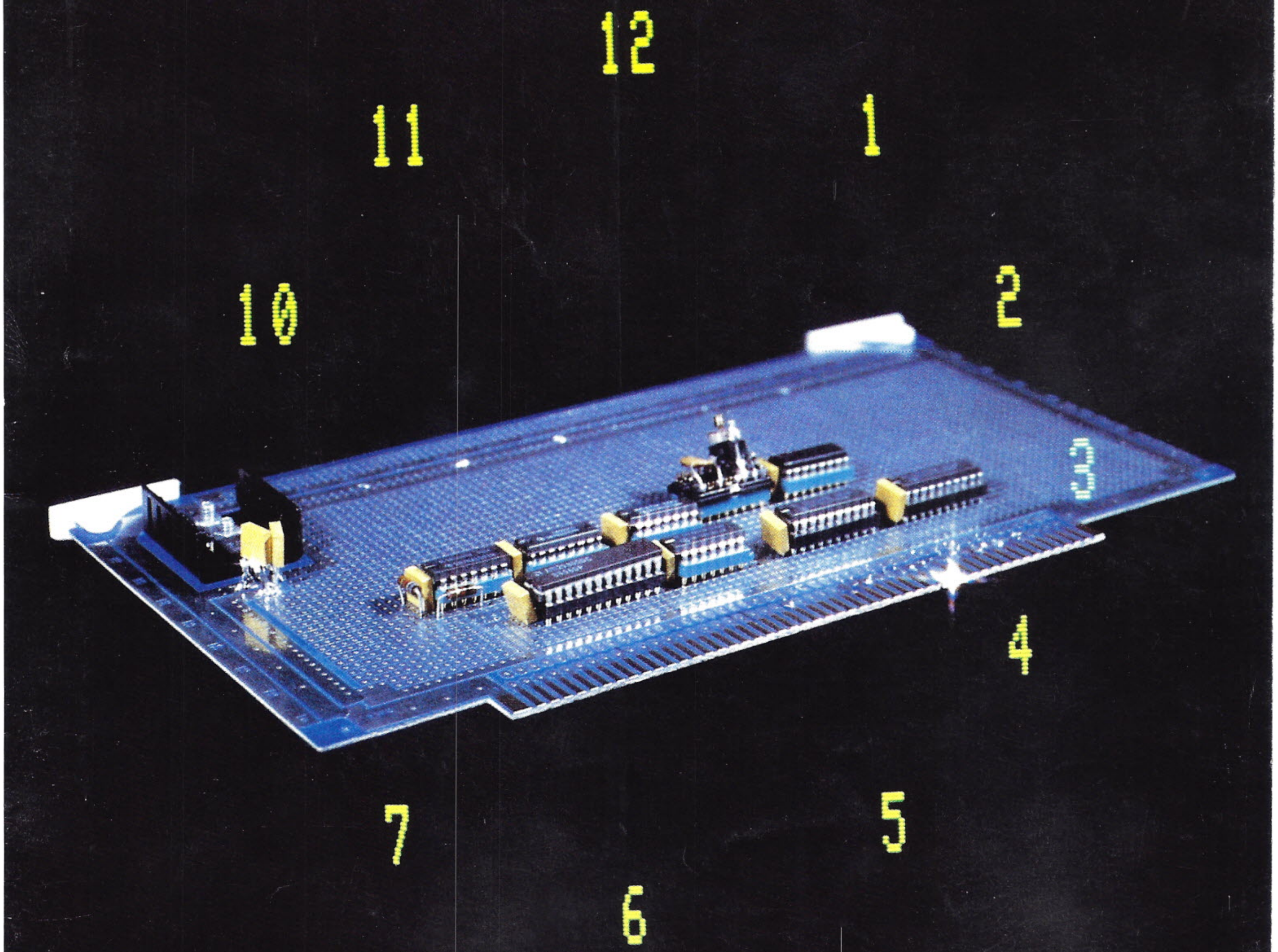
$4.90

VOL. 1, NO. 3 • SPRING 1986

S-100 journal



16-BIT SINGLE BOARD COMPUTERS TO GO

INTERCONTINENTAL MICRO is shipping solutions today for all your S-100 BUS 16-bit and PC network needs. Of course, we also offer a complete line of 8-bit and interface/controller products as well as the sophisticated TurboDOS™ multi-user operating system.

SYSTEM FLEXIBILITY. With 8 MHz 16-bit and 4 or 6 MHz 8-bit Single Board Computer (SBC)/Masters; 10 MHz 80186 and 8 MHz 8086 16-bit Slaves; ARCnet™ links between systems; and a complete line of memory, interface and controller boards, INTERCONTINENTAL MICRO SYSTEMS gives you and your customers flexibility—to grow and change.

TurboDOS™ allows you to construct true multi-user systems with CP/M,™ PC-DOS,™ MP/M,™ MS-DOS,™ and CPM-86™ compatibility. ARCnet,™ made possible with our exclusive TurboLAN™ software, make local area networks with up to 4000 users possible—and very cost effective.

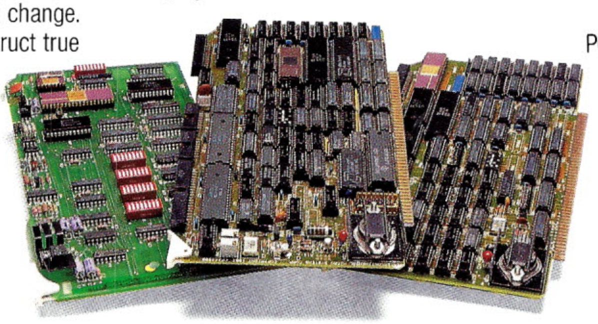
SPEED AND POWER. Features are great—But they don't mean much unless they help you do your job better and faster.

CPZ-186—

8 MHz 80186 SBC Master Processor, 2 sync or async serial I/O channels, 20 parallel I/O lines, 256K RAM expandable to 1 megabyte, onboard floppy disk controller.

CPS-186—

10 MHz 80186 SBC Slave Processor, 4 sync or async serial I/O channels, 20 parallel I/O lines, 256K RAM expandable to 1 megabyte.



CPS-16X—

8 MHz 8086 SBC Slave Processor, 256K RAM expandable to 1 megabyte, 2 sync or async serial I/O channels, 20 parallel I/O lines.

LANPC—

Allows IBM PC's,™ XT's,™ AT's,™ or compatibles to integrate into TurboLAN™ ARCnet,™ and S-100 bus networks

LANS100—

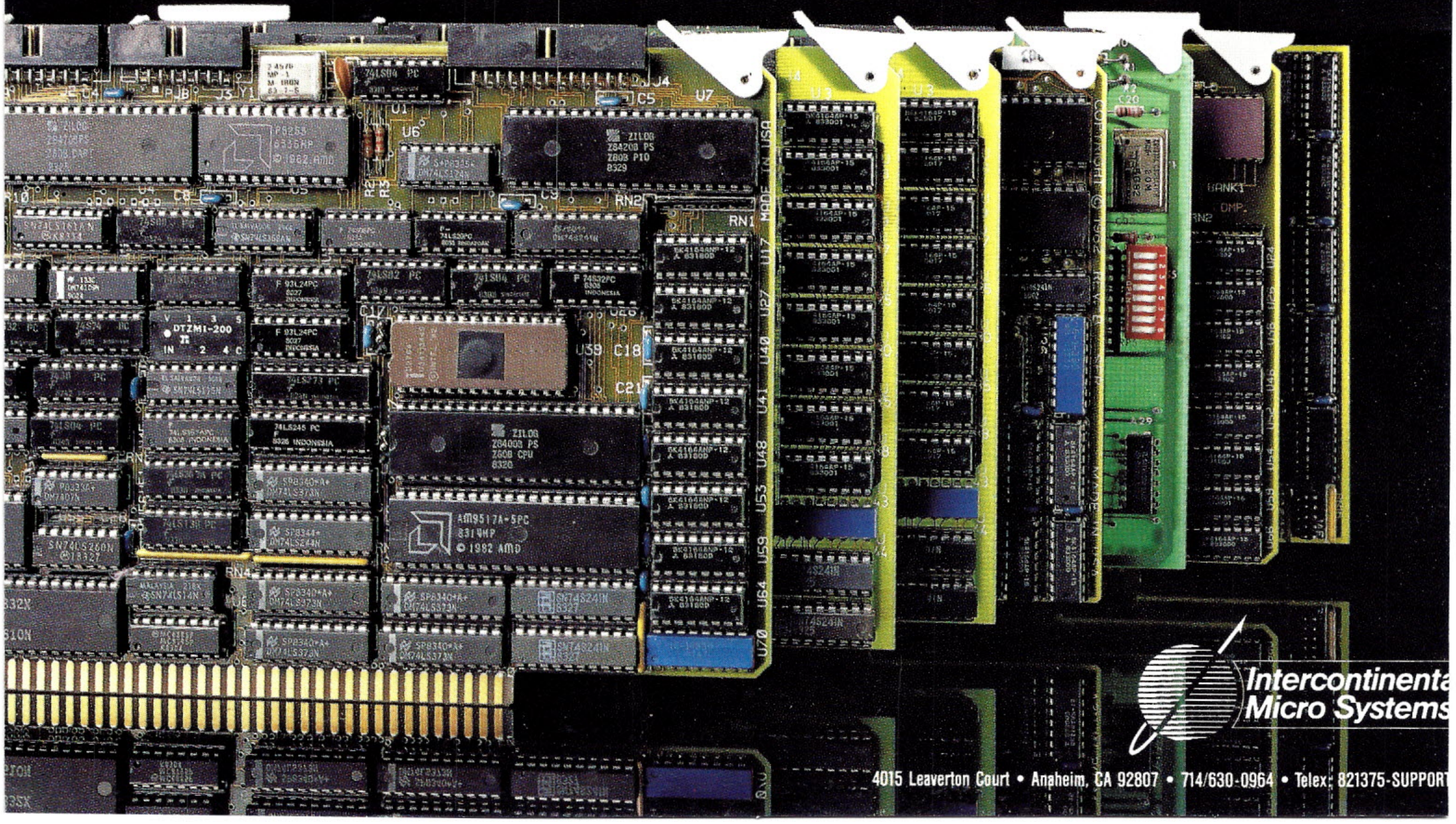
Permits networking of S-100 Bus systems to other S-100 Bus systems, Zenith Z-100,™ and IBM PC™ or PC compatible machines.

LANjr™—

Provides cost effective true multi-user PCjr™ networks with simple installation.

GET IT ALL. A comprehensive product line loaded with benefits for you and your customers. Flexibility to grow. And support that sets industry standards. Call INTERCONTINENTAL MICRO SYSTEMS today—we can help.

TurboDOS is a trademark of Software 2000, ARCnet is a trademark of the Datapoint Corporation, CP/M, CP/M-86 and MP-M are trademarks of Digital Research Corporation, MS-DOS is a trademark of MicroSoft, IBM PC, XT, AT, PCjr and PC-DOS are trademarks of International Business Machines, Zenith Z-100 is a trademark of the Zenith Corporation, TurboLAN is a trademark of Intercontinental Micro Systems.



Intercontinenta
Micro Systems

4015 Leaverton Court • Anaheim, CA 92807 • 714/630-0964 • Telex: 821375-SUPPORT

S-100 journal

VOL. 1 NO. 3 SPRING 1986

Editor
Jay Vilhena

Associate Editor
Robert Petersen

Business Manager
Linda Pereira

Advertising
Jay Vilhena

Technical Consultant
Steve Renshaw

Art Director
Joslyn L. Johanna

S-100 Journal is published quarterly by Octoplus Corporation, 2426 Wade Avenue, Raleigh, NC 27607. Telephone (919)839-0115.

Subscriptions: U.S.A.: $14.00 for 1 year, $25.00 for 2 years. Canada, England, and Mexico: add $3.00 per year. Other countries: add $6.00 per year. Foreign payments must be in U.S. currency. Send orders and payments to S-100 Journal, P.O. Box 12881, Raleigh, NC 27605.
Advertising: Contact S-100 Journal Advertising, P.O. Box 12881, Raleigh, NC 27605.
Postmaster: Send address changes to S-100 Journal, P.O. Box 12881, Raleigh, NC 27605.

outline

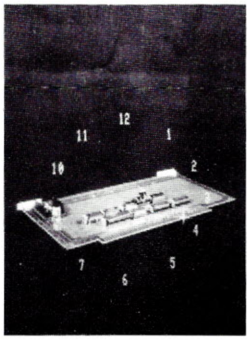
12 S-100 @ COMDEX

18 A REAL-TIME CLOCK FOR YOUR S-100

29 INSTALLING A RAM-DISK IN A CP/M 2.2 SYSTEM

52 MACROTECH'S V-RAM

3 EDITORIAL
4 EDITOR INTERFACE
8 696 BUS
24 MULTIUSER OS
42 HOMEBREWING
63 BITS
64 S-100 DIRECTORY



OUR COVER

A Real-Time S-100 clock board can be built with very few parts and little hardware-building experience. See page 18.

WHY THE S-100 BUS? A Tour for the Uninitiate

'Why have you started a magazine for a computer system that no one has ever heard of?' is a question that you might ask if S-100 Journal is your first contact with the world of S-100 computers. Or, you might think that S-100 systems are some sort of obsolete machines that used to be around in the late 70's.

Well, S-100 computers are neither unheard-of nor obsolete. But they did use to be around in the 70's. In fact, they have existed for 11 years. Today, there are hundreds of thousands of installed S-100 systems. Although individual components of an S-100 do become obsolete, the system itself can always be kept up with the latest technology.

You see, S-100 systems are modular. First, they are modular on the **outside**. There is a unit that is the computer itself, a unit that is the terminal, a unit that is the disk drive or the hard disk, etc. You can select (or have your dealer select) components from different manufacturers as your needs require, and you can replace each unit individually as your needs change and better technology becomes available. (Exactly like fine-quality stereo systems!)

Second, and most important, S-100 computers are modular **inside**. They consist of a long computer board, called the motherboard, that has up to 20 slots on it. Into these slots plug the S-100 boards that you see all over this magazine. This is what's known as a bus system, in this case the S-100 bus. There are other bus systems. However, in many other bus systems, like the IBM-PC bus, the motherboard contains the main logic chips for the system. Since all other boards and the basic computer operation are tied to those logic chips, when they become obsolete, the whole computer is obsolete.

In an S-100 system, however, the motherboard has no chips, so it never becomes obsolete. The whole computer logic resides on the S-100 boards.

The S-100 boards, each with a different function, are the key to the system's longevity. When a board becomes obsolete, you simply replace that board. Most of your investment remains in place. S-100 manufacturers are always trying to outdo each other by being the first to implement a new chip or other improvements. Since all they have to design is a board, not a whole computer system like many nonS-100 vendors, the new technology gets implemented very quickly. The result: the latest technology is always available on an S-100 board. 10-MHz S-100 systems, for example, have been in use for years.

There's more. Because S-100 boards are fairly independent from each other, manufacturers have designed boards to do just about any function they can think of. There are basic boards in an S-100 system, like CPU boards, memory boards, or disk-drive controller boards, but there are also boards to control tape drives, boards to control computer networks, clock boards, boards to program EPROMs, modem boards, boards for data collection, boards to control lights and motors, boards to control instruments, boards to produce speech, boards to interface to other computers, boards to produce color graphics, boards to emulate disk drives, and even boards to emulate other computers.

There is an S-100 CPU board for practically every major microprocessor chip. Here is a partial list: Intel 8080, 8085, 8086, 8087, 8088, 80186, 80286, Zilog Z80, Z80A, Z80B, Z80H, Z8000, Motorola 6800, 68000, 68008, 68010, NS16032, NEC 70108, Hitachi HD64180. Talk about a choice! To take advantage of this plethora, over 20 different operating systems have been configured for S-100 computers. Your S-100 system can become exactly what you want it to be.

One of the best features of the S-100 bus is that different CPU boards can be used together in the same computer. There is one master CPU board, and other CPU boards are implemented as slave boards or as coprocessor boards. This allows a single S-100 computer to behave like several, running several operating systems, and having access to a huge variety of software.

You may think that this all sounds like a dream. And it is. But it's a reality of a dream. Because of their tremendous versatility and resistance to obsolescence, S-100 computers are used as advanced single-user systems, powerful multiuser supermicros, host systems in sophisticated networks, and controllers of real-world processes. One can start with an inexpensive single-user S-100 and later upgrade to more advanced features or to multiuser, as the needs increase.

Because there are so many boards, and so many different things to do with an S-100 system, there is the need to communicate different applications, share information on how to interface boards with each other, present software that takes advantage of hardware features, and describe new products. This is a reason why S-100 Journal exists. We try to present a balance of all applications, sometimes describing multiuser systems and networks, other times solving single-user problems, other times teaching how to assemble systems or even how to build your own boards. Depending on your understanding of computers, you may find some articles difficult to follow, but there will always be articles that are easier to understand. As you continue reading S-100 Journal and learning about S-100 systems, the S-100 jargon will become more clear, and soon you will be able to look back on older issues and find the information even more useful. By then, you will have become an S-100 enthusiast.

At S-100 Journal, we are very fond of S-100 systems. I think that after reading this editorial you understand why. S-100 computers really are THE BEST SYSTEM OF THEM ALL.

Jay Vilhena

editor interface

Again I want to thank those who have written giving us encouragement and expressing support for the Journal. We are unable to publish or reply to all the letters, but we do consider everyone's opinion when selecting future articles. If you have a comment, question, or suggestion, please write to Editor Interface, S-100 Journal, P.O. Box 12881, Raleigh, NC 27605. Please print or type your letters.

It has been next to impossible to separate subjects between the Reader I/O and the Editor Interface columns. The mail has arrived addressed to one or the other, but the contents overlap considerably. We realize now that it was probably a mistake to try to separate the two, so we've dropped the Reader I/O, and merged its would-be material into Editor Interface. This will simply become a more interactive column where readers will communicate not only with me but also with each other. Feel therefore free to reply to any letters that you see published here.

A few letters that we've received disagreed with our editorial direction restricting MS-DOS coverage. We maintain our decision of ***not*** *allowing S-100 Journal to become another MS-DOS magazine. My reply to some of the letters will further clarify our position.*

The Future of S-100

As Emerson said to Whitman, 'I greet you at the beginning of an auspicious career!' The first issue of *S-100 Journal* looks beautiful, and it reads very well. A veteran English teacher doesn't often say that about computer journals. But most of all I admire your fighting spirit. IEEE-696 provides the only truly viable base for the future development of computing. The 'IBM standard,' continually undercut by IBM's attempts to get a proprietary hold, never will be the means of liberating the creative energies of our computing community. Countless man-hours have been wasted to get around its hidden quirks, and only the most advanced engineers with millions to waste can play this game. But despite S-100's obvious superiority, I believe it is doomed unless those who believe in it rise to the occasion.

I have been hacking about in computerland since 1965. As you will see if you read *My Personal Computer* (book enclosed), I like to work things out on my own terms. Now the principal makers of S-100 boards — Cromemco, CompuPro, Lomas, MacroTech, and their dealers — are telling me that I can't make my S-100 system IBM-compatible for less than it takes to buy an off-the-shelf clone! Their literature is cryptic and incomplete. Their tech support people can't answer my questions. This state of affairs is unbelievable, suicidal, intolerable, and immoral. IBM has virtually ruined the market for state-of-the-art Z80 software. Now it endangers the S-100 bus itself. Apple has been badly hurt by Steve Jobs' insistence that the Mac be a closed box, and IBM is tightening its stranglehold. By controlling the kind of hardware and software that evolve, they can control what kind of **information** is processed. That's why inertia in the S-100 community may bring the end of liberty in computing.

The only way for S-100 to beat IBM is to join the crowd, and help the clone makers take the IBM standard away from IBM. The user who wants state-of-the-art software absolutely must become IBM-compatible. Ideally, he needs a box in which he can mount any available piece of software for any processor at a lower price than it would take to buy a new computer. Look at the HiFi business. Early in the game the closed box gave way to separate components. Americans love to do it themselves. They refuse to be force-fed. S-100 is the American way. But S-100 suppliers will lose the fight unless they wake up to the fact that the mass market is there for the taking. Their best weapons are better boards, better documentation, lower prices, and mass advertising. Do they have the guts, or will they take their money and run?

Ben Ross Schneider, Jr.
Department of English
Lawrence University
Appleton, Wisconsin

I did read and enjoy your book. It is not often that one reads a story where the principal character is an S-100 computer (Ben's book — My Personal Computer *from Macmillan Publishing Company, 1984 — is not only fun but also a lighthearted introduction to microcomputers in general and the S-100 in particular).*

S-100's capability of being configured to run any software is of course its most important feature. Therefore, there is no question that S-100 computers should run IBM-compatible software. A good part of the problem has already been solved because most new operating systems (and recent updates to many older ones) include MS-DOS emulators that permit users to run nongraphics IBM-compatible software. The only remaining problem seems to be with IBM-graphics software.

In my opinion, the solution to the graphics problem has been approached from the wrong end. In the early days of S-100 computing, video boards were popular. Then stand-alone terminals became affordable, and everyone went that route because it was simpler, more modular, and more standard. Now that the IBM-graphics question has surfaced, S-100 manufacturers who tackled the problem started from the bottom again by trying to introduce IBM-compatible video boards. This is an imperfect solution; not only does it require

Industrial grade S-100 boards don't cost more, anymore!

Dual Systems has just reduced the prices on its industrial grade IEEE-696/S-100 boards for 16-bit microprocessor systems. Whether running under UNIX or any other operating system, these boards bring high performance and years of field-proven reliability to your computing environment. Each board is rigorously tested and burned-in for 168 grueling hours. If it can't bear the heat, it won't bear our name.

High Performance System Boards

Model WDC-SMDX The WDC-SMDX Hard Disk Controller is specially designed for high throughput in large, heavily-loaded multi-user UNIX systems. It offers 16-bit throttled DMA data transfers and disk transfers up to 15 Mb/sec. Also features dual-ported, full-track, look ahead cache, and on-board microprocessor. Interfaces with one or two SMD drives. $2195.

A 10 Mb/sec. version is also available. $1995.

Model S104-DMA The most advanced, intelligent, 4-port serial I/O board available for the IEEE-696/S-100 bus, this module features 256 bytes of FIFO buffer for input characters, and provides DMA transfers for output. A built-in 8085A processor greatly reduces system overhead. $595.

Model DMEM Features 256K bytes of memory and either 8- or 16-bit data paths. 24-bit addressing, and parity checking on each byte. DMEM has no S-100 wait states. $595.

Model EPROM Capable of either 8- or 16-bit data transfers, this 32/64K EPROM offers the versatility of running with 68000, Z-8000, 8086, 16000, and other 16-bit processors. It accepts industry-standard 2732 and 2716 EPROMs. 64K RAMS may be mixed with 2716 EPROMs for use as a RAM/EPROM board. $295.

Model CPU-68000M High-performance CPU board with 16-bit data path, 10 MHz CPU operation, and MC68451 MMU for multi-tasking applications. $1195.

Model CMEM This non-volatile CMOS memory board provides easy-to-use 8- or 16-bit data paths and 32K bytes of memory with dynamically movable write/protect window. On-board lithium battery holds data for 3-10 years with power off. $725.

Model EMEM-1MB Features 1 megabyte of memory and either 8- or 16-bit data transfers. 24-bit addressing and parity error checking on each byte. Runs at 6 MHz for both 8- and 16-bit systems. No S-100 wait states except during refresh. $1995.

Model TCON Nine-track tape controller supports industry-standard IBM-ANSI formatter interface and DMA on read and write to tape. 24-bit addressing on DMA transfers; 8-bit data on DMA transfers. 512-byte FIFO data buffer. Supports tape densities up to 6250 BPI. $1250 with software drivers; $850 alone.

Data Acquisition and Control Boards

Model CLK-24C Clock calendar features a LSI CMOS chip and on-board, long-life lithium battery. $295.

Model AIM-12 A highly reliable A-to-D converter with 35 microsecond maximum conversion time, 12-bit resolution and accuracy, and 32 channels single-ended/16 channels differential. $725.

Model AOM-12 This D-to-A converter offers I/O-mapped port address, 12-bit ± ½ L.S.B. accuracy (0-70°C), and voltage outputs of 0 to 10 volts, ±5 volts, and ±10 volts. $675.

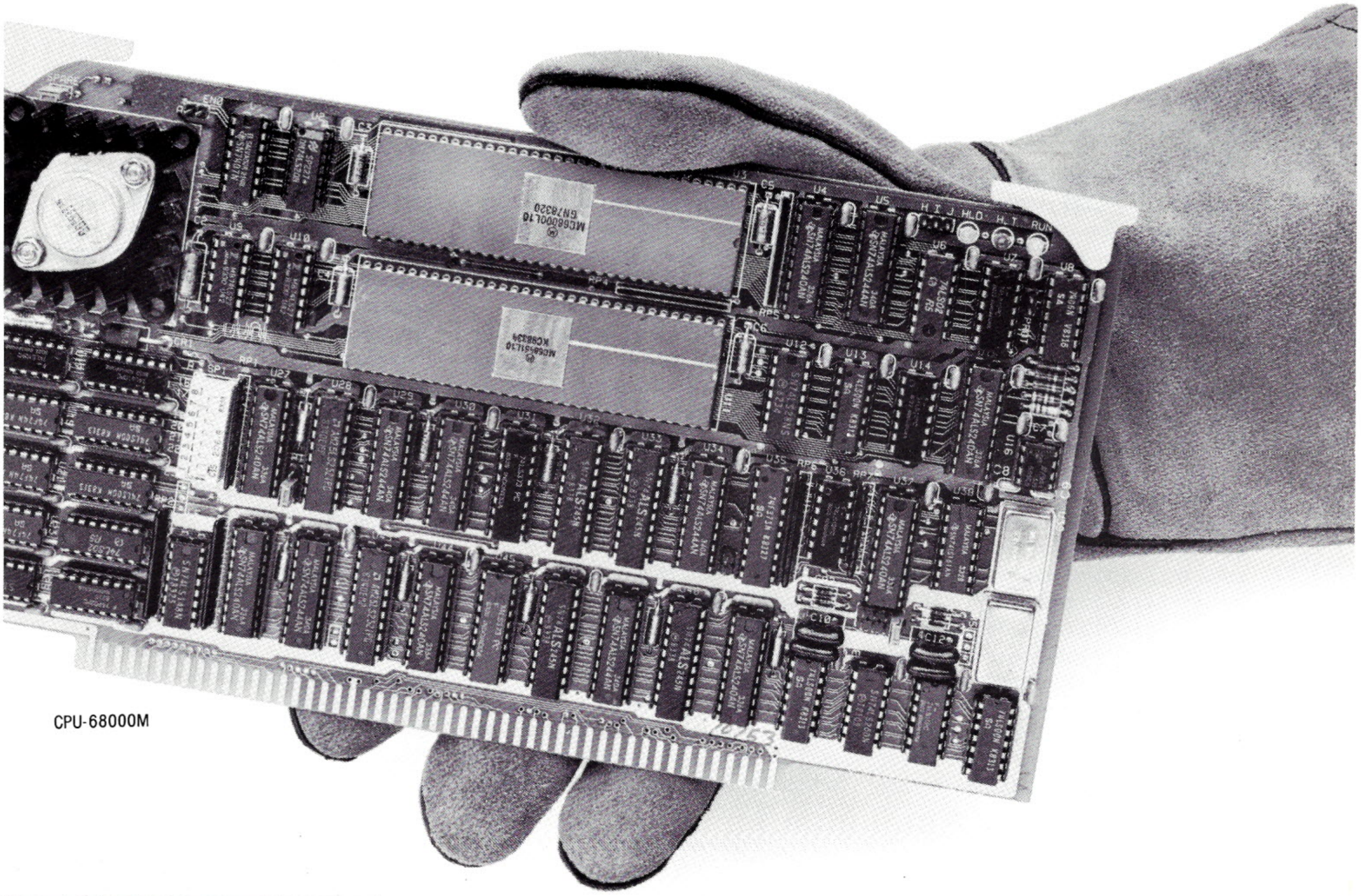
Model VIC 4-20 Converts voltage outputs from AOM-12 into four separate 4-20MA current outputs. Module also provides overvoltage protection on all current output, plus transient protection per ISA standards. $600.

Dealer and volume discounts available.

For more information, call (415) 549-3854.
Dual Systems Corp., 2530 San Pablo Avenue,
Berkeley, CA 94702

THERE'S ONLY ONE

DUAL



CPU-68000M

special software (usually tied to that manufacturer's other S-100 hardware), but you need to buy a monitor and an IBM keyboard. After spending all that money (and assuming that you've been able to integrate the S-100 video board into your present system) you still cannot run all the IBM-graphics software.

Initially, I was excited about these S-100 PC video boards, but I realize now that the graphics compatibility is a problem related more to terminals than to S-100 boards (not unlike the original stand-alone terminal situation). What we need is a ***terminal*** *that can run the silly graphics. A partial solution is to buy a cheap IBM clone (some sell for $600) and use it as a terminal. The preferred solution would be a terminal especially designed for this purpose. Terminal manufacturers have recently introduced a few products that approach this solution. Obviously, not only S-100 users but also the mini and mainframe people would be interested. I will look more closely at the products available in this area and provide more details in a future issue.*

I too believe that the S-100 community needs to join forces and invade the mass market. This is an important reason why we started S-100 Journal, to share information among S-100 users and provide a focus of cohesion from where to expand to higher goals.

Many of the advertising strategies used to sell S-100 products are very inadequate. Too often, suppliers see only two user markets for S-100. One made up of hackers and hobbyists who buy board-level products to experiment with, and another made up of businesses and corporations only interested in fully-assembled systems. Now that the first of these markets is dry, they direct their attention to the second. All along they ignore S-100's natural market: the thousands and thousands of small business owners and of professionals like you and me who, as you say, 'like to work things out on our own terms.' We are little interested in knowing why chip ZKLY has eight pins instead of ten, but even less interested in buying a closed-box. We want an easy-to-assemble system for which we can pick the boards that ***we*** *choose* ***when*** *we need them. Then, after we buy a board, we expect to be able to install it ourselves with user-oriented, easy-to-read manuals, and we expect it to work well with any of the boards that we already have, in order to fulfill the* ***application*** *that we have in mind.*

When S-100 vendors as a whole finally realize that this is the best market for S-100 products, start catering to it, and finally break down inter-company sectarian barriers, their sales should soar well beyond the bounds of the most optimistic estimates.

• Jay

Congratulations on a job well done! It looks as though someone has finally taken the initiative in the horrendous task of unifying and promoting the S-100 industry as a whole — and have done it in a big way. Having been associated with the I.A.C.U. for the past six years, and having worked closely with I/O NEWS for the last two, I have some idea of what you are up against.

I especially enjoyed the tenor of your editorial — your 'straight from the hip' approach in dealing with the MS-DOS mumbo jumbo. And then once this was dispensed with, you proceeded to provide the informative, practical, and otherwise useful articles that are so in line with the nature of S-100 users. Both you, and the rest of the staff, are to be commended on the quality, format, and content of these first issues of S-100 Journal.

There is much that we have in common, and much that can be done in the way of mutual support — both for ourselves and the S-100 industry at large. It is certainly a challenge!

I look forward to exploring the many ways in which we can assist each other in achieving our respective and shared goals.

William E. Jaenicke
Editor & Publisher
I/O NEWS
Irvine, California

Thank you, Bill, for your compliments on S-100 Journal. I also look forward to our mutual support. S-100 Journal is ready to support I.A.C.U. and other associations of S-100 users in any way that we can, including making free space available in our pages for announcements and whatever else you may require. (Note: I/O NEWS is the publication of The International Association of Cromemco Users. It's a very nice and informative publication and it is sent bimonthly to members of the Association. If you use a Cromemco, you definitely need to join. Their address is P.O. Box 17658, Irvine, CA 92713, phone 714-955-0432). *• Jay*

More on MS-DOS

I very much enjoyed your new 'S-100 Journal'. I feel that it fills a real need in the market. The IEEE 696 bus supports all of the modern processors, languages and operating systems. It is a very rich and inclusive system. I applaud your commitment to one of the most versatile and expandable small computer architectures. I also applaud your insistence on not becoming just another IBM magazine. There is, however, one point in your advertising and review policies which I feel that you should reconsider.

