FEBRUARY 1978 VOLUME 3, Number 2 $2.00 in USA

BYTE

the small systems journal

You can now have the industry's finest microcomputer with that all-important disk drive

The professional-grade microcomputer for professionals

Shown with optional bench cabinet

YOU CAN GET THAT ALL-IMPORTANT SOFTWARE, TOO

Loading your programs and files will take you only a few seconds with the new Cromemco Z-2D computer.

You can load fast because the Z-2D comes equipped with a 5″ floppy disk drive and controller. Each diskette will store up to 92 kilobytes.

Diskettes will also store your programs inexpensively—much more so than with ROMs. And ever so much more conveniently than with cassettes or paper tape.

The Z-2D itself is our fast, rugged, professional-grade Z-2 computer equipped with disk drive and controller. You can get the Z-2D with either single or dual drives (dual shown in photo).

CROMEMCO HAS THE SOFTWARE

You can rely on this: Cromemco is committed to supplying quality software support.

For example, here's what's now available for our Z-2D users:

CROMEMCO FORTRAN IV COMPILER: a well-developed and powerful FORTRAN that's ideal for scientific use. Produces optimized, relocatable Z-80 object code.

CROMEMCO 16K DISK BASIC: a powerful pre-compiling interpreter with 14-digit precision and powerful I/O handling capabilities. Particularly suited to business applications.

CROMEMCO Z-80 ASSEMBLER: a macro-assembler that produces relocatable object code. Uses standard Z-80 mnemonics.

ADVANCED CONTROLLER CARD

The new Z-2D is a professional system that gives you professional performance.

In the Z-2D you get our well-known 4-MHz CPU card, the proven Z-2 chassis with 21-slot motherboard and 30-amp power supply that can handle 21 cards and dual floppy drives with ease.

Then there's our new disk controller card with special features:

- Capability to handle up to 4 disk drives
- A disk bootstrap Monitor in a 1K 2708 PROM
- An RS-232 serial interface for interfacing your CRT terminal or teletype
- LSI disk controller circuitry

Z-2 USERS:
Your Z-2 was designed with the future in mind. It can be easily retrofitted with everything needed to convert to a Z-2D. Only $935 kit; or $1135 for assembled retrofit package.

We're able to put all of this including a UART for the CRT interface on just one card because we've taken the forward step of using LSI controller circuitry.

STORE/FACTORY

Contact your computer store or Cromemco factory now about the Z-2D. It's a real workhorse that you can put to professional or OEM use now.

Kit: Z-2D with 1 disk drive (Model Z2D-K) .... $1495.
Assembled: Z-2D fully assembled and tested (Model Z2D-W).... $2095.
Additional disk drive (Model Z2D-FDD) .... $495.

SOFTWARE

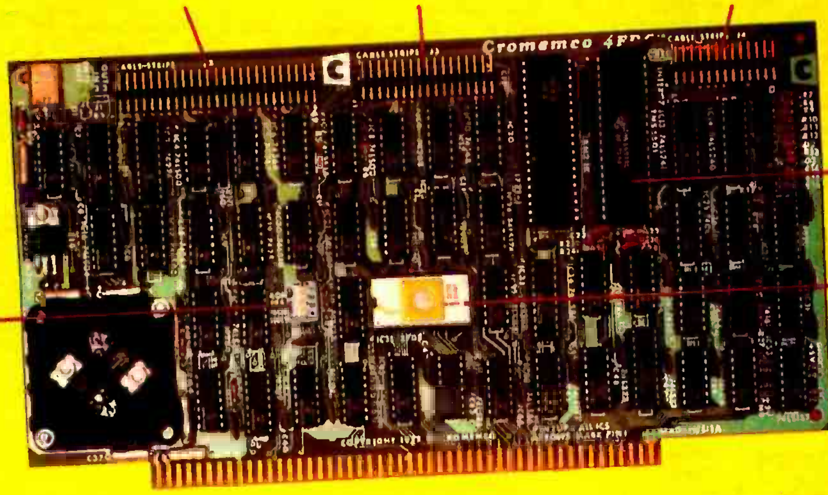
(On standard IBM-format soft-sectored mini diskettes)

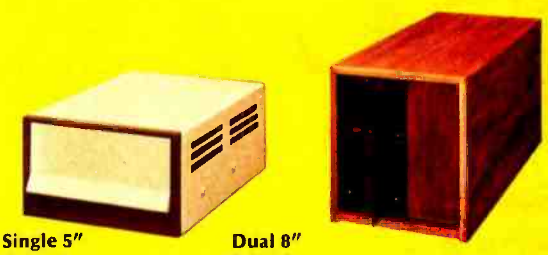
16K BASIC (Model FDB-S)......... $95
FORTRAN IV (Model FDF-S)........ $95
Z-80 Assembler (Model FDA-S).... $95

Cromemco
incorporated
Specialists in computers and peripherals
2400 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400

Circle 34 on inquiry card.

The easy way to get disk storage, FORTRAN IV, and other programming power





Single 5" disk drive

Dual 8" disk drive

Here's a new disk controller and disk drive combination that will set you up for truly powerful disk storage.

The new controller is extremely versatile. You can use it with **either** our **new 5" single disk drive** or our **8" dual disk drive.** In fact, the controller will **interface up to three** 5" or **four** 8" drives.

That means you can have enormous disk storage since the new controller puts 92 kilobytes on each side of a 5" diskette and 256 kilobytes on an 8" diskette. Recording is in soft-sectored IBM format.

FORTRAN IV AND MORE

You can get still more Cromemco disk operation aids. For example, we also offer FORTRAN IV for our computer users.

And as in so many things, we are the first manufacturer in the field to offer this advanced program for the Z-80 μP.

Besides FORTRAN IV we also offer our special BASIC (14-digit precision), our Z-80 Assembler, and now an entertainment diskette with over a dozen of our Dazzler® games.

KEYBOARD CONTROL

The new Model 4FDC disk controller (supplied in our Z-2D) is for our Z-2 computer or any S-100 bus computer using our Z-80 CPU card.

You should also know about these other capabilities of the new controller:

- Its PROM-resident Disk Operating System (RDOS) gives you keyboard control of your disk drive and also includes a bootstrap to load our powerful CDOS disk operating system supplied on all Cromemco diskettes.
- The controller will **interface your CRT terminal** through its RS-232 serial port. May save you an I/O.
- It has 5 programmable interval timers.
- It has vectored interrupts.
- And it has an 8-bit parallel input port and an 8-bit parallel output port.

LOOK TO THE FUTURE

This new disk controller equips you for the future as well as for now. Not only can you now have very large storage, but the features of the controller and the standard IBM format protect you from early obsolescence.

STORES/FACTORY

This new card and the disk drives are in production and available.

So contact your computer store or the factory today and you can have the power of FORTRAN IV and a large memory right away.

PRICES

Model 4FDC-K Disk Controller kit	$ 395
Model 4FDC-W Disk Controller assembled	$ 595
Model WFD 5" single disk drive assembled	$ 495
Model PFD-K 8" dual disk drive kit	$1995
Model PFD-W 8" dual disk drive assembled	$2495

Disk drives are complete with power supply, case and cables.

SOFTWARE

Purchasers of Cromemco computers or drives may purchase software on 5" or 8" diskettes as follows:

	5" Diskette Model	8" Diskette Model	Price
FORTRAN IV	FDF-S	FDF-L	$95
Z-80 Assembler	FDA-S	FDA-L	$95
16K BASIC	FDB-S	FDB-L	$95
Dazzler® games	FDG-S	FDG-L	$95

Cromemco
incorporated
Specialists in computers and peripherals
2400 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400

Circle 34 on inquiry card.

In the Queue

BYTE February 1978
Volume 3 Number 2

Foreground

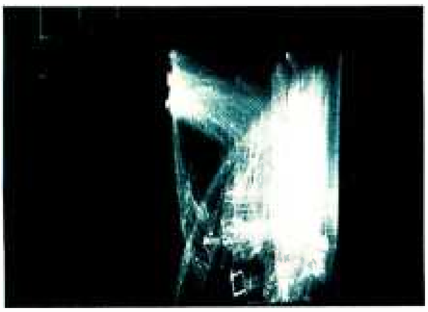
Background

Nucleus

BYTE is published monthly by BYTE Publications Inc, 70 Main St, Peterborough NH 03458. Address all mail except subscriptions to above address: phone (603) 924-7217. Address all editorial correspondence to the editor at the above address. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of BYTE. Address all subscriptions, change of address, Form 3579, and fulfillment complaints to BYTE Subscriptions, PO Box 590, Martinsville NJ 08836. Second class postage paid at Peterborough NH 03458 and at additional mailing offices—USPS Publication No. 102410. Canadian second class registration No. 9321. Subscriptions are $15 for one year, $27 for two years, and $39 for three years in the USA and its possessions. In Canada and Mexico, $17.50 for one year, $32 for two years, and $46.50 for three years. $25 for a one year subscription by surface mail worldwide. Air delivery to selected areas at additional rates available upon request. $25 for a one year subscription by air delivery to Europe. Single copy price is $2.00 in the USA and its possessions, $2.40 in Canada and Mexico, $3.50 in Europe, and $4.00 elsewhere. Foreign subscriptions and sales should be remitted in United States funds. Printed in United States of America.

In This BYTE

Page 6



With good reason, many computers these days have no front panel for low level data entry and display. But sometimes a real time pattern "signature" of an executing program can be useful. This month, Steve Ciarcia shows how to build **A Penny Pinching Address State Analyzer** which can be used with an X-Y oscilloscope to monitor the address bus of your computer in real time. If you build this state analyzer, you'll see a unique pattern corresponding to each "steady state" loop of an executing program.

When is a personal computer more than a personal computer? When it is plugged into a network of personal computers for purposes of message transfer via phone lines, sharing of programs, and perhaps even execution of multiple player logical games. In this issue, Mike Wilber begins a three part series of articles on the concept of **CIE Net: A Design for a Network of Community Information Exchanges.** *Page 14*

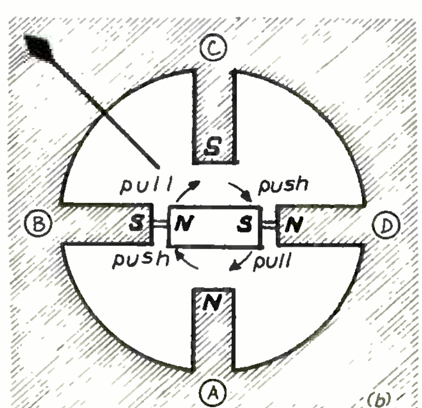
Are you looking for a driver for your model railroad's roundhouse turntable? Perhaps you need some motive power for a robot. For generation of controlled rotary motion, stepper motors as described in Robert E Bober's article **Taking the First Step** are essential. He provides readers with valuable background information on these fascinating mechanical outputs for personal computers. *Page 35* →

When is a boat like a raft of bottles? When it is conceptually chopped into many individual buoyancy elements for the purpose of simulating its performance on a choppy water surface?

In this issue Stephen P Smith continues his series of articles on the simulation of motion in personal computers with a discussion of motion of extended objects in **Simulation of Motion, Part 4: Extended Objects, Applications for Boating.** Turn to Stephen's article for more details and a BASIC program which simulates rolling or pitching motions of an arbitrary boat hull cross section. *Page 42*

Interactive editing is enhanced when a light pen can be used to zero in on a text location. See how to **Add a $3 Light Pen to Your Video Display** using the combination of hardware and software techniques provided by John Webster and John Young in this issue. *Page 52*

If you own a KIM-1 computer, here's an answer to the perpetual problem of entering and debugging large programs. Dan Fylstra's article **SWEETS for KIM** shows you how to add a mini text editor and assembler that fits in the KIM's 1 K bytes of programmable memory and still leaves room for your programs. *Page 62*



Page 162

In past BYTEs Mike Wilber and Dave Fylstra have suggested the concept of a "Community Information Exchange." Read Jeff's **Personal Computers in a Distributed Communications Network** for a discussion of some of the technological (and political) aspects of such a concept, which is well within reach of our present personal computing hardware and software. *Page 80*

Last month, we began Ernest W Kent's series of articles on **The Brains of Men and Machines.** The discussion continues this month with the next installment, **How the Brain Controls Output.** Aspiring robotics hackers will find this to be an invaluable background input on the information systems found in nature, which can serve as a source for ideas on new information systems designed by humans. *Page 84*

The minifloppy has arrived, as many readers probably know, and its popularity is increasing with time. If you'd like to take advantage of its low cost, then read David Allen's **Minifloppy Interface** and try your hand at adding a minifloppy to your system. *Page 114*

Entomology is the study of bugs. Gary McGath provides some introductory insight into various species of programming bugs, and some general design guidelines to prevent their occurrence in his background article on **Programming Entomology.** *Page 162*

Have you ever needed to experiment with a circuit and ended up rewiring it again and again? Wouldn't it be nice to have a program that simulated the circuit and could be easily modified to change the parameters? Read Robert Grappel's **A Simple Digital Filter** and find out all about filter simulation on your own computer. *Page 168*

About the Cover. . .

