs with CP/M* but unlike CP/M* this operating system was written specifically for the Z80* and takes full advantage of its extra powerful instruction set. In other words its not warmed over 8080 code! Available for TRS-80* (Model I or II). Tarbell, Xitan DDDC, SD Sales "VERSA-FLOPPY", North Star (SD&DD), and Digital (Micro) Systems. $79.95/$25.

SYSTEM MONITOR BOARD (SMB II)
A complete I/O board for S-100 systems. 2 serial ports, 2 parallel ports, 1200/2400 baud cassette tape interface, sockets for 2K of RAM, 3-2708/2716 EPROM's or ROM, jump on reset circuitry. Bare board $49.95/$20.

ROM FOR SMB II
2KX8 masked ROM of Zapple monitor. Includes source listing $34.95/$15.

PAYROLL (source code only)
The Osborne package. Requires C Basic 2.
5" disks $124.95 (manual not included)
8" disks $ 99.95 (manual not included)
Manual $20.00

ACCOUNTS PAYABLE/RECEIVABLE (source code only)
By Osborne, Requires C Basic 2
5" disks $124.95 (manual not included)
8" $99.95 (manual not included)
Manual $20.00

GENERAL LEDGER (source code only)
By Osborne. Requires C Basic 2
5" disks $99.95 (manual not included)
8" disks $99.95 (manual not included)
Manual $20.00

C BASIC 2
Required for Osborne software. $99.95/$20.

SYSTEM/6
TPM with utilities, Basic I interpreter, Basic E compiler, Macro I assembler, Debug I debugger, and ZEDIT text editor.
Above purchased separately costs $339.75
Special introductory offer: Only $179.75 with coupon!!

$160.

This Coupon Is Worth One Hundred And Sixty Dollars Toward The Full Price Of The SYSTEM/6 Package
System/6 with this coupon is only $179.95. This is a limited time offer.

$160.00

ORDERING INFORMATION
Visa, Master Charge and C.O.D. O.K. To order call or write with the following information.

1. Name of Product (e.g. Macro I)
2. Media (e.g. 8" CP/M)
3. Price and method of payment (e.g. C.O.D.) include credit card info. if applicable.
4. Name, Address and Phone number.
5. For TPM orders only: Indicate if for TRS 80, Tarbell, Xitan DDDC, SD Sales (5¼" or 8"). ICOM (5¼" or 8"), North Star (single or double density) or Digital (Micro) Systems.
6. N.J. residents add 5% sales tax.

Manual cost applicable against price of subsequent software purchase in any item except for the Osborne software.

For information and tech queries call
609-599-2146
For phone orders ONLY call toll free
1-800-327-9191
Ext. 676
(Except Florida)

OEMS
Many CDL products are available for licensing to OEMs. Write to Carl Galletti with your requirements.

* Z80 is a trademark of Zilog
* TRS-80 is a trademark for Radio Shack
* TPM is a trademark of Computer Design Labs. It is not CP/M*
* CP/M is a trademark of Digital Research

Prices and specifications subject to change without notice.

DEALER INQUIRIES INVITED.

COMPUTER DESIGN LABS
342 Columbus Avenue
Trenton, N.J. 08629

16-Bit Microcomputer Disk Operating Systems

by Sol Libes

The following is a compilation of Disk Operating System (DOS) packages currently available for 16-bit microprocessor-based computer systems. While most can be purchased separately from hardware, the XENIX and 9900 Disc Executive packages cannot. These two have been included because they are, or are expected to be, implemented on S-100 based systems. There are many other 16-bit DOS packages currently on the market that I have not included here because they are furnished only as part of turnkey systems which are not S-100 based.

I was amazed to find that there are already fourteen suppliers furnishing 27 different 16-bit DOS packages that range from low cost (typically $450) single-user development DOS, all the way up to a 32 user system capable of handling 256 tasks.

Naturally, the 8086, being the oldest of the current generation of 16-bit microprocessors, has the largest number of available packages, many of which have been in use for well over a year. Most of the Z8000 and 68000 DOS listed were not yet released when my questionnaire was returned by the company.

Reviews of two 8086 DOS packages appear in this issue of *Microsystems* (CP/M-86 and Seattle Computer Products' DOS). We plan to review some of the Z8000 and 68000 DOS in future issues. Readers interested in writing such reviews should contact me. ■

16-Bit Microcomputer DOS Suppliers

Seattle Computer Products
1114 Industry Dr.
Seattle, WA 98188

Telecompute Systems, Inc.
251 Spadina Ave.
Toronto, Ont
Canada M5T 2E2

Industrial Programming, Inc.
100 Jericho Quadrangle
Jericho, NY 11753

Onyx Systems, Inc.
73 East Trimble Rd.
San Jose, CA 75132

Marinchip Systems Ltd.
16 St. Jude Rd.
Mill Valley, CA 94941

Central Data Corp.
713 Edgebrook Dr.
Champaign, IL 61820

Intel Corp.
3065 Bowers Ave.
Santa Clara, CA 95051

Hemenway Associates
101 Tremont St.
Boston, MA 02108

Central Systems, Inc.
1317 Central Ave.
Kansas City, KS 66102

TSC
Box 2570
W. Lafayette, IN 47906

Microsoft
10800 NE 8th St.
Bellevue, WA 98004

Digital Research
801 Lighthouse Rd.
Pacific Grove, CA 93950

Phase One Systems, Inc.
770 Edgewater Dr.
Suite 710
Oakland, CA 94621

Systems & Software, Inc.
2801 Finley Rd.
Donners Grove, IL 60515

8086/8088 Disk Operating Systems

DOS Name	86-DOS	CP/M-86	MP/M-86	iRMX88	iRMX86	MTOS-86
Vendor	Seattle Computer	Digital Research	Digital Research	Intel Corp.	Intel Corp.	Industrial Prog.
Price	$195	$250	$500	$2000	$7500	$5500-$18,000
Size	12K min.	11K min.	20K min.	4-32K	10-128K	8-24K
Maximum number of:						
Users	1	1	255	1	1	1
CPU's	1	1	1	1	1	16
Tasks	1	1	255	any number	any number	4000
Network Protocols	none	via CP/NET-86	yes	none	to be released	none
Real Time Provisions	none	user configurable	yes	yes	yes	yes
Memory Size (Max.)	1M	1M	1M	1M	1M	1M
Disk Storage (Max.)	15M 15 drives	128M 16 Drives	8G 16 Drives	Supports iSBC drives	Supports iSBC drives	1G
Supports:						
Floppy Disk	yes	yes	yes	yes	yes	yes
Hard Disk	yes	yes	yes	yes	yes	yes
CRT	yes	yes	yes	yes	yes	yes
Printer	yes	yes	yes	yes	yes	yes
Line Printer	yes	yes	yes	no	no	yes
Other	—	IEEE-488 Paper tape	Mag. tape	—	—	—
Comments:	Includes Assembler, Debugger and Utilities. Basic optional.	—	Compatible with CP/M allowing clusters of systems. Requires Real Time Clock.	Supports 8087 math processor and bubble memory.	Supports 8087 math processor and bubble memory.	Includes Source Code. User guide $15.

DOS Name	MSP/8086	SP/8086	OASIS-8086	REX-80	XENIX-8086
Vendor	Hemenway	Hemenway	Phase One Sys.	Systems & Software	Microsoft
Price	$750	$500	$1495	$3750	?
Size	32K	16K	64K	4K min.	82K min.
Maximum number of:					
Users	1	1	32	User configurable	4 to 20
CPU's	1	1	1	1	1
Tasks	8 in 32K	1	256	any number	20 to 100
Network Protocols	yes	yes	none	none	yes
Real Time Provisions	yes	yes	128 max.	yes	limited
Memory Size (Max.)	1M	1M	1M	1M	1M min.
Disk Storage (Max.)	80M	80M	2.8M/Vol. 32 Volumes	User option	2M min.
Supports:					
Floppy Disk	yes	yes	yes	yes	yes
Hard Disk	yes	yes	yes	no	yes
CRT	yes	yes	yes	yes	yes
Printer	yes	yes	yes	yes	yes
Line Printer	yes	yes	yes	yes	yes
Other	paper tape	paper tape	Mag. tape Cartridge tape	A/D & D/A	—
Comments:	Includes Macro Assembler, Linking Loader, Basic & Pascal	Includes Macro Assembler, Linking Loader, Basic & Pascal	Supports bubble memory	Supports 8087 math processor and PL/M.	Expanded version of Labs UNIX Ver. 7.

Note: K = *Kilobytes;* M = *Megabytes;* G = *Gigabytes*

Z8000 Disk Operating Systems

DOS Name	ZMOS	SP/Z8000	XENIX-Z8000	ONIX	OASIS-Z8000	MSP/Z8000	Z8000 Disc Executive	TIS-APL
Vendor	Central Data	Hemenway	Microsoft	Onyx Systems	Phase One	Hemenway	Marinchip	Telecompute Sys.
Price	$450	$500	?	$1500 (4 users) $2500 (8 users)	$1495	$750	$500	$840
Size	96K	16K	82K min.	80K	64K	32K	9K	30K
Maximum number of:								
Users	32	1	4 to 20	8	32	1	1	1
CPU's	1	1	1	1	1	1	1	1
Tasks	175	1	20 to 100	255	256	8 in 32K	1	1
Network Protocols	none	yes	yes	2780, 3780, Ethernet	yes	yes	none	yes
Real Time Provisions	none	yes	limited	yes	128 levels	yes	none	N/A
Memory Size (Max.)	16M	8M	1M min.	1M	16M	16M	64K	256K
Disk Storage (Max.)	250M	80M	2M min.	10-40M	2.8M/Vol. 32 Volumes	80M	4M	120M
Supports:								
Floppy Disk	yes	yes	yes	yes	yes	yes	yes	4 drives
Hard Disk	yes	yes	yes	yes	yes	yes	no	no
CRT	yes	yes	yes	yes	yes	yes	yes	yes
Printer	yes	yes	yes	yes	yes	yes	yes	yes
Line Printer	yes	yes	yes	yes	yes	yes	no	no
Other	—	Paper Tape	—	—	Mag. Tape	Paper Tape	—	A/D & D/A
Comments:	Works with CDC memory management hardware.	Includes Macro Assembler, Linking Loader, Basic and Pascal.	Expanded version of Bell Labs UNIX Ver. 7.	Based on Bell Labs UNIX.	Includes Basic, Editor, Diagnostic & Communications package.	Includes Macro Assembler, Linking Loader, Basic and Pascal.	Includes Basic, Assembler, Editor, Linker and Utilities.	Integrated DOS & APL.

Note: K = *Kilobytes* M = *Megabytes* G = *Gegabytes*

68000 Disk Operating Systems

DOS Name	MSP/68000	SP/68000	MTOS-68K	UniFLEX	UNIX	XENIX-68000
Vendor	Hemenway	Hemenway	Industrial Prog.	TSC	Control Systems	Microsoft
Price	$750	$500	$9500	$800	not yet set	
Size	32K	16K	8K	32K	128K	82K min.
Maximum Number of: Users CPU's Tasks	 1 1 8 in 32 K	 1 1 1	 1 16 Any number	 Any number 1 Any number	 50 1 Any number	 4 to 20 1 20 to 100
Network Protocols	yes	yes	X-25	none	optional	yes
Real Time Provisions	yes	yes	yes	none	60 Hz Interrupt	limited
Memory Size (Max.)	16M	8M	16M	8M	?	1M min.
Disk Storage (Max.)	80M	80M	4 single-sided double density floppies	unlimited 8M/drive	?	2M min.
Supports: Floppy Disk Hard Disk CRT Printer Line Printer Other	 yes yes yes yes yes paper tape	 yes yes yes yes yes paper tape	 yes no yes yes no —	 yes yes yes yes yes —	 5″ & 8″ yes yes yes yes —	 yes yes yes yes yes —
Comments:	Includes Macro Assembler, Linking Loader, Basic and Pascal	Includes Macro Assembler, Linking Loader, Basic and Pascal	Source code furnished. Users Guide $15.	Hierarchical file system, password & file protection. Requires memory-mapping hardware.	Designed for CSI systems.	Expanded version of Bell Labs UNIX Vers. 7.

Note: K = *Kilobytes* M = *Megabytes* G = *Gegabytes*

9900 Disk Operating System

DOS Name	M9900 Disc Executive	NOS/MT
Vendor	Marinchip	Marinchip
Price	included with hardware	$250
Size	9K	16-36K
Maximum Number of: Users CPU's Tasks	 1 1 1	 any number 1 one/user
Network Protocols	none	none
Real Time Provisions	none	user provides
Memory Size (Max.)	60K	56K/user
Disk Storage (Max.)	4M	no limit
Supports: Floppy Disk Hard Disk CRT Printer Line Printer Other	 yes no yes yes no —	 yes yes yes yes no —
Comments:	Requires Marinchip hardware.	Requires Marinchip hardware. I/O drive source supplied. Includes Assembler, Editor, Linker, Basic, Utilities, Output processor. Completely user configureable.

Note: K = *Kilobytes* M = *Megabytes* G = *Gegabytes*

Input Queuing For North Star Double Density

by Robert T. Armstrong

As a lawyer from 'Down Under' I have been using a North Star system for bookkeeping purposes for over two years. The basic programs I wrote have been annoying because after inputing various values the system took some seconds to process that data, update running balances and write results to disk. Inputing and operating time was wasted.

The problem was aggravated when the delay caused the disk drive to stop; then after inputing the drive had to build up speed again—a minor matter—but seconds add up. I kept taking comfort in the hope that 'shortly' a compiler for North Star Basic would become available.

My interest was aroused by the articles *Queueing* and *Polling* in the May 1979 edition of *Byte*.