Your refusal to review or advertise 'generic' MS-DOS applications is like cutting off your nose to spite your face, throwing the baby out with the bathwater, and any one of a number of trite old maxims! I own two S-100 systems, and I am working on a third. One of these systems, a Heath/Zenith H-100, does run those same generic MS-DOS programs that you refuse to advertise and review. (This same machine also runs under CP/M 2.0 and CP/M 86.) This letter was written with Wordstar, and proofread with the Wang / Random House Proofreader, both are generic MS-DOS applications running on the H-100. There are also a number of S-100 bus manufacturers who are working on both hardware and software products to allow even PC-DOS specific programs to run on the S-100 bus. Anything that works with or adds to the S-100 bus should be covered by your magazine, no matter who designed it, or in spite of the fact that it will also run on other hardware configurations.

Perhaps a small change in policy is in order: Review and advertise those products which will work with generic S-100 systems, i.e., MS-DOS, but not PC-DOS products. Then state explicitly in the review or advertisement which hardware and software configurations the product is known to work with. I realize that you can't directly control an advertisers layout and copy, but you can refuse to run an ad if it does not meet your specifications.

Scott Christensen
St. Paul, Minnesota

I thought I'd tell you about something in S-100 Journal that annoys me.

Like the Ostrich Complex, S-100 Journal is going to kill itself with its own phobia. IBM and its offshoots cannot go away. Pretending can only make the pretender disappear.

I prefer my S-100 machines to IBM products, but that doesn't make me stupid. The truth is that the most elaborate software runs on IBM and nowhere else, that software developers write for the million dollar markets, and users of non-IBM configured machines pay through the nose for their idiosyncrasy.

Few hobbyists can afford the premiums noncompatibility costs, and no professionals can. Development software, compilers, business software, and even hardware is too expensive when the manufacturer is limited to a small market base. It costs several thousand dollars to buy the boards and software to emulate the cheapest IBM clone.

The manufacturer who brings out an IBM-compatible S-100, which means a box that holds IBM-compatible components as well as the futuristic wherewithal to leave IBM gasping for air, is the only hope for S-100.

I hope this isn't my last subscription to S-100 Journal. But the circle-the-wagons attitude I saw in your first issue isn't at all promising.

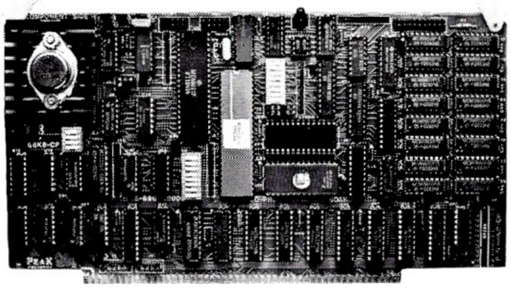
C. M. Kotlan
O'Brien, Oregon

A clarification of our editorial position concerning MS-DOS is in order: We have indicated on both of the previous issues, as we have in several places on this one, that we ***do*** *support the S-100 bus running MS-DOS, PC-DOS, or* ***any*** *other operating system. The versatility of being configured to run any kind of software is what the S-100 bus is all about. I want to emphasize that we* ***will*** *accept articles and publish other information that relates to implementing MS-DOS or PC-DOS on S-100 or helps solve problems encountered when running these operating systems on S-100. We will also publish driver software listings to allow running specific S-100 boards under MS-DOS.*

CP/M has been, and continues to be, a favorite of single-user S-100'ers, including several of our authors. For this reason, many articles use CP/M BIOS alterations as examples of how to implement device drivers, but this will not be the case for every article. We do want to cover every OS as it relates to S-100.

A quite different aspect is how much to cover MS-DOS itself, as independent from S-100 systems. This is where we hold off. S-100 Journal covers mostly S-100 hardware and software that is intimately tied to S-100 hardware. Some room is left to cover general purpose software (i.e., software that was not written with an S-100 system in mind), but we prefer to give that room to software that is ill supported by other magazines. There are many fine magazines that cover MS-DOS extensively. However, there are over 20 other operating systems (many of them more powerful and more versatile than MS-DOS) that run on S-100 systems, and most of them are

(continued on page 33)



DETAILS OF THE FINAL STANDARD

Don Pannell

696 Bus is a regular column that concentrates on the hardware aspects of the IEEE-696 bus (i.e., the S-100 bus) and answers questions that readers might have about the IEEE standard.

Don Pannell, our 696 bus columnist, is an S-100 hardware enthusiast and is coauthor of the IEEE-696 standard. He bought his first S-100 system (an Altair 8800b kit) in January 1978 and has since designed and built most of his present components. These include a terminal, EPROM programmer, serial/parallel I/O cards, two designs of DMA floppy-disk controller, and a 68000 coprocessor card. (Don markets the 68000 coprocessor through his Peak Electronics company.)

If you have questions about the S-100 bus and IEEE standard, write to Don Pannell, PO Box 700112, San Jose, CA 95170-0112. Your questions can range from architectural concerns to how to interface a specific device or function with the bus. In future issues, Don will incorporate answers to the most common questions.

In the last issue, I gave an introduction to the IEEE-696 bus and briefly discussed the differences between the well-known July 1979 'IEEE Computer' magazine draft and the approved standard. In this issue, I will cover these differences in more detail. The previously mentioned discussion on the 'fast wait state generator' circuit will appear in the next issue.

DRAFT TO FINAL STANDARD THE CHANGES

The proposed standard, published in the July 1979 'IEEE Computer' magazine, was known as 'draft D2.' This draft succeeded in better defining and vastly improving the functionality and speed of the S-100 bus. However, it still contained some deficiencies. Improvements were made in both the architectural and operational portions of the standard.

The Architectural Changes In Review

Architectural changes between draft D2 and the final standard were in two major areas: 1) the 16-bit data bus interface, and 2) the temporary master interface. The changes in these two sections of the specification, I discussed in detail in the previous issue of S-100 Journal. However, I will cover the highlights here.

1. The draft's 16-bit data bus was labeled as having high- and low-byte significance. This created many problems since there are two ways of storing 16-bit numbers in byte-addressable memory. The approved specification corrected this problem by removing the byte significance on 16-bit data transfers. The two 8-bit data buses are now called 'even' and 'odd' (as opposed to 'high' and 'low') when they are combined for 16-bit reads and writes. The even byte is assigned to the even-numbered memory address while the odd byte goes to the odd address. NOTE: More than just a name change took place in the 16-bit bus interface. The bus steering logic was also modified slightly.

2. The temporary master interface was changed mostly in name. The signals were renamed from DMA0* through DMA3* to TMA0* through TMA3*. This change was made because DMA stands for Direct Memory Access, which implies that only memory operations can be performed. In fact, temporary masters can perform any type of bus operation. Therefore, the name was changed to Temporary Master Access (TMA). In addition to the name change, a bug in the example arbitration circuit was removed.

The Operational Changes In Detail

Operational changes, between the final specification and draft D2, mostly cleaned up timing specifications and defined better some of the bus signals' functions. A new IDLE bus cycle was added as well.

EARTH LAUNCHES NEW STARS

EARTH COMPUTERS launches two of the "Hottest" new stars in the S-100 Universe. Both the **TURBOMASTER 8**™ and **TURBOSLAVE I**™ are Star performers, featuring high speed Z-80H CPUs.

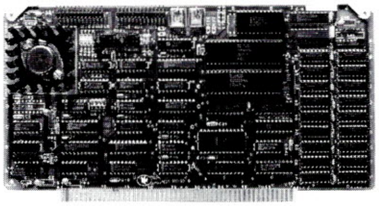
EARTH's new stars are fully compatible with the Multi-user TurboDOS operating system, and will operate in most S-100 systems, including pre-IEEE 696 systems.

TURBOMASTER 8

This outstanding new 8-Bit Single Board Computer offers features that are out of this world:

- On-board ST-506 Winchester Controller
- TurboDOS, CP/M, MP/M compatible
- 5-¼″ and 8″ Floppy Controller
- Up to 256KB of memory
- 8 MHz, Z-80H CPU
- 2 Serial ports
- 1 Parallel port

TURBOSLAVE I

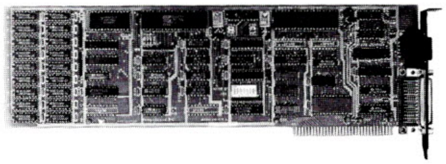


The perfect companion to the TURBOMASTER 8 or other 8/16-Bit Master processor. This high speed slave utilizes an 8 MHz Z-80H CPU and offers extensive on-board diagnostics... an industry exclusive.

- No paddle boards
- S-100, IEEE 696 compatible
- 128KB of RAM
- 2 RS-232 ports, 50–38.4K Baud
- FIFO communications
- Data transfers to 1 MB

EARTH COMPUTERS also manufacturers a growing line of PC compatible stellar performers such as:

- **TURBOSLAVE PC**™—an 8 MHz



Z-80 single board slave processor that runs CP/M applications on a PC and is compatible with the TurboDOS multi-user operating system.

- **TURBOACCEL 286**™—a high performance 80286 accelerator that boosts PC performance up to five times.
- **EARTHNET PC**™ and **EARTHNET S-100,** the low cost, ARCNET-compatible way to tie PC and S-100 systems together.

To put these stars to work for you, call or write EARTH COMPUTERS. **BE SURE TO ASK ABOUT HOW YOU CAN WIN A FREE Z-80 CO-PROCESSOR BOARD.**

EARTH COMPUTERS

"Building Blocks For The Super Micro"

P.O. Box 8067, Fountain Valley, CA 92728 ● TELEX: 910 997 6120 EARTH FV ● PHONE: (714)964-5784

SUBSCRIBE

PLEASE ENTER MY SUBSCRIPTION TO S-100 JOURNAL.
ENCLOSED IS □ $14.00 FOR 1 YEAR. □ $25.00 FOR TWO YEARS.

(Canada, Mexico, and England please add $3.00 per year. Other countries please add $6.00. All payments must be in U.S. currency.)

NAME ____

____

ADDRESS ____

____

PLEASE MAIL TO: S-100 JOURNAL, PO Box 12881, Raleigh, NC 27605

SPRING 86

The final specification varies from draft D2 in the following operational ways:

1. An IDLE bus cycle was defined.

An IDLE bus cycle can be used by a bus master when it wants to generate the bus strobes and have all the devices on the bus respond to it as a NO-OP. This type of cycle would most likely be generated by a dummy bus master, since a dummy master only waits for temporary masters to request the bus. IDLE bus cycles are optional. However, all bus slaves should be able to support IDLE bus cycles.

An IDLE bus cycle is identical to all other bus cycles; it looks just like a bus read or a bus write. The only difference is the value of the status bus during the cycle (Table 1).

NOTE: IDLE bus cycles are very different from IDLE bus states. All bus cycles consist of three or more bus states; and any bus cycle can be followed by one or more IDLE bus states. IDLE bus states are simply dead time between bus cycles.

2. The PHANTOM* signal was initially part of the utility bus only. It is now mentioned with the address bus as well, with timing specifications added.

The PHANTOM* signal allows overlapping of memory devices. It is usually used during the boot process to place a boot-up EPROM at an address location already occupied by memory. After booting, the EPROM is disabled and the overlapped system-RAM becomes available.

The key additions in the specification are:

- All memory slaves shall be capable of being disabled by the PHANTOM* signal.
- The memory slaves must be disabled for BOTH reading and writing whenever PHANTOM* is asserted.
- PHANTOM* must be stable from 30ns before to 30ns after the bus-read or bus-write strobe.

3. POC* (Power On Clear) is better defined. Timing specifications were added.

POC*'s new definition is: *'POC*, power-on clear, is active at power-on and following the rising edge of PWRFAIL*, and shall cause SLAVE CLR* and RESET* to be asserted. The POC* signal is specified as having a minimum active period of 10ms; RESET* and SLAVE CLR* are specified as having a minimum active period of 5us.'*

4. Powerfail specifications were changed.

The major change here is that PWRFAIL*, if supported, must go low 16us before the on-board voltage regulators go out of specification. And, after PWRFAIL* goes low, it must stay low for at least 16us.

This is an important specification. PWRFAIL* must stay low long enough for the POC* circuit to fully discharge. If the circuit isn't allowed to discharge, a short power brown-out could pulse PWRFAIL* so quick that POC* won't work, causing the system to hang instead of rebooting.

5. Termination methods of open-collector lines were changed.

The draft called for open-collector lines to be terminated exactly like all the other signal lines (i.e., 180 ohms to +2.6 volts). In addition, open-collector lines required a 1.5K-ohm resistor to +5.0 volts.

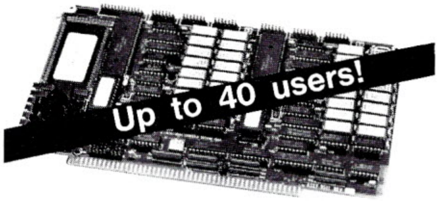
The approved standard removed the 1.5K-ohm pull-up resistor on

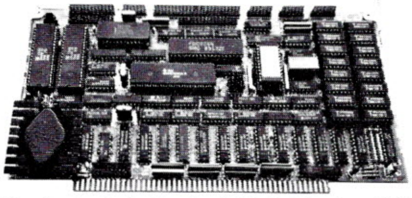
(continued on page 35)

sMEMR	sM1	sWO*	sOUT	sINP	sHLTA	sXTRQ*
L	L	H	L	L	L	X

H=High L=Low X=Don't Care

Table 1. *Value of status bus during an IDLE bus cycle.*





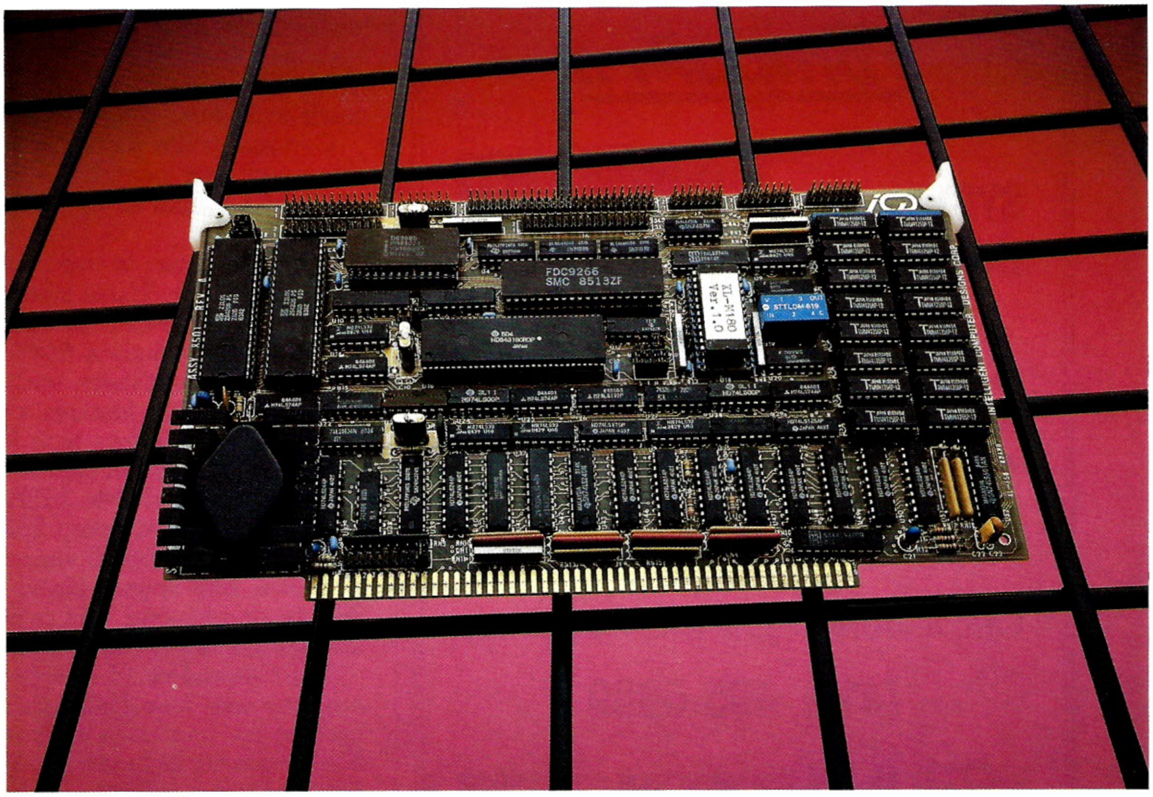
special report

S-100 @ COMDEX

Despite stern competition from an immensity of other computers, S-100 systems continue to prosper. Over twenty new S-100 boards were introduced during the past year, numerous new complete systems have been integrated, more operating systems continue to be ported to S-100 computers, and sophisticated LAN (Local Area Network) schemes are now offered by many S-100 companies.

The success of individual S-100 companies (and by extension that of the S-100 bus in general) seems to depend more on marketing strategies and how the market demands are perceived than on the actual demands of the market itself. There should be no surprise that a high demand continues to exist for S-100 systems, in spite of a nearly total disregard for their existence by the dominant microcomputer press. After all, the S-100 bus is the only system that offers unlimited modularity, unlimited expandability, always the latest performance, and is truly nonproprietary.

S-100 Journal visited the COMDEX last fall, and we were pleased to find a wealth of S-100 systems and boards. Recently, a subscriber wrote to express his happiness over the publication of S-100 Journal, and he asked in his letter: 'The S-100 bus is really not dead, is it?' We hope that the following sample of what we saw at COMDEX will answer his question.



The XL-M180 Super Single Board Computer from ICD. Based on the HD64180 microprocessor, this board features 512K of memory, 3 serial ports, 4 parallel ports, and controllers for hard disk, tape, floppy, and LAN. Several single or multiuser operating systems are offered preconfigured for the board.

ACTION COMPUTER ENTERPRISE

The Discovery X3 is the latest high-performance multiuser/network system from Action Computer Enterprise, Incorporated. An IEEE-696 S-100 computer, the Discovery X3 allows up to 16 users, up to 6 parallel or 14 serial printer ports, and both 8-bit and 16-bit processing.

The user stations can be either a PC or a terminal. At COMDEX, Action Computer had a Discovery X3 set up with several terminals and PC/terminals. PC/terminals (typically an IBM-compatible personal computer) connect to the S-100 master processor (an 80186 CPU board) and have full access to both the Discovery X3 resources and the PC/terminal resources. Users with simple terminals have a dedicated S-100 slave board in the system and have full access to the board's resources and to the Discovery X3 resources.

ACE offers a choice of user slave boards which can be used in any combination depending on each user's needs. They are available with an 8-bit CPU (Z80A) or with a 16-bit CPU (8088 or 8086/8087). Two serial ports are available per board, allowing users to have their own local printer or other serial peripheral. The 16-bit slaves can have up to 1 Megabyte of on-board memory, while the 8-bit slaves can have up to 96K.

Users with 16-bit slaves can run Personal CP/M (CP/M-86) or Concurrent DOS (includes MS-DOS emulator). CP/M Plus is available for 8-bit users. The PC/terminals can of course also run programs specific to their local hardware.

When users access system resources, the host operating system (ACE's dpc/os-86) manages the access with advanced multiuser features such as file and record locking, password protection, and print spooling. The use of the system's resources is not limited by system-to-terminal distance.

The Discovery X3 is available configured for the number of users required by the customer and, as needs grow, it can later be upgraded to more users, more storage, or more peripherals. A system/network based on the Discovery X3 would typically be able to run all the software available for CP/M, CP/M-86, MS-DOS, and PC-DOS. For more information, fill out and send one S-100 Journal's Editorial Feature Reply Cards to ACTION COMPUTER ENTERPRISE, 430 N. Halstead St., Pasadena, CA 91107.

COMPUPRO

CompuPro/Viasyn had an impressive booth at COMDEX and was giving away a cruise each day. The company has slackened considerably in the promotion of board-level products and was energetically promoting its fully-assembled multiuser systems. Nevertheless, it seems that a good number of visitors to the booth were expressing strong interest in boards.

CompuPro has placed its marketing emphasis in three multiuser integrated systems: the CompuPro 286, the CompuPro 10 Plus, and the CompuPro 816/C2.

The 286 comes in two configurations, the 286-10 and the 286-40. These systems are based on CompuPro's 80286 8-MHz processor board (CPU 286). Both systems include a 5¼″ floppy drive and word processing, spreadsheet, and communications software. The 286-40 features a 40-Megabyte hard disk with 10-Megabyte tape back-up, 768K of RAM, and nine serial ports. The 286-10 features a 10-Megabyte hard disk, 512K of RAM, and four serial ports. The standard 286-40 system supports eight users, the 286-10 supports four users. The 286-10 is fully upgradable to the features of the 286-40 and beyond. List price is $9995 for the 286-40 and $6495 for the 286-10.

The CompuPro 10 Plus is an 8088-based machine with four Z80B slave processors, one for each of the four users supported by the system. The computer includes 1 Megabyte of memory, a 10-Megabyte hard disk, a 5¼″ floppy drive, seven serial ports, and one Centronics port. List price is $4995.

The CompuPro 816/C2, the only one of the three systems that is IEEE-696 standard (the others have unregulated S-100 boards), is an 80286-based system that supports up to 14 users, each with an 8-MHz Z80H slave processor. The system includes 512K of RAM, 10-Megabyte hard disk, 5¼″ floppy drive, and nine serial ports. List prices start at $7495.

All CompuPro systems are expandable. Some of the available options are more memory, RAM-disk, tape back-up, larger hard disks, IBM/PC-compatible graphics, and networking (using CompuPro's NET 100 LAN board) to other multiuser systems and to personal computers.

CompuPro systems run under Concurrent DOS 8-16, CompuPro's implementation of Digital Research's multiuser, multitasking operating system. It supports both 8-bit and 16-bit applications. A typical CompuPro system can run CP/M-80, CP/M-86, and MS-DOS applications. A unique feature of CompuPro's C-DOS implementation is the availability of software to read diskettes written in a variety of formats.

The company has traditionally based its products on the Intel line, but claims to maintain good relations with Motorola. The release of boards based on newer Motorola CPUs and implementations of UNIX-like operating systems are apparently planned.

For more information on CompuPro systems and board products, send a completed Editorial Feature Reply Card to Viasyn Corporation, 26538 Danti Court, Hayward, CA 94545-3999.

INDEPENDENT BUSINESS SYSTEMS

IBS showed its Ultraframe line of fully integrated IEEE-696 S-100 supermicros. The new systems allow up to 32 users and PC/terminals.

Each Ultraframe features either a Z80B or an 80186 master CPU. Each user has full access to either an 8-bit or a 16-bit slave board. Slave boards include on-board memory and are available with a 6-MHz Z80B, an 8-MHz 8086, or an 8-MHz 80186 microprocessor. In addition, the system's main resources are available to all users.

Users on PC/terminals (personal computers used as terminals) access the system's main hardware and software via an S-100 ARCnet controller board. PC/terminals can be useful to run MS-DOS graphics software locally.

A choice of two operating systems is offered: TurboDOS and IBS p-NET. TurboDOS includes advanced multiuser features, networking support, and the ability to run all CP/M and nongraphics MS-DOS software. IBS p-NET is a multiuser multitasking OS compatible with UCSD Pascal and comes with a Pascal compiler. As an added benefit of IBS p-NET, two users can share one slave board. Therefore, with the same hardware, more users are possible under p-NET than under TurboDOS.

A variety of disk configurations is available for the Ultraframe. Floppy disk drives can be either 315K 5¼-inch or 1.2-Megabyte 8-inch. IBS offers hard disk drives of 10 to 300 Megabytes. Several hard disks can be used simultaneously for up to 1,160 Megabytes of storage. In addition, cartridge tape and video tape subsystems, with respective controller and S-100 interface, are available.

Like all S-100 systems, the Ultraframe is an expand-as-your-needs-grow computer. IBS offers an impressive 3-year warranty on components sold as part of a system. All components, including individual S-100 boards, are also sold separately (they carry a 1-year warranty).

To obtain more information, complete and send an Editorial Feature Reply Card to IBS, 5915 Graham Court, Livermore, CA 94550.

INTELLIGENT COMPUTER DESIGNS

A relatively newcomer to the S-100 market, ICD has already produced an outstanding line of S-100 boards.

The company's top product, introduced at COMDEX, is the XL-M180. This IEEE-696 super single board computer features the new Hitachi HD64180 CPU, up to 512K of memory, 3 serial ports, 4 parallel ports, hard disk and tape controller, floppy disk controller of up to four drives (8″, 5¼″, or 3½″), and even an ETHERNET local area network controller, all on a single board. Simply apply power, connect peripherals, and run.

The ULTRAFRAME line of multiuser systems from IBS allows up to 32 users and can control a network of personal computers. ULTRAFRAME systems are available in a variety of 8/16-bit configurations.

The heart of the XL-M180 is the HD64180. This 8-bit Hitachi chip is upwardly compatible with the Z80-8080-8085 instruction set. The chip is actually more than a CPU, packing in serial ports, timers, DMA channels, and memory management. This is extremely convenient to increase available board real estate. Unlike previous 8-bit CPUs, The HD64180 can directly address 512K of memory. A 64180 is said to be 30% faster than a Z80 of equal clock speed. Since ICD offers the board with clock speeds of up to 10 MHz, the company claims that its board can outperform many 16-bit boards.

ICD offers several operating systems preconfigured for XL-M180: Single and multiuser TurboDOS, CP/M, the Z-System (stand-alone ZCPR3), and THEOS (Oasis). A single user system (board and operating system) starts at $645. A multiuser system (board and operating system) starts at $945.

To complement its XL-M180 master, or any other S-100 master, the company offers a growing line of 2-user slave boards. The XL-DLZ80 is a dual Z80H 8-bit CPU board with 512K of memory, and four serial ports. The XL-DL88 is a dual NEC 70108 16-bit CPU board (the NEC 70108 processor is compatible with Intel's 8086 and 8088). It also includes up to 512K of memory and four serial ports. ICD's new dual slave is the XL-DL180, similar to the dual Z80, but using two HD64180 chips instead. This slave board can run at speeds up to 10 MHz, and features 512K of nonbanked memory, three serial ports, and expansion ports. With one of these dual-slave boards, two extra users can be added to a multiuser system at a cost of only $250 per user (plus terminals).

ICD's products provide an attractive low-cost option to integrate powerful single-user or multiuser systems that can run a wide variety of software and have LAN capabilities. For additional information, send a completed Editorial Feature Reply Card to Intelligent Computer Designs Corp., 23151 Verdugo Drive, Suite 113, Laguna Hills, CA 92653.

INTERCONTINENTAL MICRO

Taking advantage of Hitachi's HD64180 compactness, Intercontinental Micro has accomplished an unprecedented feat: four users on a single S-100 card. Each user has his/her own on-board HD64180 processor, 128K of RAM, and two serial ports. The new board, called the CPS-Q6A, runs at 6 MHz with provisions to run at 10 MHz. An I/O mapped board, the CPS-Q6A can be added to practically any S-100 system without hardware modification. As an extra feature, any one or more of the four user circuits can be used instead as a printer server while the remaining circuits remain as users. ICM includes free TurboDOS drivers with the board. List price is $1995.

InterContinental Micro also introduced a dual 16-bit CPU slave processor, the CPS-216. This board features two 8086 processors, 512K of RAM, and four serial ports. It permits the addition of two new 16-bit users to most multiuser S-100 systems. The board could be used to add two 16-bit workstations to an 8-bit system for example, thus making it possible to run CP/M-86 and nongraphics MS-DOS software. List price for the CPS-216 is $1695, including TurboDOS drivers.

InterContinental has developed a very complete set of products that allow building sophisticated local area networks (LANs) around S-100 computers. The S-100 computers typically control the network and act as file servers. Individual workstations can be standard terminals connected to a slave board in an S-100 system or they can be PC/terminals with local computing power. Up to 255 masters (S-100) and up to 4,000 workstations are possible under ICM's networks. In these ARCnet LANs running under TurboDOS, each workstation has master-controlled access to the system's general resources and to local peripherals.

ICM offers network controller and network interfacing boards for S-100 systems, including the Zenith Z-100, and for other micros. To demonstrate their products at COMDEX, they had a master S-100 multiuser system, with both 8-bit and 16-bit workstations, networked with a Z-100, several hard disk and tape drives, and two PC/terminals (AT and PCjr).

For additional information on ICM's line of S-100 boards and/or network components, fill out and send an Editorial Feature Reply Card to InterContinental Micro, 4015 Leaverton Court, Anaheim, CA 92807.

JC INFORMATION SYSTEMS

JCIS provides multiuser upgradable systems based on a Z80B, Z80H, or an 80186 master running at either 8 or 10 MHz. Slave processor boards are also available with either of the above CPUs, allowing both 8-bit and 16-bit workstations. JCIS slave boards have the additional capability of functioning as local masters in a network. 8-bit slave boards carry up to 128K of memory; 16-bit boards up to 256K.

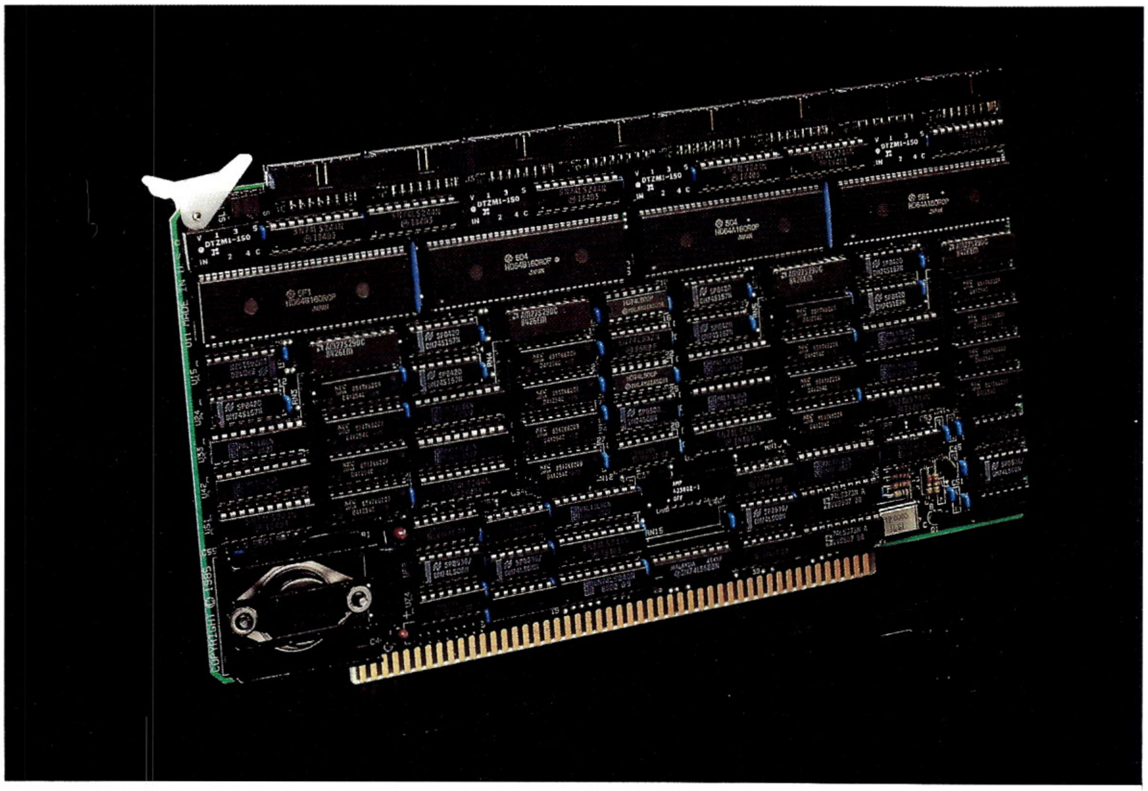
JC systems offer up to 140 Megabytes of hard disk storage, 8″ or 5¼″ floppy disk drives, back-up tape storage, RAM-disk capability, serial and parallel ports, and run under TurboDOS. The systems include unique software to read a variety of diskette formats. A typical system can run CP/M, CP/M-86, and nongraphics MS-DOS software.

A LAN controller board, the JCL*80, allows the master S-100 to act as a file server to up to 127 PC/terminals. Sophisticated networks can be created using this board and other JCIS components.

JCIS also offers a complete line of expansion S-100 boards. For more information on systems or components, send an Editorial Feature Reply Card to JC Information Systems Corp., 161 Whitney Place, Fremont, CA 94539.