The February issue contains two articles on the subject of communications networks for the personal computer user. The idea here is to create a number of noncommercial networking applications based on individual experimenters talking to individual experimenters. One of those articles, by Mike Wilbur (who spends his professional time at Stanford Research Institute), is a hypothetical design specification in several parts for a "Community Information Exchange" network. Mike's article was written as a prelude to work of the PC Net committee headed by Dave Caulkins and including a number of the Homebrew Computer Club members in and around Palo Alto CA.

Using Mike's preliminary design as a model, we asked Robert Tinney to create this fantasy on a theme of Community Information Exchanges. The "Community Information Exchanges" are processing nodes of the CIE network, talking locally with telephone lines and globally via such long distance media as amateur radio (including perhaps satellite links via AMSAT). The "walls" of CIEland represent the boundaries of the protocol of the network: through an appropriate interface the users of the CIE NET protocol can talk to the PC NET protocol through a PC NET "gate," to an ARPA net protocol through an appropriate gate arranged by some friendly local university's computer science department or artificial intelligence laboratory, or to the "X" network where X in the picture is "MUSIC" but could be arbitrary. The CIE as originally envisioned is an amateur and noncommercial group growing out of a common interest in compuer to computer communications. It would in some sense serve as a prototype and innovative test bed for what might later become low cost commercial networks in much the same way that numerous amateur radio innovations are now part of the technology of radio communications. . .CH■

Articles Policy

BYTE is continually seeking quality manuscripts written by individuals who are applying personal computer systems, designing such systems, or who have knowledge which will prove useful to our readers. For a more formal description of procedures and requirements, potential authors should send a selfaddressed, stamped envelope to BYTE Author's Guide, 70 Main St, Peterborough NH 03458.

Articles which are accepted are purchased with a rate of up to $50 per magazine page, based on technical quality and suitability for BYTE's readership. Each month, the authors of the two leading articles in the reader poll (BYTE's Ongoing Monitor Box or "BOMB") are presented with bonus checks of $100 and $50. Unsolicited materials should be accompanied by full name and address, as well as return postage.■

Ciarcia's Circuit Cellar

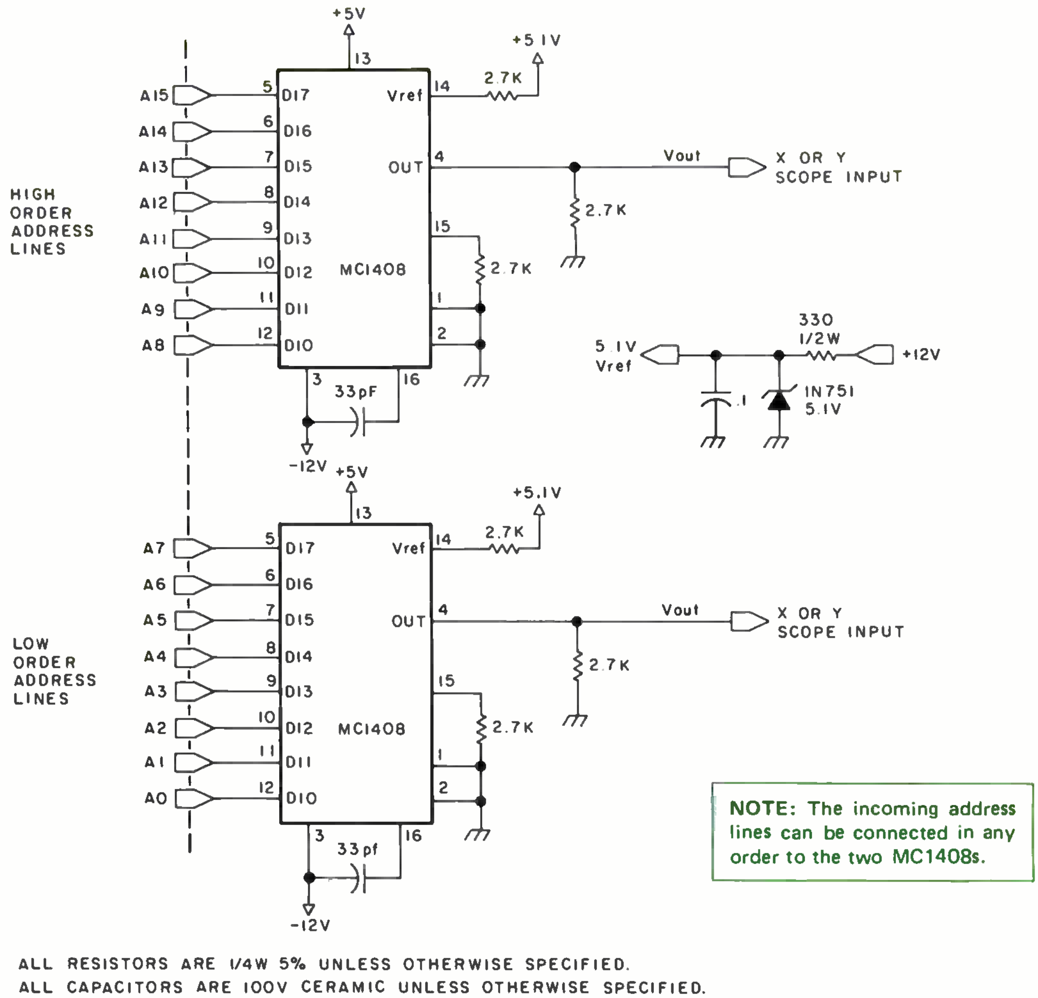


Figure 1: A 256 by 256 point address state analyzer that displays dynamic fluctuations of a computer's 16 bit memory address bus. Two 8 bit digital to analog converters drive the horizontal and vertical inputs of an oscilloscope with the analog equivalent of the eight high order and eight low order address lines. The display gives a visual "signature" of the computer in action. Accessing of unexpected memory locations which distort this "signature" becomes instantly visible for troubleshooting purposes.

A Penny Pinching Address State Analyzer

Steve Ciarcia
POB 582
Glastonbury CT 06033

Three years ago I got my first home microcomputer, a Scelbi 8H. This was before the advent of widespread interest in personal computers and it was naturally based upon the Intel 8008 processor. Back then I was satisfied with the tedious task of hand toggling a program into the computer and watching the front panel memory address and data buffer lights twinkle, signifying that the program was executing something. After that I bought more memory which consisted of 2102s. That gave me enough space to write only the simplest of monitor programs, again using the front panel as the display medium. At the end of its evolution, my 8008 did have a rudimentary video display and 300 bps cassette interface; but, if there was one major physical characteristic of the first generation home computers, it was the predominance of the front panel display and data entry switches. The concept of the integrated home computer "system" was yet to be seen. A computer required display and data entry switches if it was to be powered up and exercised. Additional IO devices such as video displays and keyboards were luxuries.

Well, it was inevitable. The prices of components have dropped drastically in the past few years and the experimenter now thinks in terms of a home computer

Photo 1: The author's computer system, showing the address state analyzer in operation with a BASIC program (see oscilloscope in center of picture). The program is printing an integer sequence of ASCII characters on the display to the right using the function CHR$(X). On the left is a 50 K byte Digital Group Z-80 system.

system incorporating a processor, cassette interface, read only memory systems software, keyboard and video display. Fewer and fewer microcomputers have front panels that display memory lines or data buffers. The memory address in addition to the contents of all other pertinent processor registers is now usually available through a monitor program command. The cost effectiveness of the front panel lights and toggle switches has diminished to the vanishing point.

I would never advocate a return to front panels, by any means. But recall how many times you have checked the rhythmic pulsations of particular lights to assure yourself that your program was executing correctly. Or, how many times have you recognized that the program had obviously vectored off into the unknown by the graphical representation of the 16 address bus lights? Adding 16 lamps on the memory address lines can be done on any microcomputer, and this would give us some indication of what the program is doing. But the LEDs are truly readable only when the processor is in a hold state, halted, or otherwise not changing the memory address. The chances of obtaining a recognizable visual pattern on the LEDs are small when running programs written in languages like BASIC that jump around in memory as they interpret each statement. And with LEDs there are only 16 graphical elements; this gives poor resolution.

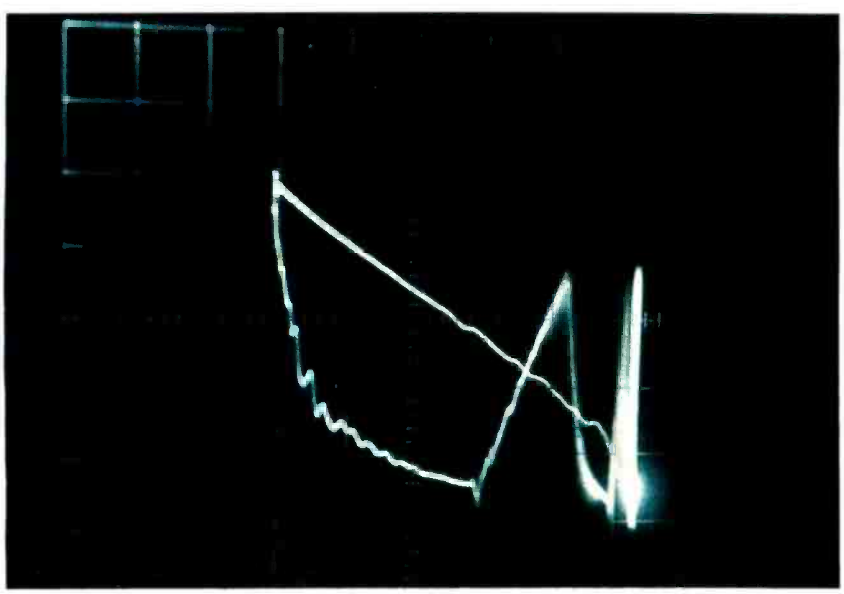


Photo 2: A close-up of the penny pinching address state analyzer display showing the result of a power on reset in the computer. Execution begins in the 256 byte EROM bootstrap program; the program continually vectors to the cassette input port 001 (thin excursion line to the upper left in the photo) to see if data is present.

A $15 Video Analyzer

There is another way to watch the internal program sequence that far exceeds a 16 bit lamp display: a 256 by 256 point analyzer that displays the dynamic fluctuations of the 16 bit memory address bus. This gadget can be added using only two integrated circuits and any X-Y oscilloscope with sufficient bandwidth. The result is a graphical presentation of the computer in action. It is not graphics in the classic sense: no pictures can be drawn, and alphanumeric capability is nonexistent. It is instead a point plot of the memory address states, dynamically changing during the execution of a program.