The question was how to ensure that the keyboard was checked 'often' while a North Star basic program was running? Two facilities are available:

• First, Basic regularly checks through the 'contc' routine to see whether a control C (to stop the basic program) has been depressed. This is accessed regularly except when a disk access is taking place.

• Second, double density DOS has available an 'often' routine which is called at least once every 40 milliseconds—no doubt incorporated for this very purpose.

The North Star manual gives us warning of the only problem (but, of course I did not read it carefully and had to find out for myself) and this is that 'often' will be called at bootstrap load time, even before the 2900H personalization block is loaded. The answer is to originally patch a 'return' and change this to 'jump' in the initialization routine.

A full listing of alterations to DOS is enclosed, the procedure for double density would be:

1. "LF DOS 5000 {CR} "—put present DOS at 5000H.
2. Bytes 2007H-2018H in my list to be loaded at 5007H-5018H.
3. Bytes 2900H-29FFH in my list to be loaded at 5800H-58FFH.
4. 'SF DOS 5000 {CR} '—get new DOS from 5000H.

This technique has cut operator input time considerably, and will hold a maximum of 32 characters in queue, more than enough for bookkeeping purposes.

The character is output twice. Once when put in queue and again when the system (basic) takes it from the queue.

At any time the following keys have special uses:

Control E—jump to bootstrap load at E800H
Control O—jump to DOS
Control B—non destructive jump to basic
Control R—'run' basic program

I still look forward to a compiler. There are no doubt thousands of good working North Star basic programs in the field—all debugged and finalized—but which would welcome the extra speed of a compiler. But in the meanwhile this queueing technique is saving us a lot to time. ■

Robert T. Armstrong, P.O. Box 263, Toronto, Australia 2283

```
2007                      ORG     2007H    ;DOS
2007  C9                  RET              ;ORIGINALLY 'RETURN'
2008  2729                DW      OFTEN    ;AFTER INIT THEN JUMP TO OFTEN
200A  C36220              JMP     2062H
200D  C31C29              JMP     CONSOUT
2010  C35F29              JMP     CONSIN
2013  C30829              JMP     INIT
2016  C33329              JMP     CONTC

2900                      ORG     2900H
29FF  =         Q1        EQU     29FFH    ;TOP RAM TO HOLD 1ST IN QUEUE
2900  00        TEMP      DB      0
2901  FF29      Q         DW      Q1       ;PLACE FOR NEXT IN QUEUE LHLD-SHLD
2903  AE40AE3700DATA      DB      0AEH,40H,0AEH,37H,0
                ;-------------------------------------------------------
2908  210229    INIT      LXI     H,DATA-1
290B  23        INIT1     INX     H
```

Input Queuing, cont'd...

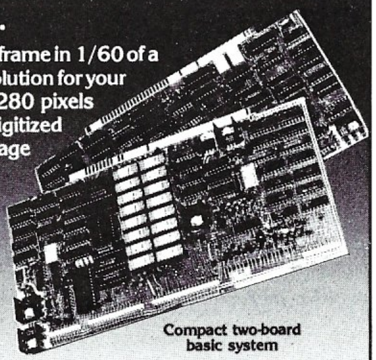
```
290C 7E                 MOV    A,M
290D D303               OUT    03        ;INIT CONSOLE
290F D305               OUT    05        ;INIT PRINTER
2911 FE37               CPI    37H       ;IS IT THE LAST
2913 C20B29             JNZ    INIT1
2916 3EC3               MVI    A,0C3H    ;JMP INSTRUCTION FOR 'OFTEN' ROUTINE
2918 320720             STA    2007H
291B C9                 RET
                ;------------------------------------------------------------
                        ;INSERT PRINTER OUTPUT ROUTINE AS NECESSARY
291C DB03       CONSOUT IN     03
291E E601               ANI    01
2920 CA1C29             JZ     CONSOUT
2923 78                 MOV    A,B
2924 D302               OUT    02        ;OUT TO CONSOLE
2926 C9                 RET
                ;------------------------------------------------------------
2927 DB03       OFTEN   IN     03
2929 E602               ANI    02
292B C8                 RZ               ;RETURN IF NO KEY HIT
292C DB02               IN     02
292E E67F               ANI    7FH
2930 C34129             JMP    OFT1
                ;------------------------------------------------------------
2933 DB03       CONTC   IN     03
2935 E602               ANI    02
2937 EE02               XRI    02
2939 C0                 RNZ              ;RETURN IF NO KEY HIT
293A DB02               IN     02        ;INPUT CHAR
293C E67F               ANI    7FH
293E FE03               CPI    03        ;IS IT CONTROL 'C'
2940 C8                 RZ               ;IF SO RETURN
2941 CD9929     OFT1    CALL   JUMPS
2944 E5                 PUSH   H
2945 2A0129             LHLD   Q         ;GET QUEUE LOCATION
2948 77                 MOV    M,A       ;PUT CHAR IN QUEUE
2949 7D                 MOV    A,L
294A FEDF               CPI    0DFH
294C CA2820             JZ     2028H     ;IF MORE THAN 32 CHAR IN QUEUE THEN TO'DOS'
294F DB03       CC1     IN     03
2951 E601               ANI    01
2953 CA4F29             JZ     CC1
2956 7E                 MOV    A,M
2957 D302               OUT    02        ;PRINT CHAR @ CONSOLE ONLY
2959 2B                 DCX    H
295A 220129             SHLD   Q         ;SAVE NEXT QUEUE LOCATION

295D E1                 POP    H
295E C9                 RET
                ;------------------------------------------------------------
295F E5         CONSIN  PUSH   H
2960 2A0129             LHLD   Q         ;GET QUEUE LOCATION

2963 7D                 MOV    A,L
2964 FEFF               CPI    0FFH      ;IS THERE A QUEUE
2966 C27829             JNZ    SPECIAL   ;GOTO SPECIAL IF A QUEUE
2969 E1                 POP    H
296A DB03       CC2     IN     03
296C E602               ANI    02
296E CA6A29             JZ     CC2
2971 DB02               IN     02        ;GET CHAR FROM CONSOLE AS NORMAL
2973 E67F               ANI    7FH
2975 C39929             JMP    JUMPS     ;TO RETURN
```

```
2978 D5         SPECIAL PUSH   D         ;TO GET CHAR FROM QUEUE
2979 C5                 PUSH   B
297A 11FF29             LXI    D,Q1
297D 01FE29             LXI    B,Q1-1
2980 3AFF29             LDA    Q1        ;THIS IS CHAR TO BE SENT
2983 320029             STA    TEMP      ;SAVE THIS UNTIL SEND
2986 0A         DO      LDAX   B
2987 12                 STAX   D         ;MOVE THE QUEUE UP
2988 0B                 DCX    B
2989 1B                 DCX    D
298A 7B                 MOV    A,E
298B BD                 CMP    L         ;IS THIS LAST CHAR IN QUEUE?
298C C28629             JNZ    DO        ;IF NOT MOVE REST OF QUEUE UP
298F 23                 INX    H
2990 220129             SHLD   Q         ;SAVE NEXT QUEUE LOCATION
2993 C1                 POP    B
2994 D1                 POP    D
2995 E1                 POP    H
2996 3A0029             LDA    TEMP      ;RETURN WITH CHAR
2999 FE0F       JUMPS   CPI    0FH       ;IS IT ^O
299B CA2820             JZ     2028H     ;GO DOS
299E FE05               CPI    05H       ;IS IT ^E
29A0 CA00E8             JZ     0E800H    ;GO BOOTSTAP LOAD
29A3 FE02               CPI    02        ;IS IT ^B
29A5 CA142D             JZ     2D14H     ;GO TO BASIC
29A8 FE12               CPI    12H       ;IS IT ^R
29AA C0                 RNZ
29AB AF         BASIC   XRA    A
29AC 320F2D             STA    2D0FH
29AF C3002D             JMP    2D00H     ;RUN BASIC
```

CAT-100 FULL COLOR GRAPHICS

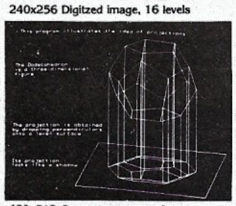
The original 256-color imaging system with high resolution video FRAME GRABBER for the S-100 bus.

Capture and digitize a video frame in 1/60 of a second. Select the best resolution for your application, from 256 to 1280 pixels per TV line. Display your digitized or computer processed image with 256 gray levels or 256 colors on standard B&W, NTSC or RGB color TV monitors.



Compact two-board basic system

240x256 Digitized image, 16 levels



480x512 Computer-generated

Features:

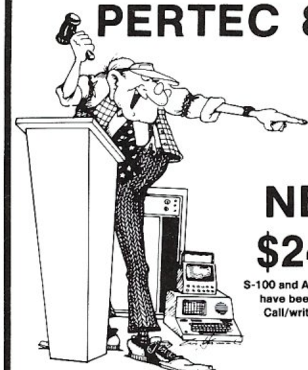
- Highest possible quality 480x512x8 digital video image presently available on the market
- Input capability from TV camera or other sources
- Variety of synchronization choices
- 2 selectable video A/D conversion circuits
- Choice of 1, 2, 4, 8, 16 or 32 bits per pixel
- 32K-byte image memory on the basic system
- 32, 64, 128 & 256K byte system capacity
- Lightpen input
- Photographic trigger control input
- Software selectable system parameters
- Interfaces for TRS-80 and other processors
- Comprehensive line of accessories, monitors and support software

SEND FOR FREE CATALOG

DIGITAL GRAPHIC SYSTEMS
441 California Ave., Palo Alto, CA 94306 415/494-6088

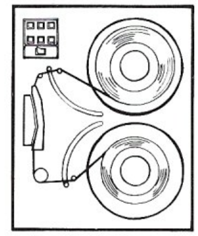
ALMOST AN AUCTION

9-TRACK TAPE DRIVES 800 BPI, 45 IPS PERTEC 8840A-9-45



NEW $2400

S-100 and APPLE interfaces have been developed. Call/write for details.



- **BARGAIN PRICED MINI COMPUTER UPGRADE**
- **READ THE NEW 9 DIGIT ZIP CODE TAPES AVAILABLE FROM USPS**

A large OEM's large overstock makes these industry standard drives available at a fraction of their current list price. Full size drives handle up to 10.5 inch reels of standard 1/2 inch mag tape. 19 inch rack mount or use right out of the box on steel shipping frame.

SPECIFICATION SUMMARY: 9-Track, 800 BPI, dual head (read-after-write), 45 ips read/write, 200 ips rewind, BOT/EOT sensing, 110 VAC/60Hz, recent manufacture, all i/o signals TTL/DTL compatible, tension arm tape buffering, full control panel, compatible with many other drives including lower priced Pertec 6840s.

PRICING: Current OEM list price is $4800+ quantity 1, $3400+ quantity 100. Cash price, new in original box with full manual, guaranteed, is $2400 FOB Morris Plains. Large quantity in stock in our warehouse, we are not brokers.

ELECTROVALUE INDUSTRIAL INC.
P.O. BOX 157 S
MORRIS PLAINS, NJ 07950
Formerly Electravalue Industrial

Phone reservations and questions are welcome.
201/267-1117

CATCH THE S-100 INC. BUS!

Spectacular Savings — July only.

Deduct an extra 5% from our already low prices for all prepaid orders.

We will be closed from July 25th until August 10, 1981.

Watch our August ad for a special announcement.

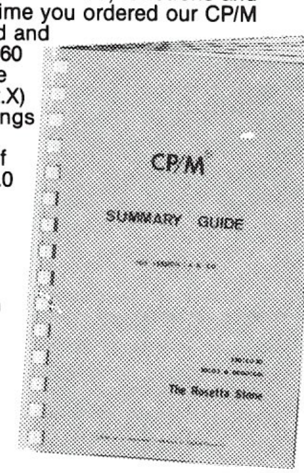
Subject to Available Quantities • Prices Quoted Include Cash Discounts. Shipping & Insurance Extra.

We carry all major lines such as S.D. Systems, Cromemco, Ithaca Intersystems, North Star, Sanyo, ECT, TEI, Godbout, Thinker Toys, SSM, Tarbell, Tecmar, Central Data, Micropro Int'l, Scion, CCS, Shugart.

S-100, inc.
7 White Place, Clark, N.J. 07066
201-382-1318
Hours: Mon. - Fri. — 10 a.m. to 6 p.m.

CP/M SUMMARY GUIDE

Tired of fanning through your CP/M manuals or writing notes that remind you of the commands, functions and error codes? Well it's about time you ordered our CP/M Summary Guide! Spiral bound and handy to hold, our guide is a 60 page booklet summarizing the features of CP/M (Ver. 1.4 & 2.X) and 2 totally alphabetical listings of the commands, functions, statements and error codes of MICROSOFT BASIC-80 Ver. 5.0 and CBASIC™ -2. Areas summarized are in table form and include all direct and transient commands plus MAC™, DESPOOL™ and TEX™. Our booklet is a much needed supplement to any of the literature currently available on CP/M and has been recommended by Digital Research.



P.S. Over 4000 users can't be wrong!