(continued on page 49)

The CPS-Q6A from InterContinental Micro allows four users on a single S-100 slot. Each user circuit has its own HD64180 CPU, 2 serial ports, and 128K of RAM. TurboDOS drivers are included with each board.



A REAL-TIME CLOCK FOR YOUR S-100

Brian Smithgall

have you ever wanted to build your own S-100 board? Maybe you've wished that you could build a circuit to interface to some peripheral device or that you could learn more about how your computer works inside. In this article, some of the internal works of the S-100 bus are explained by building a time-of-day clock board. We'll discuss S-100 bus cycles, how to build the clock-support circuits, and how to interface these circuits to the bus.

This article is aimed at the S-100 owner who has familiarity with using software and wishes to know more about the hardware. The article is written in a step-by-step manner so that you can learn as we go.

WHAT YOU SHOULD ALREADY KNOW

In order to start on a common ground, I will have to assume that you are already fairly familiar with your computer. I'll also assume that you know about logic functions and their symbols (if you don't, you can probably pick it up along the way). You should also know basic electronic concepts and know how to read schematics and data sheets. Otherwise, you may want to start with a book on elementary electronics.

Hopefully, you are somewhat adventurous — if you have never taken the lid off your computer to see where the boards are, maybe you should stop here. You also need to be a neat, careful worker. Failure to double check a circuit has caused many computer boards to end in a fiery death. But, if you're still interested, clean off some desk space, and let's get started.

PARTS AND TOOLS THAT YOU WILL NEED

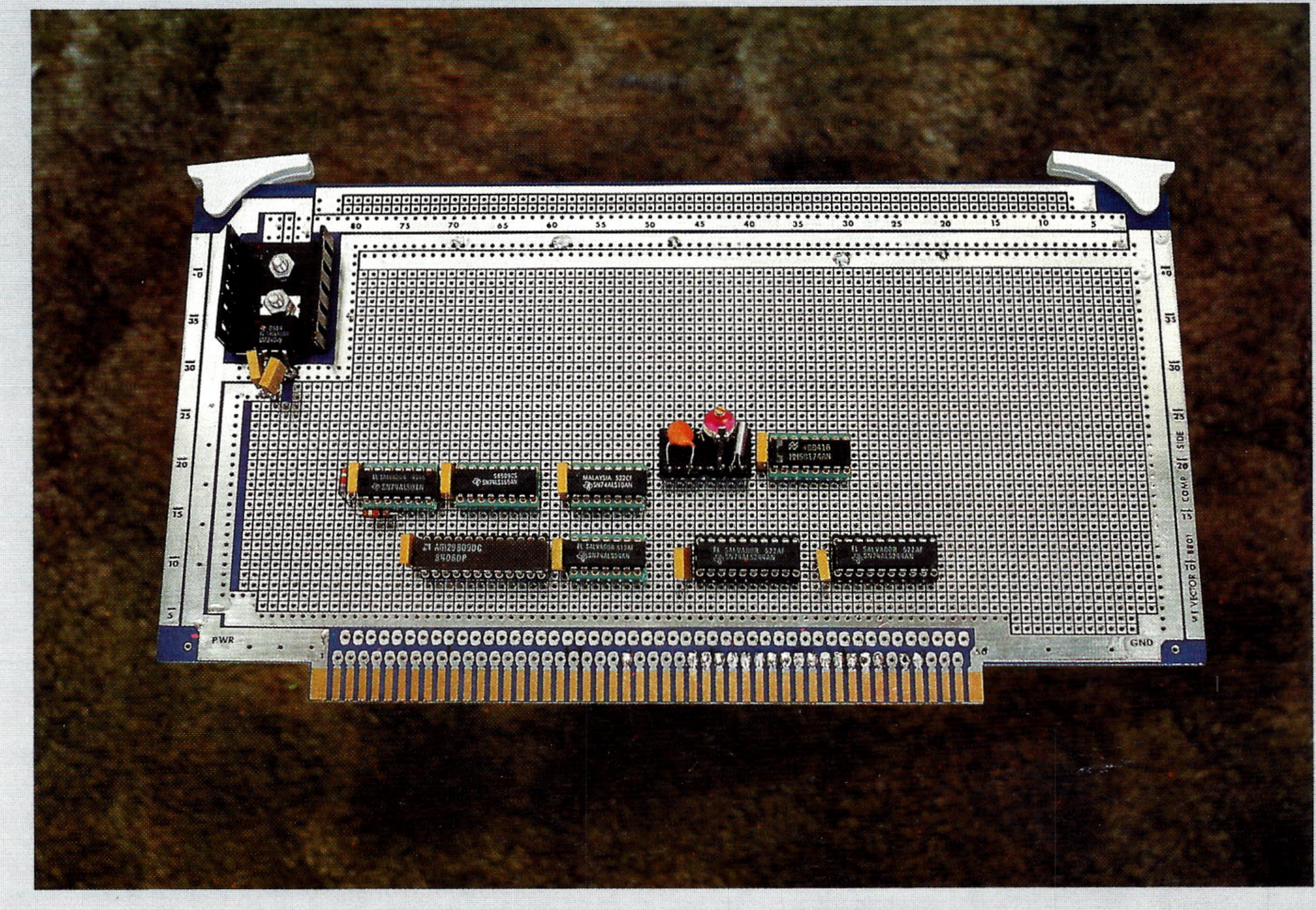
You will have to obtain some parts and tools to build the circuits described here. The parts that you need are listed in Table 1. In addition, you'll need a soldering iron and solder, a wire wrap tool and wire, and some type of electronic measuring instrument. An oscilloscope is preferable, especially for learning, but a voltmeter and logic probe will probably suffice. If you don't own a scope, maybe you know somebody with one who might enjoy helping you and learning at the same time.

It is advisable to find a source of reference books on the parts that you buy. Some distributors will provide a data sheet on purchased parts. A library is another possibility. Or you might be able to ask someone who works in an electronics company if he or she can get you the information you need. Often they throw away last years data books when the new ones come in, so they may even be able to save several of these older books for you. If all else fails, the books may be purchased directly from the manufacturer (TI, National, etc).

The items required for this project are pictured on page 20. The parts should cost about $60, with half of that being the prototype board.

ABOUT THE PARTS

The parts that we will be using belong to a logic family called TTL (bipolar Transistor-Transistor Logic). Each of these compatible parts performs a logical function, based on signals supplied to input pins, and provides the proper response at the output pins. Hundreds of chip functions are



available ranging from primitive logic functions, such as AND or OR, to complex microprocessors. These chips can be combined to create just about anything you can think of (and more).

Signals pass to and from the chips at nominal levels of 0 (<0.3) and 5 (>3.5) volts. These low and high voltages are represented by the logic levels 0 or 1.

Output signals from these chips are usually generated by one of three TTL circuits within the chip. Standard TTL outputs provide either high or low signals and may not be wired together. Open collector outputs may be connected together; the resultant signal is low if any one of the combined lines is low, and high if all would otherwise be high. Tristate outputs have high and low states as above plus a third high-impedance state which can be connected to either high or low signals without affecting them. This third type is useful when connecting output signals to provide an input to another chip. All lines except one are put into a high-impedance state, and the signal from that (non-tristated) one is passed unaltered to the next chip. The chosen chip is then said to be enabled.

TTL is somewhat forgiving about accidently hooking these outputs together for short periods. However, one sure way to kill a chip is to put a negative voltage into its input. Take care not to do this.

The chips specified are from the LS (Low Power Schottky) family. If you can find ALS parts, the circuit will consume less power. Either way, power consumption will not be a problem on such a small board.

The resistors and capacitors are readily available from a local electronics store. Most of their values are not critical, so you may use whatever close values you can easily find. But the crystal frequency and the 5-volt regulator are critical. Most parts can be cross-referenced to other manufacturers if desired. If the AMD comparator is too difficult to find, use two 74LS688 8-bit comparators. Their use should be clear from their data sheets. Almost all other parts are available from JDR Microdevices.

NOMENCLATURE OF LOGIC SIGNALS

The labels for signals on schematics are sometimes followed by an asterisk (eg. BDSEL*). This means the signal is active low or, in other words, that the described signal ('board select' in this case) is true when the signal is low (=0 volts). If the signal is run through an inverter, the output would be (BDSEL), active high. Sometimes the active low state is indicated by a bar over the letters instead of the asterisk.

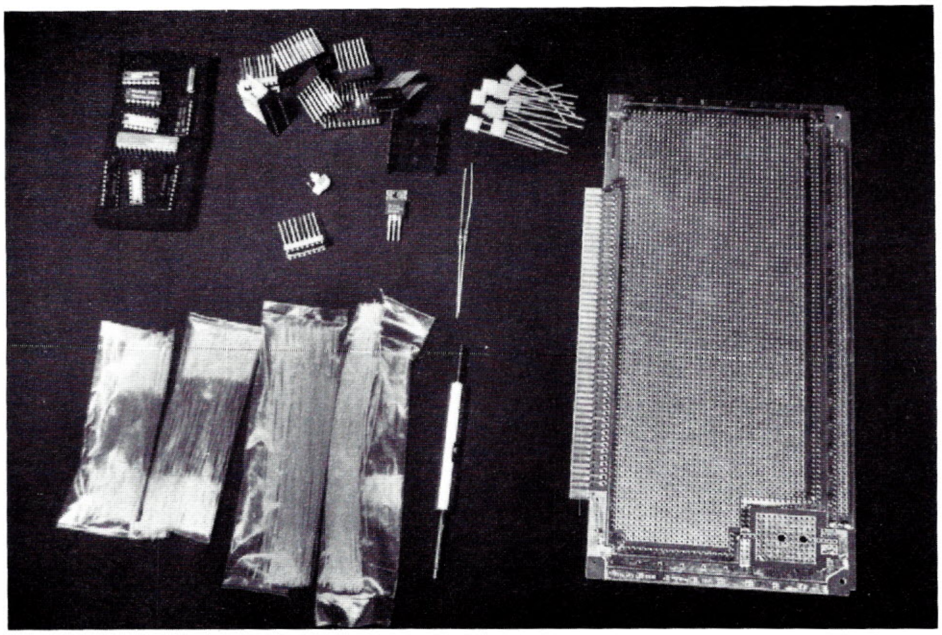
THE S-100 BUS AND ITS CYCLES

Some attributes of the S-100 bus that may be relevant to this project are highlighted below.

On-board regulation: The 8 and 16 or so volts on the motherboard are dropped to 5 and other voltages as required on every board. This makes the boards a little hot, but keeps them more electrically isolated from each other.

Master and Slave boards: Boards

The parts that you will need to build the real-time clock board.



Qty.	ID	Part Description
3		14-pin wire wrap sockets
2		16-pin wire wrap sockets
2		20-pin wire wrap sockets
1		24-pin wire wrap socket
1	U1	MM58174 (National) clock chip
1	U2	74LS01 (TI) 4 dual-input NAND gates (open collector)
1	U3	74LS04 (TI) 6 inverters
1	U4	74LS10 (TI) 3 triple-input NAND gates
1	U5	74LS165 (TI) 8-bit parallel-input serial-out shift resister
2	U6, U7	74LS244 (TI) 8-line tristate line drivers
1	U8	AM29809 (AMD) 9-bit comparator for address decoder
1		LM340-5 (TI) 5-Volt regulator
1		S-100 (Vector Elect.) prototype board. (The type with separate pads per hole is best.)
1		Heat sink for regulator
1	X_1	32.768 kHz crystal (Hughes) for oscillator circuit
1		1000 Ω resistor - pullup for open collector to LS chip
1		4700 Ω resistor - pullup for open collector to bus
2	C_1, C_2	1 uF 25 volt capacitors for power supply
1	C_3	18 pF capacitor for oscillator
1	C_4	5-45 pF trimmer capacitor to tune oscillator
8	C_D	0.1 uF decoupling capacitors (1 for each chip)

Table 1. *List of parts for the S-100 real-time clock.*

on the S-100 bus are defined as either master or slave boards. Usually one board is defined as the bus master, and the other boards function as slaves, although these functions may be exchanged (rarely). In most cases, the primary CPU board is treated as a master, and the other boards, such as our clock board, act as slaves.

Data transfers are defined according to the master. A READ bus cycle is defined as a transfer from the slave board to the master board. Conversely, the WRITE bus cycle is a transfer from the current bus master to one or more slaves. The INPUT and OUTPUT lines are similarly defined with respect to the master.

Motherboard: The Motherboard allows bus signals to pass between boards. The 100 signal lines of the motherboard are outlined in Table 2.

S-100 Bus Cycles

The S-100 interface is controlled by several bus signals. Figure 1 shows the relationship between the key signals used in this article. A master clock (called Φ, read Phi), usually generated by the master CPU, controls the timing of the other signals. Those signals involving data transfers are synchronized with the master clock. The clock is divided into three primary bus states (BS1, BS2, and BS3 in Figure 1), with an optional fourth (BSW) called a wait state. Each bus state corresponds to at least one clock cycle. We will now examine the flow of signals, first without wait states.

A bus cycle (see Figure 1) starts with Bus State 1. Here the master asserts its status and address lines. The pSYNC signal indicates the beginning of a bus cycle. The address and status lines become stable towards the end of Bus State 1. The status signals include sINP and sOUT which indicate whether the processor is in a read or in a write cycle (i.e., reading or writing data).

If the bus is in a read cycle, as indicated in Figure 1, the master asserts the read strobe pDBIN. The addressed slave uses this to turn on its data input bus drivers so the master can read the result. The master reads the data during Bus State 3.

If the bus is in a write cycle, the master puts the data on the bus during Bus State 2. After the data becomes stable, the master lowers pWR*. In Bus State 3, the slave will take the data available on the bus.

When the slave requires more time to take or deliver the data, it may tell the master so by driving the RDY signal low, delaying Bus State 3. The clock cycles intervening between State 2 and the now-delayed State 3 are called wait states. Our board requires several wait states, depending on the processor speed.

After completing Bus State 3, the processor moves to its next operation in State 1 of the next bus cycle.

Other signals are available for more complicated transfers, such as 16 bits or DMA; these will not be discussed.

Table 3 lists the S-100 signals used here. In general, the signals starting with 's' are status signals (telling what), and those starting with 'p' are strobe signals (telling when). They are combined to get the read/write qualifier signals.

Address lines	24 lines
Data	16 lines
Status	9 lines
Control output	5 lines
Control input	6 lines
Interrupt + DMA	18 lines
Miscellaneous	22 lines (includes power, reset, clocks, and unassigned)
Total	100 lines (connectors on bottom of card)

Table 2. *Summary of the S-100 bus signal lines.*

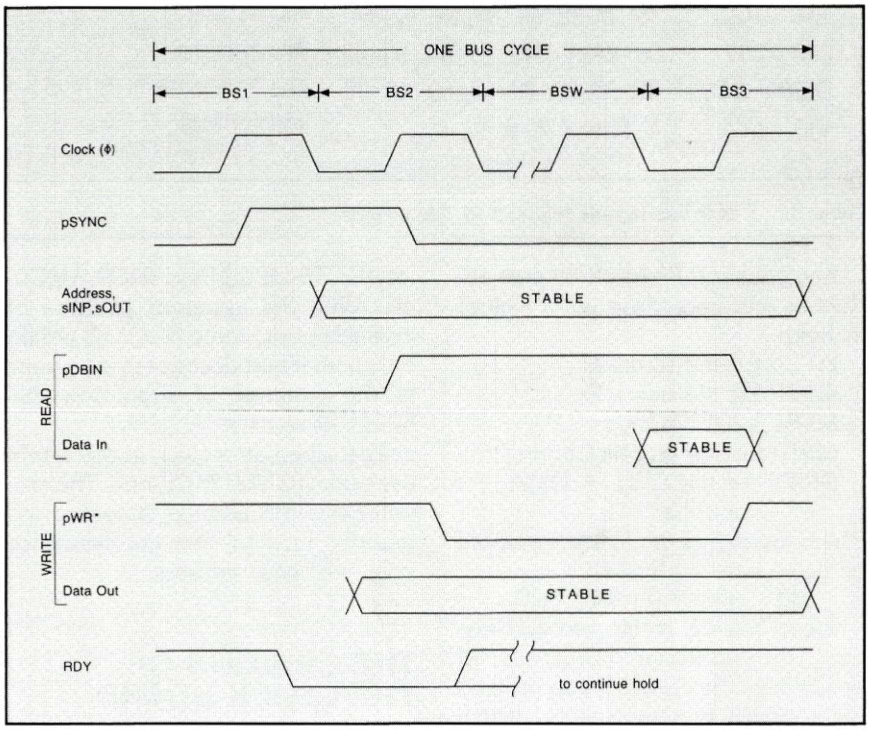


Figure 1. S-100 bus cycle and timing relationships.

HOW TO WIRE WRAP

For those who have never used a wire wrap tool and are unfortunate enough to buy one without directions, some instructions are in order.

Neatness is important. Strip about 1 inch of insulation from the wire, taking care not to weaken the wire. Some tools have a blade in the middle that works great. Prestripped wire lengths make things easy.

Use only as much wire as needed, routing the wire to facilitate changes. Slide the wire up the slot in the tool, place over socket post, and wrap clockwise. The photo on page 22, shows some good wire wraps. Practice until you become good at it.

WHAT YOU NEED TO FIND OUT FIRST

Even though S-100 boards are mostly standard, you will need to find out several things about your computer. First find out the speed of your processor (probably 2, 4, 6, or 8 MHz). Next, you need to determine the number of ports addressable by your system. The answer is either 256 or 65536, corresponding to 8- or 16-bit addresses. More recent systems have 16-bit port addresses. You also need to find out what ports are in use — your CPU uses ports for things like terminals, printers, disk drives, etc.

Make a list of the ports in use. From this list find 16 consecutive port addresses, with the lowest address divisible by 16, to use for the clockboard. The lowest address is called the base address. An example follows.

SIGNAL	PIN NUMBER	FUNCTION
Φ (clock)	24	main system clock (CPU speed)
pSYNC	76	indicates start of bus cycle
sINP	46	status line for read cycle
sOUT	45	status line for write cycle
pDBIN	78	strobe for read cycle
pWR*	77	strobe for write cycle
RDY	72	hold request (slave to master)
+8	1, 51	unregulated power
ground	20, 50, 53, 70, 100	
A0 - A7	79, 80, 81, 31, 30, 29, 82, 83	address lines
A8 - A15	84, 34, 37, 87, 33, 85, 86, 32	address lines
A16 - A23	16, 17, 15, 59, 61, 62, 63, 64	address lines
DI0 - DI7	95, 94, 41, 42, 91, 92, 93, 43	data to master
DO0 - DO7	36, 35, 88, 89, 38, 39, 40, 90	data from master
VI0 - VI7	4, 5, 6, 7, 8, 9, 10, 11	interrupt lines

Table 3. *S-100 bus signals relevant to this project.*

Say your system has 8-bit port addresses and uses these ports (values in hex):

20-26 for the terminal
40-7F for the aux I/O
80-81 for the printer
00-10 for the graphics board
D0-FF for the CPU + DMA controller

Then any one of the following would be good base addresses:

30 90 A0 B0 C0

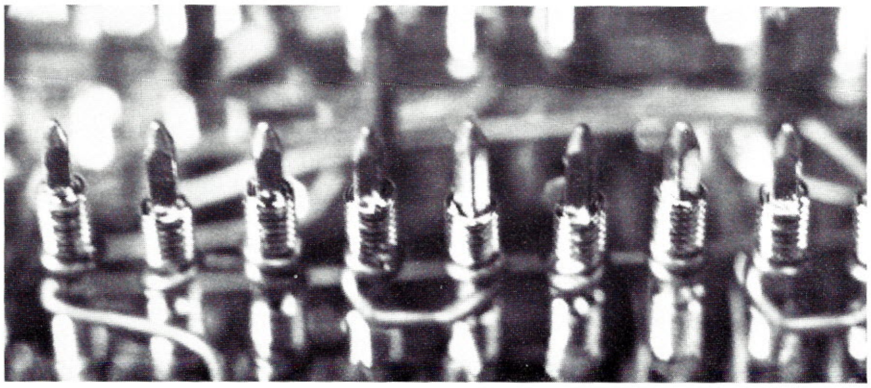
If you have a 16-bit port address, you should have hundreds of possibilities. Be careful if you have an older board in your system which decodes only 8-bit port addresses. For instance, a graphics board which responds to port address C0 (hex) with 8-bit decoding will also respond to the 16-bit address 01C0, 02C0, etc. With the apparent plethora of available ports, some designers will do only partial port decodes to save parts at the expense of large unusable blocks of ports.

The schematics given in this article use ports E070 to E07F hex. The section below on address decoding will describe how to alter the circuit for your port base address.

THE CIRCUITS OF THE CLOCK BOARD

To simplify matters, I am describing the project in sections: the power supply, the address decoding, the read/write logic, the bus interface, the clock function, and the wait state generator. Knowledgeable designers can probably come up with other circuits that will do the same job with less parts or with some other design criteria, but the purpose here is to present the parts logically so the reader may learn.

Figure 2 shows a block diagram of the whole project. The 5-sided symbol indicates S-100 bus connections and the direction of data flow. The address lines are split into two groups. The 4 least significant bits (A0-A3) go directly to the clock for function selection. The most significant bits (A4-A15) go to an address selection circuit to develop the board-select signal (BDSEL*). The input and output data lines are combined onto an internal bus with 4 least significant bits going to the clock. These are used to read and write time digits. Various read and write qualifiers are developed by miscellaneous logic on the board. The various sections are described below and detailed in Figures 3 through 7.



Examples of good wire wrapping.

Power Supply

Figure 3 shows the schematic for the power supply. This section supplies a constant 5 volts to the other chips on the board, for current loads up to 1 amp. This is the only voltage we need. A heat sink under the regulator prevents overheating. The S-100 prototype board (from Vector Electronic) that I used has special traces and comes complete with heat sink, nuts, and bolts.

The other schematics will refer to power as 5V and to ground as the inverted triangle symbol.

Address Decoding

Above you were asked to determine if your system uses 8- or 16-bit port addressing. Two alternative circuits (Figure 4 and Figure 4a) are given here, one for 16-bit addressing and the other for 8-bit. Both circuits develop an active low signal when all but the 4 least significant address lines match a preset address. This preset address is called the base address because any one of the 16 addresses starting with the base address will

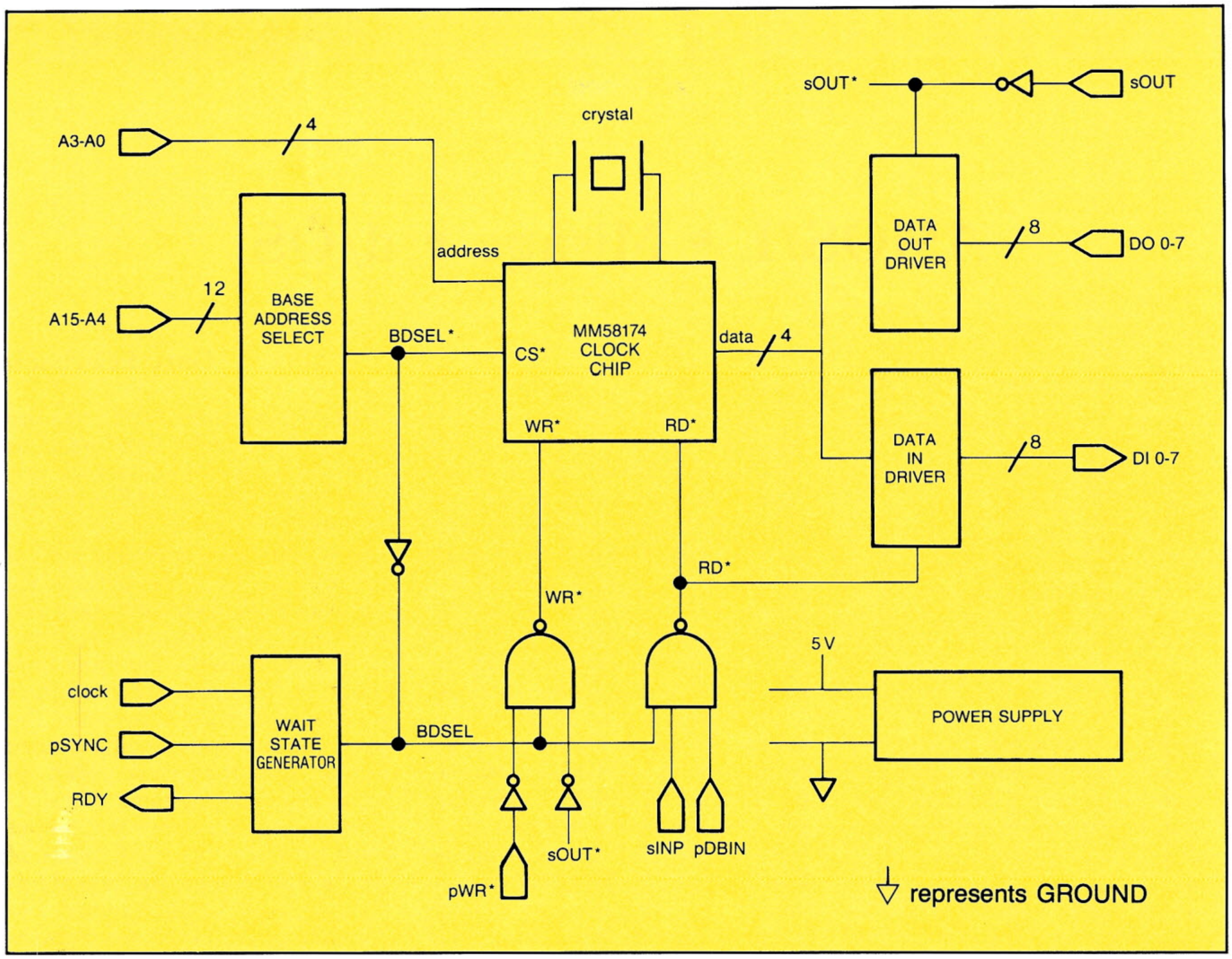


Figure 2. *Block diagram of the circuits required for the clock board.*

access the board. For example, in an 8-bit port addressing system, if the preset address values to be matched by address bits A7-A4 is 0101 binary, then the base address would be 50 hex and all addresses from 50 to 5F hex would generate the board-select signal. Similarly, a 16-bit port address would have a 12-bit preset address which would indicate the base address for 16 consecutive addresses.

Let's look at a specific example. If you have a 16-bit port address, the circuit shown in Figure 4 decodes the 12 most significant address lines to generate a board-select signal (BDSEL* active low). The base address is E070. Note how pins 15-23 determine the most significant bits (111000000 respectively). Chip U4 requires 3 high inputs to develop the ENABLE* signal. When A6-A4 are high, and address lines A15-A7 agree with the preset address, the BDSEL*

(continued on page 46)

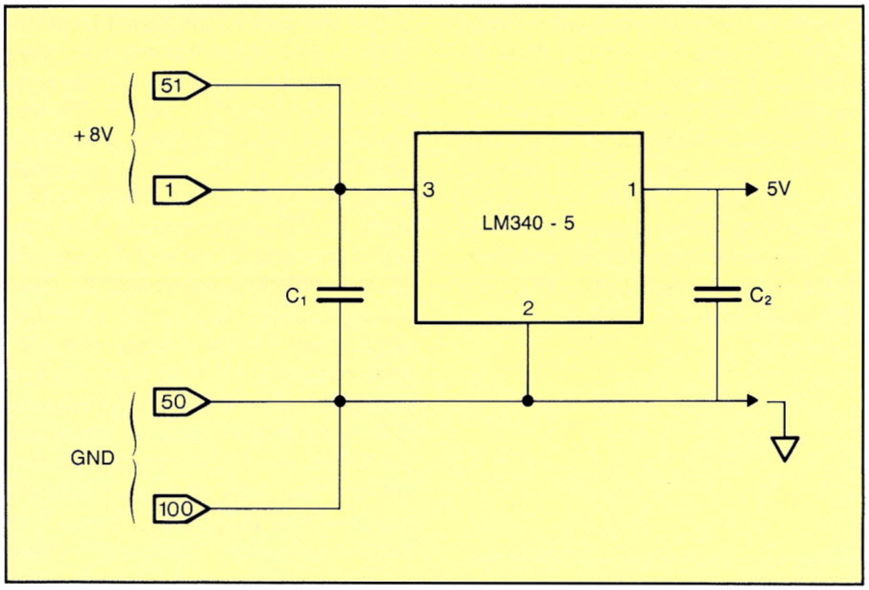


Figure 3. *Circuit for the on-board power regulator.*

multiuser os

AMOS/L • Mirage • d/OS

Gary Feierbach

Multiuser OS is the column that discusses all multiuser operating systems running on S-100 machines.

Gary Feierbach, our regular columnist for Multiuser OS, has over 20 years of experience as a computer professional and holds B.A. and M.S. degrees in Computer Science and E.E. from the University of California at Berkeley. He has worked in numerous operating-system environments and is actively providing implementations of various multiuser operating systems on S-100 machines. Gary is president of Inner Access Corporation, one of those intrepid companies that is ready to offer new S-100 products forever.

Readers are invited to write to Gary and send questions, tidbits of information, or gossip about any S-100 multiuser operating system. He will try to work them into future columns. Write to Gary Feierbach, P.O. Box 888, Belmont, CA 94002.

As I mentioned in the previous issue, many operating systems owe their lineage to the Digital Equipment RT-11 operating system. This was a sophisticated operating system for the PDP-11 line of minicomputers that allowed users to time-share the CPU and, at the same time, permitted access to all the peripherals and software facilities.

Three very similar, RT-11-derived systems for the 68000 processor are worth noting: AMOS/L, Mirage, and d/OS. The AMOS/L operating system is available from Alpha Micro Systems Inc., Mirage is available from Sahara Ltd., and d/OS is from d-Soft. When compared with each other, the performance of the three systems is very similar. All three make extremely efficient use of the CPU and do not require memory management hardware.

A short comparison of AMOS/L running on an 8-MHz 68000 on the S-100 bus with UNIX running on the VAX 11/70 will put these operating systems in perspective. Our benchmark involved the use of 24 terminals and was CPU intensive (very little disk I/O). The terminal response time under UNIX on the VAX 11/70 was almost 2 seconds, but it remained under 0.5 seconds on the 68000 under AMOS. Two seconds can be an eternity when the application involves form completion, common in most business applications. The situation improves a little in the area of disk activity since the UNIX overhead is perhaps only twice the number of executed instructions as AMOS/L. Here, disk and controller hardware, disk caching algorithms, and memory allocated to cache can make as much as a 100-to-1 difference versus a less than 2-to-1 difference for the operating system.

Any multiuser operating system has to check each user periodically (usually on an interrupt triggered by a timer) to see if there is any activity. The more users, the more time spent by the CPU switching from user to user, even if several of the users haven't done anything since the last time they were polled. In some operating systems (Unix included), terminal input and output must go through several layers of logic to check, translate, or perform some special function when encountering certain groups of one or more characters. Both user polling and terminal I/O have been greatly simplified in the AMOS/L-like systems without sacrificing the versatility required in business and industrial environments.

In AMOS/L-like systems, all system functions and the user programs are RAM resident. UNIX, on the other hand, is continuously swapping pieces of the operating system with the user programs, adding enormously to the system overhead.

Table 1 shows some advantages of AMOS-like systems over UNIX. Since UNIX and its variants will be treated in future columns, the good, the bad, and the ugly of UNIX will be saved for your future reading pleasure.

Now that I've made these AMOS/L-like operating systems look good, you may be wondering which one is best for your applica-

UNIX	AMOS/L
CPU Intensive	Low CPU overhead
Requires memory management	No memory management required
Complex to use (Shell required for most business applications, adding to overhead)	Easy to use (No shell required)
Adding drivers difficult	Add drivers extremely easy
Requires 256K RAM	Requires 128K RAM
Requires 20Mbytes disk	Requires 1Mbyte disk
No general resource locking	File and record locking
Cannot force modules resident	Easy to make modules resident
Major portions of O/S not resident	O/S entirely resident
No contiguous random access supported	Contiguous random access files supported
Limited to no task (user) communication	Intertask and interuser com. easily accomplished

Table 1. *General features of Unix compared to AMOS/L and similar operating systems.*

tion or environment. Since their performance is so similar, we need to compare features to distinguish them. There is a danger in doing so since new versions are in the works that are likely to change these comparisons. Perhaps I can include news flashes in future columns to keep you up-to-date.