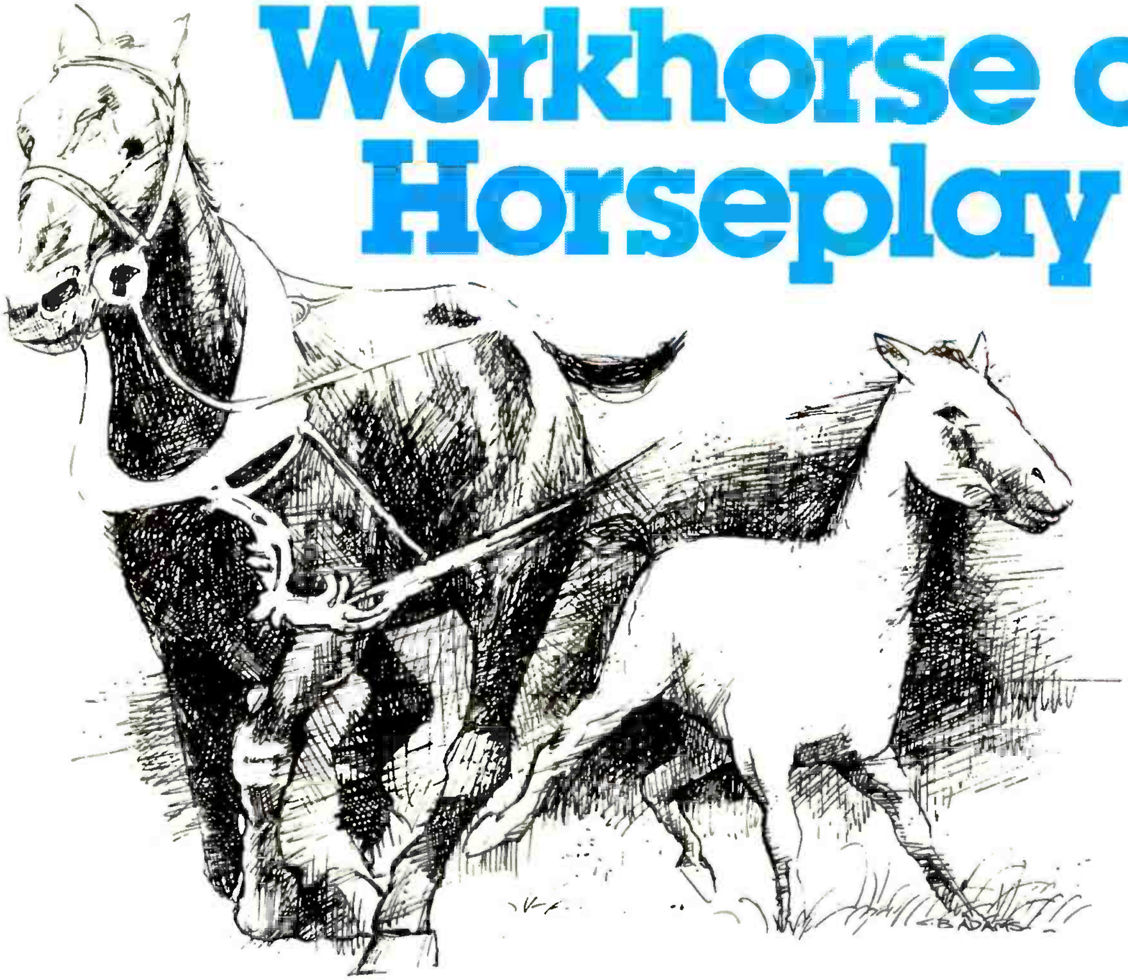
The 6800, 8080, Z-80, 6502 and other processors all have 16 bit address buses. They directly address 64 K bytes of memory (ie: there are 65,536 possible address combinations). The address bus can be divided into eight bits of high order address and eight bits of low order address.

If either of these address portions is attached to the eight input lines of a pair of digital to analog converters, two unique analog voltage values are produced for each address location. The two voltage outputs, *one for high address and one for low address*, can then be attached to the vertical and horizontal inputs of an oscilloscope. The result is a fascinating animated display of a computer in action.

Constructing the State Analyzer

It isn't often that I can outline a design in which layout, physical components, absolute voltages, input and output polarities, or input attachments are so flexible. This 2 chip circuit can be hooked up any way you

Workhorse or Horseplay?



Software systems from TSC are designed for tough business and industrial uses on the job or just plain fun off the job. Whether you are looking for a system to be used primarily in a working situation or a system for the home, look into TSC software.

Assembly Language Programs (Include Source Listings)

SL68-29	6800 Text Processing System*	$32.00
SL68-24	6800 Text Editing System*	$23.50
SL68-26	6800 Mnemonic Assembler*	$23.50
SL68-19	6800 Micro BASIC Plus*	$15.95
SL80-9	8080 Space Voyage	$12.00
SL68-5	6800 Space Voyage*	$12.00
SL68-27	6800 Disassembler	$ 9.00
SL68-28	6800 Program Relocator	$ 8.00
PD80-2	8080 Game Package II	$14.00
SL80-8	8080 Blackjack	$ 6.50
MUB-68	Multi-User System—Write for details	

*Kansas City Standard object code cassette tape available for an additional $6.95. Paper tapes available for some programs. Send 25¢ for complete catalog.

TECHNICAL SYSTEMS CONSULTANTS, INC.
BOX 2574 W. LAFAYETTE INDIANA 47906
317-742-7509

SPECIALISTS IN SOFTWARE & HARDWARE FOR INDUSTRY & THE HOBBYIST

Program-of-the-Month Club™
One year membership for $2.00. Discounts offered with no obligations.

To Order: Include 3% postage, $1.00 handling on orders under $10.00, and Indiana residents add 4% sales tax. Check your dealer!

TSC Monthly Feature:

8080 Text Editing System
At Last! An 8080 version of the famous TSC 6800 Text Editing System.

Written in assembly language, approximately 5k is required allowing local, global, content, character and line operations. For example, with one command you can change the second occurrence of the string "12345" in the next 35 lines to the string "XYZ".

Included are Tabs, Overlay, block Move and Copy, Header, Append and Zones. This Editor is actually better than many large scale computer Editors! Source listing included.

SL80-10	**8080 Text Editing System**	**$28.50**
PT80-3	**Optional Paper Tape**	**$ 9.00**

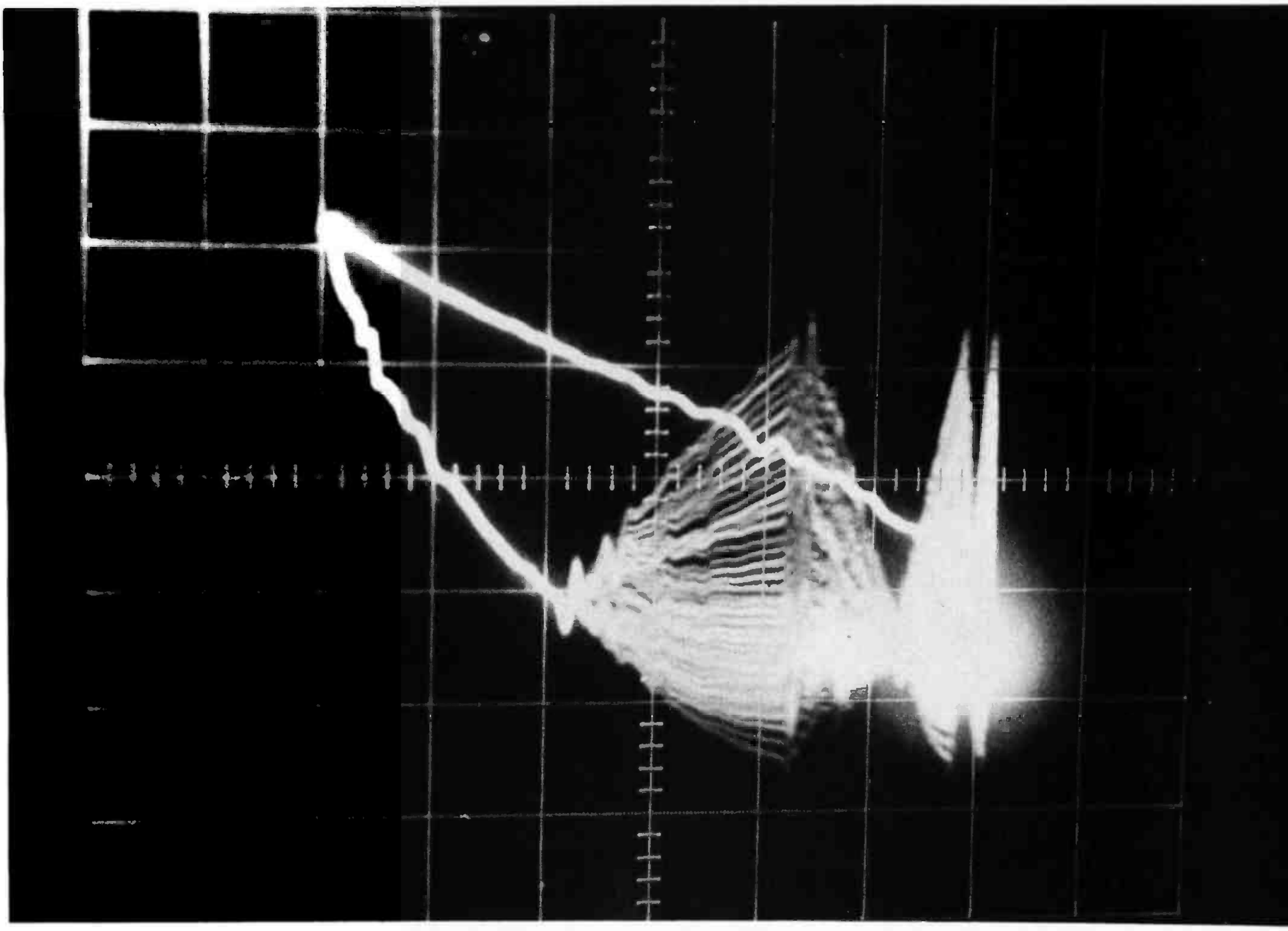


Photo 3: Effect of a carriage return while executing TDL 12 K BASIC. The long streamer shows the program's reference to the input port of the keyboard.

want it. You can mix the address bits between the two digital to analog converters. If you separate the address bytes as I have and attach one byte to each converter, the display tends to dwell in a narrow region along a vertical line.

The 8 bit digital to analog converter I have chosen is the Motorola MC1408. The L6, L7 or L8 version can be used since absolute accuracy is not important. What is being produced is a system signature unique to your system and your programming. Figure 1 illustrates the schematic of the circuit as I attached it to my computer address lines. For a more complete description of how the MC1408 works, see my previous article "Control the World" in September 1977 BYTE, page 30.

Evaluating the Results

The oscilloscope traces in photos 1, 2, 3 and 4 are particular to my system and software only; but similar, though not exact versions, should be produced on other systems. As diagrammed in the schematic, I have used the high order address lines to drive the X axis and the low order address lines to drive the Y axis. Another peculiarity of my system is a logically inverted address bus. The result is that the display moves in the opposite direction from what one might expect. The higher the address, the lower the output voltage. Again, as I stated earlier, human pattern recognition, not methodology, is important.

After attaching this video drive to an oscilloscope, turn on the power. In my case the pattern displayed (as in photo 2) illustrates that the computer is operating in the region of memory occupied by the monitor software; it regularly vectors to another address, that of the cassette input (the 8 bit low order memory address lines of the Z-80 or 8080 are also used to address input and output ports) at port 001. Later, when running BASIC, repeated addressing of keyboard input port 000 can be recognized as in photo 3 taken after TDL 12 K BASIC was loaded.