Ask your local computer store for our guide or send $6.95 plus $1.00 (postage and handling) to:

THE ROSETTA STONE, P.O. BOX 35, GLASTONBURY, CT 06025 (203/633-8490)

Name ______________________________

Street ______________________________

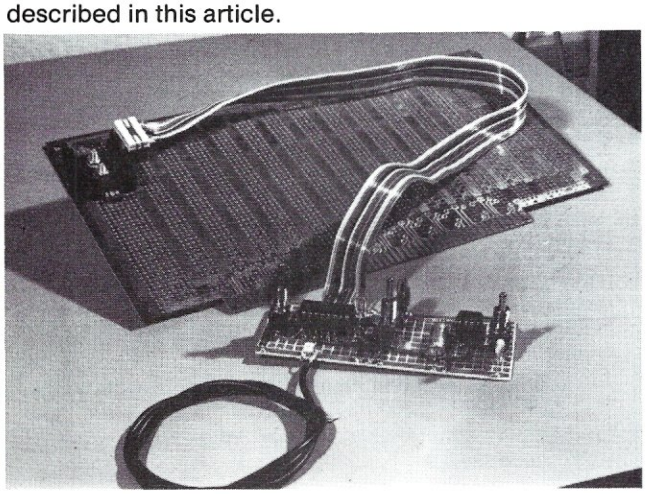
City ____________ State ________ Zip ________

CP/M™, DESPOOL™, MAC™ are registered trademarks of Digital Research. CBASIC™ is a registered trademark of Compiler Systems.

Variable Speed Automatic Slow Step For the Imsai 8080

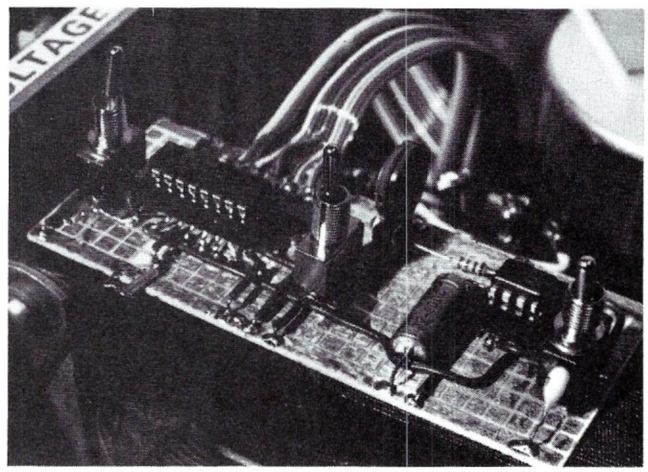
by Joseph W. Long

For some time I had been interested in adding an automatic slow step function to the front panel of an Imsai 8080 computer used by the Chemistry Department at Broome Cumminity College. Getting a look at one of the new Intersystems mainframes (the Electrical Department at BCC purchased a number of them) with its slow stepping front panel finally prompted me to see what could be done with the old Imsai. In the August 1977 *Kilobaud* I found one solution. An article by Howard Bendrot illustrated a simple modification for the Imsai front panel which required only one part and cutting a few traces on the front panel. Bendrot's approach, while simple, suffers from the problem that the slow step speed is not variable. That fact, coupled with my desire not to make irreversible modifications to the front panel, led me to develop the variable speed slow step circuit described in this article.



Complete slow step system. Multiple conductor is used only for +5V and GND. Note uncluttered layout of S-100 board.

Joseph W. Long, Broome Community College, Box 1017, Binghamton, NY 13902.



Closeup of slow step circuit mounted on Imsai Transformer. RG 174 coax was used to connect output to front panel. Double sided tape and C2 were added after photo was taken. Left to right, switches are S2, S3 and S1.

Study of the Imsai front panel schematic shows that the only requirement for single stepping the Imsai is to pull pin 1 of U17 to logic low. With the Imsai in the stop mode, I found that connecting a square wave generator to pin 1 produced slow stepping at the square wave frequency. A direct connection is not really practical however, because it interferes with the normal single step operation of the front panel. One solution to this problem is to run the generator through a tristate buffer. Disabling the buffer completely isolates the clock sign from U17. To keep the entire circuit internal to the Imsai, I decided to build in a clock, using a 555 timer. The clock circuit is very simple, requiring ony a few parts beyond the 555. Figure 2 shows the final circuit.

The range switch is necessary to give a wide range of stepping rates. The range covered by both capacitors is from about one step per ten seconds to over 400 steps per second. C2 on U17 (Figure 1) limits the maximum slow step speed to around 400 steps per second. Decreasing its value should allow higher slow step speeds, but I have not tried changing the capacitor. I'm not sure it wouldn't foul up the normal front panel single step function.

Operation of the circuit is very simple. If S2 and S3 are both open, the front panel operates in the normal way. With the front panel in the stop mode, and either S2 or S3 closed, the computer will slow step at a speed depending upon the setting of S1 and R1.

The circuit works well and causes no glitches or problems that I am aware of. While it's more complex than Bendrot's circuit, it is more versatile and requires no modification of the front panel. The circuit even has an advantage over the Intersystems computer—on that machine, you must remove the front panel cover to change the slow step speed.

I gave a lot of thought to where to position the finished circuit. Finally, I decided to mount it inside the Imsai, since the cover is usually off when I want to single step or slow step. The power transformer is a convenient mounting place—I provided a little breathing space for the transformer by securing the circuit with a couple of strips of a half-inch thick double sided tape. In order to keep everything modular, I must confess that I set up an entire Vector S-100 breadboard card to supply the 5V needed by the circuit. No doubt other people could come up with a more reasonable power source.

An experience I had with the circuit may be of some interest. The Imsai computer runs Cromemo's Control Basic and I thought it would be interesting to slow step through a simple program to see how many machine language steps are really involved. (I had tried this previously, using the single step switch without success.) I used the loop 10 PRINT "HELLO", 20 GOTO 10. This program running in the slow step mode at 100 steps per second took about 45 seconds to loop. That comes out at 4500 machine language steps. I hadn't realized how much interpreting actually goes on in an interpreter! Interesting speed comparisons are possible; the same program run under identical conditions using Northstar Basic required approximately thirty seconds to execute.

I would like to express my appreciation to my brother, David Long, for his advice on the 555 timer portion of this project and to John Young, the Broome Community College photographer for his photographic efforts. ■

Figure 1: Original IMSAI circuit. Added wiring shown with dotted line.

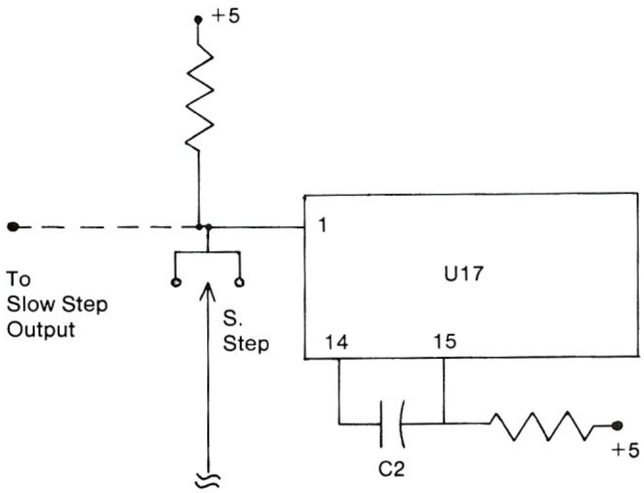
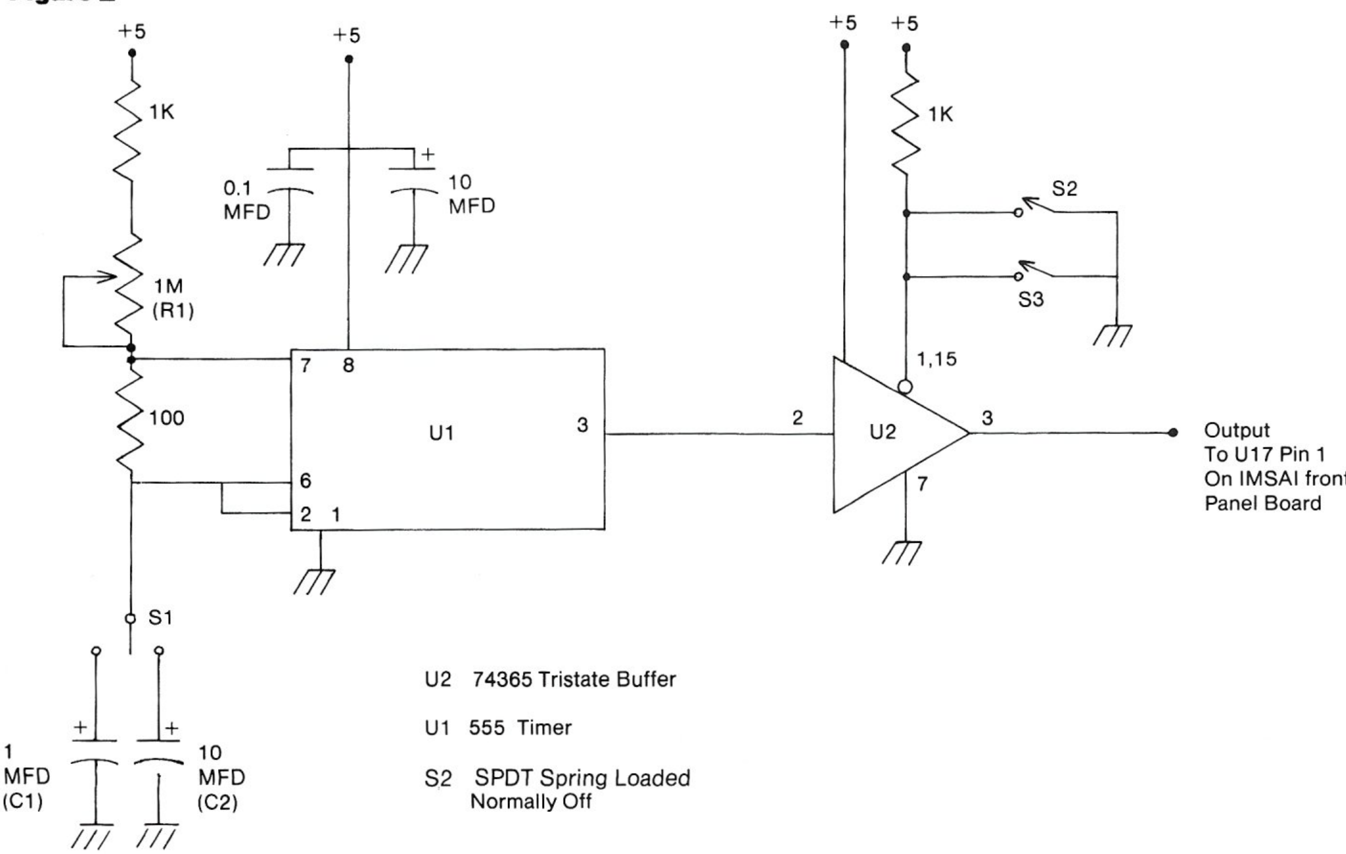


Figure 2



U2 74365 Tristate Buffer

U1 555 Timer

S2 SPDT Spring Loaded
Normally Off

Hardware Product Review

The Televideo 920-C Terminal

by Glenn A. Hart

The Televideo 920-C serial terminal is the flagship in a line of low cost yet highly flexible serial video terminals. While it does have certain problems, the Televideo terminals allow both the microcomputerist on a budget and the professional user requiring multiple terminals to achieve a level of performance previously unattainable at such a reasonable price.

All four Televideo models are based on the same chassis and electronics, differing primarily in the keyboard layout. The 920 models include eleven special function keys, six editing keys and two transmission keys. Each function key can generate two code sequences depending on whether the shift key is depressed, so a total of twenty-two special codes are available. (The physical design does not provide any convenient place above the function keys to indicate the functions assigned to each key, a desirable feature found on some more costly terminals.) The 912 models do not include the special function keys, but all models have 14-key numeric keypads, six cursor movement keys and various other special keys for excellent flexibility. All keys will repeat at a 15 CPS rate when held down.

Both the 912 and the 920 series are available with a choice of keyboard layout, indicated in the model number by either a -B or -C suffix. The B models have a layout loosely based on a Teletype keyboard, while the C models have a Selectric-based layout with oversized RETURN and TAB keys. The location of several characters is completely different (", ′ ,@,] ,etc.). The Selectric layout is easier to use and more familiar to traditional typists, but if you are used to the computer layout it may take a while to make the transition. The C models also cost quite a bit more; the choice is up to you.

The screen displays the traditional 24 lines by 80 characters. The full 96 character ASCII character set is generated in a 7 X 10 matrix with 12 X 10 resolution, resulting in a type font that is elegant and easy to read, with lower case descenders and the ability to underline. No special graphics characters are provided. The clarity of the on-screen characters is reasonably good. While it is definitely far better than many low cost terminals I have used, it is not the equal of some more costly terminals. I have used the unit for six and eight hour stretches without eye fatique, so the 12-inch black-and-white CRT certainly provides reasonable video performance. Keyboard feel is a bit on the firm side compared to some other terminals, but provides a good level of feedback to the operator.

All Televideo terminals operate at a choice of nine Baud rates from 75 Baud to 9600 Baud. Documentation for earlier units indicated that 19,200 Baud could be used. There is an obvious switch position for this speed, but evidently there were operating problems and the documentation supplied with newer units does not mention 19,200 Baud. Even, odd, mark, space or no parity is available, and the terminals can be used in either normal RS-232 or 20ma current loop modes. An RS-232 printer port is supplied. Both full and half duplex conversational as well as block mode is available.

The 920-C is microprocessor and software driven. Intel's 8035 microprocessor provides much of the operating flexibility of the terminal, with the software routine stored in a ROM. There is a price to pay for this flexibility, but more about that later.

The list of functions available is impressive (see Table 1). Most of the codes are Escape sequences, with few using control characters for compatibility. I am told that the commands resemble those of the ADM-31 terminal.

The average user will concentrate his attention on the normal cursor movement commands and few of the special formatting options. Absolute cursor addressing is handled in a normal fashion and the position of the cursor can also be read by a program. Various attractive and useful formats can be designed by combining the half-intensity, reverse video, blinking and underlining features. All these commands can work on a character-by-character basis for careful control. One peculiarity is the extra character position that some of these commands take when they execute; this sometimes requires a bit of juggling in the formatting routines.