Table 2 summarizes the discussion that follows.

There is virtually no software limit on the number of users these systems can accomodate. Many AMOS/L and Mirage systems have in excess of 100 attached users. In such cases, disk average access times and the size of disk cache memory are the most critical factors limiting the number of users. Most business applications are not processor intensive, so the speed of the processor is secondary to disk speed.

There is virtually no software limit on the number of users these systems can' accomodate

All three systems allow partitions to be of any size up to the memory limitations of the hardware. In addition, memory can be shared through the use of reentrant software that many users can use simultaneously. Mirage allows the operating system itself to be extended through the use of 'bolton' routines which provide system calls that can be shared by all users. Both AMOS/L and d/OS use 16-bit file pointers, limiting file and logical disk drive sizes to 32Mb (allocation in 512-byte blocks). Mirage, on the other hand, uses 32-bit file pointers, allowing the use of files and logical disk drives up to 4Gb. With laser disk technology arriving on the scene, even this size may not be adequate for long.

All three systems have two-level directories. After getting lost in tree-structured directories in UNIX and MS-DOS 2.0 and above, I have come to love this limitation. Oddly enough, if I had more than two levels, I would use them to my own detriment and to the detriment of others that follow in my footsteps. Is this a case where more is not necessarily better? Two levels appear to be sufficient as long as a user can be associated with a group of directories for access security purposes (as in Mirage and d/OS).

File names in AMOS/L and d/OS are 6 characters long, with 3-character extensions. This is rather limiting in large systems since extensions are preempted in most cases to designate various file usages. Mirage provides an 8-character file name and a 4-character extension which is a profound improvement. If they had made it a 9-character file name and 3-character extension, it would have been even better since extensions are again usage oriented, and 3 characters are generally sufficient for that purpose.

Each of the three systems supports command files that are flexible enough to be used as a shell to provide an applications environment. However, all three allow the user to exit to the operating system. AMOS/L and d/OS have additional shell generating packages for making shell menus that allow no escape from an applications enviroment. This provides additional protection against users, and the user is prompted in a familiar environmental envelope.

Security differs on each of the three systems. AMOS/L has the least security with only password protection in each directory. Once into the system, the other directories may be accessed if the user knows their passwords. Additional protection can be provided by using the AMOS/L shell program to create a user environment that doesn't allow the user to escape to the operating system. In addition, d/Soft offers Tass (the security package used with d/OS) for AMOS/L. Mirage has password protection and four access levels: read only, execute only, group access, and supervisor. D/OS has password protection and three

AMOS/L and d/OS have shell generating packages for making shell menus that allow no escape from an applications environment levels: specific account access, group access, and supervisor.

The three systems feature a HELP subsystem that provides useful information about commands and programs. D/OS has a multilevel HELP system that allows a user to select the level of detail that he or she desires.

Multitasking is now supported by the three operating systems. This permits a user to start one task and to operate it concurrently with another task. The output can be directed to a pseudoterminal which can be redirected at will. AMOS/L has a windows utility in beta test which should augment the multitasking and diminish the problem of where to redirect task output. Full-screen window support is already offered by d/OS as part of multitasking.

Multitasking is now supported by the three operating systems

The sophistication of the printer spoolers is something to behold. All allow multiple printers. And all use disk queues to maintain a record of the files not printed even in the event of power failure. The available options fill three typewritten pages and include the ability to delete files from the queue, make multiple copies, print a range of pages, print banner pages, delete files after printing, set file printing priorities, and so on.

An area of some importance is disk caching. Disk caching, if it is the look-ahead variety, can often buy a substantial increase in speed even for sequential files. Look-ahead caching will read a number of records, or even a full track of information, in anticipation of it being likely to be used. Many extravagant claims are made for disk caching, including up to 20-fold improvements in throughput. The truth is that startling improvements can be observed, but it depends a great deal on the native hardware speed and nature of the processsing being done. I have personally observed improvements in the range of 100% to 400% for look-ahead caches. I have never observed a degradation although this is possible in theory.

All three operating systems offer caches but none are the look-ahead variety (a look-ahead cache system is available from an independent supplier for AMOS/L, and it can be used to advantage in conjuction with the operating system cache). The caches supplied can be turned on or off, and selected files and directories can be locked into cache. Also, the caches can be set to read-only so that writes bypass the cache mechanism. In practice, disk caching is very memory hungry. The requirement for memory goes up almost linearly with the number of users unless there is a great deal of commonality in the data used by each. But, with memory prices tumbling to less than $500/megabyte, this should not be a concern. Cache memory carried to an extreme becomes RAM-disk.

A RAM-disk as a formal device is supported by both Mirage and d/OS, and is available from a third party for AMOS/L. This simply declares a chunk of RAM as a disk drive, and the operating system treats it exactly that way. Files may be copied to and from it, and it has the same two-level directory structure. Utilities are provided to save and restore RAM-disk to a real drive. RAM-disk is particularly useful in an environment with only floppies or with slow winchester drives.

Although all three systems use drivers that are independently assembled and linked to the operating system using DEVICE commands, Mirage is truly unique in allowing the user to attach new devices at any time. This is a system builders dream come true. In AMOS/L and d/OS the DEVICE commands must be part of the system-initialization file used at bootup. Even the CRT commands (escape sequences for cursor positioning, etc.) within each system are all generic and are translated by means of CRT drivers . Each system comes with a great number of these drivers for most popular terminals. In any case, they are trivial to write.

Mirage is truly unique in allowing the user to attach new devices at any time

Now to strike the hot topic of our day — LANs (Local Area Networks). Ethernet is available for all three (on S-100). This is the high-priced spread which costs about $1000/node plus the cost of cable which runs high for this technology. AMOS/L also has its own Alphanet using its proprietary video board which also doubles as a VCR backup device. Mirage has an IEEE-488 network support package, and an independent vendor has an ARCnet package planned. ARCnet costs only $500/node but is about half of the bandwidth. This turns out not to be too severe a limitation since Ethernet, when heavily loaded, wastes much of its bandwidth on retransmissions. The coaxial cable used for ARCnet is also much less expensive. These networks can be used to communicate with a mixture of other like computers and IBM compatibles. The IEEE-488 network can be used to communicate with a variety of instruments as well.

All three packages include a Basic compiler, an assembler, and a host of utilities

For those wanting to use IBM compatibles as terminals and still have access to Sidekick and other PC software, both AMOS/L and Mirage offer packages for this purpose. Alphamate and Planet must be evoked by name while in PC-DOS (MS-DOS) and provide terminal emulation and file transferring capabilities between the PC and the S-100 system. Mirage's MITS package provides the same kind of capabilities but with a distinct difference: it is a resident program evocable with a single key stroke. This may not seem significant until it is realized that it allows you to be running a program on a PC, running a program on Mirage, switching back

FEATURE	AMOS/L	Mirage	d/OS
Latest version	V1.3	V1.2.300	V0.9(7)
Number of users	>100	>100	>100
Maximum partition	16Mb	16Mb	16Mb
Largest file	32Mb	4Gb	32Mb
Largest logical disk drive	32Mb	4Gb	32Mb
Directory levels	2	2	2
Command files	yes	yes	yes
Shell support	yes	yes	yes
Multitasking	yes	yes	yes
Windows	in beta test	none	yes
Multiple printer spooling	yes	yes	yes
Disk cache	yes	yes	yes
RAM disk	yes*	yes	yes
Adding drivers	system boot	any time	system boot
Device independence	yes	yes	yes
Access security	passwd (1 level)	passwd (4 level)	passwd (64K level)
Help subsystem	yes	yes	yes
LAN available	Alphanet Ethernet*	Ethernet IEEE-488 net	Ethernet*
Comm. software to PCs	Alphamate, Planet*	MITS	Planet*
Electronic mail	in beta test	in beta test*	within system
RJE support	2780, 3780	none	none
Compilers available	AlphaBasic SMC Basic Fortran 77 (IEEE) Alpha C Whitesmith C* UCSD Pascal* Alpha COBOL	SW-Basic* SwifteBasic Fortran 77 (IEEE) Lattice C UCSD Pascal* Forth 83* Micro APL	SW-Basic Fortran 77* Whitesmith C* UCSD Pascal*
CP/M-80 support available	AM330 board	Newtons board*	Micro Concepts*
MS-DOS support available	none	Newtons board*	none
Floating Point hardware	Sky board*	planned*	planned*

*Available from independent suppliers

Table 2. *Comparative features of the operating systems AMOS/L, Mirage, and d/OS.*

and forth watching them run, pop into Sidekick (while in either environment) and record an appointment on the appointment calendar, make some notes, then return to the two jobs (one on Mirage and the other on the PC) to see how they're progressing. We currently have MITS and Sidekick coresident. One programmer typically prepares programs on an editor on the PC, ships them to Mirage, pops up his notepads and appointment calendar when answering the phone, then returns to his program editing and testing. It is an extremely impressive tour-de-force.

Electronic mail is a feature of UNIX and has a heritage from the large timesharing networks like the DARPA-sponsored ARPANET. Electronic mail allows users on all the systems in a network to send messages to each other. Both Alpha Micro and Network Nexus (a supplier of Mirage software) have electronic mail packages in beta test. Additional communication software (and hardware) is available from Alpha Micro for RJE (Remote Job Entry) support emulating the IBM 2780 and 3780 workstations.

A number of compilers are available for both Mirage and AMOS/L (see Table 2). D/OS is in the weakest position with only third-party compilers from Softworks. Alpha C is a UNIX-compatible version and Alpha COBOL is ANSI 77 and RN COBOL compatible. SW-Basic, by Softworks, is Alpha compatible allowing AlphaBasic programs to be compiled under Mirage and d/OS. The IEEE version of Fortran 77 uses the IEEE-standard floating-point format for floating point numbers. All of the Mirage compilers use the same IEEE-format floating point via a reentrant bolton routine called IEEEFPAK. This will make it extremely easy to switch to hardware floating point using the 68881 chip. Such support is planned for Mirage in the near future. Alpha Micro uses a S-100 board made by Sky Computers Inc. that supports floating point for the Fortran 77 compiler.

What are the two most frequently asked questions? 'Will PC-DOS (MS-DOS) programs run on this operating system?' For the second most frequently asked question, substitute CP/M for PC-DOS. AMOS/L supports a S-100 board (the AM330) with a Z-80 which will run CP/M-80 programs while AMOS/L acts as a file and printer server. Newtons Laboratory has a board that sports both a Z-80 and a 8088 processor. This S-100 board will run CP/M-80 and PC-DOS programs and use the Mirage operating system as a file and printer server. Only one user is supported per slave card, but it allows the attached user to run in all three enviroments. It is purported to be much faster than TurboDOS which is not surprising since the master is a 68000.

Price and availability: The approximate retail prices (on S-100 systems) for each of these packages are $1500 for AMOS/L, $995 for Mirage, and $995 for d/OS. All three packages include a Basic compiler, an assembler, and a host of utilities. Mirage without the Basic compiler is $695. Both Mirage and d/OS are available to run on any 68000 hardware for a very modest port charge. AMOS/L is a separately priced product for Alpha Micro hardware, and the manufacturer has been reluctant to see it run on other hardware. This policy may change in the future since it is bound to sink in that popularizing AMOS/L can only help their sales in the long run.

All three are fine operating systems and seem to be relatively bug free. ■

ADDRESSES

d/Soft
310 Cedar Lane
Teaneck, NJ 07666
(201) 652-1413

Sahara Ltd.
Unit 1F
Tideway Industrial Estate
87 Kirtling St.
London, England, SW8 5BP
(01) 627-1733

Alpha Micro
3501 Sunflower St.
Santa Ana, CA 92704
(714) 957-8500

INSTALLING A RAM-DISK IN A CP/M 2.2 SYSTEM

Leonard C. Schwab

The installation of a RAM-disk is a quick, simple, and cost-effective way to upgrade the overall performance of an older S-100 computer. Modern systems will benefit as well. However, unless the RAM-disk board comes with the proper software for your system, the installation process may be difficult. And even if you have factory-supplied drivers available for your RAM-disk, you may want to add features and capabilities not included with the stock drivers.

I have written this article to share my experiences installing and using a CompuPro M-Drive/H 512K RAM-disk board in an IMSAI VDP-80 computer. The information should also be useful to owners of other S-100 systems. The article describes the software necessary to drive the M-Drive/H. It includes listings of a skeletal CP/M BIOS, a RAM-disk formatting program, and a RAM-disk loading batch procedure.

RAM-DRIVE CONCEPTS

A RAM-disk, or RAM-drive, is a block of read-and-write memory that is made to appear to the system as if it were a floppy disk drive. RAM-disks are also called 'disk emulators.' To the user, this storage device acts precisely like a mechanical disk drive but at RAM speed.

There are three fundamental approaches to creating a RAM-drive. We could simply set aside a block of the system memory. This is the method most often used by RAM-disk software in the MS-DOS environment. However, in a non-banked eight-bit system there is simply not enough memory available to make this approach practical.

The second method of implementing a RAM-drive involves bank-switched memory or extended-address memory beyond the first 64K available under CP/M 2.2. This approach is possible only if your main processor board already has the bank-switching or extended-address capability. Furthermore, it is less than satisfactory because the burden of managing the RAM-drive memory is placed on your main processor.

The third approach, used by the M-DRIVE/H board, SemiDisk, and others, is to make the RAM-drive memory accessible to the main processor by way of one or more I/O ports. This method allows the RAM-drive board to handle most of the chores involved in managing the RAM-drive.

OBJECTIVES OF RAM-DRIVE INSTALLATION

The objective of installing RAM-disk is of course to increase disk-access speed. In my particular case, this is what I wanted to accomplish and, ultimately, was able to achieve:

1. Speed up the execution of application programs that make frequent accesses to disk. This includes programs that store and read data and those that use overlays, like compilers, WordStar and Supercalc.

2. Accelerate the operation of SUBMITted procedures. I use SUBMIT extensively to chain related programs, especially when coding and testing new programs (see reference 1).

3. Reduce warm-boot time. Under the Fischer-Freitas version of CP/M 2.2, which I use, warm-boots are very slow. Also, since a warm-boot occurs frequently during SUBMIT procedures, this alone would improve the performance of SUBMIT.

4. Operate the system, after an initial loading sequence, with **no** floppy disks mounted, thus reducing the wear and tear on my valued, venerable, and expensive PerSci drives.

5. Learn more about the computer and its operating system.

THE IMSAI VDP-80

My VDP-80 has been in constant use since 1978. It is an integrated system, weighing almost 100 pounds, and includes a 10″ CRT display, keyboard, dual 8″ PerSci floppy drives, and a 7-slot S-100 card cage (pre IEEE-696 specification). The main processor is an 8085, running at 3 MHz. Originally equipped with 32K of dynamic memory, the machine has been upgraded to 64K of CompuPro static memory.

I found a substantially discounted M-Drive/H board at a West Coast Computer Faire and purchased it. I have never regretted that purchase. Today, the same board is readily available for about half of what I paid, making the RAM-drive even more desirable.

COMPUPRO'S M-DRIVE/H BOARD AND SOFTWARE

The M-Drive/H board consists of 512K bytes of dynamic RAM, a dynamic RAM controller circuit, and electronics for addressing the data in the RAM array. Multiple boards may be installed in a system.

The M-Drive appears to the CPU as a pair of I/O ports. It creates no bus-assignment problems and uses no memory space. The M-Drive/H is well designed, well made, and well tested.

Technical information supplied with the board is skimpy. On-board addressing is described in a general fashion in a couple of sentences, but the precise software interface is never explicitly explained. There is nothing in the manual explaining how to format the board at power-up, except for an implication that formatting will be necessary.

INDEX OF LISTINGS DESCRIBED IN THIS ARTICLE

Listing 1 — Page 37
Listing 2 — Page 59
Listing 3 — Page 58

Software programs are available from the vendor which will patch RAM-disk driver routines into a non-CompuPro version of CP/M 2.2 and format the RAM-disk on power-up. With this software, the RAM-disk appears as the M: logical disk.

However, use of this software would not satisfy one of my important objectives - running SUBMIT procedures from the RAM-disk. CP/M 2.2 will run SUBMIT from the A: logical disk only (see reference 2, pp 127-128). I needed the ability to make my RAM-disk appear as the A: drive in my system. Furthermore, the CompuPro drivers eat up 1K of working memory, which I could ill afford. This meant that I had to write my own drivers.

My first task was to write a stand-alone program, in assembly-language, to be a test-bed for routines. Fortunately, the manual includes a set of 'sample read/write routines' without which it would have been impossible to proceed.

After learning how to access the M-Drive, how to manage the addressing function (track and sector), and verifying that I could retrieve what I had written, I set about designing the BIOS modifications that would provide the functions that I wanted.

RAM-DISK SUPPORT ROUTINES

In CP/M, procedures that are related to the specific hardware installed in a system are contained in the BIOS (Basic Input-Output System). General procedures that manage the filing system are contained in the BDOS (Basic Disk Operating System). In most cases, it is unnecessary and undesirable to change the BDOS.

Listing One shows those parts of the BIOS affected by the presence of the M-Drive/H. I omitted the details of procedures that are dependent upon other hardware devices (console, printer, floppy disks, etc.).

Salient features of the RAM-disk software design follow:

Warm Boots

In order to be able to warm-boot from the RAM-disk, a copy of the CP/M 2.2 system image (CCP and BDOS, 5632 bytes total) must be stored on the RAM-disk. The first six tracks (6K bytes) of the RAM-disk have been reserved for that purpose.

When the RAM-disk is initially formatted, a copy of the system image is read from the floppy into the RAM-disk (see Listing One, Cold-Boot section). Subsequent warm-boots read this image into memory virtually instantaneously, shaving several seconds off the warm-boot time. (see Listing One, Warm-Boot section).

Access to Tracks, Sectors, and Memory (DMA)

When data on a disk is to be accessed, the BDOS first asks the BIOS to set the disk number, track number, sector number, and memory buffer address. The BIOS is responsible for telling the physical device, usually a floppy disk controller, how to set up for the access. The device is not actually accessed until BDOS sends a read or write request to BIOS. The disk number, track number, and memory buffer addresses are reset for subsequent accesses only when necessary.

To set up a track for access, the BDOS sends a two-byte physical track number to the SETTRK (set track number) routine in the BIOS. I found that the BIOS in my system only used the low-order byte of the track number, assuming that no drive would have more than 256 tracks. The RAM-disk, however, is organized as 512 tracks. Each track contains 1K bytes, divided into eight 128-byte sectors (see Listing One, RAM-Disk-Data section). Obviously, SETTRK had to be modified to use all 16 bits in the track number, in order to be able to deal with the 512 tracks (track numbers 0000H through 01FFH) on the RAM-drive.

No modification was required for the BIOS routines named SETSEC (set sector number) or SETDMA (set direct memory access address). (See Disk-Address-Pointers section).

Other disk-related functions — SELDSK (select disk), READ (read from disk), and WRITE (write to disk) — must include a test to determine whether a floppy disk or RAM-disk is being accessed and must include appropriate branching instructions (see Select-Disk and Disk-I/O Sections).

Sector Translation

In order to maximize speed when accessing floppy disks, the logical sectors are not stored on the disk in sequence. A technique known as interleaving is used which requires the BIOS to convert the logical sector numbers sent by the BDOS into physical sector .numbers appropriate to the interleaving scheme in use by the system. This conversion is known as sector translation. It is the responsibility of the SECTRN routine in the BIOS.

Because the RAM-disk is not a rotating device, like a floppy drive, no sector translation is necessary. Hence, SECTRN must be modified to include an escape branch when the RAM-drive is being accessed (see Sector-Translation section).

Cold Boots

The code in Listing One will generate two separate versions of the BIOS. The version generated is determined by the setting of compilation-time switches RAM$A and RAM$C. In one version, which I call RAMCSYS, the floppy disks are designated A: and B:, the RAM-disk is C:. This version is generally operative only during the initial formatting and loading of the RAM-disk after a power-on. In the second version, RAMASYS, the RAM-disk is A:, and the floppies are B: and C: respectively.

The two separate versions are necessary in order to have a system that will boot from a floppy (RAMCSYS) and another system that will support batch processing with SUBMIT from the RAM-disk (RAMASYS).

The cold-boot routine in the BIOS must include a subroutine that moves a pristine copy of the system from high-memory into the reserved area of the RAM-disk. This copy will be pristine because it will be moved after it is read from the disk but before the CCP is entered (see Cold-Boot section).

Formatting the RAM-disk

It is necessary to write a format program for the RAM-disk (see Listing Two). This program is run once, and only once, immediately after the system has been powered up and cold-booted from a floppy disk.

It is important that this program **not** be autoloaded on cold-boot. It must be manually started. The format program will destroy any data on the RAM-disk, and you want to retain the capability of doing a cold-boot, to recover from some error, without reformatting the RAM-disk.

The format program places

'no-data' symbols (0E5H) into the directory area on the RAM-disk. It is immaterial whether or not the remainder of the RAM-disk is formatted, but it is important not to overwrite the reserved area in which the system image has been stored.

Parity checking

Although the CompuPro manual shows a method for parity-checking all RAM-disk I/O, I decided to omit that technique. In more than a year of constant use, I have observed no I/O errors during RAM-disk accesses. In any case, the presence of parity checking will only prevent access to what is probably substantially correct data.

ASSEMBLING AND INSTALLING THE SYSTEM

In the skeletal BIOS code (Listing One, RAMSYS.ASM), elements which are in uppercase are specifically applicable to RAM-disk operations. Other code, in lowercase, is presented only for the purpose of indicating the context of the relevant code. The only relevant items in RAMSYS.ASM that are system-dependent are the MSIZE and NDISKS constants. The code is written for Digital Research's ASM.COM assembler and must be edited into the source code for your system's BIOS before assembly.

Some method for installing the modified BIOS into memory will be needed. This will vary from system to system.

The Fischer-Freitas version of CP/M 2.2 includes a transient program, LDBIOSxx.COM (where xx designates the size of the system). LDBIOS is autoloaded during cold-boots by special code in the CCP-BDOS image, but it may be executed by the user at any time to bring a new BIOS into memory.

In order to load the two versions of my RAM-drive BIOS, I created two separate versions of the LDBIOS program. One is named LDBIOS56.COM, so that it will be found and loaded by the cold-boot routine. This version loads the RAMCSYS system. The second version of the BIOS loader is named RAMASYS.COM. Obviously, it loads the RAMASYS system and makes the RAM-disk appear to be the A: drive to the system.

For other systems, you must find a way to install the RAMCSYS into your cold-boot loader. A method for doing this is described in the CP/M 2.2 manual, Section 6, 'CP/M 2 Alteration.'

If you want to follow the two-system approach described above, you will also have to write a simple loader to replace the RAMCSYS BIOS with a RAMASYS BIOS for work purposes. Alternatively, a program may be written which patches the BIOS code in memory, thus converting from one BIOS version to the other. The sections of code to be patched are those which are bracketed by IF and ENDIF in Listing One.

USING THE RAM-DISK

I have adopted the following operating procedure when using the RAM-disk:

1. Power on.

2. Cold-boot. RAMCSYS BIOS is loaded. The RAM-disk is C:.

3. Run batch file RAM.SUB (Listing 3). RAM.SUB performs the following tasks:

a) Runs the RAM-disk format program, RAMFMT.COM.

b) Transfers general utilities from the boot-disk to the RAM-disk.

c) Runs the BIOS loader RAMASYS.COM which installs the final working BIOS, making the RAM-disk appear as A:. At this point, the two floppies, A: and B:, are converted to B: and C:, respectively.

In my system, the above procedure takes approximately 100 seconds. This includes transferring about 42K of utility software to the RAM-disk. The system may now be used without floppy-disk support, if desired.

VERY IMPORTANT. Before turning the power off, any new or edited files on the RAM-disk must be transferred back onto a floppy. Any data on the RAM-disk will be lost when the power is removed. Frequent transfers of modified data from RAM-disk to floppy are a prudent operating procedure.

Because the RAM-drive memory is physically separate from working memory, it is possible to cold-boot the system or to load and run versions of CP/M that don't support the RAM-disk. Obviously, if an operating system does not support the RAM-disk, it will be impossible to access it. But, as long as the RAMFMT program is not rerun and the power is not turned off, the RAM-disk data may be made to reappear merely by running the RAMASYS or RAMCSYS system loaders.

The installation of the M-Drive/H has given my old system many more years of very useful life. I hope that this case-study will help you achieve similar benefits. ■

References:

1. Schwab, Leonard. Minimizing the Inconvenience of Compiled Languages under CP/M. *Microsystems,* May/June 1982.
2. Cortesi, David E. *Inside CP/M.* Holt, Rinehart and Winston, 1982.
3. *CP/M Operating System Manual.* Digital Research, 1982.

(continued from page 7)

poorly, if at all, supported by the press. We want to be the magazine that gives you otherwise not available information on these operating systems and their related software.

Then there is the question of accepting ads on MS-DOS software. I suppose that theoretically it might be possible to accept these ads and maintain a nonMS-DOS editorial orientation, but in practice this is unlikely. As the magazine would depend more and more on such ads as a source of revenue, we would become severely prone to run reviews, and soon major articles, on the products advertised. Next, to widen the sales base of our advertisers, other hardware would little by little find its way into the magazine and then good-bye S-100 Journal. We've all seen many cases of magazines that sacrifice quality and editorial orientation in order to please financial advertising interests. This, we will not allow to happen to S-100 Journal, so it seems best to cut the problem at its roots.

We are neither hiding nor pretending that IBM and its compatible family of hardware don't exist. We simply have chosen to support a different class of machines. The copy-IBM mania is a loosing game with negative feedback effects. The S-100 bus needs to establish itself in the mass-market not by imitating lesser systems but by virtue of its own superiority. • *Jay*

Operating Systems for the 68000

I had to write to say how delighted I am with the quality and content of your magazine. The first issue was *deja vu*, as I had put together a CompuPro system very similar to the one presented. My motivations were to stop buying outgrown all-in-one machines, to have the power of a 'micro-mainframe,' and to get on board (no pun intended) with the 68000 microprocessor. Putting my own system together was a very satisfying and economical approach. And it has been gratifying to see the computing community coming around to the rational 68000 architecture after fully appreciating how bad the IBM-supported Intel approach is.

Prior to the advent of S-100 Journal, the quest for information on S-100 hardware and software had been maddening. I have subscribed to so many magazines in my attempt to find info that I may just be catching up with this year's issues when I retire. For example, I have yet to even see mention of the highly-impressive S1 operating system in any other magazine. Your *Multiuser OS* column is a welcome addition. Though I'm a single user, the column covers multitasking OS's which are highly sought after. I very much look forward to escaping the confines of CP/M-68K to a REAL operating system such as S1, from a company which is interested in supporting its customers.

Keep the information flowing. You have a lot of devoted subscribers who appreciate the value of the S-100 standard.

Richard Sims
Arlington, Massachusetts

I use CP/M-68K in my main system and, like you, I have been looking for something more substantial for the 68000. I had a few very positive contacts with the people from Multi Solutions (who sell S1) and at one time I was ready to write a full article on S1. However, they want to sell it only to OEMs in large quantities, so I lost interest. I still don't understand why a company will willingly restrict distribution of its operating system when it is obvious that, if it becomes popular among users, it will eventually attract the attention of the OEMs. In the case of S1, they have it there fully configured for the CompuPro 68K. All they'd need to do would be to make copies, ship them out, collect the money, and hire a person for telephone technical support.

Presently I'm exploring THEOS and OS-9, both now available for the 68000. I've tried UNIX a few times, and I have to agree with those that say it is too cryptic. The operating systems that Gary describes in this issue's Multiuser OS *are another alternative.*

The problem with all these operating systems is that they ask $1,000 or more for them. I'm just not convinced into paying that much for using the operating system with one or two users. The first of these mainframe-like, user-friendly, multitasking operating systems that sells for under $300 will become the standard. This has happened over and over with software, from CP/M to Borland's Turbo Pascal, and still they fail to grasp it. The least they could do would be to charge on a per-user basis. For example, a system with 1 or 2 terminals would pay $250 for the OS, while a system with 6 terminals would pay $750.

In the meantime, I'll stay with CP/M-68K (found some more nice software for it - a review coming up as soon as I can fit it in). CompuPro has a new improved version, Revision 1.3, of CP/M-68K. The new revision includes a full C compiler (with floating point) and supports their SPUZ Z-80 slave processor. They ask $125 (too steep!) for the upgrade. To get an upgrade, call CompuPro (415-786-0909) and ask for an RMA number to send back your old CP/M-68K. • *Jay*

Zenith H/Z-100

Please have articles on applications to the S-100 bus as implemented in the Heath/Zenith H-100. This is **my** machine and what I want things to apply to.

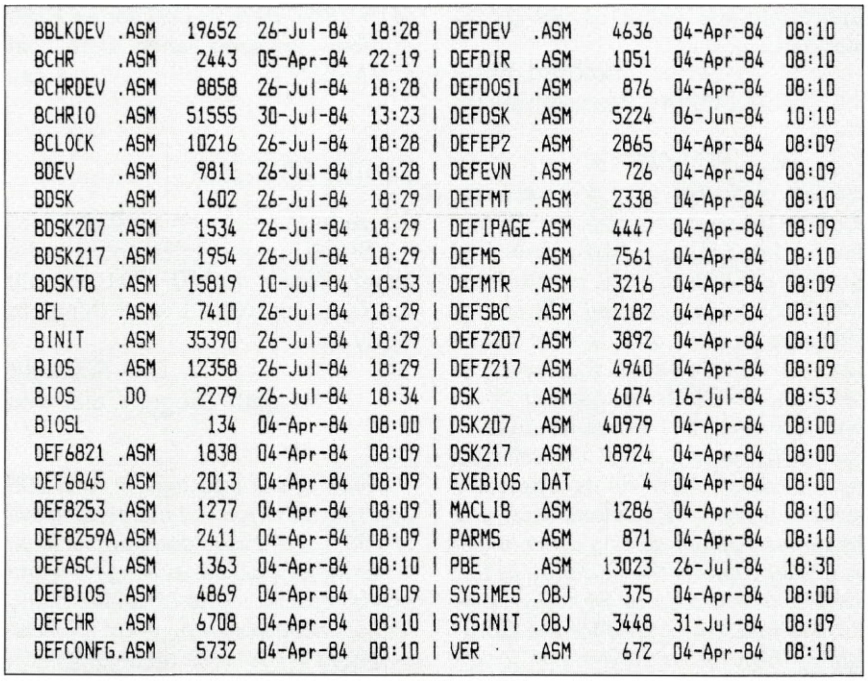
Lon Allbright
San Diego, California

Much of the information in S-100 Journal, although not mentioning the H-100 specifically, does apply to it. Nevertheless, we are looking for some articles specific to the H/Z-100. If any readers have done any good projects, or would like to write up an article on a H/Z-100 subject, please contact me for possible publication. (For example, is it possible to modify the H/Z-100 to make it IEEE-696 compatible? I don't know, but, if so, that would be a project that I would publish, since it would make all other add-on S-100 boards available to Zenith 100 users.) • *Jay*

This letter is basically a response to the letter from Tony Price in your Fall 1985 issue.

I chance to be numbered among some 'Heath Freaks' who chose to buy the Heath H-100 because it used the S-100 bus. I was even able to get a customer at the National Bureau of Standards to buy a Zenith Z-100 for a basic experiment for the same reason.