One of the programs which best illustrates this new visual dimension of the

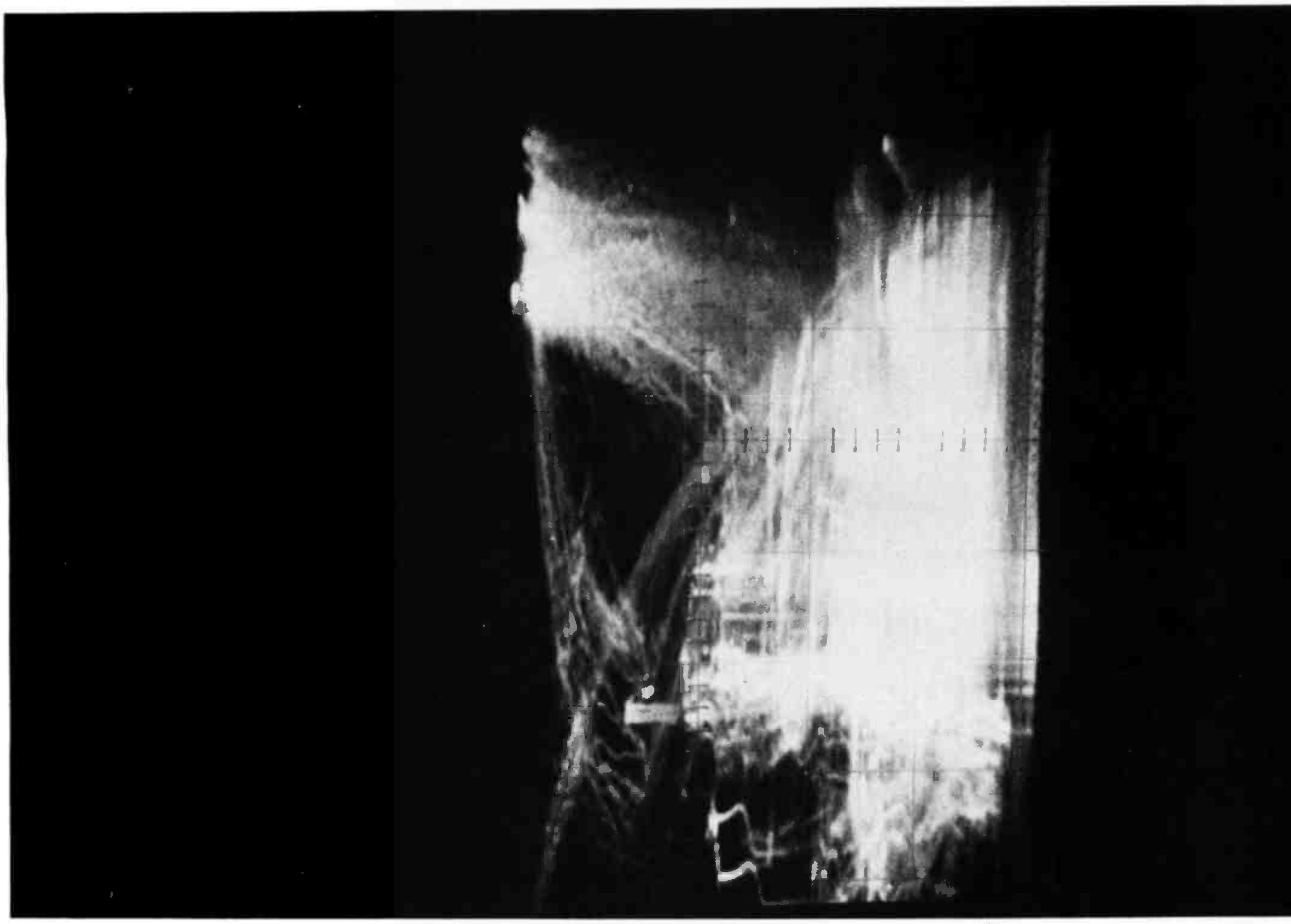
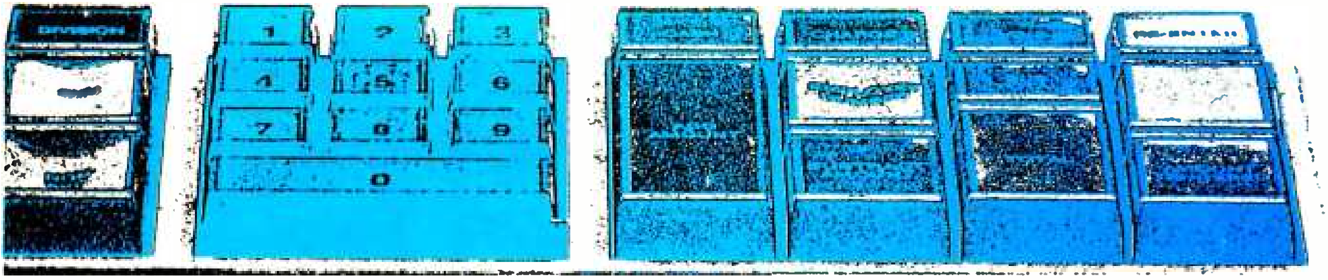


Photo 4: A complex bus addressing pattern during the execution of a BASIC program.

computer is a basic memory test program as it scans through memory. Dynamically varying displays such as these are very difficult to photograph and would appear as blurs. The photos I have included are those of programs with addressing sufficiently repetitive so that the pattern appears stable (see photo 4).

There is one particular instance that proved the worth of the address state analyzer on my system. I had received and was in the process of checking out the TDL 12 K Super BASIC software package distributed by Micro COM for the Digital Group Z-80 system, and was having trouble getting the software to execute in 26 K of memory. Rather than call the company and complain of a possible bad tape, I turned on the address state analyzer and loaded the tape. I could see the computer cycling through the cassette input section of the monitor and depositing it in increasingly higher portions of memory. At its conclusion, the words "Highest Memory" appeared on the screen. I promptly typed in 26000 and hit a carriage return. The computer took off and started doing a scan across memory in a pattern similar to that of a memory test program. Following this, the computer went into visible convulsions (or the electronic equivalent) on the ocilloscope and never returned to the display. I loaded the program once again and this time answered the question with 20000. The result was an introductory blurb indicating that BASIC was fully operational. A quick scan of the 2 K bytes of memory on the processor board verified that they were wired for something other than 24 K to 26 K. The address state analyzer (in which I now had considerably more faith) told the complete story. After replying to the "Highest Memory," the program apparently scanned memory and tried to verify that the typed input was indeed plausible. In a false case it got hung up. Resetting the memory bank decoding circuit for 24 K to 26 K, of course, solved the problem.

Next month: "Programming EROMs with BASIC."■



INTRODUCING SCELBI's 8080 STANDARD EDITOR

Here, at last, is an efficient way to edit text when preparing program source listings or other text material. You'll need an 8080 computer, with a minimum of 2K memory (of which at least 1K should be RAM); a text input device, like a keyboard; and a display/text output device.

OPTIONAL HARDWARE

Additional memory beyond 2K allows expanded text buffer storage area. Recommend 4K-8K for practical applications. Bulk storage I/O devices allow text to be saved for future use/modification.

SOFTWARE REQUIRED

User provided I/O driver routines for whatever I/O devices will be utilized. Each I/O device is linked to the program by a single vector for ease in adapting the program to individual systems.

MEMORY UTILIZED

The assembled listing provided in the manual resides in pages 01 through 05 (hexadecimal which is 001 through 005 octal). Pages 00, part of 05, and all of 06 (hexadecimai-000, part of 005 and all of 006 octal) are left available for user provided I/O routines. Pages 07 (hexadecimal-007 octal) through available memory used for text buffer.

MNEMONICS UTILIZED

This program is written in 8080 machine language standard industry accepted mnemonics for the 8080 CPU (such as MOV A, B; INX H; CALL; etc.) (Note: SCELBI is discontinuing its use of special 8080 compatible mnemonics which have characterized its 8080 programs in the past.)

PROGRAM OPERATION

This is a standard line-oriented text editing program intended for use in the creation of source listings and similar text manipulations. The program operates in two modes; the Text Entry mode for entering text into the text buffer and the Command mode used to specify operator directives. Information in the text buffer may be manipulated using the Command directives and the contents of the text buffer transferred to an external storage device or filled from an external storage device.

PROGRAM COMMANDS

APPEND (A) text to the text buffer; CHANGE (C) text; DELETE (D) text; INSERT (I) text; LIST (L) text; character SEARCH (S); READ (R) from or WRITE (W) to an external storage device; CLEAR text buffer; plus single character deletion, tab (spacing), and various character search directives.

DOCUMENTATION

In the famous SCELBI tradition. The program manual describes the operation of the editor, presents detailed discussions of all major routines with flow charts, contains two completely assembled listings (one with addresses and object code in hexadecimal notation and one in octal notation), and of course includes operating instructions and tips on enhancing the program if desired.

SPECIAL FEATURES

Because the program has been carefully organized and written with all memory references assigned labels, it may be readily reassembled to reside in any general area in memory. This program may even be assembled to reside in just 1K of ROM provided that some RAM area is available for scratch pad and text buffer use!

OPTIONS

A punched paper tape of the object code for this editor (as described in the documentation) is available. The object code tape is provided in the widely accepted "hexadecimal format." Also, the complete, commented source listing of the program as presented in the documentation is available in straight ASCII format on punched paper tape. Fan-fold paper tapes are provided for ease in handling. Additionally, opaque paper tape is supplied to facilitate the use of low cost optical paper tape readers now in widespread use. NOTE: Paper tapes are sold only as optional supplements to the documentation.

OPTIONAL PAPER TAPE NOW AVAILABLE

An optional object code on punched paper tape is available. Specify 8080ED-OPT, $6.00. And you can order optional commented source listing on paper tape too. Specify 8080ED-SPT, $20.00.

ORDER YOUR COPY TODAY...$12.95

SCELBI COMPUTER CONSULTING INC.
Post Office Box 133 PP STN
Milford, CT 06460

Price shown for North American customers. Master Charge, Postal and Bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing, specifications, availability subject to change without notice. SCELBI Books are available in many fine Computer Stores. **IMPORTANT!** Include 75¢ postage/handling for each item delivered by U.S. Mail Book Rate; or $2 for each item shipped First Class or via UPS.

Circle 102 on inquiry card.

About the Author:

Mike Wilber is a member of the techqical staff at SRI International in Menlo Park CA. He has been actively working on the concept and definition of the Community Information Exchange for more than two years, and has a history of work at the frontiers of computer science in areas including network design and artificial intelligence.

CIE Net:

A Design for a Network of Community Information Exchanges

Mike Wilber
920 Dennis Dr
Palo Alto CA 94303

Part 1: The Beginnings

The need for a personal computer telecommunications network is rapidly becoming inescapable.

Motivation and Background: Why Build a Network?

A good, cheap and practical telecommunication network can be extremely useful to the personal information processing community. It can provide a means by which people exchange programs and files of data. For example, a respectable dictionary can be built by 1000 people who each contribute 20 words. Just as important, a good telecommunication facility can help people talk to one another, for instance, to advertise the presence of a good data file, or to explain just when one technique is superior to another. These considerations and others are explored in more detail in an earlier BYTE article. (See "Homebrewery vs the Software Priesthood," by Wilber and Fylstra, October 1976 BYTE, page 90.)

The need for a personal computer telecommunication network is rapidly becoming inescapable. Now that personal computers are economically feasible, manufacturers are selling cheap reliable systems in astounding quantities to personal users of information processing, each of whom stands to gain from freely shared interactive experience. Already, hobbyist clubs and other, more primitive, information exchanges have sprung up to fill the void. Telecommunication can greatly facilitate the free exchange of ideas and data that currently take place on a limited but increasing scale.

How This Effort Got Started

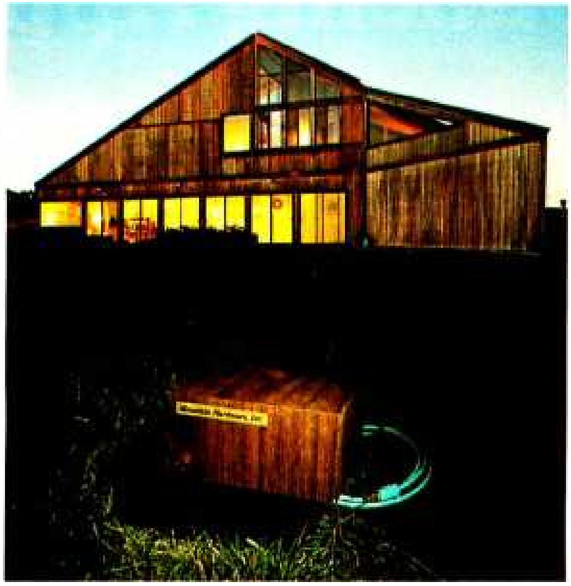
In response to this need, I designed a network for presentation at the First West Coast Computer Faire in April of 1977. That work was unfinished as the Faire proceedings press deadline loomed, so I wrote up the design considerations for publication there. Since then, I have finished the design, of which the main part of this series of articles is a detailed exposition.

I was not alone in feeling this need and responding to it with action. After the Faire, Dave Caulkins organized a group to design and implement a personal computer network; this group has thus taken PCNET as its name. The PCNET committee started from many of the premises I feel are important, but it has identified a slightly different set of problems, and it has solved almost every one of them differently, and so it is developing a design that differs considerably from mine in its details.

The PCNET design was rapidly developing at the press deadline for this article and is thus not detailed here. Its broader aspects

Continued on page 138

The Ultimate Turn-on



On/off control everywhere-by computer over the AC wiring

Now it's simple and economical to control AC devices remotely from an S-100 or Apple II computer. Mountain Hardware's new Introl™ system delivers on/off commands over the existing AC lines — so you don't have to string a foot of wire!

Control at any AC outlet. The Introl system impresses a code-modulated 50 KHz control signal on the house wiring. Then decodes the signal at any outlet to switch AC devices on and off. You can control lights, refrigerators, TVs, solenoid valves, sprinklers, burglar alarms—and many other things we leave to your fertile imagination. With the addition of input sensors to your computer system, you can automatically control variables such as temperature and soil moisture.

Here's how it works. You plug in a single AC Controller board at the computer bus and connect the AC Interface Adapter to any convenient 115 VAC outlet. The AC Controller is now connected to address as many as 64 channels remotely. But it's completely isolated from the 115v power, so there's no chance of short or shock.

At any outlet where you seek control, plug in a Dual Channel AC Remote. Then plug one or two devices to be controlled into the box. Every AC remote has two independent 500 watt channels. When commanded by the computer, the Dual Channel AC Remote turns the devices on and off independently. When polled by the computer, the Dual Channel AC Remote sends a signal back, telling the computer the status of each device. Bidirectional communication provides error free operation.