Glenn A. Hart, 51 Church Rd., Monsy, NY 10952.

Many of the other editing and special features are not available in the normal conversational mode and are intended for block mode use. Since CP/M and other microcomputer operating systems are character oriented, these functions are of little practical use. For the mainframe user, a full spectrum of editing and block mode features is available.

While the editing features would not normally be used in a microcomputer environment, they are often used by applications software. Commands like Erase to End of Line, Insert Line and Erase to End of Screen are often issued by word processors and other programs to speed on-screen activity. The Televideo terminals have problems when such commands are issued by the computer. The terminals were designed to handle *keyboard* entry of these commands correctly, but the microprocessor/software combination is simply a bit too slow to react correctly. Word-Star, for example, will send several consecutive commands when it is necessary to scroll, position the cursor, insert lines, etc., and the TVI will almost always drop at least one character. The Musicraft music entry system uses Erase to End of Line and Insert Line frequently; the terminal will often sound its bell and garbage up the screen with any Basic or other high level languages; presumably they are either slow enough to avoid the problem or don't make use of the troublesome commands at all.

The design engineer at Televideo explained that the 8035 is running at its full designed clock speed. (Some of the terminals using a Z-80 may respond faster.) TVI sent me a new ROM with somewhat faster routines which completely solved the Musicraft problem but still could not totally handle Word-Star. The answer with Word-Star is to disable the use of the special functions by patching the program. This reasonably easy step causes Word-Star to generate the required actions in software instead and results in perfect, although very slightly slower, operation. I don't know TVI's policy on upgrading older units, but I would assume that all new production uses the faster ROM.

I also experienced some reliability problems. Several hours after first powering up the terminal, the power supply module blew a capacitor. I was chagrined to find that TVI warranty covers only in-factory repair, which would have meant sending the unit back to California. TVI's ads inicate that General Electric field service is available. This is true, and service contracts can be purchased to cover the period after the 90 day warranty expires. However, TVI doesn't authorize GE to perform warranty service, so repairs during the warranty period must either be at the factory or at the owner's expense. Some other terminal manufacturers have made field service arrangements similar to TVI's, but they evidently also permit in-warranty repairs at a replacement power module immediately. I don't know if this is something they would do for all customers, but it certainly helped me out tremendously.

All in all, I have been quite happy with the 920-C. It is flexible and easy to use, and has provided many long hours of dependable service once its initial problems were sorted out. Considering the heavy discounts at which the entire Televideo line is commonly sold, TVI terminals offer a very positive cost/performance ratio. The 920-C has more features than many terminals selling for much more, and is a clear winner when compared with terminals selling at or near its price. ■

Table One
TVI-912/TVI-920 Command Sequences

Function	Sequence	Function	Sequence
Beep	Control-G	Read Cursor	Escape ?
Cursor Left	Control-H	Set Block Mode	Escape B
Cursor Down	Control-J	Set Conversation Mode	Escape C
Cursor Up	Control-K	Print Page	Escape P
Cursor Right	Control-L	Character Insert	Escape Q
Home Cursor	Control-~ or Control-^	Character Delete	Escape W
Tab	Control-I	Line Insert	Escape E
New Line	Control-_ (underscore)	Line Delete	Escape R
Protect Mode On	Escape &	Line Erase to Space	Escape T
Protect Mode Off	Escape '	Page Erase to Space	Escape Y
Start Half Intensity	Escape)	Back Tab	Escape I
End Half Intensity	Escape (	Toggle Page	Escape K
Set Column Tab	Escape 1	Start Blink Field	Escape ^
Clear Tab	Escape 2	Start Blank Field	Escape _ (underscore)
Clear All Tab	Escape 3	End Blink/Blank	Escape q
Send Line Unprotect	Escape 4	Tab	Escape i
Send Page Unprotect	Escape 5	Start Inverse Video	Escape j
Send Line All	Escape 6	End Inverse Video	Escape k
Send Page All	Escape 7	Start Underline	Escape l
Clear All to Space	Escape + or Escape Z	End Underline	Escape m
Clear All to Null	Escape *	Line Erase to Null	Escape t
Clear Foreground to Null	Escape :	Page Erase to Null	Escape y
Clear Foreground to Space	Escape ; or Control-Z	Auto Flip On	Escape v
Keyboard Enable	Escape "	Auto Flip Off	Escape w
Keyboard Disable	Escape #	Extension Port On	Escape @
Load Cursor	Escape =	Page Print Mode On	Escape A

An 8086/8088 Reference Book

by Chris Terry

The 8086 Book, by R. Rector and G. Alexy. Osborne/ McGraw-Hill, 608 pp., $16.99. 1980.

This substantial book is a very good value for the money, and I have a strong feeling that it will become the standard 8086/8088 reference work. As is true of all the books that I have seen from Osborne Associates, it is well organized, cleanly and clearly written, and loaded with diagrams. Good paper, a very readable typeface, and judicious use of boldface enhance the communication, making *The 8086 Book* a pleasure to use. The book is divided into ten chapters, the first six discussing software and the instruction set, and the last four concerned with the hardware aspects.

Software Aspects

Chapter 1, "Programming," is a crisp, pertinent, and sometimes amusing exposition of the six aspects of the programming task: Specification, Design, Implementation, Testing, Documentation and Maintenance. There's nothing new here, but it's a valuable reminder of what it takes to create a good program.

Chapter 2, "Some Program Examples," discusses the design aspects, at the flow-chart level, of a sort program and associated I/O routines. This is a preparation for chapter 6, "Examples of 8086 Assembly Language Programming," which shows the implementation. Chapter 6 is very valuable; it does not merely supply code, but shows alternative ways of coding certain functions, and discusses their impact on storage space and execution speed.

Chapter 3, "The 8086 Instruction Set," is the longest chapter in the book. After a seventeen-page introduction mainly concerned with design considerations for an I/O driver using the 8251 USART (which I think would have been better placed in chapter 2), we get down to business. First comes a description of the 8086 registers, and how various groups of instructions affect the Status Register flags. Next, there is a detailed description of the six basic addressing options; Immediate, Direct, Direct Indexed, Implied, Base Relative, and Stack. This section includes the mechanisms by which addresses are computed, and the part played by the segment registers. Finally we come to a detailed description of each 8086 instruction, in alphabetical order of mnemonics. Here, the very clear diagrams detail what the instruction does; notes provide clarification and indications of the practical uses of the instruction.

Chapter 4 groups the instructions according to their functions: Data Movement, Arithmetic, Logic, String Primitives, Program Counter Control, I/O, Interrupt, and Rotate and Shift. The information here is mainly tabular, and promotes a better understanding of the instructions by discussing them with a different slant.

Hardware Aspects

Chapter 7 is a clear and detailed description of basic 8086 system concepts and architecture, with particular reference to the use of the data and address buses. Chapter 8 discusses operating modes, interrupts and a timing in single-CPU system. The excellent diagrams include configurations for DMA (Direct Memory Access). Chapter 9 discusses the Intel Multibus, and describes the function of each line. And finally, Chapter 10 discusses multiprocessor configurations.

Of the four appendices, A and B list the instruction set alphabetically by mnemonic, and numerically by hex value of the operation code. Appendix C contains data sheet reprints giving AC and DC signal characteristics and signal waveforms for the 8086, 8088, and support chips of the same family. Appendix D discusses the differences between the 8086 and the 8088; the instruction sets are identical, but the 8088 operates with an 8-bit data bus and therefore uses two bus cycles instead of one to access 16 bits of data.

Comments

The descriptions of how the various addressing modes operate are detailed and as comprehensible as one could expect considering their complexity. The same goes for the use of the segment registers. However, I long for some indications of the *purpose* behind it all. Although not a professional, I consider myself a moderately competent 8080 programmer, and I can see the point of indexed and relative addressing. But why would anyone want to use base relative, direct, indexed stack addressing? Obviously someone does, or it would not be included. But *who*—and what for? Similarly, what is the advantage of having a Code segment, a Data segment, a Stack segment, and an Extra segment of memory? For multi-user systems? Maybe, since this element of purpose is something that gets left out of manuals all too often. Without it, the mass of detail on "what" and "how" tends to overwhelm a reader who has no experience with comparable procedures, because he or she is working in an application vacuum. Some guidelines for when

Chris Terry, 324 East 35th St., New York, NY 10016.

and how to use these features would have been far more valuable than the elementary material that now constitutes chapter 5.

I question the value of the first three pages of chapter 1 (which contains highly simplified remarks about the functions of Assembly Language and assembler programs) and chapter 5 (which contains elementary descriptions of the functions of an editor, an assembler, and a debugger). I have a suspicion that the material was included to appease some editor who complained that terms were being used without being defined. I can only say that anyone who does not have a firm grasp of this material at a much more detailed level is not ready to struggle with the complexities of the 8086. This material cannot possibly prepare a neophyte adequately for the rest of the book, and is just padding for any programmer with more than a few weeks experience with assembly language.

The index is generally useful, although it has a few quirks (e.g., the sort program of chapters 2 and 6 is listed under "Shell sort," not "Sort," and only the chapter 6 reference is listed). I found a few typographical errors and a reference to a non-existent procedural step—but such flaws are few and very minor.

Don't, on any account, let my complaints and wish-list stop you from rushing out to buy this book if you are considering using the 8086, or if you have one already. It's a fine piece of work. And nobody has ever managed to write a book for which someone else (with the benefit of hindsight) could not suggest improvements! ■

An S-100 Clock/Calendar Circuit

by Fred J. Deadrick

I recently put together a very simple and reliable clock and calendar interface for my IMSAI-8080 system. The board uses a new chip designed by OKI Semiconductor, 1333 Lawrence Expressway, Santa Clara, CA 95051. The IC is called the MSM5832 Microprocessor Real-Time Clock/Calendar. It is oriented to microprocessor use and provides 4-bit data of seconds, minutes, hours, day-of-week, month and year. Data access is controlled by 4-bit addressing. It includes 12/24 hour selection, leap year identification and manual plus or minus 30 second correction. The chip comes in an 18 pin DIP package, designed for crystal control frequency reference, and can use standby battery backup. I bought the IC from Concord Computer Products, 1973 South State College, Anaheim, CA 92806, (714)937-0637. The cost is $8.50 plus tax and shipping.

The following is a sample of the output format from my clock/calendar program:

22:42:45 Tuesday 01-JUL-81
08:31:52 Wednesday 09-JUL-81

I've designed the software to generate a 30 character ASCII string which is displayed in the upper corner of my memory-mapped video terminal, print on my assembler listings, and use anywhere I need to document the time and date.

Hardware

I constructed the clock/calendar circuit on an S-100 prototyping board. The circuit uses only seven IC's and occupies only a quarter of the board, leaving room for future projects. The interface to the S-100 bus follows a design by Condra, using an 8255 Programmable Peripheral Interface IC in a bi-directional mode to communicate with the clock chip. Two latched output ports and one input port are needed for the interface. The clock/calendar IC also requires a 32.768 KHz crystal for its internal clock circuit to operate. I extracted one from an old LED wristwatch I had lying around. I also used the small trimmer capacitor from the watch for the time adjustment trimmer.

For battery backup I decided not to fool around with using a NICAD re-chargeable battery. Instead I selected an alkaline 4.5 Volt photoflash battery I purchased in a local drug store. I used the Mallory PX21. The capacity of this battery is 580 Ma-Hrs; at the measured current drain of the clock chip (20 micro-amperes), the clock should keep on running for 3.3 years!

Once the trimmer is adjusted, the accuracy of the clock is excellent. I have run my board for nearly a year and found it to be accurate to better than five seconds/month. No glitches have been observed during the times I turned the computer power on or off. I can even remove the board from my mainframe without affecting the time of the clock.

Software

The software I use to read the clock circuit is shown in Listing 1. The part of the program specific to the 8255 PPI-IC is the CLKRD subroutine. If you build the circuit with some other interface, this part of the software will have to be altered to fit the IC used. CLOCK is a subroutine which generates a 30 character ASCII string containing the time, day and date (as was shown earlier). On entry, H&L registers are set to point to the location where the ASCII string is to be stored. I use the CLOCK subroutine to display the time and date on my video terminal, and periodically update the time by calling the CLOCK routine while in the keyboard input status wait loop.

Use of the interface is not restricted to assembly language routines. Listing 2 shows a very simple program written in North Star Basic to read and display the clock/calendar data. It would be a very simple task to re-format the output to the needs of a user.