I am of course disappointed that Heath/Zenith is vigorously going down the compatible track. Nevertheless, there remains something of the Heath tradition of overwhelming documentation. This might interest Mr. Price. As soon as it was available, I bought the 'MS-DOS Vers. 2 Development Kit,' Catalog # CB-5063-16, which lists at $149 in the current Heathkit catalog. This kit comes with the BIOS sources on three disks for three versions of the Heath/Zenith machines. For the S-100-based H/Z-100, the directory of the disk is:



```
BBLKDEV .ASM   19652  26-Jul-84  18:28 | DEFDEV  .ASM    4636  04-Apr-84  08:10
BCHR    .ASM    2443  05-Apr-84  22:19 | DEFDIR  .ASM    1051  04-Apr-84  08:10
BCHRDEV .ASM    8858  26-Jul-84  18:28 | DEFDOSI .ASM     876  04-Apr-84  08:10
BCHRIO  .ASM   51555  30-Jul-84  13:23 | DEFDSK  .ASM    5224  06-Jun-84  10:10
BCLOCK  .ASM   10216  26-Jul-84  18:28 | DEFEP2  .ASM    2865  04-Apr-84  08:09
BDEV    .ASM    9811  26-Jul-84  18:28 | DEFEVN  .ASM     726  04-Apr-84  08:09
BDSK    .ASM    1602  26-Jul-84  18:29 | DEFFMT  .ASM    2338  04-Apr-84  08:10
BDSK207 .ASM    1534  26-Jul-84  18:29 | DEFIPAGE.ASM    4447  04-Apr-84  08:09
BDSK217 .ASM    1954  26-Jul-84  18:29 | DEFMS   .ASM    7561  04-Apr-84  08:10
BDSKTB  .ASM   15819  10-Jul-84  18:53 | DEFMTR  .ASM    3216  04-Apr-84  08:09
BFL     .ASM    7410  26-Jul-84  18:29 | DEFSBC  .ASM    2182  04-Apr-84  08:10
BINIT   .ASM   35390  26-Jul-84  18:29 | DEFZ207 .ASM    3892  04-Apr-84  08:10
BIOS    .ASM   12358  26-Jul-84  18:29 | DEFZ217 .ASM    4940  04-Apr-84  08:09
BIOS    .DO     2279  26-Jul-84  18:34 | DSK     .ASM    6074  16-Jul-84  08:53
BIOSL            134  04-Apr-84  08:00 | DSK207  .ASM   40979  04-Apr-84  08:00
DEF6821 .ASM    1838  04-Apr-84  08:09 | DSK217  .ASM   18924  04-Apr-84  08:00
DEF6845 .ASM    2013  04-Apr-84  08:09 | EXEBIOS .DAT       4  04-Apr-84  08:00
DEF8253 .ASM    1277  04-Apr-84  08:09 | MACLIB  .ASM    1286  04-Apr-84  08:10
DEF8259A.ASM    2411  04-Apr-84  08:09 | PARMS   .ASM     871  04-Apr-84  08:10
DEFASCII.ASM    1363  04-Apr-84  08:10 | PBE     .ASM   13023  26-Jul-84  18:30
DEFBIOS .ASM    4869  04-Apr-84  08:09 | SYSIMES .OBJ     375  04-Apr-84  08:00
DEFCHR  .ASM    6708  04-Apr-84  08:10 | SYSINIT .OBJ    3448  31-Jul-84  08:09
DEFCONFG.ASM    5732  04-Apr-84  08:10 | VER  ·  .ASM     672  04-Apr-84  08:10
```

The directories of the other two disks show equivalent contents. The point of this is that there are lots of BIOS source code to study.

Mr. Price might also like to take a look at an article by Cochran and Sweger in the January/February issue of *Micro/Systems Journal.* It bears the title: 'Implementing PC-DOS on Non-IBM Compatible Computers — Part I.'

I find that *Micro/Systems Journal,* plus *PC Tech Journal,* with *Byte* thrown in on the side, give me as much as I can handle in reading about the MS-DOS world.

Thus, keep your present 'not MS-DOS' course and don't sweat production delays as far as I am concerned. My only real fear is that SITA and you might become sufficiently successful at increasing circulation that Ziff-Davis or McGraw-Hill might become interested. We don't need additional 'regrets' letters as the MBAs advising those people cancel quality publications and/or force editorial changes which demolish their quality.

I hope the new department by Gary Feierbach works out. I wonder if there could be an article on the S1 operating system. A year or so ago, I wrote Dr. Godbout and the S1 people correlated letters about S1 in response to some splashy ads characterizing UNIX as a 'dinosaur.' I was sort of curious about whether or not the Viasyn CPU16032 might be a good target for S1. The people in New Jersey sent me a courteous put-down reply because they were only interested in dealing with major OEMs. Since then, I have seen some passing references to S1. I note that Mr. Feierbach's table couples S1 with CompuPro hardware (I assume CompuPro = Viasyn). Incidentally, when I compare Mr. Feierbach's table with the content of his company's ad on page 3 (of the Fall 85 issue), I wonder why he didn't put Inner Access hardware down for OS-9. I'll send him a copy of this letter along with a request for more information about the stuff advertised.

Blanton C. Duncan
Washington Grove, Maryland

Thanks for the information. About S-100 Journal being sold, fear not. S-100 Journal is published by Octoplus Corporation which in turn is owned by Robert Petersen, Linda Pereira, and me. And we aren't selling. The reason why we are doing this is because we believe in the S-100 bus and its great advantages for users. Neither of us had any prior publishing experience, but believe me, we are learning very fast.

See my reply above concerning S1. We will probably still publish something on it. In any case, Gary should be talking about it in a future issue.

CompuPro is the actual name for what people erroneously think is now called Viasyn. Although CompuPro products do come from Viasyn Corporation. Confused? They brought it upon themselves. • *Jay*

▶ 696 BUS

(continued from page 11)

open-collector lines and 'recommends' that they **not** be terminated. A 360-ohm pull-up resistor to +5.0 volts should be used instead.

6. A 10″ double-high board was defined.

Realizing that board real estate was at a premium, the standard committee defined a 10″-high board as well as the normal 5″-high board. Unfortunately this taller board has not caught on, mainly due to the lack of enclosures that allow such a tall board.

The board's outline also changed slightly. The draft called for a clear area in the top center of the board. This clear area became very confusing. It was intended so that a hold-down rod could lock the boards into the bus, but some vendors placed a notch in their boards instead. This clear area is absent from the approved standard, so now the entire top of the board is available for connectors or other components.

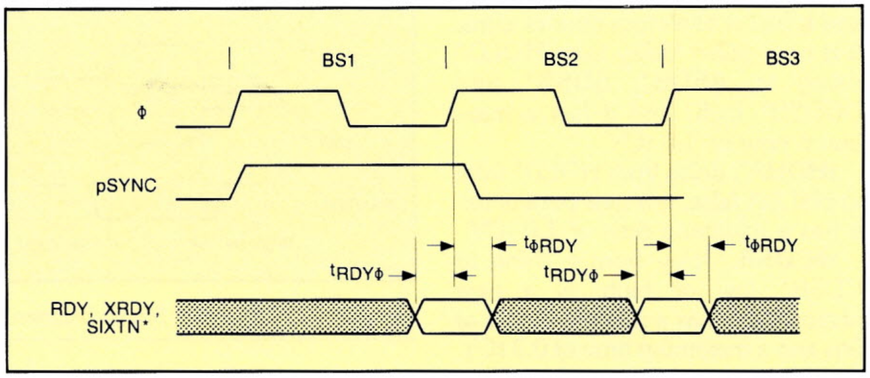


Figure 2. *RDY Timing Diagram.*

7. Some timing parameters were changed and others were added. Please refer to Figures 1 through 4 for the locations of the timing parameters that were added or changed. These are discussed next.

tSTSY, pSTVAL* falling edge prior to pSYNC high, was removed. It was replaced with **tSYST***, delay from pSYNC high to pSTVAL* low, which has a minimum time of 30ns.

tAΦ, addresses stable prior to Φ (PHI) high during pSYNC high, is a new parameter. It has a minimum timing value of 80ns.

tSTΦ, status stable prior to Φ high during pSYNC high, is a new parameter. It has a minimum timing value of 50ns.

tDBZ, delay from pDBIN high to slave DI drivers active, was previously defined to have a maximum time of 25ns + 0.1 tCY. This maximum time is now 70ns.

tDB*Z, delay from pDBIN low to slave DI drivers HI-Z (High-Impedance), was previously defined to have a maximum time of 25ns + 0.1 tCY. The maximum time is now 70ns.

tSDB*, data valid setup time to pDBIN low, was listed in the draft but no timing parameters were given. tSDB* does not exist in the approved standard.

tRDYΦ, setup time RDY, XRDY, SIXTN* to Φ rising, was defined to have a minimum time of 70ns. The minimum time is now 20ns.

tΦRDY, hold time RDY, XRDY, SIXTN* after Φ rising, was defined to have a minimum time o 70ns. The minimum time is now 20ns.

tPOV, overlap of PHANTOM* and pDBIN or pWR*, is a new parameter. It has a minimum timing value of 30ns.

tSET, delay pHLDA to ADSB*, SDSB*, DODSB* low, was defined to have a minimum time of 30ns. The minimum time is now 0ns.

tOV, time both temporary and permanent master drive the control output lines, was defined to have a minimum time of 0.5 tCY. It is now 0.4 tCY.

tREL, was called 'setup time end of bus transfer to pHLDA rising edge,'

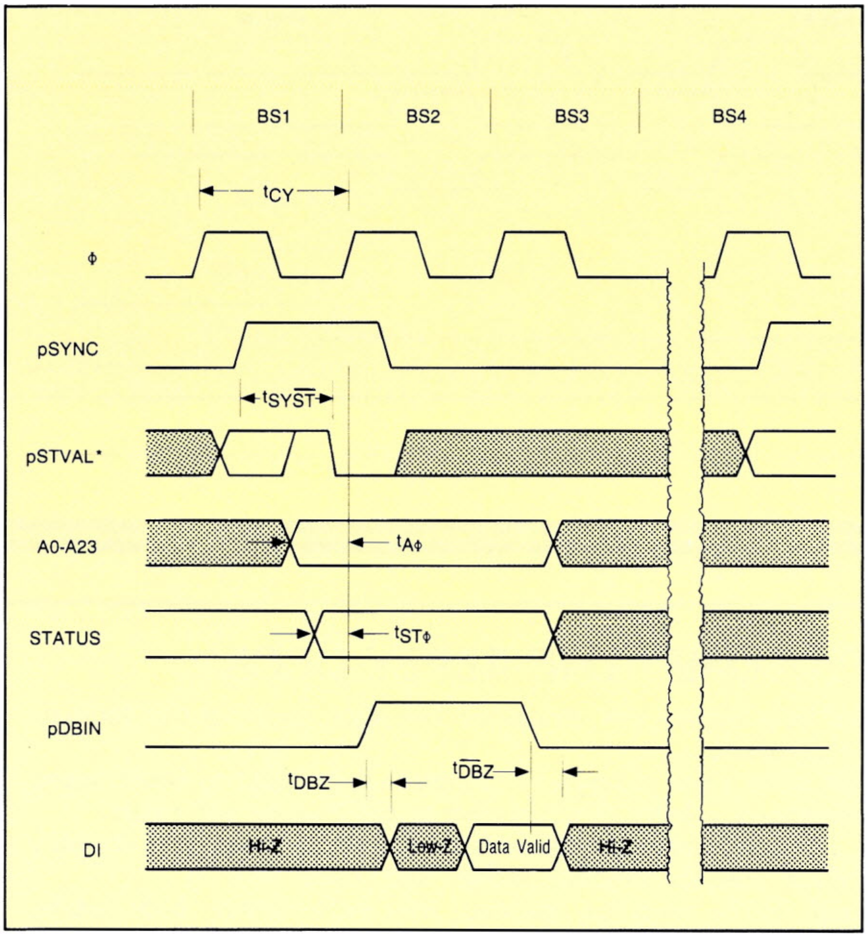


Figure 1. *Read Cycle Timing Diagram.*

and it had a minimum time of 20ns. It is now called 'delay from HOLD* rising to ASDB*, SDSB* and DODSB* high,' and it has a maximum time of 1.0 tCY.

tHDHA*, delay from HOLD* false to pHLDA false, is a new parameter. It has a minimum time of 1.0 tCY.

tΦCDSB, delay from Φ rising to CDSB* low or high, is a new parameter. It has a minimum time of 0ns and a maximum time of 0.3 tCY.

tHD*HA, delay from HOLD* falling to pHLDA rising, is a new parameter. It has a minimum time of 1.0 tCY.

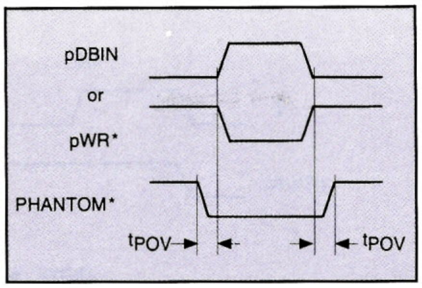


Figure 3. *PHANTOM* Timing Diagram.*

In Conclusion

The July 1979 *IEEE Computer* magazine draft of the IEEE-696 standard greatly improved the understanding and functions of the original S-100 bus. The final IEEE-696-1983 standard fine tuned the specification further. But, as with any document, the final standard still has a few inadequacies in it. In my next article, I will discuss one of these oversights when I describe a 'fast wait state generator' circuit.

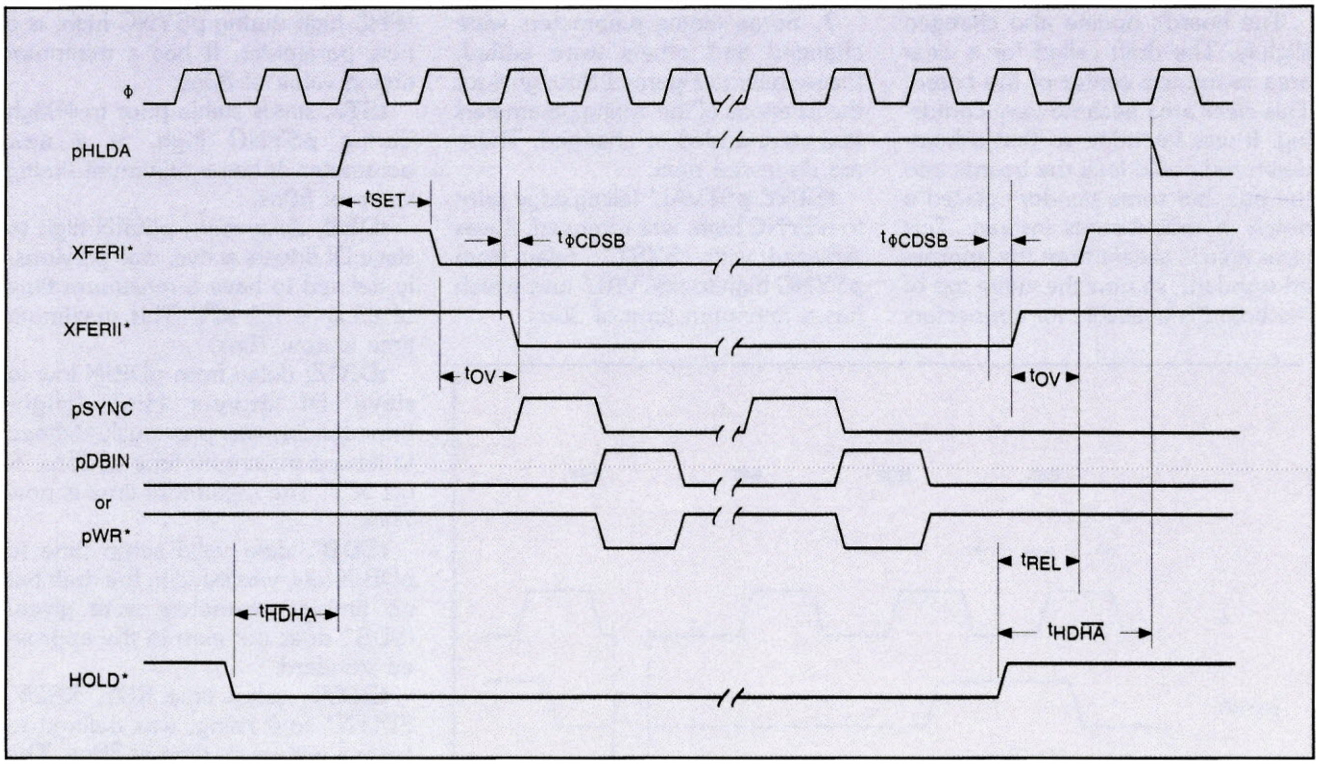


Figure 4. *Bus Exchange Timing Diagram.*

PLEASE WELCOME: THE REPLY DECK

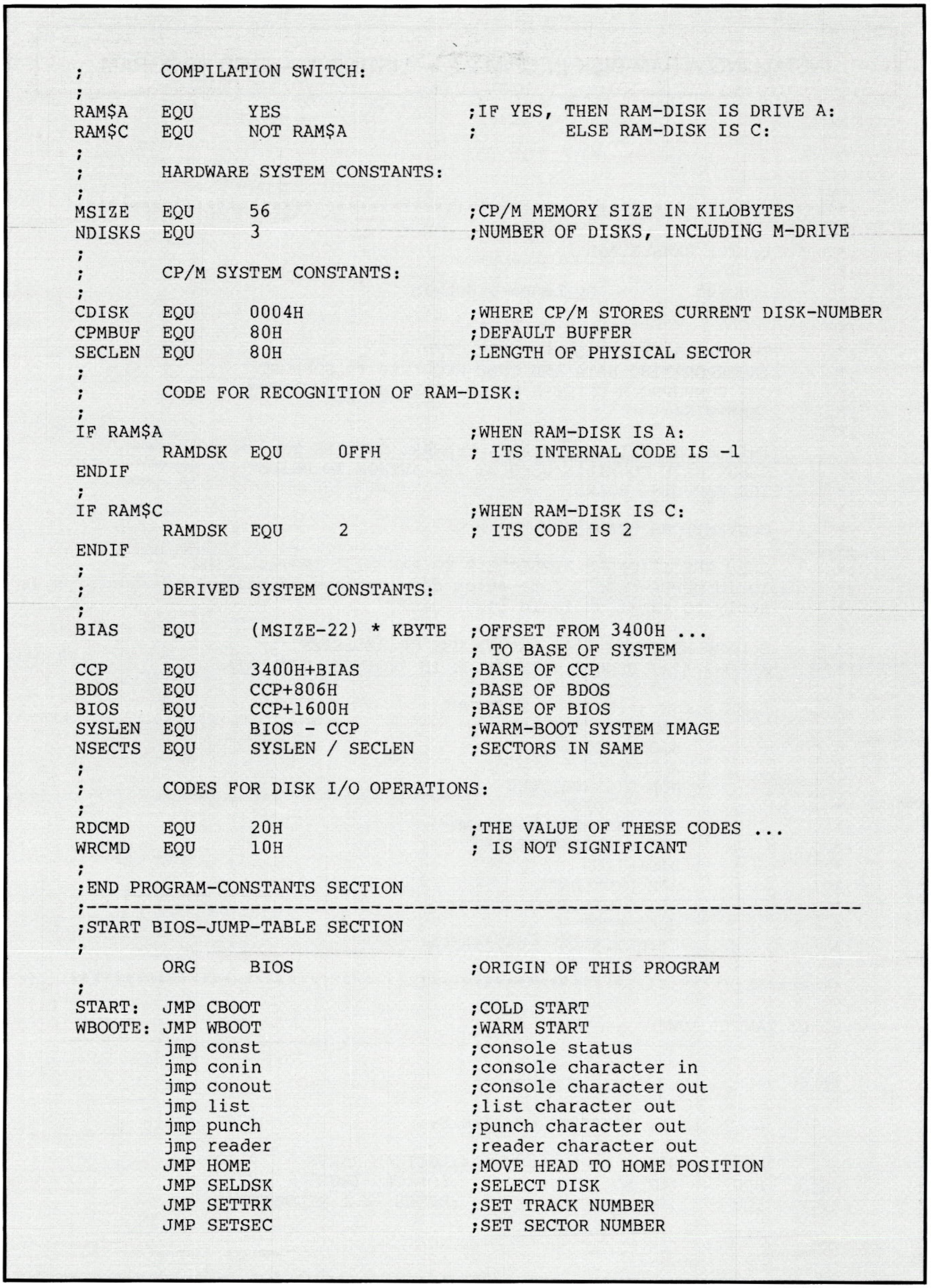
Starting with this issue, there will be a Reply Deck distributed with each copy* of the Journal. There are 3 types of cards in the reply deck:

1. A card for each advertisement of 1/3 page or larger. This card is to be used to obtain more information from S-100 Journal advertisers. Simply fill in your name and address and return.
2. Editorial Feature Reply Cards. These are blank-address cards to obtain information from vendor's mentioned or featured in articles. Fill in both the vendor's address and yours.
3. Reply cards to S-100 Journal. To be used for renewals, new subscriptions, back issues, changes of address, etc.

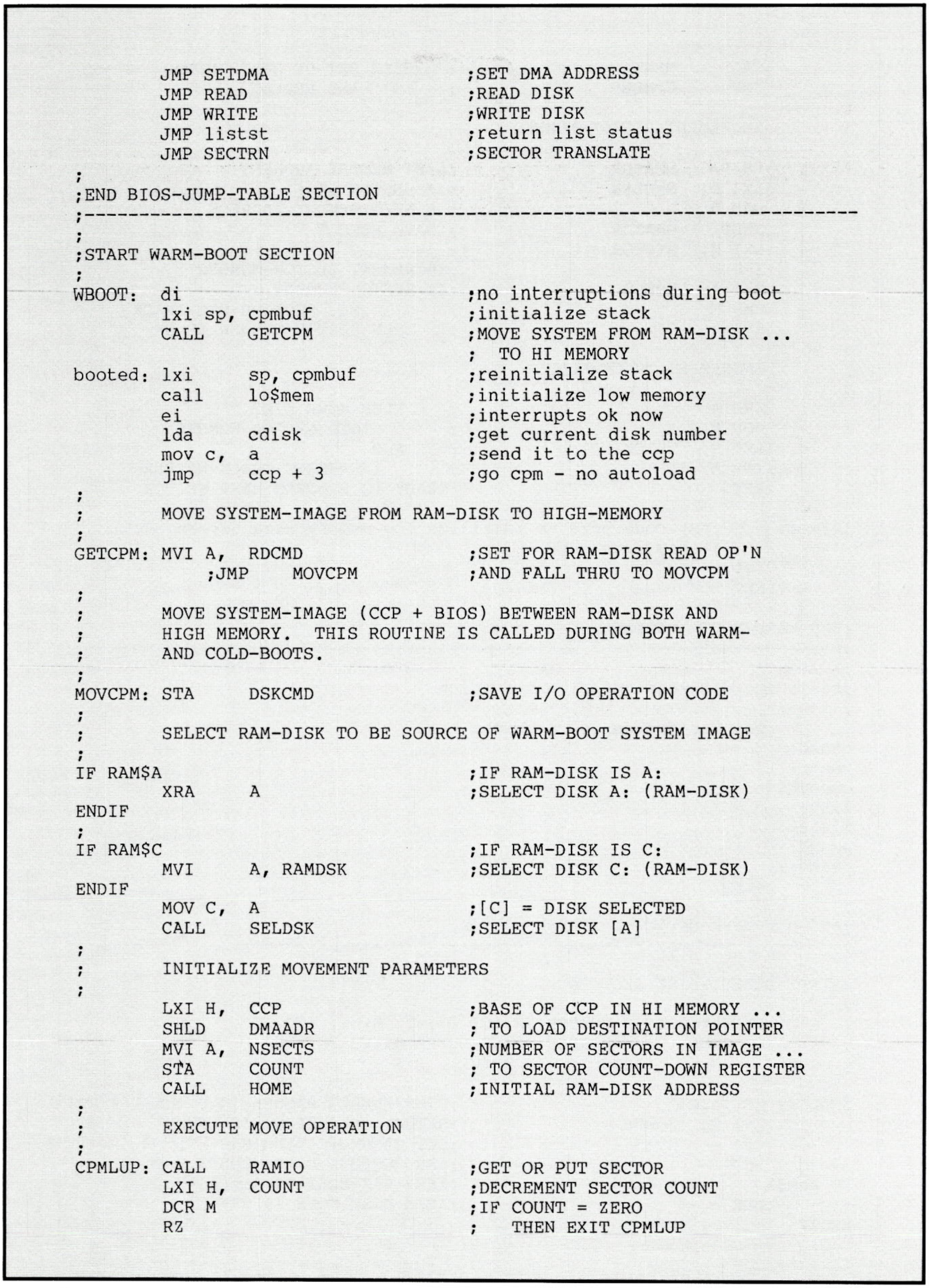
*This first mailing of the deck is experimental. Only a limited number of copies will be distributed.

INSTALLING A RAM-DISK IN CP/M 2.2 — LISTING 1 — BIOS PROGRAM

```
;***********************************************************************
;*
;*      title: RAMSYS.ASM
;*
;*      15Nov85         by Leonard Schwab
;*
;*
;*      THIS IS A SKELETAL BIOS FOR CP/M 2.2
;*      INCORPORATING DATA AND CODE REQUIRED TO SUPPORT
;*      ONE COMPUPRO M-DRIVE-H SOLID STATE DISK EMULATOR
;*      (RAM-DISK).
;*
;*      THIS BIOS IS INTENDED AS A GENERAL OUTLINE TO
;*      ILLUSTRATE ROUTINES USED BY THE AUTHOR TO DRIVE
;*      THE RAM-DISK BOARD.
;*
;*      CONVENTIONS USED:
;*
;*      CODE SPECIFICALLY APPLICABLE TO RAM-DISK INSTALLATION
;*      IS IN UPPER-CASE.  Code which does not specifically
;*      apply to ram-disk is in lower-case.
;*
;*      SQUARE-BRACKETS ([]) DENOTE USE OF REGISTER.
;*      ARROWS (>>) DENOTE THAT VALUE IS 'POINTER TO' DATA.
;*
;*      RAMSYS.ASM - INDEX OF SECTIONS:
;*              PROGRAM-CONSTANTS
;*              BIOS-JUMP-TABLE
;*              WARM-BOOT
;*              NON-DISK-DRIVERS
;*              SELECT-DISK
;*              DISK-ADDRESS-POINTERS
;*              SECTOR-TRANSLATION
;*              DISK-I/O
;*              RAM-DISK-DATA
;*              RAM-DISK-I/O
;*              COLD-BOOT
;*              UNINITIALIZED-DATA-AREA
;*
;***********************************************************************
;
;START RAMSYS.ASM
;-----------------------------------------------------------------------
;
;START PROGRAM-CONSTANTS SECTION
;
;       LOGICAL AND PHYSICAL CONSTANTS:
;
NO      EQU     0                       ;LOGICAL FALSE
YES     EQU     NOT NO                  ;LOGICAL TRUE
KBYTE   EQU     1024                    ;BYTES IN 1 KILOBYTE
;
```



```
;       COMPILATION SWITCH:
;
RAM$A   EQU     YES                     ;IF YES, THEN RAM-DISK IS DRIVE A:
RAM$C   EQU     NOT RAM$A               ;        ELSE RAM-DISK IS C:
;
;       HARDWARE SYSTEM CONSTANTS:
;
MSIZE   EQU     56                      ;CP/M MEMORY SIZE IN KILOBYTES
NDISKS  EQU     3                       ;NUMBER OF DISKS, INCLUDING M-DRIVE
;
;       CP/M SYSTEM CONSTANTS:
;
CDISK   EQU     0004H                   ;WHERE CP/M STORES CURRENT DISK-NUMBER
CPMBUF  EQU     80H                     ;DEFAULT BUFFER
SECLEN  EQU     80H                     ;LENGTH OF PHYSICAL SECTOR
;
;       CODE FOR RECOGNITION OF RAM-DISK:
;
IF RAM$A                                ;WHEN RAM-DISK IS A:
        RAMDSK  EQU     0FFH            ; ITS INTERNAL CODE IS -1
ENDIF
;
IF RAM$C                                ;WHEN RAM-DISK IS C:
        RAMDSK  EQU     2               ; ITS CODE IS 2
ENDIF
;
;       DERIVED SYSTEM CONSTANTS:
;
BIAS    EQU     (MSIZE-22) * KBYTE      ;OFFSET FROM 3400H ...
                                        ; TO BASE OF SYSTEM
CCP     EQU     3400H+BIAS              ;BASE OF CCP
BDOS    EQU     CCP+806H                ;BASE OF BDOS
BIOS    EQU     CCP+1600H               ;BASE OF BIOS
SYSLEN  EQU     BIOS - CCP              ;WARM-BOOT SYSTEM IMAGE
NSECTS  EQU     SYSLEN / SECLEN         ;SECTORS IN SAME
;
;       CODES FOR DISK I/O OPERATIONS:
;
RDCMD   EQU     20H                     ;THE VALUE OF THESE CODES ...
WRCMD   EQU     10H                     ; IS NOT SIGNIFICANT
;
;END PROGRAM-CONSTANTS SECTION
;-----------------------------------------------------------------------
;START BIOS-JUMP-TABLE SECTION
;
        ORG     BIOS                    ;ORIGIN OF THIS PROGRAM
;
START:  JMP CBOOT                       ;COLD START
WBOOTE: JMP WBOOT                       ;WARM START
        jmp const                       ;console status
        jmp conin                       ;console character in
        jmp conout                      ;console character out
        jmp list                        ;list character out
        jmp punch                       ;punch character out
        jmp reader                      ;reader character out
        JMP HOME                        ;MOVE HEAD TO HOME POSITION
        JMP SELDSK                      ;SELECT DISK
        JMP SETTRK                      ;SET TRACK NUMBER
        JMP SETSEC                      ;SET SECTOR NUMBER
```



```
        JMP SETDMA                   ;SET DMA ADDRESS
        JMP READ                     ;READ DISK
        JMP WRITE                    ;WRITE DISK
        JMP listst                   ;return list status
        JMP SECTRN                   ;SECTOR TRANSLATE
;
;END BIOS-JUMP-TABLE SECTION
;-----------------------------------------------------------------------
;
;START WARM-BOOT SECTION
;
WBOOT:  di                           ;no interruptions during boot
        lxi sp, cpmbuf               ;initialize stack
        CALL    GETCPM               ;MOVE SYSTEM FROM RAM-DISK ...
                                     ;  TO HI MEMORY
booted: lxi     sp, cpmbuf           ;reinitialize stack
        call    lo$mem               ;initialize low memory
        ei                           ;interrupts ok now
        lda     cdisk                ;get current disk number
        mov c,  a                    ;send it to the ccp
        jmp     ccp + 3              ;go cpm - no autoload
;
;       MOVE SYSTEM-IMAGE FROM RAM-DISK TO HIGH-MEMORY
;
GETCPM: MVI A,  RDCMD                ;SET FOR RAM-DISK READ OP'N
            ;JMP    MOVCPM           ;AND FALL THRU TO MOVCPM
;
;       MOVE SYSTEM-IMAGE (CCP + BIOS) BETWEEN RAM-DISK AND
;       HIGH MEMORY.  THIS ROUTINE IS CALLED DURING BOTH WARM-
;       AND COLD-BOOTS.
;
MOVCPM: STA     DSKCMD               ;SAVE I/O OPERATION CODE
;
;       SELECT RAM-DISK TO BE SOURCE OF WARM-BOOT SYSTEM IMAGE
;
IF RAM$A                             ;IF RAM-DISK IS A:
        XRA     A                    ;SELECT DISK A: (RAM-DISK)
ENDIF
;
IF RAM$C                             ;IF RAM-DISK IS C:
        MVI     A, RAMDSK            ;SELECT DISK C: (RAM-DISK)
ENDIF
        MOV C,  A                    ;[C] = DISK SELECTED
        CALL    SELDSK               ;SELECT DISK [A]
;
;       INITIALIZE MOVEMENT PARAMETERS
;
        LXI H,  CCP                  ;BASE OF CCP IN HI MEMORY ...
        SHLD    DMAADR               ; TO LOAD DESTINATION POINTER
        MVI A,  NSECTS               ;NUMBER OF SECTORS IN IMAGE ...
        STA     COUNT                ; TO SECTOR COUNT-DOWN REGISTER
        CALL    HOME                 ;INITIAL RAM-DISK ADDRESS
;
;       EXECUTE MOVE OPERATION
;
CPMLUP: CALL    RAMIO                ;GET OR PUT SECTOR
        LXI H,  COUNT                ;DECREMENT SECTOR COUNT
        DCR M                        ;IF COUNT = ZERO
        RZ                           ;   THEN EXIT CPMLUP
        CALL    NEXT                ;   ELSE SET-UP NEXT SECTOR
        JMP     CPMLUP              ;        AND REPEAT
;
;       INCREMENT SECTOR-NUMBER
;
NEXT:   LHLD    DMAADR              ;LAST MEMORY ADDRESS
        LXI D,  SECLEN              ; + SECTOR SIZE
        DAD D                       ; = NEXT MEMORY ADDRESS
        SHLD    DMAADR              ; SAVE RESULT
        LXI H,  SECTOR              ;
        INR M                       ;INCREMENT SECTOR-NUMBER
        MVI A,  HDSPT               ;IF SECTOR-NUMBER
        CMP M                       ;       .NE. SECTORS-PER-TRACK
        RNZ                         ;   THEN SECTOR-NUMBER IS OK
;
;       INCREMENT TRACK-NUMBER
;
        XRA A                       ;    ELSE MOVE ZERO
        MOV M,  A                   ;       INTO SECTOR-NUMBER
        LXI H,  TRACK               ;    AND
        INR M                       ;      INCREMENT TRACK-NUMBER
        RET                         ;READY TO PROCESS NEXT SECTOR
;
lo$mem:     ;put code here to initilize low-memory area before
            ;entry into ccp
        ret
        RET
;
;END WARM-BOOT SECTION
;-----------------------------------------------------------------------
;
;start non-disk-device-drivers section
;
;       put various device routines here
const:
conin:
conout:
list:
listst:
punch:
reader:
        ret
;
;end non-disk-device-drivers section
;-----------------------------------------------------------------------
;
;START SELECT-DISK SECTION
;
;       EXPECTS: DISK-NUMBER IN [C] (A:=0, B:=1, ETC.)
;       RETURNS: DPH ADDRESS OF SELECTED DRIVE IN [HL]
;                 (ZERO IF BAD DRIVE)
;
SELDSK: MOV A,C                     ;DISK-NUMBER PASSED IN C (BASE ZERO)
        LXI H,  0000H               ;RETURN [HL] ZERO, IF ERROR
        CPI     NDISKS              ;SET CARRY IF < NDISKS
        RNC                         ;ERRORONEOUS DISK NUMBER
IF RAM$A                            ;WHEN RAM-DISK IS A: [A]=0
        DCR A                       ; RAM-DISK CODE IS 0FFH
ENDIF
```

```
        STA     DISK                    ;SAVE DISK NUMBER
        LXI H,  DPHRAM                  ;IN CASE RAM-DISK IS BEING SELECTED
        CPI     RAMDSK                  ;IF RAM-DISK IS NOT BEING SELECTED
        cnz     floppy                  ;   then process request for floppy
        RET                             ;RETURN [HL] = >>DPH
;
floppy:     ;put floppy-disk select routine here
        ret
;
;END SELECT-DISK SECTION
;-----------------------------------------------------------------------
;
;START DISK-ADDRESS-POINTERS SECTION
;
;       'home' drive on first track/sector
;
home:   lxi b,  0                       ;pass sector zero ...
        call    setsec                  ; to setsec
                                        ; and fall thru to select first track
;
;       set track to [bc]
;
settrk: mov h,  b
        mov l,  c
        shld    track
        ret
;
;       set sector to [bc]
;
setsec: mov a,  c                       ;use low byte only
        sta     sector
        ret
;
;       set i/o buffer address to [bc]
;
setdma: mov h,  b
        mov l,  c
        shld    dmaadr
        ret
;
;END DISK-ADDRESS-POINTERS SECTION
;-----------------------------------------------------------------------
;
;START SECTOR-TRANSLATION SECTION
;
;       EXPECTS: LOGICAL SECTOR-NUMBER IN [BC]
;                >>SECTOR-TRANSLATION TABLE IN [DE]
;                IF [HL] ZERO, THEN NO TRANSALATION REQUIRED
;       RETURNS: PHYSICAL SECTOR-NUMBER IN [HL]
;
SECTRN: MOV H,  B                       ;[B], [H], ALWAYS ZERO
        MOV L,  C                       ;[C], [L] = LOGICAL SECTOR-NUMBER
;
;       TEST FOR [DE] = ZERO
;
        MOV A,  D
        ORA E                           ;
        RZ                              ;NO TRANSLA'N; PHYSICAL SECTOR IN [HL]
;
```