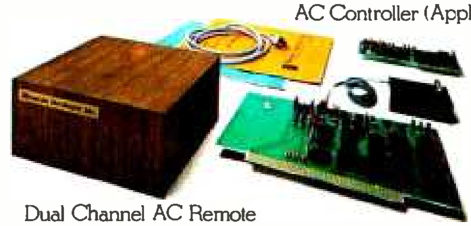
Simple programming. You write your control program in BASIC or Assembler language. Software subroutines for the control programs come with the equipment — along with complete documentation. If you have an S-100 computer, you can program on/off commands at any day and time using our optional 100,000 day Calendar/Clock Board. A self contained power source assures fail safe operation.

Modest prices. The AC Controller, for both the S-100 and Apple II computers, costs $149 in kit form or $189 completely assembled and tested. Each Dual Channel AC Remote costs $99 as a kit or $149 assembled and tested. Thus, a fully operative system in kit form can be yours for as little as $248.

The Calendar/Clock Board for S-100's costs $179 in kit form, $219 assembled and tested.

All prices are f.o.b. Ben Lomond, CA. Prices are USA Domestic. California residents add 6% sales tax.

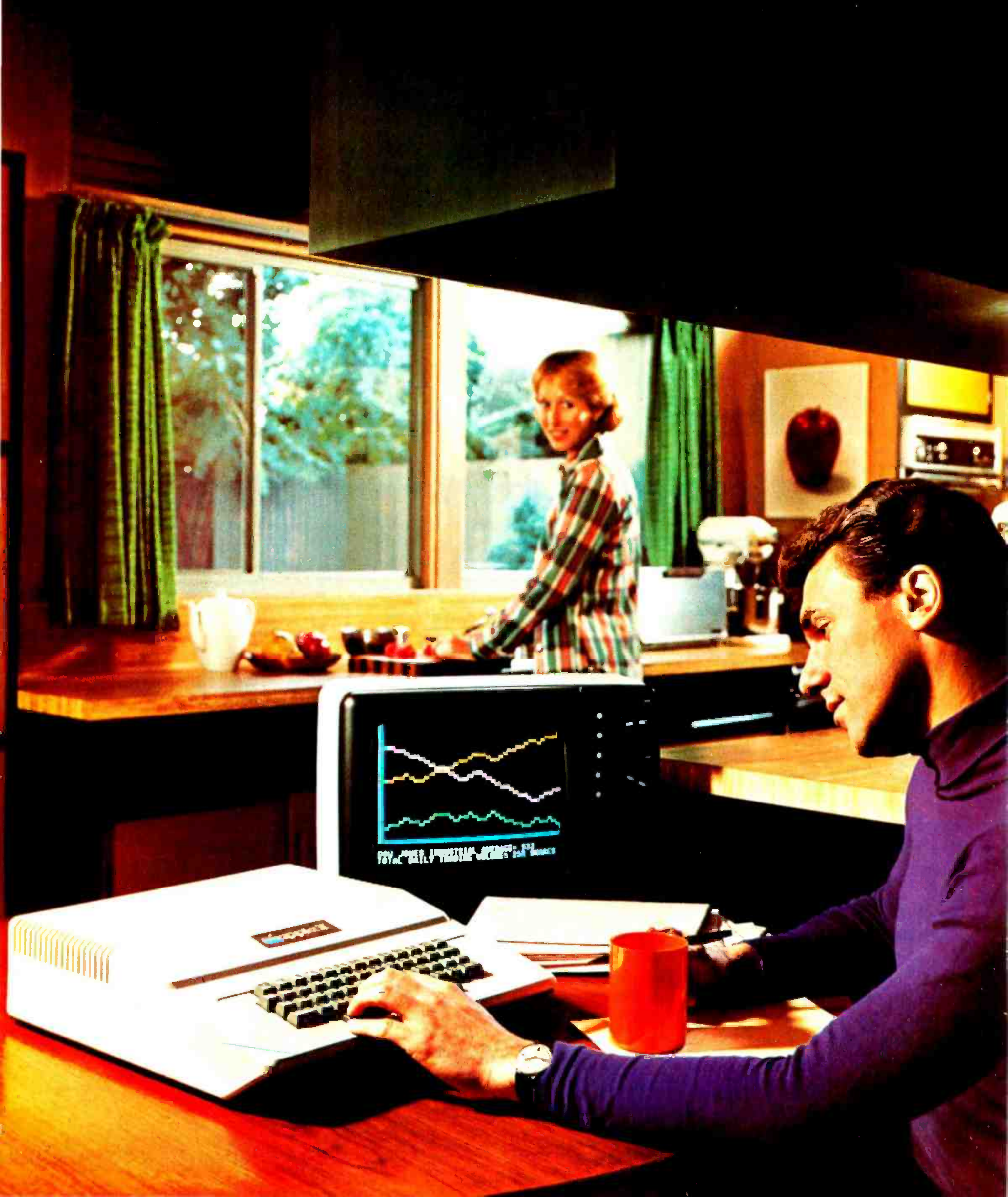
Where to find it. The Introl system can now be found at computer shops throughout the U.S. and Canada. Drop by and ask for a demonstration. Mountain Hardware, Inc., may be reached at Box 1133, Ben Lomond, CA 95005. Phone (408) 336-2495.



Mountain Hardware

Circle 78 on inquiry card.

Introducing Apple II.™



Complete ASCII

David M Ciemiewicz
533 N Holly St
Elizabethtown PA 17022

Most of the time when you see a magazine article that requires an ASCII table, the table accompanying the article is either incomplete or is in a numeric system that you cannot use without converting it.

The table I have devised is complete 128 character ASCII. Each character is accompanied by its binary, octal, decimal and hexadecimal equivalent.

This table has proven invaluable to me, as I am sure it will to you.■

128 CHARACTER ASCII TABLE

Character	Binary Bit 7 to Bit 0	Octal	Decimal	Hexadecimal	Character	Binary Bit 7 to Bit 0	Octal	Decimal	Hexadecimal
NUL	00000000	000	000	00	0	00110000	060	048	30
SOH	00000001	001	001	01	1	00110001	061	049	31
STX	00000010	002	002	02	2	00110010	062	050	32
ETX	00000011	003	003	03	3	00110011	063	051	33
EOT	00000100	004	004	04	4	00110100	064	052	34
ENQ	00000101	005	005	05	5	00110101	065	053	35
ACK	00000110	006	006	06	6	00110110	066	054	36
BEL	00000111	007	007	07	7	00110111	067	055	37
BS	00001000	010	008	08	8	00111000	070	056	38
HT	00001001	011	009	09	9	00111001	071	057	39
LF	00001010	012	010	0A	:	00111010	072	058	3A
VT	00001011	013	011	0B	;	00111011	073	059	3B
FF	00001100	014	012	0C	<	00111100	074	060	3C
CR	00001101	015	013	0D	=	00111101	075	061	3D
SO	00001110	016	014	0E	>	00111110	076	062	3E
SI	00001111	017	015	0F	?	00111111	077	063	3F
DLE	00010000	020	016	10	@	01000000	100	064	40
DC1	00010001	021	017	11	A	01000001	101	065	41
DC2	00010010	022	018	12	B	01000010	102	066	42
DC3	00010011	023	019	13	C	01000011	103	067	43
DC4	00010100	024	020	14	D	01000100	104	068	44
NAK	00010101	025	021	15	E	01000101	105	069	45
SYN	00010110	026	022	16	F	01000110	106	070	46
ETB	00010111	027	023	17	G	01000111	107	071	47
CAN	00011000	030	024	18	H	01001000	110	072	48
EM	00011001	031	025	19	I	01001001	111	073	49
SUB	00011010	032	026	1A	J	01001010	112	074	4A
ESC	00011011	033	027	1B	K	01001011	113	075	4B
FS	00011100	034	028	1C	L	01001100	114	076	4C
GS	00011101	035	029	1D	M	01001101	115	077	4D
RS	00011110	036	030	1E	N	01001110	116	078	4E
US	00011111	037	031	1F	O	01001111	117	079	4F
SP	00100000	040	032	20	P	01010000	120	080	50
!	00100001	041	033	21	Q	01010001	121	081	51
"	00100010	042	034	22	R	01010010	122	082	52
#	00100011	043	035	23	S	01010011	123	083	53
$	00100100	044	036	24	T	01010100	124	084	54
%	00100101	045	037	25	U	01010101	125	085	55
&	00100110	046	038	26	V	01010110	126	086	56
'	00100111	047	039	27	W	01010111	127	087	57
(	00101000	050	040	28	X	01011000	130	088	58
)	00101001	051	041	29	Y	01011001	131	089	59
*	00101010	052	042	2A	Z	01011010	132	090	5A
+	00101011	053	043	2B	[	01011011	133	091	5B
,	00101100	054	044	2C	\	01011100	134	092	5C
—	00101101	055	045	2D	]	01011101	135	093	5D
.	00101110	056	046	2E	Λ	01011110	136	094	5E
/	00101111	057	047	2F	–	01011111	137	095	5F

Character	Binary Bit 7 to Bit 0	Octal	Decimal	Hexadecimal	Character	Binary Bit 7 to Bit 0	Octal	Decimal	Hexadecimal
'	01100000	140	096	60	p	01110000	160	112	70
a	01100001	141	097	61	q	01110001	161	113	71
b	01100010	142	098	62	r	01110010	162	114	72
c	01100011	143	099	63	s	01110011	163	115	73
d	01100100	144	100	64	t	01110100	164	116	74
e	01100101	145	101	65	u	01110101	165	117	75
f	01100110	146	102	66	v	01110110	166	118	76
g	01100111	147	103	67	w	01110111	167	119	77
h	01101000	150	104	68	x	01111000	170	120	78
i	01101001	151	105	69	y	01111001	171	121	79
j	01101010	152	106	6A	z	01111010	172	122	7A
k	01101011	153	107	6B	{	01111011	173	123	7B
l	01101100	154	108	6C	¦	01111100	174	124	7C
m	01101101	155	109	6D	}	01111101	175	125	7D
n	01101110	156	110	6E	~	01111110	176	126	7E
o	01101111	157	111	6F	DEL	01111111	177	127	7F

Note: The bit 7 in the binary column is sometimes a 1 or is sometimes used as a parity bit.

Abbreviations for Control Characters:

NUL — null, or all zeros
SOH — start of heading
STX — start of text
ETX — end of text
EOT — end of transmission
ENQ — enquiry
ACK — acknowledge
BEL — bell
BS — backspace
HT — horizontal tabulation
LF — line feed
VT — vertical tabulation
FF — form feed
CR — carriage return
SO — shift out
SI — shift in
DLE — data link escape
DC1 — device control 1
DC2 — device control 2
DC3 — device control 3
DC4 — device control 4
NAK — negative acknowledge
SYN — synchronous idle
ETB — end of transmission block
CAN — cancel
EM — end of medium
SUB — substitute
ESC — escape
FS — file separator
GS — group separator
RS — record separator
US — unit separator
SP — space
DEL — delete

Letters

CELTIC PIANO NOTES

I was very interested in the article, "Notes on Interfacing Player Pianos" (September 1977 BYTE, page 112), especially since this was the subject of a student project here in 1973-74 by Stephen Cowles. The piano, which I bought for the university laboratory, was a simple MORS 88 note nonreproducing model, and we found the electronic/pneumatic interface to be the most difficult part. I was delighted therefore to hear about the Reisner relay.

The original system was a dual track cassette for serial clock and data plus circuitry which looks very much like figure 4 of your article, and in my opinion having constructed the special TTL electronics and got it to work, there is a better solution.

This year we are building a general interface which can be connected up to any player piano having rubber tubing, and strongly believe that the cheapest solution, bearing in mind building time and fault finding, etc, is to use a microcomputer with about 98 parallel output lines (say Motorola PIAs) driven by a microprocessor which receives the serial data in any of the standard ways.

Our original system only played "Three Blind Mice" rather shakily from a PDP-11/45, but we did write a PDP-11 interactive compiler which constructed keyboard images directly from typed-in sheet music, and we shall keep that system for making the magtapes.

We will let you know how the new system comes along.

Prof F G Heath
Gavin Weir
Heriot-Watt University
31-35 Grassmarket
Edinburgh SCOTLAND EH1 2HT

ORGANS, MUSIC AND PROGRAMMING

Chances are, it's occurred to you long ago: there's an amusing philosophical similarity between organ stop lists and computer machine language instruction lists.

Nicholas Bodley
300 W 108
New York NY 10025

One might consider the organ stop list as being equivalent to initialization data for a complicated program. Consider the score to be the equivalent of a music program. The parallels are very, very strong and tend to drive a number of people into music as well as programming. We've heard it said by some people *responsible for hiring programmers that musicians make excellent programmers.*

BAR CODE ON A BALL

Regarding the letter on page 12 of the October 1977 BYTE concerning an IBM ball that will print bar code and letters together: Datatype Corporation, 1050 NW 163rd Dr, Miami FL 33169, (305) 625-8451, had such a system. Bar code was ASCII. I don't know if they're still in business.