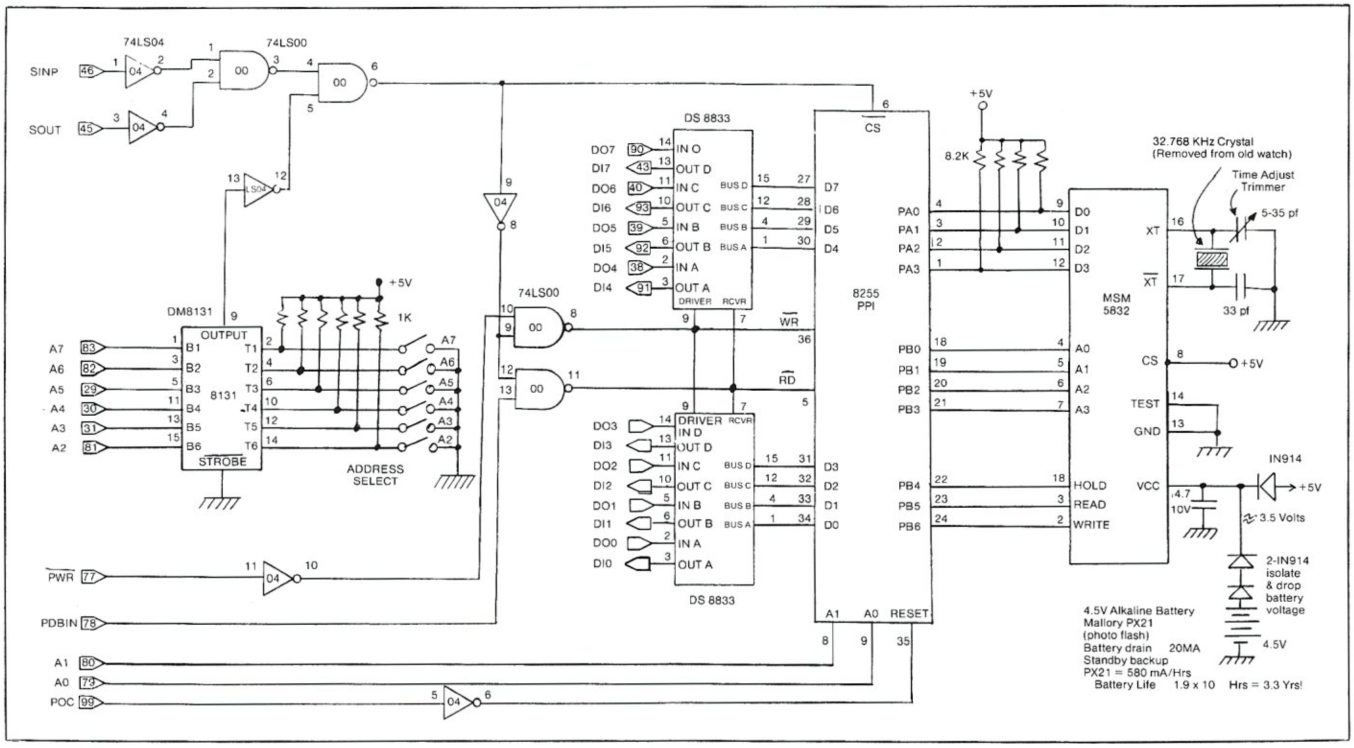
Finally, I've included in Listing 3 the program I used to initialize the clock/calendar IC. I've only used this program a few times because the chip keeps such good time, but it is needed to get your chip going.

Give this simple clock/calendar interface a try. You will be surprised how easy it is to build—and how handy it is to have the time and data available to your S-100 system. ■

References

Condra, David L., "Interfacing the S-100 Bus with the Intel 8255," *Byte*, Vol. 4/No. 10, October, 1979.

Fred J. Deadrick, 2544 Leeds Court, Livermore, CA 94550.



Listing 1

```
;
;RDCLK IS A DUMMY DRIVER PROGRAM USED FOR CALLING THE CLOCK SUBROUTINE.
;IT PUTS THE 30 CHARACTER ASCII TIME, DAY AND DATE STRING UP IN THE UPPER
;RIGHT HAND CORNER OF THE SCREEN SPLITTER MEMORY MAPPED VIDEO DISPLAY.
;
STRING  .EQU    0F032H  ;PUT TIME ON CRT DISPLAY
;
;
        .ORG 01000H
RDCLK:  LXI H,STRING
        CALL CLOCK      ;GO READ CLOCK
        RET             ;RETURN BACK TO MONITOR
;
;
;
;
;CLOCK--IS A SUBROUTINE TO GENERATE A 30 CHARACTER STRING CONTINING
;       THE TIME, DAY-OF-WEEK, DAY, MONTH, AND YEAR IN THE FORMAT
;       12:34:56  WEDNESDAY  29-JUN-80.
;
;
;INPUT:  H&L POINT TO A 30 CHARACTER ASCII STRING OUTPUT BUFFER
;        ALL REGISTERS ARE USED AND DESTROYED BEFORE RETURNING.
;
;
CLOCK:  PUSH    H       ;SAVE CLOCK STRING ADDR
        LXI H,CLKTBL    ;POINT TO CHIP DATA BUFFER
        CALL CLKRD      ;GO READ CLOCK CHIP
        LDA CLKTBL+5    ;MASK OFF 24 HR BIT
        ANI 3H
        STA CLKTBL+5
        LDA CLKTBL+8    ;MASK OFF LEAP YR BIT
        ANI 3H
        STA CLKTBL+8
;
;CONVERT TIME DATA TO ASCII STRING
;
        LXI B,CLKTBL+5  ;START AT H10
        POP H           ;GET STRING BUFFER POINTER
        LXI D,0302H     ;3 GROUPS OF 2 DIGITS
TIME:   LDAX B
        ADI 30H         ;CONVERT TO ASCII.
        MOV M,A
        INX H
        DCX B
        DCR E
        JNZ TIME        ;GET UNITS VALUE
        DCR D
        JZ DAY          ;DONE WITH TIME, DO DAY
```

```
        MVI E,2
        MVI M,':'       ;PUT IN COLON
        INX H
        JMP TIME
;
;DAY-OF-WEEK  THE 7TH BYTE IS THE DAY OF THE WEEK DIGIT
;             0=SUNDAY, 6=SATURDAY
;
DAY:    MVI M,' '       ;PUT 2 SPACES IN STRING
        INX H
        MVI M,' '
        INX H
        PUSH H          ;SAVE STRING POINTER
        LDA CLKTBL+6    ;GET DAY DIGIT
        LXI D,9         ;9 CHARACTERS PER DAY
        LXI H,DAYTBL
DAY0:   DCR A           ;COMPUTE DAY TABLE ENTRY
        JM DAY1
        DAD D
        JMP DAY0
DAY1:   POP B           ;BC=ASCII STRING PTR
DAY2:   MOV A,M
        STAX B          ;XFER DAY STRING
        INX H
        INX B
        DCR E
        JNZ DAY2        ;CONTINUE FOR 9 CHAR
        JMP DATE
DAYTBL: .ASCII 'Sunday   '
        .ASCII 'Monday   '
        .ASCII 'Tuesday  '
        .ASCII 'Wednesday'
        .ASCII 'Thursday '
        .ASCII 'Friday   '
        .ASCII 'Saturday '
;
;DATE--CONVERT THE CHIP DATA TO DAY-MONTH-YEAR
;
DATE:   MVI A,' '       ;STORE 2 MORE SPACES
        STAX B          ;BC NOW STRING POINTER
        INX B
        STAX B
        INX B
        LXI H,CLKTBL+8  ;GET DAY*10 OF MONTH
        MOV A,M
        ADI 30H         ;CONVERT TO ASCII
        STAX B
        INX B
        DCX H
        MOV A,M         ;GET DAY*1 OF MONTH
        ADI 30H
        STAX B
        INX B
        MVI A,'-'
        STAX B          ;PUT IN A "-"
        INX B
        PUSH B          ;SAVE ASCII STRING POINTER
        LXI H,CLKTBL+10 ;POINT TO M10
        MOV A,M
        ORA A           ;CHECK FOR 0 DIGIT
        JZ DATE0
        MVI A,10        ;ADD 10 FOR MONTHS > 9
DATE0:  DCX H
;
;
CLKRD:  MVI A,MODE8     ;PUT 8255 IN PROPER MODE
        OUT CNTRL
        MVI A,RDHLD     ;SET READ AND HOLD LINES
        OUT BPORT
        MVI B,NDELY     ;WAIT ABOUT 150 US
WAIT1:  DCR B           ; BEFORE READING CHIP
        JNZ WAIT1
        MVI B,RDHLD     ;TURN ON READ & HOLD
LOOP1:  MOV A,B
        CPI RDHLD+13    ;SEE IF DONE READING
        JZ HLDOFF       ;YES, DONE
        OUT BPORT       ;SET UP ADDR & CONTROL LINES
        IN APORT        ;GET CLOCK DATA
        MOV M,A         ;SAVE DATA
        INX H
        INR B
        JMP LOOP1       ;GET NEXT DIGIT
HLDOFF: XRA A           ;ALL DONE, RELEASE HOLD
        OUT BPORT
        RET
;
CLKTBL: .RES    13      ;CLOCK DATA RAM BUFFER
;
        .END
```

Listing 2

```
10 REM--THIS IS A NORTH STAR BASIC PROGRAM TO READ THE
20 REM--CLOCK/CALENDER INTERFACE.  THE PROGRAM ASSUMES THAT
30 REM--THE 8255 PPI I/O PORT ASSIGNMENTS ARE AS FOLLOWS.
40 REM-----CLOCK DATA I/O (4 BITS) = PORT 50H (80 DECIMAL)
50 REM-----ADDRESS & CONTROL (7 BITS) = PORT 51H (81 DECIMAL)
60 REM-----8255 MODE CONTROL (8 BITS) = PORT 53H (83 DECIMAL)
65 REM
70 DIM C(13),D$(63),M$(36)
80 D$(1,63)="SUNDAY MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY"
90 M$(1,36)="JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC"
100 REM--SET 8255 IN MODE8
110 OUT 83,144
120 REM--TURN ON CLOCK CHIP READ AND HOLD LINES
130 A1=48
140 OUT 81,A1
150 REM--READ THE 13 BYTES OF CLOCK DATA
160 FOR J=1TO13
170 C(J)=INP(80)
180 OUT 81,A1+J
190 NEXT J
200 REM--TURN OFF READ AND HOLD LINES
210 OUT 81,0
220 REM--TAKE OFF THE 24 HOUR BIT (BIT2)
230 C(6)=C(6)-8
240 REM
250 REM--NOW PRINT OUT THE TIME, DAY AND DATE
260 REM
270 PRINT %1I,C(6),C(5),":",C(4),C(3),":",C(2),C(1)
280 PRINT D$(9*C(7)+1,9*C(7)+9)
290 M=3*(C(11)*10+C(10))-2
300 PRINT M$(M,M+2)," ",%2I,C(8)+10*C(9),", ",%1I,C(13),C(12)
310 PRINT
```

Listing 3

```
;
;THIS IS A SUBROUTINE TO SET THE MSM5832 CLOCK/CALENDAR INTERFACE.
;
;
;SYSTEM EQUATES
;
COUT    .EQU    200DH           ;NORTHSTAR OUTPUT
CINP    .EQU    2010H           ;NORTHSTAR INPUT
APORT   .EQU    50H             ;8255 PORT A--CLK DATA I/O
```

```
BPORT   .EQU    51H             ;8255 PORT B--CLK ADR & CNTRL
CNTRL   .EQU    53H             ;8255 MODE CONTROL PORT
NDELY   .EQU    15H             ;DELAY CONSTANT
WRHLD   .EQU    50H             ;WRITE & HOLD BITS
MODE0   .EQU    80H             ;8255 MODE 0
HLDBIT  .EQU    10H             ;HOLD BIT POSITION
WRTBIT  .EQU    50H             ;WRITE BIT POSITION
;
;
        .ORG    1000H
;
CLKSET: LXI     H,MSG1          ;PRINT HEADER MESSAGE
        CALL    MSG
        LXI     H,TBUF1         ;POINT TO TIME BUFFER
        CALL    CIN             ;READ KEYBOARD
        CALL    CRLF
        LXI     H,TBUF1         ;POINT TO BUFFER AGAIN
        MOV     A,M
        ORI     8H              ;ADD ON 24HR BIT
        MOV     M,A
        LXI     H,MSG2          ;POINT TO DAY OF WK MSG
        CALL    MSG
        LXI     H,TBUF2         ;DAY BUFFER
        CALL    CIN
        CALL    CRLF
        LXI     H,MSG3          ;DATE MESSAGE
        CALL    MSG
        LXI     H,TBUF3         ;DATE BUFFER
        CALL    CIN
        CALL    CRLF
RDY:    LXI     H,MSG4          ;READY TO SET MSG
        CALL    MSG
        CALL    CINP            ;WAIT FOR A CR TO SET CLK
        CPI     0DH
        JNZ     RDY             ;LOOP BACK IF NOT A CR
        CALL    CLKWRT          ;OK, GO SET THE CLOCK
        CALL    CRLF
        RET                     ;RETURN BACK TO THE MONITOR
;
;SUBROUTINE TO SEND MESSAGE TO CONSOLE--H&L POINT TO MSG
;
MSG:    MOV     M,A             ;CHECK FOR 0 BYTE TERMINATOR
        ORA     A
        RZ                      ;RETURN IF 0
        MOV     B,A
        XRA     A               ;0 FOR VECTORED OUTPUT
        CALL    COUT            ;NORTHSTAR DOS OUTPUT
        INX     H
        JMP     MSG
;
;CARRIAGE RETURN-LINE FEED SUBROUTINE
;
CRLF:   XRA     A               ;0=VECTORED OUTPUT DEVICE
        MVI     B,0DH
        CALL    COUT
        MVI     B,0AH
        CALL    COUT
        RET
;
;CONSOLE INPUT ROUTINE
;
CIN:    XRA     A               ;0=VECTORED INPUT DEVICE
        CALL    CINP
        MOV     B,A
        XRA     A
        CALL    COUT            ;ECHO CHARACTER INPUT
        MOV     B,A
        CPI     0DH             ;CHECK FOR RETURN
        RZ
        ANI     0FH             ;MASK LOWER NIBBLE
        MOV     M,A
        DCX     H               ;DECREMENT DATA POINTER
        JMP     CIN             ;GET MORE INPUT TILL CR
;
;CLOCK WRITE SUBROUTINE--USED TO INITIALIZE DATA IN CLK CHIP
;
CLKWRT: LXI     H,TBUF0         ;POINT TO BUFFER
        MVI     A,MODE0         ;INITIALIZE 8255
        OUT     CNTRL
        MVI     A,HLDBIT        ;SET HOLD LINE
        OUT     BPORT
        MVI     B,NDELY         ;WAIT FOR THINGS TO SETTLE
WAIT:   DCR     B
        JNZ     WAIT
        MVI     B,HLDBIT        ;GET HOLD BIT + ADDRESS
WRLOOP: MOV     A,B
        CPI     HLDBIT+13       ;TEST TO SEE IF DONE
        JZ      HLDOFF          ;DONE, TURN OFF HOLD
        OUT     BPORT           ;WRITE ADDRESS TO CHIP
        MOV     A,M
        OUT     APORT           ;SEND DATA TO CHIP
        MOV     A,B
        ORI     40H             ;OR WITH WRITE PULSE
        OUT     BPORT           ;WRITE DATA TO CHIP
        ANI     1FH             ;MASK OFF WRITE BIT--KEEP HOLD
        OUT     BPORT
        INX     H
        INR     B               ;INCREMENT POINTERS
        JMP     WRLOOP
;
HLDOFF: XRA     A               ;TURN OFF HOLD BIT
        OUT     BPORT
        RET
;
;TERMINAL MESSAGES
;
MSG1:   .ASCII  'CLOCK/CALENDAR Initialization Program'
        .BYTE   0D
        .BYTE   0DH
        .BYTE   0AH
        .ASCII  'Input Set Time (HHMM)  '
        .BYTE   0H
MSG2:   .ASCII  'Input Day of the Week (0=Sunday, 6=Saturday
        .BYTE   0H
MSG3:   .ASCII  'Input Year, Month and Day (YYMMDD)  '
        .BYTE   0H
MSG4:   .ASCII  'When Ready, PRESS RETURN to Initialize Clock.
        .BYTE   0H
;
;BUFFER STORAGE
;
TBUF0:  .BYTE   0               ;CLOCK DATA BUFFER
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   0
TBUF1:  .BYTE   0
TBUF2:  .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   0
        .BYTE   0
TBUF3:  .BYTE   0
;
        .END
```