Listing continues on page 44

homebrewing

Wow! Judging by the letters we received, it seems that everyone and his computer really enjoyed Burt Hanagami's letter to Homebrewing *in the last S-100 Journal.*

So, we bring you more: ***S-100 in the Far North.*** *This fellow, Gord Wiggins, is up there in Labrador, in the town of Cartwright (population 658, we looked it up) having the time of his life with his S-100 system. So what if his computer's company no longer exists? So what if he owns the only S-100 system in town? These are irrelevant matters to a true S-100 enthusiast. He just keeps on filling that baby up with goodies.*

Homebrewing *is our for-fun, for-nostalgia, for-learning column. If you've had painful-pleasant problems-solutions dissecting or constructing an S-100 system, share the experience. Give us humor, gives us wit, clever solutions and unique systems.*

There are no rules for what goes into this column, except that it must be about hacking with S-100 systems. Describe your system, decisions you had to make, how you overcame problems. Photos are welcome.

Send your contribution (typed double-spaced) to Homebrewing, *S-100 Journal, PO Box 12881, Raleigh, NC 27605.*

Late in the fall of 1979, I decided that the time was ripe for me to invest in a computer. I had been subscribing to *BYTE* magazine since its conception and browsed through the ads with a wishful eye, dreaming of owning an Altair or a Cromemco or other 8080-based S-100 system advertised at that time.

A friend of mine had an Altair and was busy writing programs and keying them in on the front panel switches. They really worked! He was gaining all kinds of insights into the workings of the amazing 8080 CPU. We keyed in a program and listened to a radio tuned between stations as the Altair played 'Daisy Daisy Give Me Your Answer True' with its radio interference. A performance that would not be appreciated today by the FCC.

My problem was money. If I were going to have a computer, it had to be low-cost. Prices of $1000.00 to $1500.00 were out of the question. The S-100 bus concept was excellent and, with each issue of *BYTE*, more and more board manufacturers came to the surface.

I then began to notice the ads from Netronics Research and Development about their 'Explorer-85' computer. This was just what I was looking for. For a price of $129.95, you got a motherboard using the Intel 8085 CPU with its built-in serial I/O. This seemed like a great idea at the time but I later found it to be a real pain. There was no handshaking or method of polling the keyboard for a keypress as with conventional UART driven I/O. However, the software UART did save a chip and kept the price down.

For your $129.95, you also got a total of 256 bytes of RAM. That's 256 **bytes** of RAM not K of RAM. It seemed like a lot at the time. There was space on the motherboard for an additional 4K of RAM if you could afford the 2114 chips at a later date. There was a good monitor included as well as 8K of EPROM sockets. These could be configured for the soon-to-be-announced 16K RAM chips (6116) that at that time cost about $60.00 each.

The best thing about the Explorer-85 was that it was S-100 bus. You could not use it with S-100 boards yet. That was 'Level B' which cost more dollars to install.

However, it wasn't long before I had levels A, B, C, D, and E installed, and my computer was growing and worth a lot more than $129.95. I now had 8K Microsoft Basic-80 (cassette version) on cassette and the same thing in EPROM. I had a flimsy six-slot card cage and 4K of RAM.

I also bought the Netronics 300-baud terminal board and case. This was 64 by 16 and state-of-the-art at the time. I worked with this for four years. 300 baud and 64 by 16 seems like something that should be alongside Bell's original telephone in a museum someplace instead of in a computer in Labrador.

There were many many problems to overcome, and I was all alone up here. For three years, I struggled along like this. Then I got the idea of forming a users group. Maybe Netronics had sold a lot of these things and there were other people in the same boat.

I advertised in *BYTE* and the response was immediate. About ten times as great as I had expected. I was using an old model 35 telex machine for a printer, and printing the monthly newsletter was very time consuming and noisy to boot. Another fellow had the same idea of a users group at about the same time. We found out about each other and combined the groups. I had to give up the job of editor of the newsletter as my job did not allow the necessary time it took. Clarence Heir took it over for a while, and it eventually went to Leroy Marshall living in the Chicago area.

Leroy still heads the group now, years later, and publishes a fine Newsletter called 'Explorations' for those of us still using the Explorer-85. The group is International, and I have made many fine friends through it. Their help and technical expertise has been a godsend to someone like me working alone.

If there are any Explorer-85 users reading this, I urge them to contact Leroy at 808 Vassar Lane, Schaumburg, IL 60193. We are into our 3rd year now and going strong.

I continued to expand the Explorer by adding memory and boards and two years ago, ât last, CP/M.

CP/M opened up a whole new world of software. Prior to CP/M, I was limited to what software I could write myself in assembler and BASIC. When I could find BASIC listings in magazines, I would key them in and try to get them to run on the Explorer. I was not always successful.

Cost was always foremost in my mind. I had to wait until 8″ drives came down in price and disk controller boards dropped to where I could afford them. Two years ago, I purchased two Siemens FDD100-8 SSSD drives for $129.00 each. I obtained a disk controller from a Canadian firm for a reasonable price, and I was in business. I purchased CP/M for $179.00, and then the fun started. I had to write my own BIOS since I had all oddball equipment, including the Explorer-85. Netronics, now being defunct, I could get no help from. Other members of the users group had gone to the JADE DD disk controller board, totally different from mine, so I was on my own again.

Three months later, with much help from group members Tom Head from Kentucky and Leroy Marshall from Illinois, I had CP/M up and running. I have not looked back since. I now have over 50 disks of software, mostly all public domain. I am heavy into remote CP/M systems and downloading. There is always something new to try out or a new S-100 board to wish for.

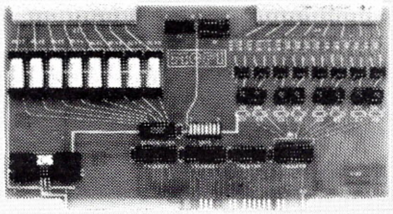
Following is a list of the boards that presently comprise my S-100 system here in Cartwright, Labrador:

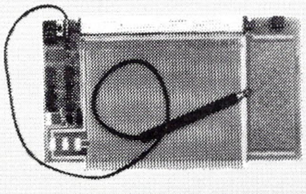
1. Explorer-85 computer levels A, B, C, D, E.
2. 64K Static RAM Board, from Memory Merchant.
3. 2793 Disk Controller Board, from Multiflex Technology.
4. 256K Solid State Disk Emulator Board, from Digital Research Computers of Garland, Texas.
5. Two Siemens FDD100-8 SSSD disk drives.
6. Multiflex 80 × 24 terminal/keyboard (9600 baud).
7. Star 103-232 acoustic Modem.
8. Netronics 'Electric Mouth' voice board.
9. SSM IO-4 two serial, two parallel I/O board.
10. SSM Programmer EPROM board.
11. E.D.A. Video Display Processor board (sprites, sound).
12. Jade 'Bus Probe' LED bus display board.
13. Three homebrew power supplies.
14. Teco high-res green monitor.
15. Motorola B/W 9″ monitor for video board.
16. Star Powertype daisy-wheel letter-quality printer.
17. Mannesmann Tally Spirit-80 dot-matrix printer.

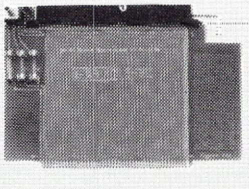
On order, but not yet received at the time of this writing, is a 1200-baud Hayes compatible modem, a CompuPro/Viasyn CPU-Z Z80 CPU board, and an Integrand cabinet with two 8″ drives and 10-slot motherboard with active termination both ends.

I am happy that the S-100 bus is alive and active and that someone has seen fit to publish this fine journal. Something that was badly needed by us S-100 bus enthusiasts.

Gord Wiggins
Cartwright, Nfld., Canada







Listing continued from page 41

```
;       OFFSET INTO TABLE AND GET PHYSICAL SECTOR-NUMBER
;
        DAD D                         ;[DE] + [HL] POINTS INTO TABLE
        MOV L,  M                     ;FETCH APPROPRIATE SECTOR
        RET                           ;RETURN TO BDOS
;
;END SECTOR-TRANSLATION SECTION
;--------------------------------------------------------------------
;
;START DISK-I/O SECTION
;
;       READ FLOPPY- OR RAM-DISK
;
READ:   MVI A,  RDCMD                 ;SET FOR READ OPERATION
        JMP     DO$IO                 ;EXECUTE READ OPERATION
;
;       WRITE FLOPPY- OR RAM-DISK
;
WRITE:  MVI A,  WRCMD                 ;SET FOR WRITE OPERATION
                                      ;FALL THRU TO EXECUTE
;
;       EXECUTE I/O OPERATION
;
DO$IO:  STA     DSKCMD                ;SAVE I/O OPERATION CODE
;
;       BRANCH ON DISK TYPE (FLOPPY/RAM)
;
        LDA     DISK                  ;LAST DISK SELECTED
        CPI     RAMDSK                ;IF RAM-DISK ...
        JZ      RAMIO                 ;   THEN USE RAM-DISK ROUTINES
                                      ;   else fall into floppy-disk code
;
;       put floppy-disk access code here
;
        ret
;
;END DISK-I/O SECTION
;--------------------------------------------------------------------
;
;START RAM-DISK-DATA SECTION
;
;       RAM-DISK PORT ASSIGNMENTS:
;
HDATA   EQU     0C6H                  ;PER ON-BOARD SWITCH SETTINGS
HADDR   EQU     HDATA + 1
;
;       RAM-DISK PHYSICAL SPECIFICATION:
;
HDK     EQU     512                   ;SIZE OF RAM-DISK (KBYTES)
;
;       RAM-DISK ORGANIZATION SPECIFICATIONS:
;
HDKIT   EQU     1                     ;SIZE OF RAM-DISK TRACK (KBs)
HDTIB   EQU     2                     ;TRACKS PER BLOCK
HDDBLK  EQU     2                     ;DIRECTORY BLOCKS
;
;       RAM-DISK DERIVED PARAMETERS:
;
HDTRK   EQU     HDK * HDKIT           ;TRACKS ON RAM-DISK
```

```
HDBIB   EQU     HDTIB * HDKIT * KBYTE
                                    ;BYTES IN BLOCK
HDBIS   EQU     SECLEN              ;BYTES IN SECTOR
HDSPT   EQU     HDKIT * KBYTE / HDBIS
                                    ;SECTORS PER TRACK
HDRES   EQU     NSECTS / HDSPT + 1
                                    ;RESERVED TRACKS (FOR SYSTEM)
HDSIB   EQU     HDBIB / HDBIS       ;SECTORS IN BLOCK
HDDTRK  EQU     HDTRK - HDRES       ;DATA TRACKS AVAILABLE
HDDSEC  EQU     HDTRK * HDSPT       ;DATA SECTORS AVAILABLE
HDDSM   EQU     HDDSEC / HDSIB      ;DATA BLOCKS AVAILABLE
HDDENT  EQU     HDDBLK * HDBIB / 32   ;DIRECTORY ENTRIES
;
;       DISK PARAMETER HEADER
;                                                     (OFFSET)
DPHRAM: DW      00                  ;>> TRANSLATE TABLE     (00)
        DW      00,00,00            ; SCRATCH AREA
        DW      DIRBUF              ;>> DIRECTORY BUFFER    (08)
        DW      DPBRAM              ;>> DPB                 (0A)
        DW      00                  ;>> CHECK VECTOR        (0C)
        DW      ALVRAM              ;>> ALLOCATION VECTOR   (0E)
;
;       DISK PARAMETER HEADER
;                                                     (OFFSET)
DPBRAM: DW      HDSPT               ;SECTORS PER TRACK      (SPT)
        DB      04                  ;BLOCK SHIFT FACTOR     (BSH)
        DB      15                  ;BLOCK MASK             (BLM)
        DB      01                  ;EXTENT MASK            (EXM)
        DW    HDDSM - 1             ;DISK SIZE - 1          (DSM)
        DW    HDDENT - 1            ;DIR ENTRIES - 1        (DRM)
        DB    11000000B             ;ALLOC 0                (AL0)
                                    ; - 2 BLOCKS FOR DIRECTORY
        DB      0                   ;ALLOC 1                (AL1)
        DW    00                    ;CHECK AREA SIZE        (CKS)
        DW    HDRES                 ;RESERVED TRACKS        (OFF)
;
;END RAM-DISK-DATA SECTION
;-----------------------------------------------------------------------
;
;START RAM-DISK-I/O SECTION
;
;       EXECUTE I/O OPERATION DESIGNATED BY [DSKCMD]
;
RAMIO:
;       SET M-DRIVE ADDRESSING LOGIC FOR TRACK/SECTOR
;
        LDA     SECTOR              ;SEND SECTOR-NUMBER
        OUT     HADDR               ; TO M-DRIVE
;
        LHLD    TRACK               ;MULTIPLY TRACK-NUMBER BY:
        DAD H ! DAD H               ;2, 4
        DAD H ! DAD H               ;8, 16
        DAD H ! DAD H               ;32, 64
        DAD H                       ;128
        MOV A,  H                   ;
        OUT     HADDR               ;SEND HIGH-BYTE
        MOV A,  L
        OUT     HADDR               ; ... LOW-BYTE
;
```

Listing continues on page 57

(continued from page 23)

line goes low. The use of U4 to assist in address decoding requires at least 3 bits in the base address to be 1's, unless inverters are used. U4 was used to help decode the address because it was nearby and unused.

To change the base address to FC70, for example, you would have to change pins 18, 19, and 20 to high (5V).

If your system uses 8-bit ports, the circuit shown in Figure 4A will decode the 4 most significant address lines to make BDSEL*. The base here is B0. The address lines A7-A4 must agree with the preset values on pins 20-23 respectively for the BDSEL* to go active (low).

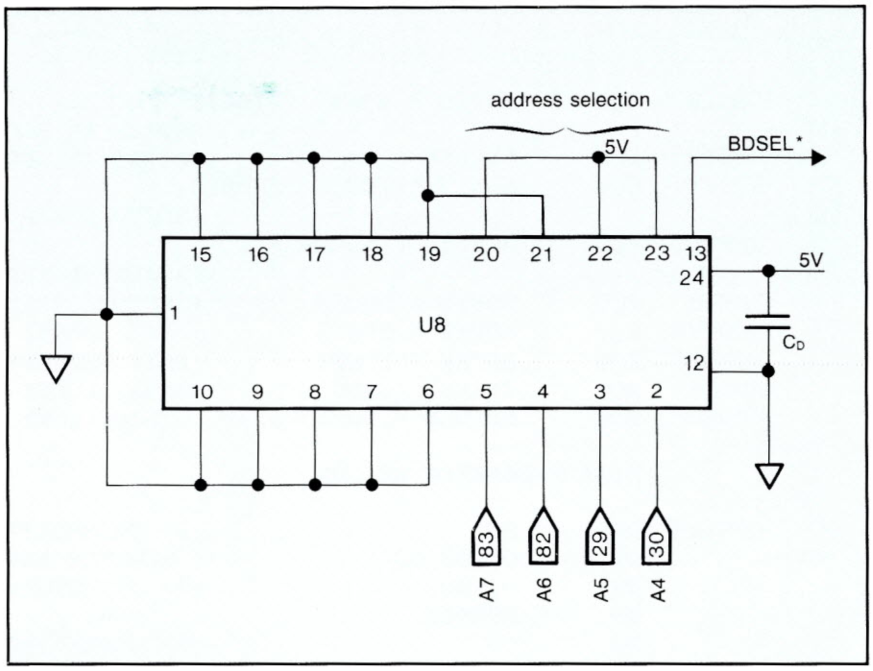


Figure 4a. *Address decoder for 8-bit port addressing. The base address here is B0 Hex.*

Read/Write Logic and Bus Interface

As mentioned earlier, the status and control lines must be combined to make the on-board read and write signals (RD* and WR* respectively). Figure 5 shows how this is done for both the read cycle (data to master) and the write cycle (data from master to our board). The signal BDSEL* is generated by the address-decoding section described above. These signals control the data flow between the clock and driver chips.

The interface to the bus is done by means of two transparent latches (U6 and U7), also shown in Figure 5. These latches are really drivers which send the signals between the bus and the other chips on the board. No latching or holding of data is done. The data input lines (DI0-DI7) must be tristated (put into high-impedance state) so that, if not in use, they do not interfere with the bus activity. The select signal for the data input latch is developed by the RD* signal. Note that a value is put on the bus only if the proper address and status signals exist. In order for RD* to go low (active), the proper address bits must match to generate BDSEL*, plus sINP and pDBIN must be high. This signal, RD*, goes to the ENABLE* inputs of the latch.

The data output lines also go through a similar latch but for a different reason. Here the latch keeps the bus lines from being loaded down by the other circuitry on the slave board. This is especially a problem if several devices on the board use these lines as inputs. The latch enable is sOUT*, which drives data lines D0-D3 during write transfers. Signal sOUT is high for both port and memory transfers, but the data doesn't 'go anywhere' if the clock chip is not selected. The sOUT signal must be inverted for the ENABLE* input on the latch. The sOUT signal could go directly into U4 for the write strobe, but, rather than loading down the bus signal sOUT by sending it to two

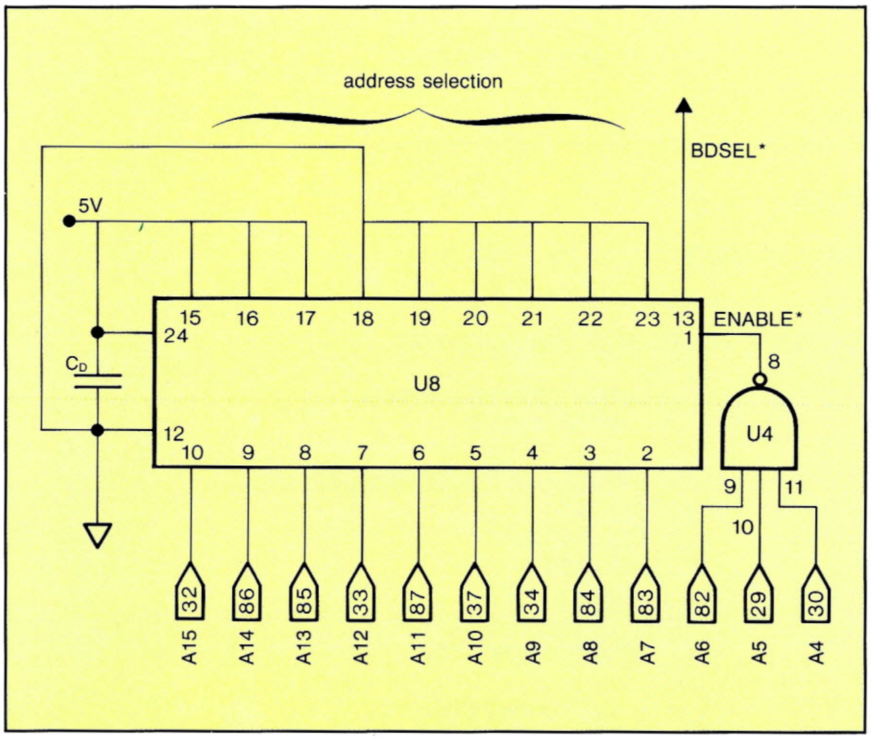


Figure 4. *Address decoder for 16-bit port addressing. The base address in this diagram is E070 Hex.*

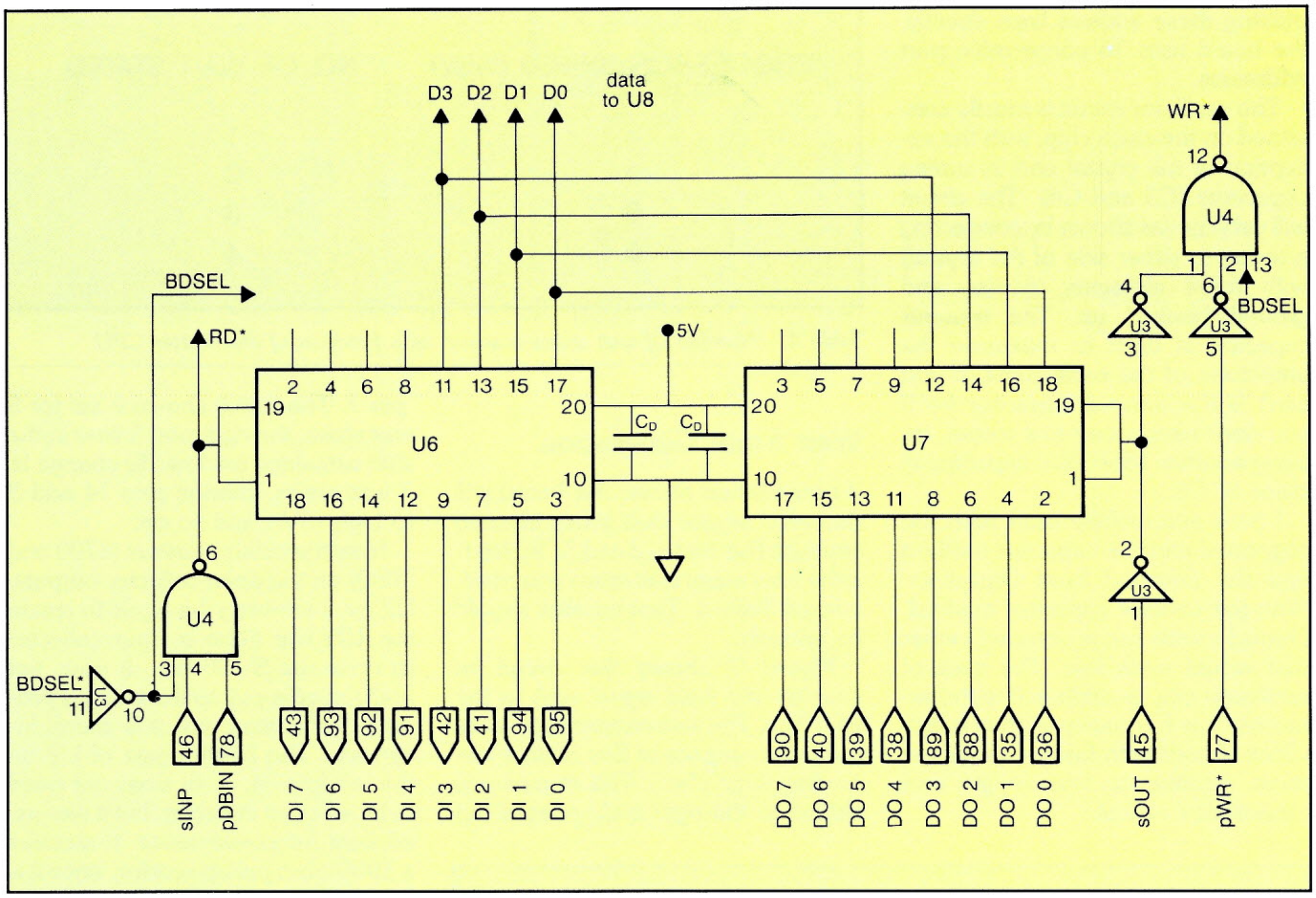


Figure 5. *The Read/Write Logic and the S-100 bus interface circuits.*

places, I reinverted sOUT* to create the U4 input.

The schematic shows both latches having 8 connections to the bus for completeness. If desired, only the 4 lsb's need be attached. However, if you use this board for other chips or future projects, the other connections may be useful.

Clock Circuitry

The clock chip needs only to be hooked up to a bidirectional data bus, its enable signals, the port select lines, the crystal, and power. This is shown in Figure 6.

The enable and timing signals are developed as described above (BDSEL* was generated by the address selection circuit, and RD* and WR* were developed by the read/write logic). The 4 address lines are used to select the digits as described in the clock data sheet. For example, address 0001 selects the tenths-of-second digit, and address 1010 selects the day-of-the-week digit. By

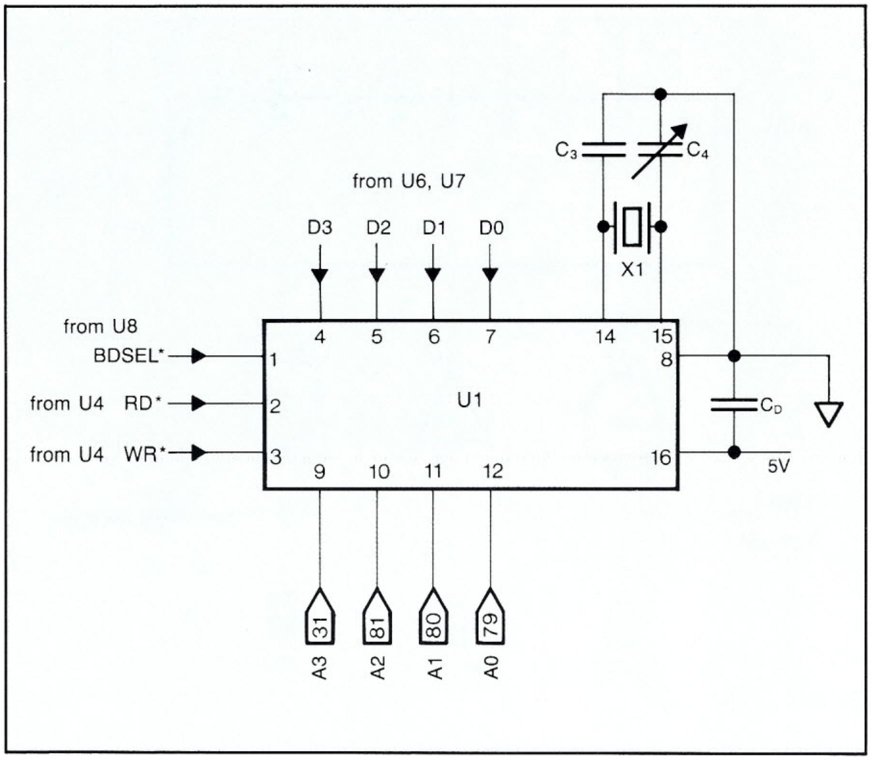


Figure 6. *The clock circuit.*

utilizing these address lines directly, the board uses 16 consecutive port addresses.

The oscillator circuit is mostly contained on the clock chip, with the exception of the crystal and its tuning capacitors (C3 and C4). The circuit will oscillate (as shown by connecting a scope to either side of the crystal) with these elements, power, and ground hooked up. The trimmer capacitor is used to maximize the amplitude of the oscillations. If you can't find a trimmer capacitor (or if you don't have access to a scope), try using another 18pF fixed capacitor in place of C4.

I have successfully used both the suggested variable capacitor method and the matched fixed capacitors. With the variable capacitor method, I found a wide range of tuned values that would work fine. The ease of oscillation may be attributed to the extremely low frequency of the crystal. Clock circuits with faster crystals are more sensitive to wire length and capacitance values.

PROCESSOR SPEED (MHz)	NO. OF WAIT STATES
2	1
4	2
6	3
8	4

Table 4. *Number of wait states required as a function of the master CPU.*

Wait State Generation

As mentioned above, the board will probably require wait states inserted between Bus States 2 and 3. To determine how many wait states you need, consult Table 4. That number should be enough.

Figure 7 shows the circuit to develop the hold signal sent to the RDY line. The shift register (U5) loads the shift sequence at Bus State 1 (according to pSYNC). This sequence is shifted to the right and inverted out pin 7. The circuit shown is set for 5 wait states; the rightmost 5 lines in the shift sequence are low. To change to 3 wait states, change pins 14 and 3 to high (5V), and so on.

Note the pullup resistors (4700 and 1000) on the open collector outputs. U2 pin 1 reinverts the signal to create the RDY line. U2 is an open collector to drive the S-100 bus. It does not really need a pullup if the bus is properly terminated, but it is cheap insurance. The NAND gate of U2 on the left (pins 4, 5, 6) does not need to be an open collector, but it was used here for convenience. It requires a 1000-ohm pullup resistor, since it is driving only one LS part.