Joe Fisher
Computer Consultant
1120 E 52nd, Room 203
Austin TX 78723

NEWT POINTS

I have some corrections on your part and on mine regarding my article about "Newt: A Mobile, Cognitive Robot" (June 1977 BYTE, page 30). First (my mistake), on page 38, the stepping motor drive circuit motor cable color designations are wrong. The single color labels "red" and "green" should be interchanged or the motor will just sit there and quiver.

In several places the shading of Newt in figure 1 is incorrectly done, making a single surface appear broken into two surfaces, etc.

On page 30, the caption on figure 1 should have also mentioned that the turret is capable of panning left and right as well as tilting up and down.

On page 44, it is not true that any mobile robot must incorporate programs for seeking electrical outlets. For example, a robot on Mars might have a hard time finding an outlet!

On page 45, paragraph 1, the phrase "such as already demonstrated by the Viking robots" is unfortunate, since the Viking landers do not qualify as cognitive robots in the context of this article. They are sophisticated teleoperators. A Mars "rover" robot project, however, is in progress at Jet Propulsion Laboratories.

I am looking forward to writing more about Newt as things progress over the next few years. One possibility is a series of four articles. The published article was an overview of the system with emphasis on the motive subsystem. Three more articles would cover in some detail the manipulator, the sensory turret vision system, and finally the software experiments with Newt. All this depends, of course, on how much we can get done in the next several years.

I have been able to resume work on Newt almost immediately upon returning from France. The hand wired stepping motor drivers have been replaced by compact printed circuit versions and next week an order will go out for about $400 worth of gears and bearings for the manipulator assembly. With luck, the manipulator should be working within six months. Other lower priority items being worked on at present by myself and other persons in the group are the power conversion electronics, ADC system, and the radio telemetry link.

Ralph L Hollis Jr
University of Colorado
Boulder CO 80309

Our thanks to Ralph Hollis for calling these errors to our attention. We eagerly await the further adventures of Newt. . .CM

MORE ON TV SHIMMY

I am moved to write this because of the unsatisfactory answer given in "Ask BYTE," page 145 of the November 1977 BYTE, with regard to cathode ray tube (CRT) image "shimmy."

The root cause of this is almost always 60 Hz power line interference with the CRT scanning waveforms as seen by the electron beam. Ripple in the internal power supplies of cheap monitors or TV sets can easily amplitude modulate the scanning waveforms, particularly the horizontal waveform. AC magnetic fields emanating from the monitor's own power transformer can also deflect the beam. Sometimes the field from a nearby computer power supply can affect the display, particularly if a "constant voltage" transformer is used. These generally emit several times more field than a standard transformer does.

Commercial TV uses a vertical scan rate which, for practical purposes, is exactly 60 Hz. This was done so that the interference pattern on the screen would be *static* and therefore less objectionable. The fact that a video display image shimmies (swims, crawls, waves) is proof that the vertical rate of the video source is not dead-on 60 Hz because of a design compromise, drifting oscillator, or other defect.

The problem may be corrected or at least reduced by attenuating the interference in the monitor. Ripple in the monitor's power supply may be reduced by increasing the main filter capacitor by a factor of five to ten. If a voltage doubling supply is used, two capacitors must be increased. Shield the power transformer with a metal box from an old military type transformer. Find a big lamination from an old transformer, cut and shape it into a cylinder, and slip it over the picture tube neck behind the deflection yoke. These measures should cut the interference by a factor of four or more and at least make the system liveable.

Hal Chamberlin
29 Mead St
Manchester NH 03104

IS VIRTUAL A VIRTUE?

Mark Dahmke suggests, in an article titled "Virtual Memory and VSAM for Micros," November 1977, page 224, that virtual memory techniques be considered for new APL interpreters. Indeed, it might be a cure for the problem of limited primary memory. However, one must ask what price must be paid for the vast increase in memory space available to the user. I submit the price is either increased cost for specialized addressing hardware to support virtual memory or a slower running machine if the virtual memory techniques are implemented in software.

No matter how one chooses to implement a virtual memory scheme, the secondary memory must still be accessed. This process takes time and effort by the machine to execute, just like the more traditional file access methods.

While the concept of virtual memory may be practical for large machines like the IBM 370 or Univac 90/80, I believe virtual memory on a microcomputer is not a practical option, given the current state of technology available.

Clayton A Dane III
423 Roberts Av
Conshohocken PA 19428

COMMENTS ON THE MCM/70 FROM A USER

I recently came across your August issue in which the desirability of implementing APL on a microprocessor is discussed with the inference that it has not yet been accomplished. I would like to inform you that an APL interpreter was written for an Intel 8008 microprocessor in 1972 by a company called Micro Computer Machines of Kingston CANADA. The resulting computer, which was called the MCM/70, was first sold commercially late in 1974.

The interpreter uses 32 K bytes of ROM and about 7 K bytes for IO and special functions. In this space, a fairly complete subset of the APL language is implemented. Functions not implemented are matrix divide, factorial/binomial coefficient for nonintegral values and lamination.

My company purchased an MCM/70 in August 1975 and it has been used extensively since then. The version we own has 8 K bytes of user available memory, two cassette drives and a Diablo Hi-type printer which can also be used as a plotter (with a horizontal and vertical resolution of 1/48th and 1/60th of an inch respectively).

One of the most interesting and useful features of the MCM/70 is a virtual operating system which allows a cassette (or floppy disk) to be used as an extension to memory. In this way jobs that would take up to 110 K bytes on a timesharing computer can be run on the MCM/70, provided no individual program is larger than about 4 K bytes. (110 K bytes is the maximum capacity of the cassette; use of a floppy disk increases this capacity to 256 K bytes.)

We have used the MCM/70 for many routine engineering problems, and estimate that it has already earned its

Look To The North Star HORIZON Computer.

HORIZON™ — a complete, high-performance microprocessor system with integrated floppy disk memory. HORIZON is attractive, professionally engineered, and ideal for business, educational and personal applications.

To begin programming in extended BASIC, merely add a CRT or hard-copy terminal. HORIZON-1 includes a Z80A processor, 16K RAM, minifloppy™ disk and 12-slot S-100 motherboard with serial terminal interface — all standard equipment.

WHAT ABOUT PERFORMANCE?

The Z80A processor operates at 4MHZ — double the power of the 8080. And our 16K RAM board lets the Z80A *execute at full speed*. HORIZON can load or save a 10K byte disk program in less than 2 seconds. Each diskette can store 90K bytes.

AND SOFTWARE, TOO

HORIZON includes the North Star Disk Operating System and full extended BASIC on diskette ready at power-on. Our BASIC, now in widespread use, has everything desired in a BASIC, including sequential and random disk files, formatted output, a powerful line editor, strings, machine language CALL and more.

EXPAND YOUR HORIZON

Also available — Hardware floating point board (FPB); additional 16K memory boards with parity option. Add a second disk drive and you have HORIZON-2. Economical serial and parallel I/O ports may be installed on the motherboard. Many widely available S-100 bus peripheral boards can be added to HORIZON.

QUALITY AT THE RIGHT PRICE

HORIZON processor board, RAM, FPB and MICRO DISK SYSTEM can be bought separately for either Z80 or 8080 S-100 bus systems.

HORIZON-1 $1599 kit; $1899 assembled.
HORIZON-2 $1999 kit; $2349 assembled.

16K RAM — $399 kit; $459 assembled; Parity option $39 kit; $59 assembled. FPB $259 kit; $359 assembled. Z80 board $199 kit; $259 assembled. Prices subject to change. HORIZON offered in choice of wood or blue metal cover at no extra charge.

Write for free color catalogue or visit your local computer store.

NORTH STAR ★ COMPUTERS

2465 Fourth Street • Berkeley, California 94710 • (415) 549-0858

Circle 86 on inquiry card.

purchase price in savings on timesharing computer costs. In particular, the MCM/70 was used to simulate the thermal performance of a cooling pond. This involved a finite difference solution of the time dependent advection-dispersion equation of fluid mechanics.

Because of the analysis involved, the calculation required several months of effort. Most people with whom I have discussed the problem found it hard to believe that it could be handled on such a small machine. It would not have been possible if it weren't for the storage economy and power of APL in conjunction with the capability of the machine to run in the virtual mode.

The program was set up to plot the results of calculations (centerline temperatures) as they were produced. In this way the effect on the temperatures, of changes in thermal input and/or climatic conditions, could be monitored with the option to change either the data or the program without restarting the run.

We have also used the machine for many other projects and find it a very useful tool for the type of throwaway programming that is typical of much engineering work.

The MCM/70 has since been upgraded to the MCM/800 through the use of a specially designed microprocessor that has a similar instruction set to the 8008 but performs APL operations about ten times faster.

R V Elliott PhD
Ontario Hydro
700 University Av
Toronto, Ontario
CANADA M5G 1X6

See page 216 of December 1977 BYTE for a current report on the MCM/70.

COPYRIGHTS: US VERSUS CANADIAN?

The long letter from the Canadian patent attorney Daniel A Mersich does little to clarify the situation, since it reveals ignorance of American law. In fact, software is protected under the American copyright law, just as new hardware is protected under patent law. Enforcement of copyright is similar to enforcement of patent: the owner takes the violator to court.

American law seems clear to me (although I am assuming that the copyright office regulations which implement the law will remain the same) in that a copyright of software must be presented in a printed form readable by a (knowledgeable) human; it cannot be just a magnetic tape or a binary listing.

Since the philosophy of the patent law is that it will provide protection while providing public access to new technology, a person attempting to patent a little black box would have to describe the new technology inside to get a patent (like bubble memory). If the function was in fact carried out by programmable sequential steps, the description would have to be carried to the copyright office.

The US copyright office has a flier describing procedures and limits for software programming.

Mike Firth
104 N St Mary
Dallas TX 75214

CLARIFYING PERSONAL USE OF PATENTED INVENTIONS

I must take issue with Daniel Mersich's comments on patents in the November 1977 "BYTE's Bits." Though his information is, for the most part, correct, he leaves readers with an impression which is quite mistaken.

It is quite true that a *commercial* producer infringes a patent even by inadvertent "reinvention." The Patent Code gives the owner of a patent an absolute commercial monopoly, good against anyone who seeks to profit from the same knowledge. The fact that another inventor develops the same device without knowledge of the patent is of no effect.

However, a patent does *not*, despite the impression left by Mr Mersich, prevent *every* person from developing and using a given device. In general, a noncommercial user may create and use an identical device, even by directly copying the patent, and regardless of lack of permission from the patent owner. Such users are free to create and operate the device so long as use is restricted to purely *personal* experimentation, amusement, instruction, education, or curiosity.

I'm afraid that Mr Mersich may have convinced some readers that even such restricted personal use infringes, so that a patent search would be required for every new development. The Patent Code is often a pain in the lower bits, but it isn't that restrictive.

C Kevin McCabe
The Lawyer's Club, Room B-14
551 S State St
Ann Arbor MI 48104 ■

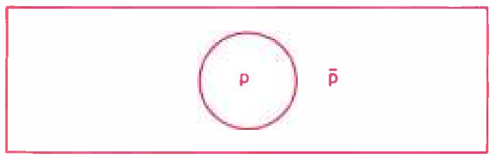


Figure 1: The Venn diagram universe is the collection of all points in the rectangle; the points inside the circle (p) are points corresponding to terminal p being H. Those outside ($\bar{p}$) refer to p=L.

Some Musings on Boolean Algebra

Dan Bunce
139 Morewood Av
Pittsburgh PA 15213

Art Schwartz
740 Broughton St
Pittsburgh PA 15213

The purpose of this article is to unify the concepts of digital electronics, the graphical representation of set theory and propositional calculus, using Boolean algebra. Our motivation for the background work represented in this article was the design of an encoder for a surplus keyboard. That was as much a problem in set theory and propositional calculus as it was in digital design.

First a note about the subject matter of this article. The availability of MSI and LSI makes the systematic reduction of many logic functions a waste of both time and money. For example you can buy an 8 bit addressable latch for about \$1.50. Synthesizing it out of small scale integration can take some 75 gates or 30 integrated circuits. No money or time savings here! The following techniques are quite general, though, and you can certainly find design problems that don't have off-the-shelf solutions. In this particular case (keyboard encoding), off-the-shelf solutions exist. But learning about logic design techniques requires illustrative examples. Encoding a large set of inputs serves well as such an example.