SOFTWARE DIRECTORY

Program Name: Energy Basic
Hardware System: CP/M 2.2 & I.D.S. Modem
Language: Machine Code
Description: Energy Basic is a high level language designed to simplify implementation of energy management systems and similar applications requiring monitoring of time, elapsed time, temperature, kilowatt demand, digital inputs, and control of devices based on such information. It provides the Basic language constructs including FILL, FOR, GOTO, GOSUB, IF, INPUT, LET, LIST, NEXT, OUT, PRINT, RETURN, REM, RUN, STOP, WAIT, ABS, CALL, EXAM, INP, RND AND SIZE. Special commands and functions include MODE, SET, ANSW, ELAP, ORIG, PSWD, TEMP and TIME. For example, X=TEMP(0) sets X to current temperature at sensor 0; T=TIME sets T to current time of day; SET causes current time of day to be set; ANSW places system modem in auto-answer mode; ORIG causes a data communications call to be established to current Originate telephone number; ELAP(A) returns time which has elapsed since A was set equal to TIME; etc. Energy Basic supports a primary system console device, an optional system printer, and an optional originate/answer modem. Energy Basic is available as a development system on 8" or resident on two 2716 type PROMs for dedicated control applications. The application program may also reside in 2716 type PROM. The Development System version of Energy Basic also supports the following commands and functions: BYE, LOAD, NEW, SAVE, and SIZE. LOAD and SAVE retrieve and store Energy Basic source programs to and from disk storage.
Release: January 1981
Price: $195, User's manual only $10
Included with price: Either 8" disk (P/N EB080) or two 2716 EPROMs (P/N EB010) and user's manual.
Where to purchase it:
International Data Systems, Inc.
P.O. Box 17269
Dulles International Airport
Washington, DC 20041

Program Name: Alpha FORTRAN
Hardware System: Alpha Micro (16-bit)
Minimum Memory Size: 32K user memory
Language: Assembler
Description: A multi-user Fortran 77 implementation that has mainframe features. The compiler produces actual assembly language code, not pseudo code, thus allowing Fortran programs to execute many times faster than Basic. Compilations can be stored into a program library, and later linked with assembler or Pascal programs. In addition, Fortran programs are directly callable from Softwork's AlphaAPL language or from Basic. Floating point hardware provides the user with 11 digit accuracy.
Releases: April 1981
Price: $600
Included with price: Language, documentation, sample programs
Where to purchase it:
Softworks Limited
607 W. Wellington
Chicago, IL 60657
(312)327-7666

Program Name: ACCESS/80 - Information Management System
Hardware System: CP/M Operating System
Minimum Memory Size: 54 K+
Language: Assembly
Description: ACCESS/80 is a high-level, non-programmer oriented system for report generation, data entry, file update, reorganization and maintenance, statistical tabulation, and applications development. Its high level funtionality is comparable to the RAMIS system on IBM mainframes. In addition to functioning as a self-contained system, ACCESS/80 will produce reports from any external file stored in ASCII character format, including Basic and Fortran files.
Price: $795
Included with price: diskette containing program and sample applications; User's Manual, 3 copies of Command Reference Card
Author: Friends Software, Inc.
Where to purchase it:
Friends Software
2020 Milvia Street, Suite 400
P.O. Box 527
Berkeley, CA 94701
(415)540-7282

Program Name: Enhanced I/O Drivers
Hardware System: NorthStar MDS or Horizon
Language: 8080 Machine Code
Description: These enhanced I/O drivers for NorthStar DOS (versions 4 & 5), Lifeboat's NorthStar CP/M (versions 1.4 & 2.2), and UCSD Pascal (version 1.5) are field tested. NorthStar DOS can now echo console output to printer, suspend console output until another key is pressed, and reassign console device. I/O drivers are available for serial devices, IMSAI's VIOC, Malibu's 160 printer, and a modem attached to a serial port with all remote I/O echoed to the local console. CP/M users now have a full implementation of I/O byte, allowing user to reassign console, list, and reader-punch to any of four devices such as CRT, printing terminal, high speed printer and modem. Includes ability to use NorthStar computer as intelligent terminal which can send or receive disk files. Special support is provided for IMSAI VIOC and Malibu 160. UCSD Pascal (from NorthStar) can detect which device is being used as console and can detect if IMSAI VIOC is present.
Release: Available now
Price: $50 per driver
Included with price: CP/M disk
Where to purchase it:
Aardvark Computer Solutions
9434 Chesapeake Drive #1210
San Diego, CA 92123
(714)292-8338

Software Directory, cont'd...

Program Name: Z-80 Floppy Disk Test
Hardware System: CP/M 2.0
Minimum Memory Size: 32 kbytes
Language: Z-80 Assembler
Description: An extremely fast, general purpose utility to test or initialize a diskette. When the program is loaded, the operator is asked a series of questions to define the test mode. Selectable options include: lock on read or write, restore original diskette data, fixed or semi-random data patterns, lock on track, lock on sector, error listings on console or printer. The program is supplied to test a standard single density soft sectored diskette, but allows the user to specify the number of tracks or sectors per track for other types of disk drives.
Release: Currently available
Price: $25.00
Included with price: Eight inch soft-sectored single-density diskette, detailed printed instructions.
Where to purchase it:
Laboratory Microsystems
4147 Beethoven Street
Los Angeles, CA 90066

Program Name: BASIC-PACK: Statistics Programs
Hardware System: Run Minimal Basic
Minimum Memory Size: 4-12K, depending on program
Language: Basic
Description: Contains 33 statistical programs written in minimal Basic. The programs are listed and documented in the book *BASIC-PACK: Statistics Programs for Small Computers*. Most of the necessary statistical programs are included for small samples. Programs are available for descriptive statistics, confidence intervals, t-test, chi-square, and two-sample tests. The book contains a description a sample run, and a listing of each program.
Price: Book $16.95
Author: Dennie Van Tassel
Where to purchase it:
Prentice-Hall, Inc.
Englewood Cliffs, NJ 07632

Program Name: STAR*TRAC BASIC Debugger
Hardware System: North Star 5.1 or 5.2 DOS
Minimum Memory Size: 16K
Language: Assembler
Description: Extention to North Star Basic 5.1 offers the first fully interactive debug monitor for *any* microcomputer Basic. Allows user to insert breakpoint in Basic program and assume full keyboard control over subsequent execution. Upon reaching the breakpoint, program control is turned over to STAR*TRAC monitor, which allows execution of any direct mode command. Program variables can be examined or altered before resuming. The Basic program can then be single-stepped, with each program source line and value of selected variables displayed before execution. Single-step feature of STAR*TRAC extends to multiple commands on a source line: each individual command is executed separately. Breakpoint can be relocated anywhere within program, or invoked after a program command has been

Software Directory, cont'd...

executed a specified number of times. Can assert a conditional breakpoint: control is assumed whenever a specified logical expression becomes true. Often a faulty program can only be identified by its results—the portion of the program responsible for the fault cannot be specified. The conditional breakpiont allows control over such a Basic program to be assumed when a specified program symptom occurs, such as when value of a variable is altered.
Release: 1980
Price: $49.00
Included with price: Basic modification; complete documentation is included and full user support is provided.
Author: Allen Ashley
Where to purchase it:
395 Sierra Madre Villa
Pasadena, CA 91107

Program Name: DATABS
Hardware System: CP/M 8"
Minimum Memory Size: 40K
Language: 8080 Object Code
Description: DATABS was inspired by CLU developed at MIT. It is a data abstraction language suitable for control and systems programming. The built-in types are boolean, character, single-byte integer, double-byte integer, and string. Data abstractions allow the implementation of user-defined types using a dynamic storage mechanism. Data abstractions are a step beyond structured programming. Programs created using DATABS are easier to design, understand, and modify. DATABS supports UNIX-style command line arguments and I/O redirection with and . A stream abstraction allows terminal and disk input/output. Disk contains the compiler, built-in type and run-time support library, stream abstraction, and command line processor.
Release: March 1981
Price: $49.50; manual only $10
Included with price: 8" disk and manual
Where to purchase it:
Softronics
36 Homestead Lane
Roosevelt, NJ 08555

Program Name: DOS/65
Hardware System: Tarbell Disk Controller, 6502 CPU
Minimum Memory Size: 16K
Language: Machine Code
Description: Disk operating system with features similar to CP/M. In addition to basic operating system, distribution disk contains a powerful disk file text editor; a disk based, two-pass assembler; a debugger; a system generation routine and a number of other transient utilities. Routines are also included which show how to modify Pittman Tiny Basic and a RAM based version of Microsoft Basic for DOS/65 including SAVE and LOAD of programs. Available with several transient starting addresses ranging from $200 to $2000 for compatibility with AIM, SYM, KIM, TIM, OSI, PET, and Apple memory allocations.
Release: January 1981
Price: $100-$150 depending on options or special modes. Manual only $30.
Included with price: 8" disk and manual
Where to purchase it:
DOS/65
1363 Nathan Hale Dr.
Phoenixville, PA 19460

Program Name: ZAS Z-8000 Development Package
Hardware System: Any 8080/Z80 standard CP/M system
Minimum Memory Size: 48K
Language: 8080 Machine Code
Description: ZAS is an assembly language development tool for Zilog's Z8001 and Z8002 16-bit microprocessors. Includes a relocatable cross-assembler, a linker/task builder, an absolute object file loader, and a Z-8000 run-time module, ZEX, which supports any Z-8000 alternate bus master (such as the Ithaca Intersystems MPU-8000). Using CP/M, ZEX creates an I/O-independent run-time environment for application code written with ZAS. The package provides a fully integrated software development environment for the Z-8000, while retaining full use of current software and hardware facilities under CP/M.
Release: March 1981
Price: $395, $25 for user manual
Included with price: ZAS Assembler, ZLK Task Builder, ZLD Object Loader, ZEX Run-Time Monitor, User Manual. (8" SD CP/M Format Floppy)
Where to purchase it:
Western Wares
P.O. Box 48
Placerville, CO 81430
(303)728-4266

NEW PRODUCTS

16-Bit Intelligent Terminal

A 16-bit, intelligent terminal optimized for word-processing and office automation applications is now available from Piiceon, Inc.

It uses an 8086 microcomputer and CP/M 86 operating system.

The Model 1000 has a 66-by-80 character video display, 64K bytes of RAM and 8K bytes of PROM, and two dual-sided, double-density floppy disk drives with 1.2 megabytes of local storage each.

The detachable keyboard consists of a full alphanumeric set of 107 keys with N-key roll over. The keyboard also includes eight function keys that can be programmed for user convenience.

The terminal has three RS-232C ports, one for communications, one for a printer, and one as an auxiliary interface. Eight transmission rates between 110 to 19.2k baud are selectable in either block or interactive mode.

The OEM quantity 25 price of the Model 1000 is $8,654. Substantial OEM discounts are available for larger volumes. Prices include CP/M 86 operating system and word processing applications software. Workstation hardware can be purchased without software at additional discounts.

Piiceon, Inc., 2350 Bering Drive, San Jose, CA 95131. (408)946-8030.

North Star Introduces New I/O Board

North Star Computers Inc. announces a new four-port serial input/output board. The HSIO-4 Board is S-100 bus compatible, and supports asynchronous and synchronous communications with either RS-232 or current loop options. Each port's baud rate is programmable with eight asynchronous or six synchronous speeds. Each port also has four interrupt sources, three of which are maskable, the fourth being enabled/disabled with an on-board jumper.

The HSIO-4 Board supports North Star's new TSS/A multi-user system, and can be easily reconfigured through header changes to support other applications. Price: $349.

For further information, please contact: Elliot Wassarman, Vice President/Marketing, North Star Computers, Inc., 14440 Catalina Street, San Leandro, CA 94577, (415)357-8500.