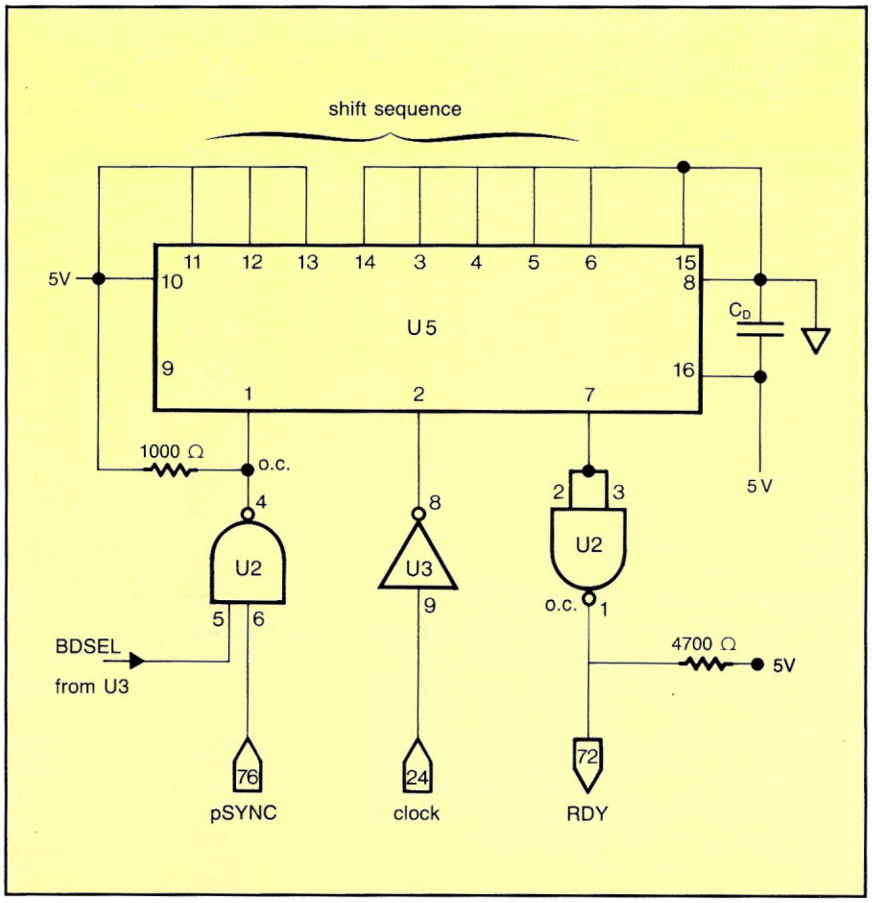


Figure 7. *Circuit for the wait state generator.*

BOARD LAYOUT

Plan your layout before starting to wire. The regulator should have an obvious place on the side of the prototype board. Organize the other chips in a way that minimizes distances. Bus drivers should go near the bottom. Locate the crystal and capacitors near the clock circuit. Each chip has a decoupling capacitor associated with it, so leave room near the chip socket for it. This capacitor goes between power and ground on each chip to remove the high-frequency noise.

Labeling the pins on the wire-wrap side helps reduce wiring errors. You can buy inexpensive socket-wrap IDs for this purpose. All chips are numbered clockwise from the notch as viewed from the bottom.

SUBSYSTEM TESTS

As you wire the circuit, it is recommended that you test each section separately. This makes it easier to find bugs, and gives a much higher confidence in the board working.

Be sure to double check each wire before applying power. Verify that power is not shorted to ground. Power up the unpopulated board and check power and ground at all chips. With the power off, add chips for each section separately.

Verify that the clock circuit oscillates as described above. Test the address section next (you'll need to write some software to address the chip). You should be able to see the BDSEL* signal go low when the address matches.

Continue by testing the read/write logic, bus interface chips, and wait-state logic separately. For the read/write logic section, verify RD* and WR*. For the bus interface section, check that data passes correctly. The wait-state circuit should generate the RDY signal of length corresponding to the wait states selected. When all these work, install all chips and check out the whole board.

BELLS AND WHISTLES

A backup battery may be connected to the clock, if desired. This would supply power for the chip to continue keeping time while the system is off. Figure 8 shows a circuit for battery backup. The diode is used for protection.

For the adventurous: the interrupt output (active low) may be wired to one of your vectored interrupt lines VI0-VI7. The same interrupt line must be directed to the CPU or interrupt controller and the appropriate software written to intercept the event. Make sure that you know what you are doing, as interrupts are tricky to use and invariably difficult to debug. (But those things make software fun!)

DRIVER SOFTWARE

The data sheet for the clock will tell you the details on setting and reading your new clock. On page 50 are some routines written in Pascal to get you started. Note that there is no range checking or other necessary things for 'good' software.

Bon Voyage!

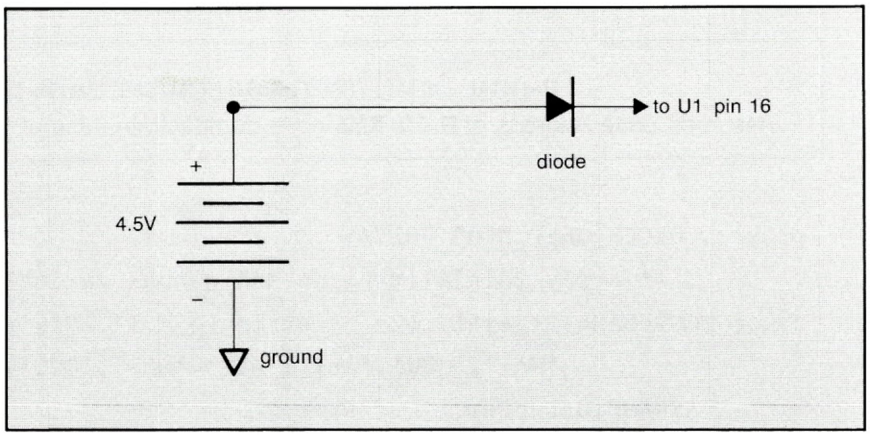


Figure 8. *Optional circuit for battery back-up.*

ADDRESSES

JDR MICRODEVICES
1224 S. Bascom Ave
San Jose, CA 95128
(800) 538-5000

VECTOR ELECTRONIC
12460 Gladstone Ave.
Sylmar, CA 91342
(818) 365-9661

▶S-100 @ COMDEX

(continued from page 17)

NNC ELECTRONICS

NNC offers several multiuser Z80B-based systems running THEOS. A variety of floppy, hard disk, and tape drives is available. Typical systems have ten serial ports and one Centronics port.

NNC also offers a line of S-100 enclosures, including 8-slot or 19-slot motherboard, fan, and some with provisions for internal drives. Front panels can be customized as required.

Send Editorial Feature Reply Card to NNC Electronics, Inc., 15641 Computer Ln., Huntington Beach, CA 92649, for more information.

L/F TECHNOLOGIES

L/F Technologies demonstrated its 800 and 1600 series of multiuser supermicros previously described in the New Products section of S-100 Journal. These are fully-expandable, TurboDOS-based systems with full network capabilities and 8-bit/16-bit processing using 80186 and Z80B CPUs. Concurrent DOS is also available for the 1600 series.

The latest system, the LFT 1650, supports 30 users, each with a dedicated 80186 slave processor and 1 Megabyte of RAM, and takes advantage of all advanced TurboDOS multiuser and network features.

For more about these systems, please send an EF Reply Card to L/F Technologies, 2800 Lockheed Way, Carson City, NV 89701.

(continued on page 63)

PASCAL DEMO PROGRAM FOR THE 58174 CLOCK CHIP

Assumes Base Address of E070. This code demonstrates use of the chip and is not bulletproof!

```
program CLOCK_CHIP_DEMO_58174;
         (* array initializations and global variables *)
const MONTHNAME:array[1..12] of string[3]= ('JAN','FEB','MAR','APR',
                'MAY','JUN','JLY','AUG','SEP','OCT','NOV','DEC');
      DAYNAME:array[1..7] of string[3]=
              ('SUN','MON','TUE','WED','THR','FRI','SAT');
      YEARCODE:array[0..3] of byte=(8,4,2,1);  (* used to set leap year *)
var YEAR:integer;           (* global variable since clock doesnt store *)
    TIME:array[1..12] of byte;  (* working space *)

procedure SNDCLOCK(ADD,VALUE:byte); (* ADD= chip address 0-15; VALUE is BCD *)
BEGIN
  port[$E070+ADD]:=VALUE and $0F;   (* mask off 4 lsb's *)
END;

function GETCLOCK(ADD:byte):byte;   (* ADD= chip address 0-15; returns BCD *)
BEGIN
  GETCLOCK:=port[$E070+ADD] and $0F;(* mask off 4 lsb's *)
END;
```

RESET CLOCK

```
procedure RESETCLK;
BEGIN
  SNDCLOCK(0,0);    (* force out of test mode *)
  SNDCLOCK(15,0);   (* no interrupts *)
END;
```

GETTIME

```
procedure GETTIME;            (* write time to terminal *)
var I:integer;
BEGIN
  I:=GETCLOCK(1); (* discard; will probably be 15 *)
  repeat
    for I:=1 to 12 do TIME[I]:=GETCLOCK(I); (* starting with 1/10 sec *)
  until GETCLOCK(1)=TIME[1];  (* reget 1/10 sec and verify no changes *)
  I:=TIME[10];                  (* weekday number *)
  if ((I>0) and (I<=7)) then write(DAYNAME[I],' ');  (* weekday *)
  I:=TIME[12]*10+TIME[11];    (* month number *)
  if((I>0) and (I<=12)) then write(MONTHNAME[I],' ');
  writeln(TIME[9],TIME[8],' ',TIME[7],TIME[6],':',TIME[5],TIME[4],':',
          TIME[3],TIME[2],'.',TIME[1]);    (* DD HH:MM:SS.T format *)
END;
```

SETTIME

```
procedure SETTIME;           (* enter current time + write to chip *)
var I,MONTH,DATE,HOUR,MINUTE:integer; C:char;
BEGIN
  writeln('ENTER YY MM DD HH MM'); write('      ');
  readln(YEAR,MONTH,DATE,HOUR,MINUTE);   (* year is global; others local *)
  TIME[12]:=MONTH div 10; TIME[11]:=MONTH mod 10;
  TIME[9]:=DATE div 10;   TIME[8]:=DATE mod 10;
  TIME[7]:=HOUR div 10;   TIME[6]:=HOUR mod 10;
  TIME[5]:=MINUTE div 10; TIME[4]:=MINUTE mod 10;
  write('ENTER DAY OF WEEK AS NUMBER SMTWTFS=1234567:');
  readln(TIME[10]);
  SNDCLOCK(14,0);  (* stop clock *)
  SNDCLOCK(13,YEARCODE[YEAR mod 4]);   (* set years from leap year *)
  for I:=4 to 12 do SNDCLOCK(I,TIME[I]);    (* cant set seconds *)
  write('HIT RETURN TO START CLOCK AT EVEN MINUTE:');
  readln;                               (* wait for return *)
  SNDCLOCK(14,255);                     (* start clock *)
  write('CLOCK STARTED  ');
END;
```

DEMO STARTS HERE

```
BEGIN                (* demo starts here *)
  RESETCLK;          (* force not test mode, no interrupts if not already *)
  repeat             (* repeat loop forever *)
    repeat
      GETTIME;                (* write time to screen repeatedly *)
    until keypressed;         (* exit from inner loop when key is pressed *)
    SETTIME;                  (* allow setting of new time *)
  until false;       (* return to time printing *)
END.
```

MACROTECH'S V-RAM

A DUAL-PURPOSE BOARD

Jay Vilhena

Half a megabyte of Static RAM for your S-100 system will cost about $800 (usually at discount prices) and from 1 to 8 slots of your motherboard. You can, with little difficulty, save the extra 6 or 7 slots by buying high-density boards, but you will not be able to shave much off the $800. However, a way exists to get a free RAM-disk tossed in the deal. It's called V-RAM.

V-RAM FEATURES

As you might have noticed, high-density Static S-100 RAM boards have started to appear. Not too long ago, the maximum Static RAM that you could get in one S-100 slot was 64K. Now your slot can accommodate from ½ Meg to... hold on till the next issue, I've been asked not to tell yet.

Anyway, MACROTECH got ahead of the pack and, as early as last year, had a 512K V-RAM on the market. Now, while other first-version high-density Static RAM boards are only starting to show up, the V-RAM is already on its second, debugged version.

The V-RAM is unique and a deal. Its price does not differ significantly from other Static RAM boards of the same size, but the V-RAM carries the extra capability of doubling as a RAM-disk with on-board battery back-up. A simple operation, removing or installing a couple of jumpers, decides whether the board becomes RAM-disk, system RAM, or both.

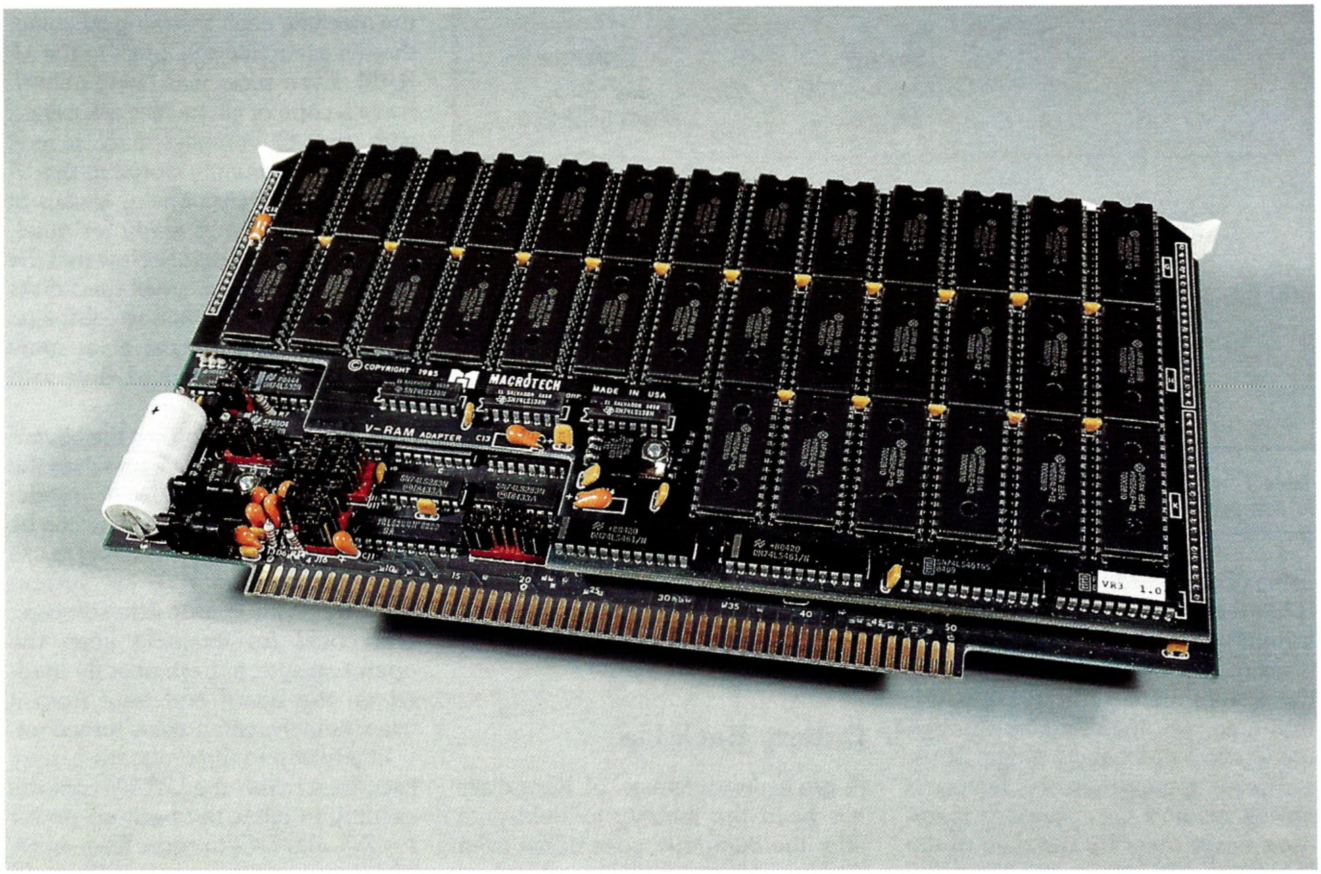
V-RAM as System Memory

When jumper J12 on the V-RAM is set to the right, the board performs as system memory. An extra ½ Megabyte of highly-reliable and fast Static RAM becomes available to your system. The V-RAM adheres to IEEE-696 specification so there should be no incompatibility problems adopting it as the system memory of any S-100 computer that follows the standard. The board can be accessed by either 8-bit or 16-bit CPUs.

For older S-100 systems, or new systems that do not fully comply with the 696 standard, the V-RAM manual describes a few common remedies to get the board to run properly. But, even if the remedy necessary for it to run on your system is not listed, I'm certain that Macrotech is willing to answer questions that you might have on how to adapt the board for your S-100 computer.

The V-RAM board can be set so that its address starts at any 64K boundary within the 16 Megabytes of extended addressing. Explanation: There are 24 lines in the S-100 bus that are used to select the address of each byte in memory. Since each line can represent a 1 or a 0, the 24 lines can represent any binary number from 000000000000000000000000 to 111111111111111111111111. That's a total of 16,777,216, or 2^{24}, possible addresses. Storing one byte at each address gives a potential memory capacity of approximately 16 Megabytes. Originally, the S-100 bus had only 16 address lines; when 24 address lines were defined, the capability to address all 24 was named extended addressing

So that multiple memory boards can be used in a computer system, each board must be assigned a starting address from the possible 16+ million. The V-RAM can be set up to start at any address that is a multiple of 65,536 (the same as 10000 in hexadecimal). Note that each block of 65,536 addresses can hold 65,536 bytes which is defined as 64K. This



is why it is said that the base address of the board can be set to any 64K boundary. For example, if you already had a 128K memory board in your system, the last address available would be address 131,071, so you would want to set the V-RAM to start at address 131,072 (20000 hexadecimal). This is the boundary where the third 64K (bank) starts. You could go on adding memory boards, with each base address starting where the last board ended, until you filled the entire 16-Megabyte memory space (you would need 32 V-RAM boards for example).

And by the way, the V-RAM is also available as a 256K memory board, but in this case there is no RAM-disk capability.

V-RAM as RAM-Disk

Set jumper J13 to the right, and your V-RAM becomes a RAM-disk or, as Macrotech calls it, a Macro-Drive. A RAM-disk acts exactly like a disk system except that, instead of a disk drive and diskettes, you have memory chips on a board, extremely fast access speed, and no moving parts.

The Macro-Drive feature of the V-RAM is designed to be compatible with any software written for CompuPro's M-Drive/H, a popular Dynamic-memory RAM-disk. Up to eight V-RAMs (or a combination of V-RAM and M-Drive/H boards) can be used together for a maximum of 4 megabytes of RAM-disk.

Like the M-Drive/H, the V-RAM Macro-Drive communicates with the master CPU through two S-100 I/O ports. This information is irrelevant if you have RAM-disk driver software (simply make certain that the port address jumpers are set correctly — see below). However, if you need to write your own drivers, examine the article and example listings by Leonard Schwab in this issue of S-100 Journal.

A RAM-disk is one of those items that, once you have one, you wonder forever how you were able to survive without it. The increase in execution speed is more dramatic, obviously, in operations that use the disk intensively, as in compiling programs. But even with trivial operations, such as loading a program or a database into memory, a RAM-disk can save enough time to more than justify its purchase.

To give a rough idea of just how much time a V-RAM Macro-Drive can save, I made two informal comparisons of operations involving disk access (V-RAM versus a QUME 8″ floppy drive). In the first comparison, I compiled and linked BILL, my C program that plucks from a subscriber file naughty S-100 Journal subscribers who have not paid for their subscriptions and sends them a bill. In the second comparison, I simply measured how long it took to load into memory a text file of about 100K. The results are in Table 1 and speak for themselves. Obviously these times are very dependent on what programs are used, the type of hardware, the operating system, and many other variables, so the absolute values are meaningless, but the relative values demonstrate the point.

RAM-Disk AND System Memory

By Setting both jumpers, J12 and J13, to the right, the V-RAM is ca-

	Using Qume 8-inch Drive	Using V-RAM Macro-Drive
Compile and Link BILL	4 min. 27 sec.	39 sec.
Load a Text File	20 sec.	12 sec.

Table 1. *The V-RAM versus a Qume 8-inch Drive. Comparison of the time required to compile and link a C program and to load a text file into memory.*

pable of performing simultaneously as RAM-disk and system memory. Unfortunately, while on this mode, there is no means of physically separating the system memory from the RAM-disk area, i.e., they overlap through the whole ½-Megabyte range. Thus, for many applications, this feature is inconsequential since memory operations easily mess up the data of the RAM-disk and vice versa. I suppose there might be some way of writing memory/RAM-disk management software to take better advantage of this feature, but it seems hardly worth the trouble.

Nevertheless, it should be very useful to leave J12 and J13 on (to the right) so that both the RAM-disk and the 512K of system memory are always accessible without having to open the computer to reset the jumpers. The key here, to avoid loss of data, is NOT to use the two features simultaneously. For example: you have 1 M-byte of memory and the upper 512K is a V-RAM board. Configure two versions of your operating system, one for 512K and the other for 1024K (1M) of memory. Keep (or install) the RAM-disk drivers with the 512K version. Now, when you want to use 512K of memory and the RAM-disk, boot with the 512K version of the operating system. And when you want to use 1024K of memory and no RAM-disk, boot with the 1024K version. You never have to bother with the jumpers.

In addition, the V-RAM capability of performing simultaneously as disk and memory, could be useful to hackers and in software development since it permits looking at the contents of a disk file while accessing the board as system memory.

Battery Back-Up

A great disadvantage of Ram-disks has been the inability to hold data after the computer goes off or even during a short power failure. You had to constantly be saving your RAM-disk files to real disks lest your files were no more. Some solutions to this problem have appeared, but usually they involve the extra purchase of a battery, connecting it to the board via a wire, finding a place to mount it on the frame, worrying about replacing it, and other hassles.

Macrotech solved the problem. A relevant feature of the V-RAM is its standard battery-support circuitry and Ni-Cd battery conveniently premounted on the board. The board circuitry takes care of automatically recharging the battery whenever the system is on, and switching the RAM to battery power when the system goes off or the power fails. According to Macrotech data, it takes about 1 hour of charge to supply 150 hours of back-up power, and a full battery will keep on supplying power for at least 50 days. This means that for systems that are used daily, or even just 2 or 3 days per week, the charge-discharge cycle is totally automatic.

I've worked on this article for 3 days, keeping my file in the V-RAM Macro-Drive. Although I copied the file to a real disk each day as a precautionary measure, when I switched on the machine each following morning, there was my file still intact in the V-RAM. Even nicer was being able to leave a copy of all the files associated with the C compiler and several utilities permanently stored in the V-RAM. There they remain, always at your fingertips, and ready for quiet, lightning-fast operation. Now that the V-RAM is available, small hard disks are obsolete (well, almost obsolete; eight V-RAMS still cost a lot more than a 4-Megabyte hard disk with controller).

I have been pointing out the advantages of the battery back-up when the V-RAM is in Macro-Drive mode. However, the back-up scheme can be equally useful for those wanting to use the board primarily as system memory. With appropriate software, one could for example keep the operating system permanently loaded on the board and boot from it every time the computer is turned on.

In addition to the on-board-battery back-up scheme, the V-RAM contains circuitry to allow the back-up power (~ 2.3 volts DC) to come from an external battery through S-100 pin 21 (Note: Pin 21 is an S-100 bus uNDEFined pin. With a minor modification, the V-RAM can instead accept external back-up power from any other uNDEFined pin). An external battery could be useful to extend the duration of back-up.

The V-RAM is also capable of lowering the RDY, PWRFAIL*, or NMI* lines of the S-100 bus to signal the CPU when the board's source of power, the +8-volt line, is unreliable for proper operation. Which of these three lines is driven is jumper-selectable.

Other Features

The V-RAM consists of a 5-layer PC host board and a 4-layer adapter board. Only the host board resides on an S-100 slot, and the attached adapter board does not interfere with the use of the next slot on the bus. The boards are populated with low-power 8K×8 Static CMOS RAM chips(HM6264LP-12).

When I first examined the V-RAM, I was disappointed to see that the RAM chips had no sockets. This of course makes it very difficult for a user to change a bad chip. I called Macrotech to find out the company's philosophy on not using sockets. Well, it seems that a large percentage of the units returned for repair (of an earlier RAM board) had failed because of bad contacts between the chips and the sockets. So they decided to eliminate the problem. Macrotech points out that the V-RAM boards are burned-in for 48 hours, that the board carries a 1-year warranty, and that it is unlikely that any chip will fail after the warranty period. Although I am still not totally convinced, the argument does seem valid.

In any case, I am impressed by Macrotech's direct support of its customers. Besides offering a 1-year warranty on the V-RAM, the company is willing to answer questions that a user might have about the board or about installing it. After the warranty is over, if for any reason the board fails, it can be returned directly to Macrotech for repair. Most out-of-warranty repairs are $50 to $100, and usually include an upgrade to the latest board version.

INSTALLING AND USING V-RAM

A set of twenty-five easily-accessible box jumpers is used to select the various options, address, and I/O ports of the board (see Table 2).

Selecting the Base Address

To use the V-RAM as system memory, you must first select a base address as explained earlier. Then you must set jumpers J4 through J11 to that address. This is done as follows: Each of these jumpers represents one bit, for a total of eight bits. If a jumper is set to the left, the value of the corresponding bit is 0; if the jumper is set to the right, the value of the bit is 1. The pattern of 0's and 1's defined by the eight jumpers represents an 8-bit binary number. That binary number represents the last 64K **after** which the address starts.

Here is an example to clarify what I just said: Suppose that you already have 256K of memory and that you want the V-RAM to start immediately afterwards. Since 256K contains

NOTICE TO SUBSCRIBERS

Please Read This

YOU DID NOT MISS AN ISSUE. There is not a Winter/86 S-100 Journal. To keep up with the real seasons, we decided that we'd better call this issue Spring (maybe it should have been Summer). Please pardon our abandon with publication schedules. We should soon conquer the production cycle and settle into a regular schedule. In the meantime, we prefer being late rather than sacrificing quality.

EACH 1-YEAR SUBSCRIPTION entitles you to **4 issues** (remember, we 'are' a quarterly). Regardless of when the issues come out, your 1-year subscription will not be up until you've received 4 issues (or 5 if you subscribed with a bonus-issue offer).

IF YOU ARE WORRIED that an issue might have missed you, or have any other subscription problem, please feel free to write or call us (919-839-0115). All written queries that we receive about a **subscription** problem are processed and all problems corrected while the magazine is in printing (the 2 weeks before the magazines go out). At that time, we also send you a reply. If you must have an earlier answer, please call.

THIS ISSUE LOOKS A LITTLE DIFFERENT from the previous two. We continue to look for ways of improving S-100 Journal. This issue has more information packed into it and more pages. 10,000 copies of this issue have been printed, compared to 6,000 for number 2. We've also introduced the REPLY DECK that most of you have received. We hope that it will make things easy for communicating with suppliers of products that you read about in S-100 Journal.

YOUR CONTINUED SUPPORT of S-100 Journal is extremely important for the magazine to remain successful. Here are a few areas where you can help a great deal:

1. Let us know your opinion. Tell us where we do well and where we goof.
2. Please renew your subscription early. We will not bother you with super-early renewal notices, but of course the sooner we have your payment, the more financial leverage we have in producing the magazine and in buying quality articles. A subscription renewal card is included in the deck.
3. If you don't see S-100 Journal in a bookstore or computer retailer that sells other computer magazines, ask for it. Let them know that we are in Raleigh, North Carolina, so they'll be able to locate us. This is often the only way to get alternative magazines like S-100 Journal into bookstores.
4. Any other ideas that you might have to support the Journal are always welcome. Thank you for your support.

4 times 64K, your V-RAM is to start immediately **after** the 4th 64K. Thus change 4 into an 8-bit binary number, arriving at

0 0 0 0 0 1 0 0.

So, the jumpers

J4-J5-J6-J7-J8-J9-J10-J11

should be set to

left-left-left-left-left-right-left-left.

In addition, the V-RAM manual contains a table showing the jumper settings for all possible base addresses.

Some operating systems automatically detect how much memory is available in the computer (provided that all boards have the addresses set right). For others, it is necessary to run a small program, or procedure, that configures a copy of the operating system for the required memory size. Consult your OS manual and installation notes to see if you need to reconfigure the system to the new memory size after you add the V-RAM.

Selecting the Port Address

To use the V-RAM as RAM-disk, the address of the first I/O port must be set correctly. This is done with jumpers J24 through J18. If the jumper is up, it corresponds to 0; if the jumper is down, it corresponds to 1. Again, the objective is to obtain the address as an 8-bit number. Note that there are only 7 jumpers; the least significant bit (the rightmost digit in a binary number) has no jumper to represent it, being always 0.

Example: For CompuPro systems, the V-RAM-disk first I/O port must be set to 0C6 (hexadecimal), which is

1 1 0 0 0 1 1 0

in binary. So the seven jumpers

J24-J23-J22-J21-J20-J19-J18

should read

1 1 0 0 0 1 1

or, in other words

down-down-up-up-up-down-down.

The address of the second I/O port is always consecutive to the first, so it needs no jumper setting.

Independently of the port address, you must tell the system how many V-RAM-disks (up to eight) are in your computer by assigning each one a number. The first board is board 0, the second board 1, etc. Then use jumpers J3-J1 to assign each board its number in binary (right=0). For example, the fourth board is board 3, or 011 in binary, so set jumpers J3-J2-J1 to right-left-left.

JUMP	FUNCTION	SETTING
J1-J3	Selects nth RAM-disk board	See text
J4-J11	Selects base address	See text
J12	Sets board as system memory	Right = Enabled
J13	Sets board as RAM-disk	Right = Enabled
J14	Pull-up resistor for PHANTOM* line	Left@ = Connected to bus
J15	PHANTOM* recognition	Right@ = Enabled
J16	Pull-up resistor for sXTRQ* line	Left@ = Connected to bus
J17	External (via bus) battery	Up@ = Disconnected from bus
J18-J24	Selects port address	See text

J25	Selects wait or interrupt signal	Open@ = not connected to bus
		Up = Drives line 12 (NMI*)
		Right = Drives line 13 (PWRFAIL*)
		Down = Drives line 72 (RDY)

Table 2. *The jumpers of the V-RAM board and their functions. Typical settings for jumpers J14-J17 and J25 are indicated by @. For setting the other jumpers see text.*

Software Drivers for RAM-Disk

If you have an operating system supplied by CompuPro (CP/M 2.2, CP/M 816, CP/M-68K, Concurrent DOS 816), you should already have the software to drive M-Drive/H. The same software will drive the V-RAM. If you have an older version of a CompuPro operating system without M-Drive software, you can usually get an upgrade (for a charge) from a CompuPro System Center. Under a CompuPro environment, the V-RAM first needs to be formatted with a program called MFORM; it then becomes the M: 'disk' drive. Drive M: is used by the user and the operating system exactly like drives A: and B:, but you will love the speed and silence.

For those with other systems, V-RAM driver software needs to be written and patched into the operating system, and a formatting program written. The article by Leonard Schwab (in this issue) demonstrates how this can be done for CP/M 2.2. The principles should hold for other operating systems. (If anyone has written or is planning to write V-RAM, or M-Drive/H, drivers for any other operating system, please contact me. S-100 Journal would be interested in publishing your results.)

WRAP-UP

Because of its dual-purpose capability and clever battery back-up scheme, the V-RAM is one of the best buys in S-100 Static RAM that I presently know. Add to that Macrotech's full 1-year warranty and the company's commendable willingness to support the user directly, and you've got a deal that you truly cannot refuse.

For more information on the V-RAM, write or send an Editorial Feature Reply Card to:

MACROTECH
9551 Irondale Avenue
Chatsworth, CA 91311
800-824-3181
818-700-1501 (Calif.)