In the various systems, digital, graphical and logical, analogous concepts are expressed differently. We're going to show the equivalences which exist. Digitally and logically we have a "system," where graphically we have a "universe." Universe and system will mean the same thing to us. Our system is composed of "states," while our universe consists of "points." The meaning of point will be clearer after we've explained state. Physically we picture a device with several terminals, some input and output, the others internal, perhaps not even accessible. Suppose there are N terminals, each of which can have its signal level high (H) or low (L). H and L are the two possible states of any terminal. We want to know how many states the system has. There are two states for the first terminal, times two states for the second, times two for the third, . . . times two for the N^{th}, giving a total of 2^N states for the system. Another way to express this is to consider each terminal of the device to be represented by some bit in a binary word. For N terminals we need an N bit binary word, the p^{th} bit representing the state of the p^{th} terminal. Since an N bit binary word can take 2^N different values, our system must have 2^N different states. Each possible arrangement of Hs and Ls on the various terminals defines a unique system state, which corresponds to a point in our equivalent Venn diagram universe. The logical analogy to the digital terminal is the proposition, merely a statement, which is true (T) = 1 or false (F) = 0.

Think of any device terminal, call it p and define the proposition which is T when p = H and F when p = L. Of course we call this proposition p also. Graphically we collect all the points of our universe which correspond to system states with terminal p = H and enclose them in a curve as in figure 1. It is the interior of this region that corresponds to p. In summary we equate propositions with regions and label them p, q, r, etc. If a device has two inputs, two outputs and four internal terminals then the universe has 2^8 = 256 possible states or points. The aforementioned keyboard encoder is a 64 input, 7 output and n internal terminal device where minimizing the unknown, n, is one way of stating the logic design problem.

Venn diagrams are an easy way to demonstrate the laws of logic. For example it's an

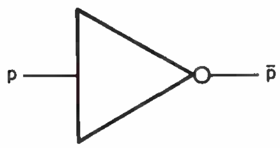


Figure 2: The output of the inverter is the complement of the input. A small circle means invert the signal. The triangle symbolizes an amplifier or "buffer" of the digital signal.

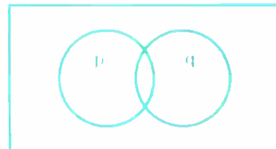


Figure 3a: The most general example of two variables. Apparently different examples are degenerate cases of this figure.

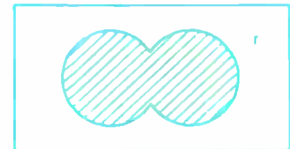


Figure 3b: The region p$\vee$q is shaded. The region r of the text is unshaded.

axiom of logic that $(\overline{\overline{p}}) = p$. (An axiom is something that you can't prove; you simply must assume it is true in order to build a system of thought.) Figure 1 makes this very plausible because p partitions the universe into the regions p and $\bar{p}$ (read as not p or complement of p). $\bar{p}$ lies outside of p. Now what lies outside of $\bar{p}$? It can't be $\bar{p}$ and it must be inside the universe. All that's left is p. It seems reasonable to accept $(\overline{\overline{p}}) = p$. The inverter or inverter gate (figure 2) is the digital device that generates the complement of its input.

The excitement begins when we label some other terminal q and associate with it the proposition which is true when terminal q = H. Then p and q together partition our universe as shown in figure 3a.

For even two variables there are many possibilities, one of which is shown in figure 4. This relation is written p => q; another common form is p<=q (read as p implies q or p is contained in q). In this example look at the areas $\bar{p}$ and $\bar{q}$. Area $\bar{q}$ is smaller than and wholly contained within $\bar{p}$. This illustrates another of the axioms of logic,

$$(p => q) => (\bar{q} => \bar{p}).$$

Of course we might have the other case, q => p. If we have p => q and also have q => p we then have the definition p = q. Implication and equation are distinct concepts.

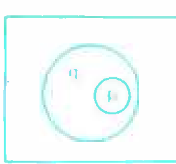


Figure 4: A degenerate case of figure 3a. Specifically, the empty set: $p\wedge\bar{q}=0$. Said differently, there are no states for this system with p=H and q=L.

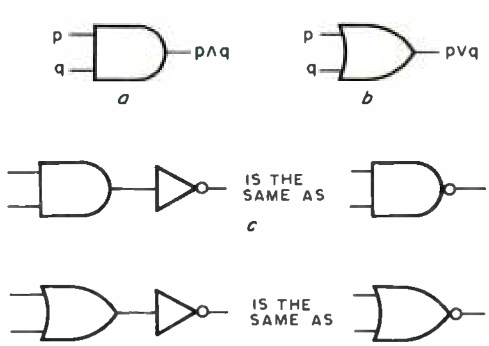


Figure 5: A 2 input AND gate is shown in (a) and a 2 input OR gate is shown in (b). Equivalences for the "inverted" forms of these gates are shown at (c), NAND; and at (d), NOR.

There are many other possibilities for two variables but they're all special cases of figure 3a. In general some points will lie in both p and q. Call this set of states s and write s = p$\wedge$q, read s = p and q. This is often written s = pq symbolic of ordinary multiplication, since associating the possible binary values of 0 or 1 with the variables p and q gives numerically correct results for s. Digitally we realize this set with the AND gate as shown in figure 5a. We'll call it the set of states for which p = T and q = F. We see that we can write t = $p\wedge\bar{q}$. A third region is u = $\bar{p}\wedge q$. What about the fourth region, that for which p and q are both L? Certainly this set r = $\bar{p}\wedge\bar{q}$. But we can describe it differently. First we form the set for which either p or q = 1. This is the region of both circles and is shown in figure 3b. It is the set p$\vee$q (p or q) which, since we're adding areas, is sometimes written p + q. It comes as no great surprise that the OR gate exists for just this purpose (see figure 5b). When we form the complement of p$\vee$q, we again get the set r!

Our diagrams have given us one of DeMorgan's laws: $\bar{p}\wedge\bar{q} = (\overline{p\vee q})$. We can get an equivalent form of the law by taking the complement of both sides of the equation, $(\overline{\bar{p}\wedge\bar{q}}) = p\vee q$. Figure 3a also gives us the other law, $\bar{p}\vee\bar{q} = (\overline{p\vee q})$, which is equivalent to $(\overline{\bar{p}\vee\bar{q}}) = p\wedge q$.

Now you begin to see where we're going. If we want our system to generate $\bar{p}\wedge\bar{q}$ from p and q we have options. We could use either a NOR gate (NOR => not OR), or an inverter each for p and q followed by a 2 input NAND gate. In a NOR gate the inverter follows the OR gate. Use figure 3a to convince yourself that $\bar{p}\vee\bar{q}$ is different from $(\overline{p\vee q})$. There are NAND gates as well and you've probably guessed what they do. It's worth noting that in the usual mode of operation a transistor inverts the signal. Consequently NANDs and NORs are more easily fabricated than ANDs and ORs, and more often used.

Now a reminder about what all this

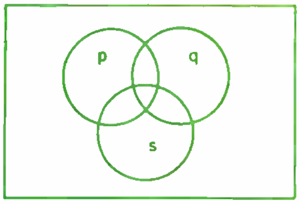


Figure 6: Diagram used to show operations are associative and distributive.

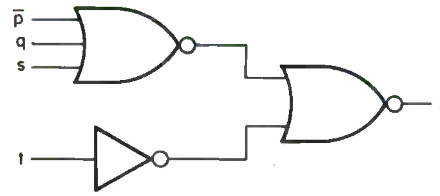


Figure 7: Circuit representing the equation: $x=[(\overline{p} \vee q) \vee s] \wedge t$.

means. The ps and qs are terminals, regions or propositions. But most of all they're 0 or 1. $p \wedge q$ is also 0 or 1. If p = 1 at the same time that q = 1 then $p \wedge q = 1$, if either (or both) is 0 then $p \wedge q = 0$. What about $p \vee q$? If both are 0, $p \vee q$ is 0, but if either is 1 then $p \vee q = 1$. One objection to the notation + for OR is that 1 + 1 = 1.

We can consider three variables by referring to figure 6. Convince yourself of the following:

$$(p \vee q) \vee s = p \vee (q \vee s) = p \vee q \vee s$$
$$(p \wedge q) \wedge s = p \wedge (q \wedge s) = p \wedge q \wedge s.$$

In words, the operation $\vee$ and $\wedge$ are associative. The next two examples are only a little harder to see:

$$p \vee (q \wedge s) = (p \vee q) \wedge (p \vee s)$$
$$p \wedge (q \vee s) = (p \wedge q) \vee (p \wedge s).$$

In mathematical parlance, the operations are distributive. The formulas show that within a set of parentheses order doesn't count; that is $p \wedge q = q \wedge p$ or $p \vee q = q \vee p$. This means the operators are commutative.

We want to write equations in more than three variables but shading the subregions of five circles creates eye strain faster than understanding. Our diagrams have been most helpful, though, for we can go on symbolically, by repeated application of what we've already learned.

Let's work out an example. To reduce $x = [(\overline{p} \vee q) \vee s] \wedge t$ we define a new variable, and continue to define new variables as often as necessary: $u = (\overline{p} \vee q) \vee s$, so the original expression takes the simplified form $x = u \wedge t = \overline{(\overline{u} \vee \overline{t})}$. What's $\overline{u}$? Just plug in $u = (\overline{p} \vee q) \vee s$, so $\overline{u} = \overline{(\overline{p} \vee q \vee s)}$. Substitute back and write: $x = \overline{[\overline{(\overline{p} \vee q \vee s)} \vee \overline{t}]}$. One circuit that generates x is shown in figure 7. We emphasize again, x is a number, either 0 or 1, and its value depends on the values of p, q, s and t.

For many variables, we'll run out of letters to designate propositions. There's an easy way around this. If we have a proposition a, and later in the problem we find we'd like to call a different proposition a also, we tack on subscripts. The first proposition becomes a(1), read a-one or a-sub-one, the second a(2). We never run out of subscripted letters, but more important subscripting allows a more compact notation. When we have $a(1) \vee a(2) \vee a(3) \vee \ldots \vee a(n)$ we write this as

$$\bigvee_{i=1}^{n} (a(i)).$$

Similarly we can form

$$b(1) \wedge b(2) \wedge \ldots \wedge b(m) = \bigwedge_{j=1}^{m} (b(j)).$$

The first time you encounter this notation it appears awkward. In fact it's very efficient, but like any unfamiliar concept it requires some mental accommodation. Whenever you feel uncomfortable with it just go back to the definition and expand it. It soon becomes second nature.

We'll pose and solve a final problem. The Teletypewriter keyboard which started this article has 64 switches. We wanted a circuit which shows when any of the switches have been pressed. The goal is, in effect, a 64 input OR gate. Nobody makes such a thing (if they did they'd need a 68 pin package to put it in), but we can synthesize it. Just follow the expansion:

$$\vee [a(i)] = \overline{\left[\bigwedge_{1}^{8} \left(\overline{\bigvee_{1}^{8} b(i, j)}\right)\right]}$$

where

$$b(1, j) = a(j),\ b(2, j) = a(8+j),\ b(3, j) = a(16+j) \ldots b(8, j) = a(56+j).$$

Once you understand how to break OR into NOR then NAND (reading from inside the parentheses since that's the way the signal flows), you can nest successively, generating:

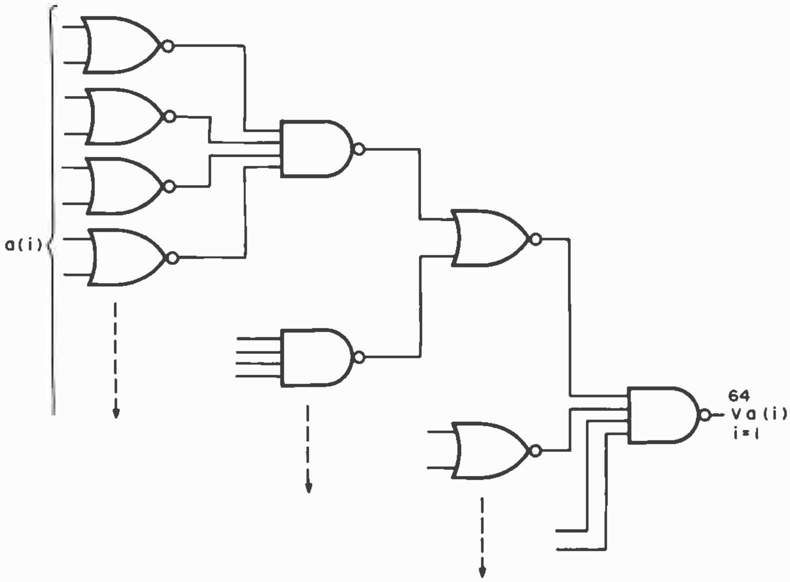


Figure 8: A 64 input OR gate. Dotted lines indicate repetition of the relevant element. For example, there are 32 2 input NOR gates at the leftmost level.

$$\overline{\bigwedge_{1}^{4}\left[\overline{\bigvee_{1}^{2}\left(\overline{\bigwedge_{1}^{4}\left[\overline{\bigvee_{1}^{2} d(i, j, k, l)}\right]}\right)}\right]}$$

Working from the inside out we need NOR gates, NAND gates, NOR gates, NAND gates. 2 input NORs are common, as are 4 input NANDs. If we feed the 64 input signals into 32 2 input NORs, their outputs into eight 4 input NANDs then on into four 2 input NORs and finally into a 4 input NAND, we have a 64 input OR. Using a logic circuit representation, figure 8 shows how this expression might be wired.