Corvus Unveils 5-Megabyte Add-On Winchester Disk Systems

Corvus Systems has announced a family of 5-megabyte Winchester disk systems available to interface to a wide variety of microcomputers—TRS-80 models I and II, Apple II and III, Altos, Alpha Micro, Intertec Superbrain, NEC PC-8001, and Ontel, as well as all S-100 bus-based computers running under CP/M or OASIS; under development are interfaces for the TRS-80 model III, PET, Zenith Z-89, Atari, and HP-85 machines.

A system package consists of the drive (same size as a 5 1/4-inch floppy), and intelligent Z80-based controller card, an intelligent interface card with firmware, software appropriate to the given model of microcomputer and power supply.

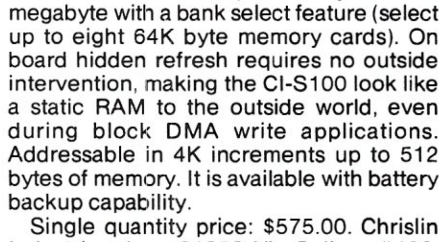
Performance specifications include an unformatted data capacity of 6.9 Mbytes (5.8 Mbytes formatted); a minimum seek time of 10 milliseconds; and average seek and latency times of 50 and 8.3 milliseconds, respectively. Power consumption is 120 W.

Further, the drives are fully compatible with Corvus' Mirror™ and Constellation™. The Mirror provides Winchester backup at a 1-Mbyte/minute rate via a standard video cassette recorder and 120-Mbyte capacity cassettes. The Constellation is a backend local network—a host multiplexer that allows up to 64 microcomputers to communicate with each other, share peripherals, and share a common Corvus disk drive.

Price is $3,750; quantity discounts are available. Corvus Systems, 2029 O'Toole Avenue, San Jose, CA 95131. (408)946-7700.

64K Byte Memory For S-100 Microcomputers

Chrislin Industries' new CI-S100 dynamic RAM memory module requires no wait states at 2 or 4 MHZ and is compatible with most S-100 bus microcomputers.



Features include expandability to a half megabyte with a bank select feature (select up to eight 64K byte memory cards). On board hidden refresh requires no outside intervention, making the CI-S100 look like a static RAM to the outside world, even during block DMA write applications. Addressable in 4K increments up to 512 bytes of memory. It is available with battery backup capability.

Single quantity price: $575.00. Chrislin Industries, Inc., 31352 Via Colinas #102, Westlake Village, CA 91361, Phone (213) 991-2254.

S-100 Color Video Processor/Programmable Sound Generator

The Color Video Processor and Programmable Sound Generator allows an S-100 bus computer to display text, graphics and animation along with sound effects or music on a color television set. The board includes 16K bytes of on-card I/O mapped video memory for storing multiple patterns. Two

New Products, cont'd...

programmable I/O ports can be used to interface to keyboards, joysticks or other external devices.

The Texas Instruments TMS9918A Video Display Processor (VDP) is used to provide a composite video signal which can directly drive a color monitor or a color television via available R.F. modulator. The VDP chip has four modes of operation: Graphics I Mode (256 x 192 dots), Graphics II Mode (Extended 256 x 192 dots), Text Mode (40 char. x 24 lines of user defined characters), and a Multicolor Mode (64 x 48 positions). Sixteen possible colors including black and transparent which can be used in various combinations in each of the above modes.

Internal counter chain in the 9918A provides a real time interrupt source of approximately 1/60th of a second rate. This signal is jumperable to any of the bus vector interrupt inputs. Documentation includes programming examples and test routines. $475.00 (assembled and tested) or $375.00 (kit). Contact Electronic Design Associates, P.O. Box 94055, Houston, TX 77018, phone (713)999-2255.

Tarbell CP/M Database System

Tarbell Electronics has developed a new Database System using CP/M. This system features variable length fields with field names that may be of any length and may include spaces. It runs under CBasic. Other features include sequential or random files and an optional index file. It also includes Interactive Programs such as: DBSETUP which creates a file, DBENTRY for entering data, DBUPDATE for changing files, DBQUERY for accessing data, DBLABEL which prints mailing labels and DBLETTER for printing irate letters. Non-Database Programs include: INV for inventory control and FLIGHT for cross-country flight planning.

The $50 price includes sources on disk. For further information, contact Don Tarbell, Tarbell Electronics, 950 Dovlen Place, Suite B, Carson, CA 90746. Or call (213)538-4251.

New Products, cont'd...

New EXPANDER Desktop Computer Introduced

Micro-Expander, Inc. has announced their new entry into the professsional microcomputer market. Called the EXPANDER, the S-100 computer requires only a video display and media storage for operation.

Lee Felsenstein, designer of the EXPANDER, is well known for his design of the SOL computer. The computer is built around a single board that contains a Z-80A CPU, keyboard circuitry, interrupt, video circuitry, real time clock, parallel printer interface, RS-232 serial interface, and full color circuitry. The unit also includes a 4-slot S-100 motherboard.

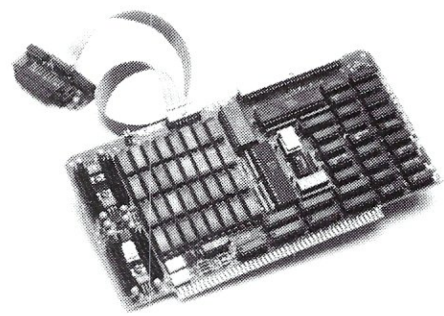
Features include standard 80 x 24 screen format, upper/lower case, 4K ROM monitor, 64K RAM expandable to 512K, video output and color graphics using 256 colors, and a complex tone generator with internal speaker. Keyboard capabilities include calculator keypad, two programmable function keys, and four cursor control keys.

The EXPANDER is sold complete with 24K Microsoft BASIC-80 (disk version) and 10K Microsoft BASIC-80 (cassette tape version). Included is *Instant Basic* by Gerald Brown, a beginner's manual.

The EXPANDER is available through dealers in the U.S. for under $2,200. A European version, called PAL, will also be available. For more information, contact Mats Ingemanson, president, Micro-Expander Inc., 7835 W. Higgins Ave., Chicago, IL 60656. Telephone (312)792-1196.

Single Board Computer Provides Multi-Processing Capability On The S-100 Bus

Net/80™, a single board microcomputer which operates as a slave processor for data processing networks, is now available from MuSYS Corporation. The device is ideal for use with CP/NET™. NET/80 performs as a Z-80 slave processor loosely coupled to an S-100 bus. Each board comes complete with 64K of RAM, a single level interrupt, a console serial port and a parallel port for communication with the S-100 bus bus master CPU. Each NET/80 slave operates independently of any others, except for resource queuing in the master. Thus, the entire system appears to be dedicated to each user, unless a large amount of shared resources are being accessed. In addition, NET/80 totally isolates the master CPU from errors in the slave processors.



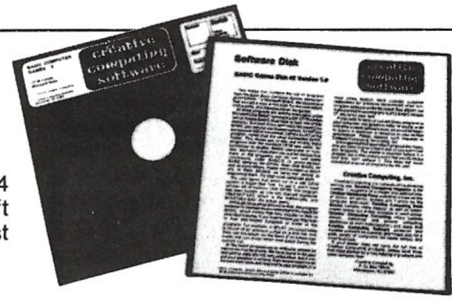
The master processor has complete control over each slave, and can reset or interrupt a slave at any time. Transfer protocol can be performed with Z-80 block I/O instructions at near DMA speeds, while retaining protection and validation capability for the master. A bootstrap PROM supplied with each slave uses this transfer technique to download the system software into RAM. The PROM is then switched out of the address space so that the entire 64K is available as RAM.

NET/80 permits the customization of each serial port for various applications. Currently, a board configured for RS-232 with the slave appearing as a null modem allows direct connection to most common CRT terminals. Many other configurations are possible, including actual modem operation and RS-449.

A unique expansion bus on each slave gives users with unusual I/O requirements the ability to access additional peripherals. The first board designed for this bus will add a second serial port, Centronics printer or 8-bit bi-directional parallel port, priority interrupt control, real time clock, and the capability to act as the IEEE S-100 permanent bus master. The system is compatible with most CP/M software. Digital Research, the author of CP/M, offers CP/NET and its MP/M operating system for the network master, while Action Computer Enterprises offers DPCOS , an operating system for the master, which runs under CP/M.

Price: $1,395.00; complete software also available. For more information, contact Mr. Bill Schultz, MuSYS Corporation, 1451 E. Irvine Blvd., Suite 11, Tustin, CA 92680. (714)730-5692. TWX:910-595-1967. Cable: MUSYSTSTN.

CP/M



All disks expect CS-9004 require 48K and Microsoft Basic. All 8" CP/M disks cost $24.95.

Basic Games-1, CS-9001

Includes the Following:

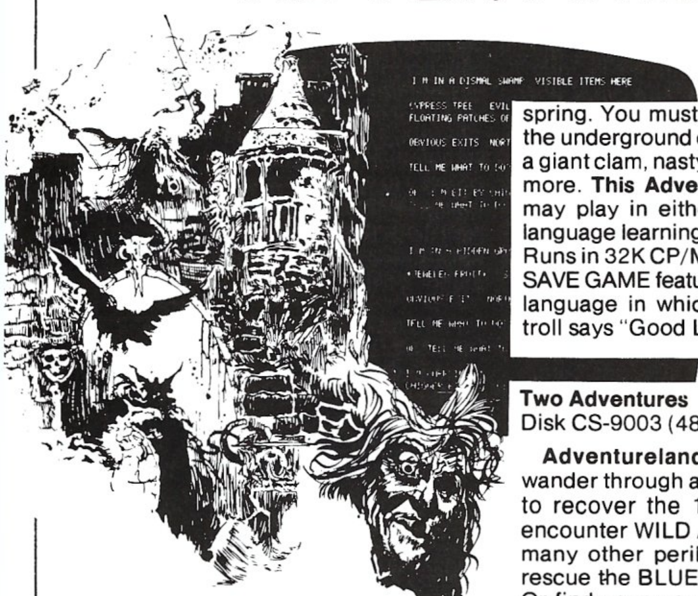
Acey Ducey, Amazing, Animal, Awari, Bagels, Banner, Basketball, Batnum, Battle, Blackjack, Bombardment, Bombs Away, Bounce, Bowling, Boxing, Bug, Bullfight, Bullseye, Bunny, Buzzword, Calendar, Change, Checkers, Chemist, Chief, Chomp, Civilwar, Combat, Craps, Cube, Depth Charge, Diamond, Dice, Digits, Even Wins I, Even Wins II, Flip Flop, Football 1, Football II, Fur Trader, Golf, Gomoko, Guess, Gunner, Hammurabi, Hangman, Hello, Hexapawn, Hi-Lo, High I-Q, Hockey

Basic Games-2, CS-9002

Includes the Following:

Horserace, Hurkle, Kinema, King, Letter, Life, Life For Two, Literature Quiz, Love, Lunar LEM Rocket, Madlib, Mastermind, Math Dice, Mugwump, Name, Nicomachus, Nim, Number, One Check, Orbit, Pizza, Poetry, Poker, Qubic, Queen, Reverse, Rocket, Rock, Scissors, Paper, Roulette, Russian Roulette, Salvo, Sine Wave, Slalom, Slots, Splat, Stars, Stock Market, Super Star Trek, Synonym, Target, Trek Instructions, 3-D Plot, 3-D Tic Tac Toe, Tic Tac Toe, Towers, Train, Trap, 23 Matches, War, Weekday, Word

ADVENTURE



Original Adventure
Disk CS-9004 (48K) $24.95

(by Crowther, Woods, Manning and Roichel)-Somewhere nearby is a colossal cave where others have found fortunes in treasures and gold, but some who have entered have never been seen again. You start at a small brick building which is the wellhouse for a large spring. You must try to find your way into the underground caverns where you'll meet a giant clam, nasty little dwarves, and much more. **This Adventure is Bi-Lingual**—you may play in either **English or French**—a language learning tool beyond comparison. Runs in 32K CP/M system (48K required to SAVE GAME feature). Even includes SAM76 language in which to run the game. The troll says "Good Luck."

Two Adventures
Disk CS-9003 (48K) $24.95

Adventureland (by Scott Adams)-You wander through an enchanted world trying to recover the 13 lost treasures. You'll encounter WILD ANIMALS, BEINGS, and many other perils and puzzles. Can you rescue the BLUE OX from the quicksand? Or find your way out of the maze of pits? Happy Adventuring....

Pirate Adventure (by Scott Adams)-"Yo Ho Ho and a bottle of rum..." You'll meet up with the pirate and his daffy bird along with many strange sights as you attempt to go from your London flat to Treasure Island. Can you recover LONG JOHN SILVER's lost treasures? Happy sailing matey....