Listing continued from page 45

```
;       SET PARAMETERS FOR EVERY RAM-DISK ACCESS OPERATION
;
        MVI C,  SECLEN                  ;BYTES TO BE MOVED = 1 SECTOR
        LHLD    DMAADR                  ;[HL] = MEMORY SOURCE OR DEST'N
        LDA     DSKCMD                  ;GET I/O OPERATION CODE
        ANI     RDCMD                   ; IF NOT READ
        JZ      PUTRAM                  ;    THEN DO WRITE OPERATION
                                        ;    ELSE FALL TO READ OP'N
;
;       MOVE [C] BYTES FROM RAM-DISK TO >>[HL]
;
GETRAM: IN      HDATA                   ;GET BYTE FROM M-DRIVE
        MOV M,  A                       ;STORE IN MEMORY
        INX H                           ;NEXT BYTE
        DCR C                           ;COUNT DOWN
        JNZ     GETRAM                  ;... TILL FINISHED
        XRA A
        RET                             ;ALWAYS RETURNS ZERO - SUCCESS
;
;       MOVE [C] BYTES FROM >>[HL] TO RAM-DISK
;
PUTRAM: MOV A,  M                       ;BYTE TO STORE
        OUT     HDATA                   ; .. TO RAM-DISK
        INX H                           ;NEXT BYTE
        DCR C                           ;COUNT DOWN
        JNZ     PUTRAM                  ;... TILL FINISHED
        XRA A
        RET                             ;ALWAYS RETURNS ZERO - SUCCESS
;
;END RAM-DISK-I/O SECTION
;-------------------------------------------------------------------------
;
;START COLD-BOOT SECTION
;
;       EXTERNAL SUBROUTINES:
;               MOVCPM, lo$mem - WARM-BOOT SECTION
;
CBOOT:  lxi sp, 100h                    ;temporary stack
        call    hwinit                  ;initialize hardware as needed
;
        LDA     WRCMD                   ;WRITE TO RAM-DISK ...
        CALL    MOVCPM                  ; FROM HI MEMORY
;
        call    sgnon                   ;display sign-on message
        call    lo$mem                  ;initialize low-memory
        xra     a                       ;
        mov c,  a                       ;let CCP set default disk A:
        jmp     ccp                     ;enter ccp - AUTOLOAD active
;
hwinit:     ;put hardware initialization here
        ret
;
sgnon:      ;put routine to display sign-on message here
        ret
;
;END COLD-BOOT SECTION
;-------------------------------------------------------------------------
;
```

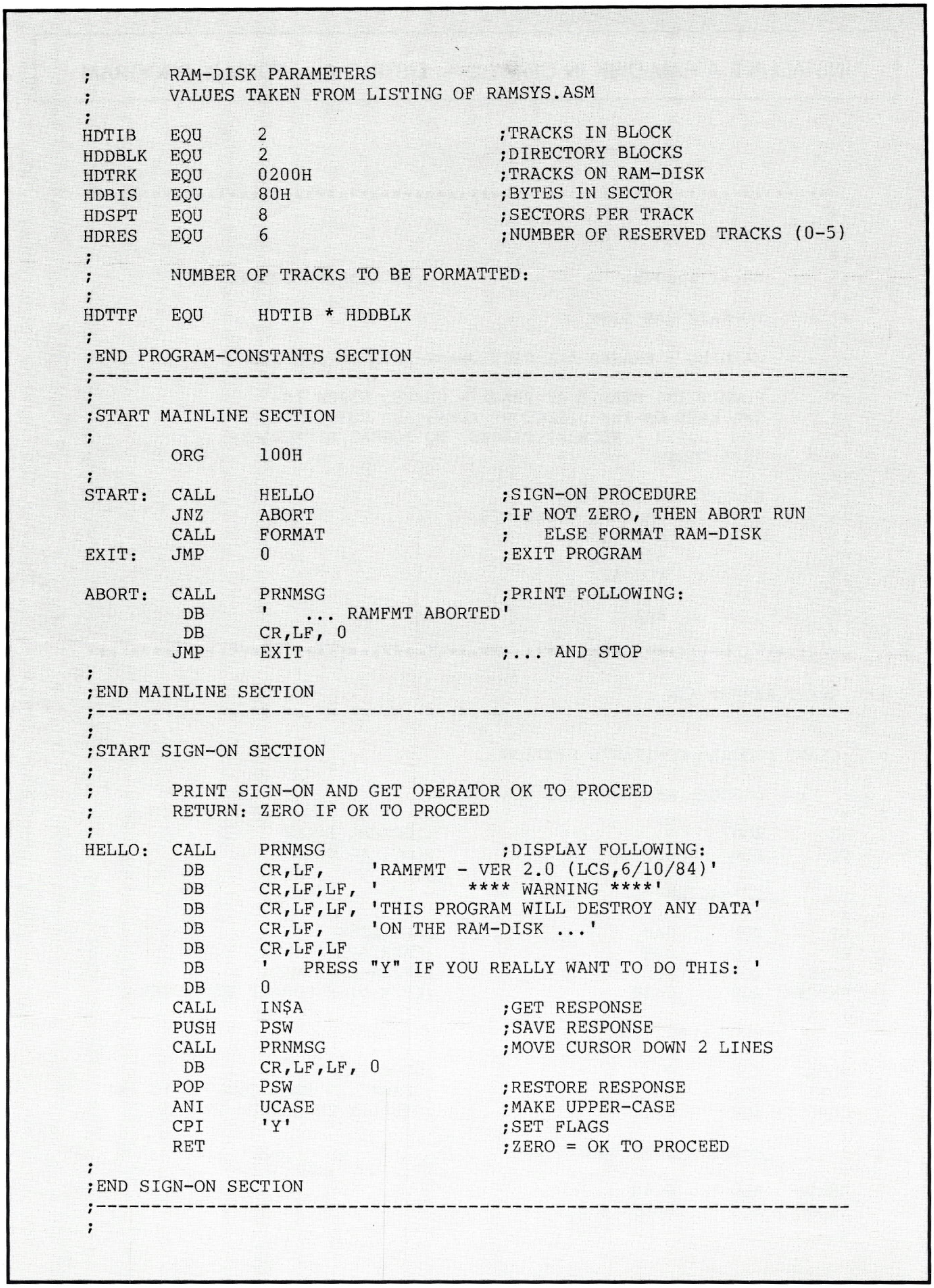
```
;START UNINITIALIZED-DATA-AREA SECTION
;
;       FOLLOWING CODE REDEFINES COLD-BOOT SECTION
;       FOR USE BY OS AFTER COMPLETION OF COLD-BOOT
;
        ORG     CBOOT
;
COUNT:  DW      0                   ;USED BY MOVCPM ROUTINE
;
;       DISK ACCESS DATA AREA
;
DISK:   DB      0                   ;DRIVE SELECTED
DSKCMD: DB      0                   ;DISK OPERATION CODE
TRACK:  DW      0                   ;TRACK-NUMBER
SECTOR: DB      0                   ;SECTOR-NUMBER
DMAADR: DW      0                   ;>> DISK I/O BUFFER
ALVRAM: DS      HDDSM+1             ;ALLOCATION-VECTOR AREA FOR RAM-DISK


DIRBUF: DS      SECLEN              ;COMMON BUFFER FOR DISK DIRECTORIES
;
ENDLOAD:
;
BIOSLEN EQU $ - BIOS                ;LENGTH OF THIS MODULE
;
;
;END UNINITIALIZED-DATA-AREA SECTION
;-----------------------------------------------------------------------
```

INSTALLING A RAM-DISK IN CP/M 2.2 — LISTING 3 — INITIALIZE RAM-DISK

```
;RAM.SUB - SUBMIT FILE TO INITIALIZE RAM-DISK
;
;RUN RAM FORMATTING PROGRAM:
ramfmt
;
;CREATE NULL TRANSIENT PROGRAM, IN ORDER TO REPEAT PIP COMMAND
save 0 @.com
;
;TRANSFER UTILITIES TO RAM-DISK
pip c:=pip.com
@ c:=sub.com
;etc
;
;LOAD WORKING BIOS - CONTROL WILL TRANSFER TO NEW BIOS
; AND RAM:DISK WILL BE A:
ramasys
;
;END RAM.SUB
```

INSTALLING A RAM-DISK IN CP/M 2.2 — LISTING 2 — FORMAT PROGRAM

```
;*********************************************************************
;*
;*      title: RAMFMT.ASM
;*
;*      date: 15NOV85                       by Leonard Schwab
;*
;*      FORMATS RAM-DISK
;*
;*      WARNING - ERASES ALL DICTIONARY-DATA ON RAM-DISK
;*
;*      FORMATTING STARTS AT TRACK = HDRES, WHICH IS
;*      THE BASE OF THE DIRECTORY-AREA, AND CONTINUES
;*      FOR (HDTIB * HDDBLK) TRACKS, TO FORMAT DIRECTORY-
;*      AREA ONLY.
;*
;*      RAMFMT.ASM SECTIONS:
;*               PROGRAM-CONSTANTS
;*               MAINLINE
;*               SIGN-ON
;*               FORMAT
;*               SYSTEM-SUPPORT
;*               DATA
;*
;*********************************************************************
;
;START RAMFMT.ASM
;--------------------------------------------------------------------
;
;START PROGRAM-CONSTANTS SECTION
;
;       LOGICAL AND PHYSICAL CONSTANTS:
;
NO      EQU     0                       ;LOGICAL FALSE
YES     EQU     NOT NO                  ;LOGICAL TRUE
;
;       CHARACTERS:
;
LF      EQU     0AH                     ;NEW LINE
CR      EQU     0DH                     ;CARRIAGE-RETURN
UCASE   EQU     5FH                     ;UPPER-CASE MASK
FMTCHR  EQU     0E5H                    ;CP/M DISK-FORMAT CHARACTER
;
;       BDOS FUNCTIONS
;
BDOS    EQU     05                      ;BDOS ENTRY
CONIN   EQU     01                      ;CHARACTER FROM CONSOLE IN [A]
CONOUT  EQU     02                      ;DISPLAY CHARACTER IN [E]
;
;       M-DRIVE PORT ADDRESSES
;
HDATA   EQU     0C6H
HADDR   EQU     HDATA + 1
;
```



```
;       RAM-DISK PARAMETERS
;       VALUES TAKEN FROM LISTING OF RAMSYS.ASM
;
HDTIB   EQU     2                     ;TRACKS IN BLOCK
HDDBLK  EQU     2                     ;DIRECTORY BLOCKS
HDTRK   EQU     0200H                 ;TRACKS ON RAM-DISK
HDBIS   EQU     80H                   ;BYTES IN SECTOR
HDSPT   EQU     8                     ;SECTORS PER TRACK
HDRES   EQU     6                     ;NUMBER OF RESERVED TRACKS (0-5)
;
;       NUMBER OF TRACKS TO BE FORMATTED:
;
HDTTF   EQU     HDTIB * HDDBLK
;
;END PROGRAM-CONSTANTS SECTION
;---------------------------------------------------------------------
;
;START MAINLINE SECTION
;
        ORG     100H
;
START:  CALL    HELLO                 ;SIGN-ON PROCEDURE
        JNZ     ABORT                 ;IF NOT ZERO, THEN ABORT RUN
        CALL    FORMAT                ;   ELSE FORMAT RAM-DISK
EXIT:   JMP     0                     ;EXIT PROGRAM

ABORT:  CALL    PRNMSG                ;PRINT FOLLOWING:
         DB     '   ... RAMFMT ABORTED'
         DB     CR,LF, 0
        JMP     EXIT                  ;... AND STOP
;
;END MAINLINE SECTION
;---------------------------------------------------------------------
;
;START SIGN-ON SECTION
;
;       PRINT SIGN-ON AND GET OPERATOR OK TO PROCEED
;       RETURN: ZERO IF OK TO PROCEED
;
HELLO:  CALL    PRNMSG                ;DISPLAY FOLLOWING:
         DB     CR,LF,    'RAMFMT - VER 2.0 (LCS,6/10/84)'
         DB     CR,LF,LF, '         **** WARNING ****'
         DB     CR,LF,LF, 'THIS PROGRAM WILL DESTROY ANY DATA'
         DB     CR,LF,    'ON THE RAM-DISK ...'
         DB     CR,LF,LF
         DB     '   PRESS "Y" IF YOU REALLY WANT TO DO THIS: '
         DB     0
        CALL    IN$A                  ;GET RESPONSE
        PUSH    PSW                   ;SAVE RESPONSE
        CALL    PRNMSG                ;MOVE CURSOR DOWN 2 LINES
         DB     CR,LF,LF, 0
        POP     PSW                   ;RESTORE RESPONSE
        ANI     UCASE                 ;MAKE UPPER-CASE
        CPI     'Y'                   ;SET FLAGS
        RET                           ;ZERO = OK TO PROCEED
;
;END SIGN-ON SECTION
;---------------------------------------------------------------------
;
```

```
;START FORMAT SECTION
;
;       FORMAT-PROCEDURE MAINLINE.
;       NOTE: INITIAL VALUES OF TRACK, SECTOR AND COUNT
;       ARE SET DURING COMPILATION (SEE DATA SECTION).
;
FORMAT: CALL    SETRAM                ;SET-UP RAM-DISK
        CALL    FMTRAM                ;FORMAT ONE SECTOR
        CALL    NEXT                  ;SET-UP NEXT SECTOR
        JNZ     FORMAT                ;REPEAT UNTIL FINISHED
        RET
;
;       INCREMENT SECTOR/TRACK
;       RETURN: ZERO WHEN SPECIFIED TRACKS ARE FORMATTED
;
NEXT:   LXI H,  SECTOR                ;LAST SECTOR-NUMBER
        INR     M                     ;NEXT SECTOR-NUMBER
        MVI A,  HDSPT                 ;IF SECTORS-PER-TRACK...
        CMP     M                     ; .NE. NEXT-SECTOR-NUMBER
        RNZ                           ;   THEN SECTOR-NUMBER IS OK
        XRA     A                     ;   ELSE START NEW TRACK:
        MOV M,  A                     ;SECTOR = 0
        LXI H,  TRACK                 ;INCREMENT TRACK
        INR     M                     ;
        LXI H,  COUNT                 ;DECREMENT...
        DCR     M                     ;  TRACKS TO FORMAT
        RET                           ;ZERO = FINISHED
;
;       SET RAM-DISK ADDRESS
;
SETRAM: LDA     SECTOR                ;SEND SECTOR-NUMBER ...
        OUT     HADDR                 ; TO RAM-DISK
        LHLD    TRACK                 ;SEND TRACK-NUMBER ...
        DAD     H                     ;* 2
        DAD     H                     ;* 4
        DAD     H                     ;* 8
        DAD     H                     ;* 16
        DAD     H                     ;* 32
        DAD     H                     ;* 64
        DAD     H                     ;* 128
        MOV A,  H                     ;  HIGH-BYTE
        OUT     HADDR                 ;    TO RAM-DISK
        MOV A,  L                     ;  LOW-BYTE
        OUT     HADDR                 ;    TO RAM-DISK
        RET
;
;       WRITE ONE SECTOR OF FORMAT-BYTES TO RAM-DISK
;
FMTRAM: MVI A,  FMTCHR                ;FORMAT BYTE
        MVI C,  HDBIS                 ;BYTES IN SECTOR
FMTLUP: OUT     HDATA                 ; .. TO RAM-DISK
        DCR     C                     ;COUNT DOWN
        JNZ     FMTLUP                ;... TILL FINISHED
        RET
;
;END FORMAT SECTION
;---------------------------------------------------------------------
;
```

```
;START SYSTEM-SUPPORT SECTION
;
;       CONSOLE INPUT
;
IN$A:   MVI C,  CONIN                   ;READ CONSOLE ...
        JMP     BDOS                    ; VIA CP/M
;
;       DISPLAY IN-LINE MESSAGE
;
PRNMSG: XTHL                            ;GET >>MESSAGE IN [HL]
        CALL    PRNLUP                  ;DISPLAY MESSAGE
        XTHL                            ;SET TO RETURN AFTER MESSAGE
        RET
;
;       PRINT MESSAGE >>[HL] TO NULL-BYTE
;
PRNLUP: MOV A,  M                       ;GET CHARACTER
        ORA A                           ;IF ZERO ...
        RZ                              ;   THEN EXIT
        PUSH    H                       ;SAVE POINTER
        CALL    OUT$A                   ;DISPLAY CHARACTER
        POP     H                       ;RESTORE POINTER
        INX     H                       ;POINT NEXT
        JMP     PRNLUP                  ;AND REPEAT
;
;       DISPLAY CHARACTER IN [A]
;
OUT$A:  MOV E,  A                       ;PASS CHARACTER [A]
        MVI C,  CONOUT                  ;  TO CONOUT FUNCTION
        JMP     BDOS                    ;  IN CP/M
;
;END SYSTEM-SUPPORT SECTION
;----------------------------------------------------------------------
;
;START DATA SECTION
;
TRACK:  DW      HDRES
SECTOR: DB      00
COUNT:  DW      HDTTF                   ;ONLY LOW-BYTE IS USED
;
;END DATA SECTION
;----------------------------------------------------------------------
        END                             ;RAMFMT.ASM
```

bits

FOR SALE

Two S-100 systems. System 1: 22 slot TEI frame; TDL ZPU (Z-80) and SMB 2; I/O card; Hayes Micromodem; 8K Bytesaver; Micropolis Mod II dual drives. System 2: 12 slot TEI frame; CompuPro Z-80 cpu; 2 CompuPro Interfacer 1's; CompuPro Disk 1; 4 Tandon 8″ DSDD drives with cases, supplies. Each system has a CompuPro RAM 17 64K memory board; Heath/Zenith H-19 terminal; CP/M 2.2; considerable software and disks; all cables. Both systems up and running, all drives like new. Asking $900 each. Harvey, (Los Angeles) (818) 784-9501.

FOR SALE

COMPUPRO 816/A Complete system - Includes: 8085/88 CPU, Disk 1, System Support, Interfacer 4, RAM22 (256K) - Comes with Two (2) 8″ Floppy Drives, Heath Terminal, all necessary cables and manuals, CP/M 816 and Supercalc. Asking $1,100.00. Also, KONAN Hard Disk Controlelr DGC-100. Asking $195.00. Call or Write: Greg Puckett, 6470 West Fork Rd., Cincinnati, Ohio 45247. (513) 761-5610.

SOFTWARE WANTED

Looking for user written or Public Domain Software for Cromemco 'Dazzler Board.' Write to: Duane Erwin, 335 So. Wabash, Bradley, IL 60915.
(PS: Running an IMSAI 8080 56K CP/M 8″ Disk)

SOFTWARE WANTED

I need help finding a CP/M 2.2 BIOS for my VERSAFLOPPY II DISK CONTROLLER BOARD. I spent $100.00 for the one from SD Systems which I have working with CP/M 1.4. The best I have been able to obtain from SD Systems is a source listing for a newer ROM which still is not CP/M 2.2 compatible. Please write or call: Ron Simpson, 122 S. Madison Ave., Louisville, CO 80027. (303) 665-2535.

This department is for publishing non-commercial small advertisements. There is no charge for subscribers to place an ad. Readers may take advantage of this service to sell or buy used hardware, trade personal programs, etc. Messages must be kept short. Send ads to S-100 Journal, BITS, PO Box 12881, Raleigh, NC 27605.

▶S-100 @ COMDEX

(continued from page 49)

TELETEK

Teletek has entered the LAN market by offering TurboNET, a full-featured local-area-network scheme implemented under the TurboDOS operating system.

At COMDEX, Teletek had a network of S-100, terminal workstations, and PC/terminal workstations. In the TurboNET LAN, PC/terminals (such as ATs, XTs, Z-100s, or S-100s) can access the full resources of the main S-100 host and, in addition, benefit from their local processing capabilities. A multiuser/network system built with Teletek components would typically be able to run all of CP/M, CP/M-86, and MS-DOS/PC-DOS software.

At the heart of the TurboNET is an IEEE-696 S-100 Network Interface Board that supports CSMA (Carrier Sense Multiple Access) standard networking protocols and fast transfer speeds.

Teletek also offers a line of quality S-100 single board computers, both master and slave, with 8-bit or 16-bit CPUs. The slave processors can be used to bring 16-bit capability to 8-bit systems (or vice versa) and to add workstations to a multiuser or network system.

The company provides a convenient 30-day evaluation of boards, with return privileges, to system integrators. To request more information on the Teletek line of S-100 products, send a completed Editorial Feature Reply Card to Teletek, 4600 Pell Drive, Sacramento, CA 95838.

This special COMDEX report substitutes the New Products section for this issue.

We have described most of the new and not-so-new S-100 products present at COMDEX. However, because of the huge size of the show, limitations on our schedule, and our involvement at COMDEX in the formation of SITA (The S-100 International Association), it was not possible to visit every single S-100 company at the show. We'll catch the others next time.

Needless to say, not all S-100 companies are interested in the market that COMDEX provides, so this report is biased toward higher-end products. Several of the products shown at COMDEX have only now (Spring 86) become available for sale, so you can still be the first on your block (or office building) to have one.

For more information on any of the products mentioned, write to the company, or, better yet, send them one of our Editorial Feature Reply Cards.

Jay Vilhena, Robert Petersen, and Linda Pereira contributed to this report.

s-100 directory

$20 **PER ISSUE BUYS THIS SPACE**

List your business or product in this S-100 Directory. All listings must be S-100 related and messages must have 30 words or less. We do all typesetting free. Send listing w/ $20 to S-100 Journal, Directory Listings, PO Box 12881, Raleigh, NC 27605.

COCHLIN COMPUTER SYSTEMS INC.

330 Townsend St. #107, San Francisco, CA 94107
(415) 495-5999

Authorized CompuPro System Center. We provide on-site service and maintenance, custom software design, multi-user system integration, local area networking, user training and more.

MENTZER COMPUTER SYSTEMS

1441 Rollins Rd., Burlingame, CA 94010
(415) 340-9363

CompuPro Full Service System Center. Custom S-100 systems for Accounting, Medical, and Legal applications. Repair and maintenance of all brands of S-100 systems.

MICRO POWER UNLIMITED

1716 Erring Rd. #102, Simi Valley, CA 93065
(805) 584-6789

Authorized CompuPro System Center. We provide on-site service and maintenance, custom software design, multi-user system integration, user training and more.

JURIS SCRIVENER, INC.

2019 B Hollywood Blvd., Hollywood, FL 33020
(305) 920-7772

System Integrator and Vendor. S-100 Granturbo Computers running TURBODOS 8 and 16 bit, will also run CP/M and MS-DOS. Systems for 4 users and up. FMS 80 Data Base Management Software for S-100 computers. Primage printers.

SMALL BUSINESS SYSTEMS, INC.

PO Box 635, 9537 Jackson Ave
Brookfield, IL 60513
(312) 485-8338

CompuPro multi-user systems, networking, peripherals, software. Fast, reliable, and versatile systems for any business computer needs.

RENARD SYSTEMS INC.

4248 Ridge Lea Road, Amherst, NY 14226
(716) 833-4758

Service offices in NYC and Rochester. Full Service CompuPro System Center specializing in word processing automation, AutoCad on S-100, laser printers, very large hard disk drives, and tape backup. Service contracts to entire North East (out of NYC). National distributors for a variety of system enhancing software products, including PrintMan, Phantom Printer, Smart-Op and Speed-Op by Data Base Administrators.

DIGITZ

3015 Hillsborough Street, Raleigh, NC 27607
(919) 828-5112

Your source for terminals, printers, and Zenith Z-100 components. Repair and Service Center. One week turnaround on most products.

CO-SY, COMPUTER UND SYSTEME

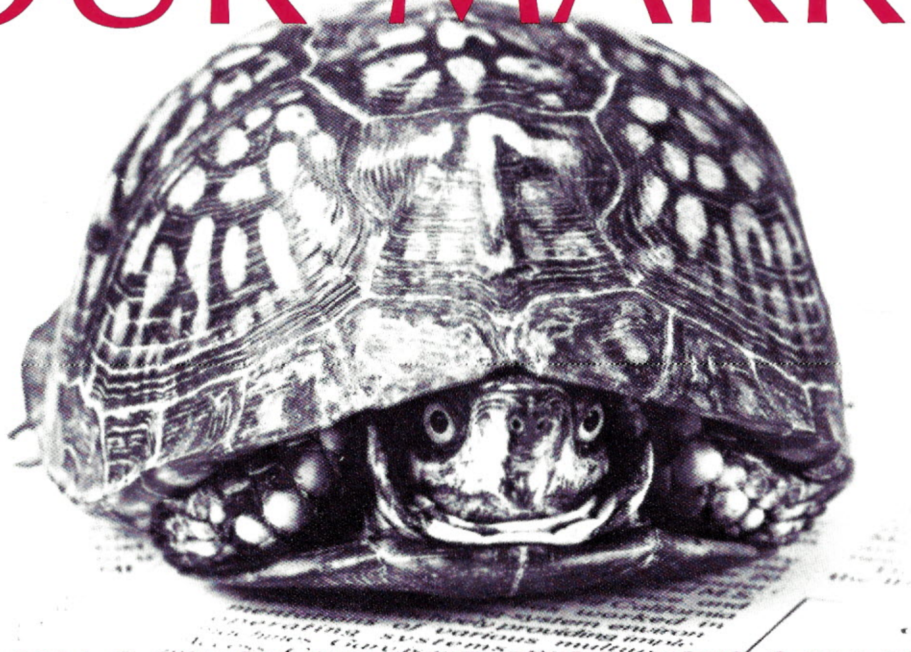
Krischerstr. 70, d-4019 Monheim, West-Germany
(2173) 52071 or 72

We are looking for cooperation with Software manufacturers-authors to adapt and translate Software into German. We specialize in UNIX and C language, MS-DOS and C, and C-Basic.

$140 **BUYS THIS SPACE FOR TWO YEARS**

That's only $17.50 per issue. All listings must be S-100 related. Messages must be 30 words or less. Message can be changed for each issue. We do all typesetting free. Send first listing with $140 to S-100 Journal, Directory Listings, PO Box 12881, Raleigh, NC 27605.

DON'T HIDE FROM YOUR MARKET



IF YOU SELL S-100 OR RELATED PRODUCTS, S-100 JOURNAL IS THE NATURAL ENVIRONMENT TO ADVERTISE THEM.

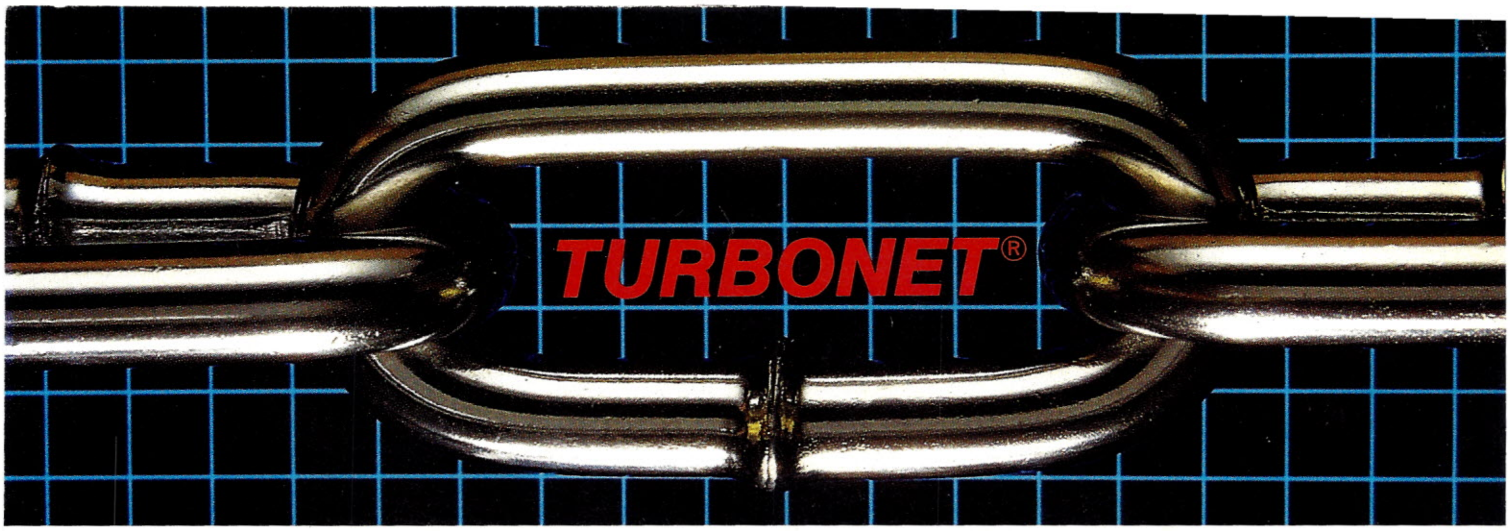
Look at all the advantages that only S-100 Journal offers:

1. Nearly 100% of our readers are either S-100 users or people considering the purchase of an S-100 system. This unusual market concentration makes S-100 Journal competitive (for advertisers of S-100 products) with magazines having circulations of 50,000 to 100,000; yet, we only charge a fraction of the price.

2. Every advertisement of ⅓ page or larger automatically receives a Free Reply Card in our unique Reply Deck. Design the card yourself, or go with our design customized to your ad. Either way, you pay only the price of the magazine ad.

3. No charge for typesetting. If your ad has less than 200 words, we will typeset it at no additional charge. Simply send us the text and any photos that you wish to go with it.

Why do we do all this? Because we are committed to bringing the S-100 bus to the prominent position that it deserves in the marketplace. If you succeed, we succeed, and all S-100 users gain. So, get out of your shell and join us. Give Jay a call today at 919-839-0115 and reserve some space out here in your natural environment.



One Strong Link Can Strengthen Your Whole System

TurboNET® is Teletek's new local area network that permits reliable high speed data transfers among computers of differing bus architectures. TurboNET is designed to be used in networks consisting of multiple S-100 based and PC based systems. It will allow up to 4000 users, including up to 255 IBM PCs or compatibles, to share a single network and all attached peripherals. The network can be organized in any number of different ways mixing Teletek's 8 and 16-bit multiuser systems and PCs in any combination.

Teletek's Networking Family consists of:

TurboNET PC: IBM-PC Network Interface Board

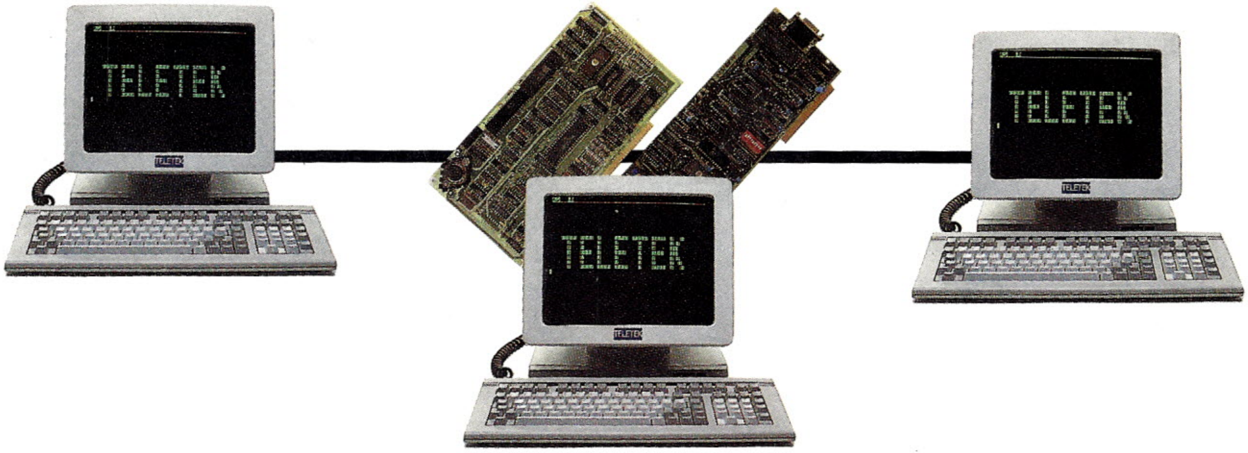
Teletek's TurboNET PC board offers IBM-PC Compatibility, CSMA industry standard protocols, 2 megabits/sec network speed, on-board CPU and communications management firmware, and media independence.

TurboNET S-100: S-100 Network Interface Board

Teletek's TurboNET S-100 board offers IEEE 696 Compatibility, CSMA industry standard protocols, 2 megabits/sec network speed, on-board CPU and communications management firmware, and media independence.

The benefits are obvious: The cost savings of shared peripherals, almost unlimited system expansion capability, and the use of existing PC workstations with the ability to run the myriad of application software written for MS-DOS and PC-DOS. This coupled with Teletek's 8 and 16-bit multiuser systems running application software written for CP/M and MP/M allows the system the ability to access almost any software library.

For more information on Teletek's TurboNET S-100 and TurboNET PC boards or on any of our full line of S-100 products, please call our Sales Department at 916-920-4600.



TELETEK

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