Basic Games-3, CS-9005

Includes the Following:

Artillery-3, Baccarat, Bible Quiz, Big 6, Binary, Blackbox, Bobstones, Bocce, Boga II, Bombrun, Bridge-it, Camel, Chase, Chuck-A-Luck, Close Encounters, Column, Concentration, Condot, Convoy, Corral, Countdown, Cup, Dealer's Choice, Deepspace, Defuse, Dodgem, Doors, Drag Race, Dr. Z, Eliza, Father, Flip, Geowar, Grand Prix, Guess-It, ICBM, Ink Blot, Joust, Jumping Balls, Keno, L Game, Life Expectancy, Lissajous, Magic Square, Man-Eating Rabbit, Maneuvers, Maze, Millionaire, Motorcycle Jump, Lewis

Basic Games-4, CS-9006

Includes the Following:

Mastermind, Masterbagels, Matpuzzle, Minotaur, Nomad, Not One, Obstacle, Octrix, Pasart I, Pasart II, Patterns, Pinball, Rabbit Chase, Roadrace, Rotate, Safe, Scales, Schmoo, Seabattle, Seawar, Shoot, Smash, Strike 9, Tennis, Tickertape, TV Plot, Twonky, Two-to-Ten, UFO, Under & Over, U.S. Population, Van Gam, Warfish, Word Search Puzzle, Wumpus I, Wumpus II, Yahtzee, 4-In-A-Row

Special Packages

Special package: BASIC Computer Games Book and Disks 1 and 2
CS-9000 $50.00

Special package: More BASIC Computer Games Book and Disks 3 and 4
CS-9007 $50.00

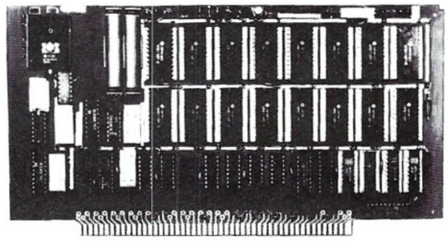
BASIC Computer Games Book, More BASIC Games Book and All four disks
CS-9008 $95.00

creative computing
P.O. Box 789-M
Morristown, NJ 07960
Toll-free **800-631-8112**
(In NJ 201-540-0445)

New Products, cont'd...

Non-Volatile Memory Modules For S-100 Bus

Non-volatile memory boards for S-100 systems are now announced by Dual Systems Control Corporation. The new boards feature high speed CMOS RAM IC's, on-board batteries, and proprietary write-protection circuitry. The result is a degree of data security approaching an EPROM board with the fast access and convenience of high speed RAM.



A software programmable "write-protect window" allows parts of the program, or selected data, to be changed without any risk of accidentally writing over protected data. For further data security, the boards generate an interrupt when a power drop is detected, enabling the system to store critical data quickly before the main power supply fails. When power is restored, the computer can resume operation as if no power failure had occurred.

Access time is 250 nanoseconds. Other features include 8 or 16-bit data transfers, bank select option, and extended memory addressing through 24-bit address lines. The batteries are guaranteed to keep programs and data intact for one year.

Prices: $1,095 for CMEM-32K with 32K bytes of memory, $895 for CMEM-16K and $695 for CMEM-8K. Dual Systems Control Corporation, 1825 Eastshore Highway, Berkeley, CA 94710; (415)549-3854 or (415)549-3890.

BIZCOMP Introduces VersaModem

BIZCOMP Corporation is introducing the Model 1084 Intelligent VersaModem™, compatible with the Bell Standard 103

protocol. It uses a patent-pending combination of automatic calling unit (ACU), custom BIZ-080 microcomputer and data modem to enable full automatic dialing and auto-answer capability controlled through a simple RS232 interface. VersaModem's unique Code-Multiplexed Design allows dialing functions to be easily implemented in high level languages such as Basic or Cobol. The unit itself has a simple command language much like the monitor commands of a minicomputer or microcomputer. Interfacing to RS232-equipped computers, word processors and programmable data equip-

Microsystems – the CP/M* and S-100 User's Journal

CP/M is the software bus!+ S-100 is the hardware bus for sophisticated microcomputer users!

If you are a CP/M user, on any system—S-100, Apple, TRS-80, Heath, Ohio Scientific, Onyx, Durango, Intel MDS, Mostek MDX, etc—after all CP/M is the Disk Operating System that has been implemented on more computer systems than any other DOS—then *Microsystems* magazine is the "only" magazine published specifically for you!

Or, if you use an S-100/IEEE-696 based computer—and the most sophisticated microcomputer systems available use the S-100/IEEE-696 hardware bus—then *Microsystems* magazine is the "only" magazine published specifically for you!

We started publishing *Microsystems* almost two years ago to fill the void in the microcomputer field. There were magazines catering exclusively to the TRS-80, Apple, Pet, Heath, etc. system users. There were also broad based publications that cover the entire field but no one system in depth. But no magazine existed for CP/M users—nor did one exist for S-100 users.

The why and what of a software bus

First of all what is a "bus?" And why do we call CP/M "the software bus?"

A "bus" is a technique used to interface many different modules. Examples are the "S-100/IEEE-696 Bus" and the "IEEE-488 Bus." These are hardware buses that permit a user to plug a bus-compatible device into the bus without having to make any other hardware modifications and expect the device to operate with little or no monification.

CP/M is a Disk Operating System (DOS). It was first introduced in 1974 and is now the oldest and most mature DOS for microcomputer systems. CP/M has now been implemented on over 250 different computer systems. It has been implemented on hard disk systems as well as floppy disk systems. It is supported by two user groups (CP/M-UG and SIG/M-UG) that have released over sixty volumes containing over 1,600 public domain programs that can be loaded and run on systems using the CP/M DOS. Add to this another 1,500 commercially available CP/M software packages and you have the largest applications software base in existence.

CP/M is the only DOS for micros that has stood the test of time (seven years) with the highest level of compatibility from version to version. And over the years this compatibility has been maintained as new features have been added.

This is why we say "CP/M is the software bus" and why *Microsystems* magazine is vital to providing CP/M users with technical information on using CP/M, interfacing to CP/M, new CP/M compatible products and for CP/M users to exchange ideas.

Why support the S-100 bus?

S-100 is currently the most widely used microcomputer hardware bus. It offers advantages not available with any other microcomputer system. Here are a few of the advantages:

S-100 is processor independent. There are already thirty different S-100 CPU cards that can be plugged into an S-100 bus computer. Nine 8-bit microprocessors are available: 6502, 6800, 6802, 6809, 2650, F8, 8080, 8085 and Z80. Eight 16-bit microprocessors are available: 8086, 8088, 9900, Z8000, 68000, Pascal Microengine, Alpha Micro (similar to LSI-11) and even the AMD2901 bit slice processor. Take your pick from the incredible offerings.

S-100 has the greatest microcomputer power. What other microcomputer system has direct addressing of up to 16 megabytes of memory, up to 65,536 I/O ports, up to 10 vectored interrupts, up to 16 masters on the bus (with priority) and up to 10 Mhz data transfer rate? You will have to go a long way to use up that computing power.

S-100 is standardized. The S-100 bus has been standardized by the IEEE (Institute of Electrical and Electronic Engineers) assuring the highest degree of compatibility among plug-in boards from different manufacturers. And, *Microsystems* has published the complete IEEE S-100/696 standard (all 26 pages).

S-100 has the greatest hardware support. There are now over sixty different manufacturers of about 400 different plug-in S-100 boards. Far greater than any other microcomputer system.

With all these advantages is it any wonder that S-100 systems are so popular with microcomputer users who want to do more than just play games?

For the serious computer user.

Each issue of *Microsystems* brings you the latest in the CP/M and S-100 world. Articles on applications, tutorials, software development, product reviews, and lots more, to keep you on top of the ever changing microcomputer scene.

And if you are an S-100 system user using other operating systems (e.g. North Star) *Microsystems* also supports you.

Get your copy today

Order your susbscription to *Microsystems*. Send $10 for one year (6 issues), $18 for two years (12 issues) or $24 for three years (18 issues). If you prefer, call our toll-free number, **800-631-8112** (in NJ 201-540-0445) to put your subscription on your MasterCard, Visa or American Express card. Canadian/Mexican and other foreign surface subscriptions are $15 and $25, respectively, per year and must be prepaid. We guarantee that you will be completely satisfied or we will refund your subscription.

Join thousands of *Microsystems* subscribers like Jim Johnstone of Los Altos CA, who said "*Microsystems* has lived up to your promises and my expectations. Congratulations."

the CP/M* and S-100 user's journal
MICROSYSTEMS™
a Creative Computing publication

39 East Hanover Avenue
Morris Plains, NJ 07950, USA
Toll-free 800-631-8112
(In NJ, 201-540-0445)

* registered trademark Digital Research + registered trademark Lifeboat Associates

New Products, cont'd...

ment requires only a 3-wire data cable. The data rate is preset to 300 baud but may be user-optioned to autobaud on rates of 110, 134.5, 150, 200 or 300 baud.

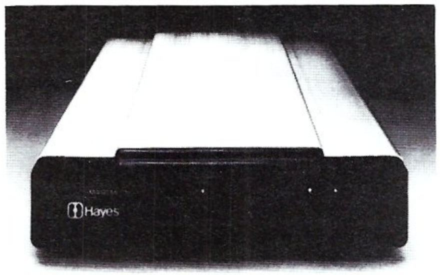
Price for the VersaModem is $299. OEM quantity discounts are available. Availability is stock to four weeks. Inquiries Manager, BIZCOMP Corporation, P.O. Box 7498, Menlo Park, CA 94025. Tel: (415)966-1545.

'Smartmodem' Data Communications System

Hayes Microcomputer Products Inc. announces the Smartmodem, designed for use with RS-232C compatible computers or terminals. The Smartmodem can be program controlled in any language by ASCII character strings. It has auto dial and auto answer capabilities and can be connected directly to the phone line.

An audio monitor permits the user to follow the progress of the call and be alerted to wrong numbers and busy signals. If a busy signal is encountered, by entering a repeat command, the Smartmodem will automatically redial the number at any time.

In addition, the unique "Set" commands allow the user to select (and change) various operational parameters such as dialing speed, escape code character and number



of rings to answer on. Price: $279.00. Hayes Microcomputer Products, Inc., 5835 Peachtree Corners East, Norcross, GA 30092. (404)449-8791.

SSM Introduces New S-100 EPROM Board

SSM Microcomputer Products has introduced the MB8A 1K-16K EPROM Board, which provides sockets to support up to sixteen 2708 EPROM's. By removing EPROM's, the board can be disabled in 1K increments. For example, with 8 EPROM's the board will act like and have the capacity of an 8K board. In addition, users can easily add or subtract memory as necessary. The user can overlay RAM and ROM at the same address in any desired increment. This provides increased flexibility when the board is used with RAM boards equipped with Phantom Disable.

SSM Microcomputer Products, 2190 Paragon Drive, San Jose, CA 95131, (408) 946-7400.

Software Vendor Directory

Micro-Serve Inc. has published the fourth edition of the *Software Ventor Directory*—a directory of microcomputer software companies. This newest edition contains the following features: 1001 software vendors, 4195 products, indexed by 80 hardware categories, and 200 software categories.

The price is $100 for the Directory and two updates (which are future new printings at 6-month intervals). The Directory alone is $57.95, and one update to that Directory is $25. A disk version is also available (under CP/M) at $78, which includes a product named "Information Master" from Island Cybernetics of Port Aransas, Texas. The *Software Vendor Directory* is available from Micro-Serve Inc., at 250 Cedar Hill Avenue, Nyack, New York 10960, telephone (914) 358-1340.

Super Isolator

Electronic Specialists' recently announced Model ISO-11 is designed to curb electrical problems. It features two individually dual-Pi filtered AC socket banks (6 sockets total). Heavy-duty spike/surge suppression is incorporated in the design. Equipment interactions are eliminated and disruptive/damaging line spikes and hash are controlled. The Model ISO-11 Super Isolator controls power line Spikes and Hash while providing interaction free microprocessor operation. Price: $94.95.

Electronic Specialists Inc., 171 South Main Street, Natick, MA 01760, Phone: (617)655-1532.

Advertiser Index

advertiser **page**

ABM Products....48
Adaptive Data & Energy....cover 3
Alpha Data Services....14
American Square Computers....8
Applied Software Technology....15
California Digital Engineering....35
Casheab....12
Computer Design Labs....49
Computer Innovations....38
Cornerstone Software....75
Creative Computing....69, 77
Data Compass....37
Digiac Corporation....33
Digital Graphic Systems....57
Digital Marketing....5
Digital Research....31
Discount Software Group....70
Dual Systems Control Corp....17
Ecosoft....25
Electronic Control Technology....13
Electronic Specialists....29
Electrovalue Industrial....57
Far East Electronics Tour....71
Godbout Electronics....cover 4
Hawkeye Grafix....72
Hayes Microcomputer Products....56
Infosoft Systems....78
Integrand....29
Interactive Microware....73
JES Graphics....55
Laboratory Computer Systems....63
Laboratory Microsystems....72
Lomas Data Products....21
Magnolia Microsystems....55
MicroDaSys....1
Microsystems....79
Micro Tech Exports....78
Midwest Micro Warehouse....8
Morrow Designs....cover 2
MuSys Corporation....7
National Computer Show....39
Piiceon....27
Potomac Micro-Magic....76
S-100, Inc....57
Seattle Computer Products....23
Snow Micro Systems....16
Software Review....73
SSM Microcomputer Products....2
Static Memory Systems....11
SZ Software....75, 78
TecMar....19
The Rosetta Store....57
Theta Labs....63
Timins Engineering....63
Used Computer Exchange....73
WW Component Supply....9

TURN MULTI-USER DREAMS INTO PRACTICAL REALITY...

...TIES IT ALL TOGETHER

Interfacer 3, an 8 port serial board, is the answer to the serial interfacing problems inherent in multi-user systems.

- All 8 channels are true RS-232 with full handshake.
- 6 asynchronous DCE interface channels hook up directly to terminals and printers (no wiring changes required).
- 2 asynchronous or synchronous interface channels (DCE or DTE) are optimized for connection to modems, printers, CRTs, and similar devices.
- Boards may be cascaded for up to 32 channels.
- Includes software programmable Baud rates and many other convenience features.

Like our best-selling **Interfacer 1** and **Interfacer 2** boards, **Interfacer 3** meets the most demanding electrical and mechanical specifications, and is built to the same stringent standards that have established our position of technical leadership in the S-100 field.

Starting in June, multi-user systems will be able to enjoy **CompuPro** quality interfacing in an extremely convenient format. **Interfacer 3** is available at finer computer stores world-wide, or order directly from us.

CompuPro™ division of GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614 (415) 562-0636

MEMBER MITA