At the first level we've used 32 2 input NORs. Four of these come in one integrated circuit package which means we need eight packages at the input level. The second level requires four packages since there are two of these gates in each integrated circuit, the third two packages and the fourth one half package. The circuit uses approximately 15 integrated circuits.

We achieve the same result more directly if we begin with the complemented signal from each switch. In practice this means the switch is wired H rather than L. It is neither more nor less difficult. We write the following:

$$\vee(a(i)) = \overline{\bigwedge_{1}^{4}\left(\overline{\bigvee_{1}^{4}\left[\overline{\bigwedge_{1}^{4} (c(j, k, l))}\right]}\right)}$$

beginning with complemented signals we need 4 input NANDs, NORs, and NANDs, as shown in figure 9a.

An even more direct means to our end is the following formula which is illustrated in figure 9b:

$$\vee(a(i)) = \overline{\bigwedge_{1}^{8}\left[\overline{\bigvee_{1}^{8} (b(j, k))}\right]}$$

where all NAND and NORs are 8 input. This requires only nine packages. The only thing wrong with the last method is 8 input NORs are unusual. They are available in newer CMOS, but not in low numbered (7400) TTL. Since there are other nearly as simple and certainly less expensive ways of doing the job it hardly pays to look for the special integrated circuits.

Having established (or more correctly, justified) the laws of logic and worked a few

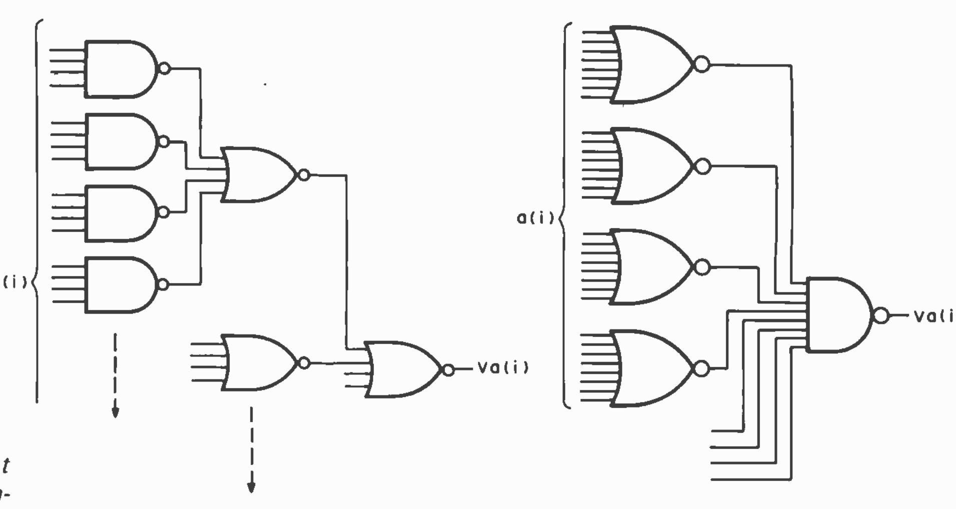


Figure 9: Two different ways of building a 64 input OR gate.

problems, we're ready to take on a multitude of simple reductions.

A more formal and complete treatment of logic (at the introductory level) is presented in the monograph *Propositional Calculus* by P H Nidditch, Dover Publications, New York. It sells for about $1, is clearly written and aimed at the general reader. Armed with this basic knowledge of combinatorial logic, we were able to implement a keyboard encoder.■

GLOSSARY

AND gate: A binary circuit with two or more inputs and a single output. The output is logic 1 only when all inputs are logic 1. The output is logic 0 if any of the inputs are logic 0 ($A \wedge B$).

Associative: When the result of an equation is independent of the groupings of the elements, provided the elements are kept in the same order, the operation performed on the equation is associative.

Axiom: A proposition regarded (with good reasons) as self-evident truth.

Commutative: When the result of an equation is independent of the ordering of the elements within the equation, the operation performed on the equation is commutative.

Complement: In a given universe, all of the elements not contained within one set are the complement of the set.

DeMorgan theorem: Inversion of a series of AND implications is equal to the same series of inverted OR implications. The inversion of a series of OR implications is equal to the same series of inverted AND implications. $\overline{A \wedge B} = \overline{A} \vee \overline{B}$ and $\overline{A \vee B} = \overline{A} \wedge \overline{B}$.

Empty set: A set containing no elements.

Gate: A circuit having two or more inputs and one output. The output depends on the combination of logic inputs.

Implication: Logical relation between two propositions.

Inverter: An operation or device which outputs the logical complement of the input. Inverting logical 1 gives logical 0; Inverting logical 0 gives logical 1.

NAND gate: Combination of a NOT circuit and an AND circuit ($\overline{A \wedge B}$).

NOR gate: Combination of a NOT circuit and an OR circuit ($\overline{A \vee B}$).

OR gate: Binary circuit with two or more inputs and a single output. The output is logic 0 when all the inputs are logic 0. The output is logic 1 if one or more of the inputs is logic 1 ($A \vee B$).

Universe: Set containing all elements relevant to a specific problem.

Venn diagram: Graph employing circles to represent logical relations between and operations on sets.

Circle 9 on inquiry card.

Languages Forum

The Case for a "Compiler Interpreter"

Richard A Rodman
3041 Patrick Henry #202
Falls Church VA 22044

Your "Technical Forum," May 1977 issue, presents a higher level language approach which, although based on ideals achievable on larger systems, is unrealizable on microprocessor devices within any realm of practicality. The character set alone of PL-SKYE is unavailable on displays to anyone save ECD Micromind or Noval 760 users, and hard copy would be a foregone conclusion.

I have been working on high level language ideas myself and have already discarded the use of a direct compiler system of home computers because of its stringent demands on mass storage peripherals. Yet, as Mr Skye realizes, the resplendent luxury of the source code in memory used by an interpreter is also far too costly for anything but the smallest programs.

However, there is middle ground in the form of a semi-compiler or "compiling interpreter." This is a program, or set of programs, which reads the source code from tape, translates it into compact symbolic code, and then interprets this code. As a single program, this is of course too large for the amount of memory space in a home computer. However, it may be broken down by task into three segments: the translator, which creates the symbol table as well; the resolver, which resolves label references, allocates memory for variables and so forth; and the interpreter. These three segments, would, each upon completion, load in the next segment. There would be an optional fourth segment, the "reconstructer," which would be able to reconstruct the source code if desired.

I feel that this arrangement would allow the use of slow speed devices, such as the audio cassette recorder, without sacrificing an enormous amount of speed. Translation would of course have to be done in a single pass, reading from one tape onto another.

As far as specifying the particular elements of the language, it would be best to start simple. I, as well as many others, share

The Computer for the Professional

Whether you are a manager, scientist, educator, lawyer, accountant or medical professional, the System 8813 will make you more productive in your profession. It can keep track of your receivables, project future sales, evaluate investment opportunities, or collect data in the laboratory.

Use the System 8813 to develop reports, analyze and store lists and schedules, or to teach others about computers. It is easily used by novices and experts alike.

Reliable hardware and sophisticated software make this system a useful tool. Several software packages are included with the machine: an advanced disk operating system supporting a powerful BASIC language interpreter, easy to use text editor, assembler and other system utilities. Prices for complete systems start at $3250.

See it at your local computer store or contact us at 460 Ward Dr., Santa Barbara, CA 93111, (805) 967-0468.

PolyMorphic Systems

UP AND RUNNING

TDL EQUIPMENT USED BY NEW JERSEY PUBLIC TELEVISION
TO PROCESS NEW JERSEY GUBERNATORIAL PRIMARY ELECTION RETURNS

John Montagna, computer engineer (above left), lead this successful network team in generating election results speedily, efficiently and reliably using predominantly TDL hardware and software. Montagna created three programs to get the job done. The text for a SWAPPER program was written and assembled using the TDL TEXT EDITOR and Z80 RELOCATING MACRO ASSEMBLER. The SWAPPER text and all debugging was run through TDL's ZAPPLE MONITOR. The relocatable object code was punched onto paper tape. A MAIN USERS program updated votes and controlled air display. An ALTERNATE USERS program got hard copy out and votes in. The latter two programs were written in BASIC. Montagna modified the ZAPPLE BASIC to permit timesharing between the two USERS programs.

Four screens were incorporated, two terminals entered votes as they came in and were used to call back votes to check accuracy. Montagna called on the power and flexibility offered by TDL's ZPU board and three Z-16 Memory boards.

Montagna's setup worked constantly for over four hours updating and displaying state-wide and county-wide results without flaw.

"I chose TDL because they have all the software to support their hardware, and it's good; it has the flexibility to do the job."
John Montagna

We salute John Montagna and NEW JERSEY PUBLIC BROADCASTING for spearheading the micro-computer revolution.

TDL's XITAN SYSTEMS have the capacity to do similar tasks for you. Write to us for XITAN information and the name of your nearest TDL dealer.

TECHNICAL DESIGN LABS

RESEARCH PARK BLDG. H 1101 STATE ROAD
PRINCETON, NEW JERSEY 08540 (609) 921-0321

Circle 115 on inquiry card.

the visions of Mr Skye, but it is best to start with the humble and glorify it, rather than to start with the glorious and humble it.

BYTE is to be commended for the publication of the Tiny Assembler in bar codes. However, the dearth of bar code hardware is certain to lead to its gracing far more bookshelves than memory chips. Furthermore, even with suitable hardware, the reliability of bar code loading remains to be demonstrated for most of us. Audio disks are more easily read but obviously more expensive and quite subject to physical damage. (One can imagine the record skipping and loading all of memory with "disposable interrupts.") *[This has not been a problem with* Interface Age's *"Floppy ROMs," according to Bob Jones. . .CM]* Surely there must be other possibilities for the publication of large programs.

There is a very fine line between standardization and ossification. Technological superiority must always come before standardization, although the primary things to keep in mind are *specific needs* and *available resources.* I feel that too often these factors have been ignored in this field. I hope this will not be the case in the future.■

APL Character Representations

David Sloan
628-555 Keenleyside St
Winnipeg Manitoba R2K 3PG
CANADA

For those readers with a problem of representing APL characters, I have found a temporary solution. For those who have a video display with reverse video selectable for each character, the functions can be very easily displayed as reverse polarity characters. This is easily accomplished by defining one of the keys as a special function key through software control. This will produce the reverse polarity required.

Since most of the APL characters are a close representation of the present keyboard characters, this will be sufficient for most applications. It also means that the lower case and special characters can still be represented.

In the APL interpreter, a special character can then be easily detected, by looking to see if the cursor bit is turned on.■

why the last bus is the best bus

If you want to be better, you have to be willing to make changes. So, when we started designing our H8 Computer back in 1975, we put aside a lot of pre-conceptions to design a *better* computer. For instance, instead of slavishly adopting an inferior bus as a "standard," we designed our own. We used fifty fully-buffered lines because that's all we really needed with the H8's built-in systems controller. The result? Clean signals, minimum noise and great expansion capabilities.

Next, we picked a male/female connector. Not only is it less expensive than edge connectors, our tests show it's also more reliable! And because it costs less, every H8 motherboard includes all the connectors you need for expansion. There's nothing extra to buy.

The connectors are mounted on the right side of the cabinet at an angle so the boards tilt back. That gives us a couple of significant advantages. First, it allows the boards to be larger without requiring a taller cabinet. So our circuit board layout is more open, which makes the boards easier to build – even if you've never built a kit before. Second, the angled mounting provides easy accessibility to all the components on the face of the boards without an extender. (Many times an extender alters the transmission characteristics of the bus, which masks the problem you're trying to solve or even causes new ones.)

And we plan to support the H8 bus. In the future you can expect to see many new and exciting accessories for the H8 bus like higher density, lower power memories and a variety of unique I/O interfaces.

We're very happy with our bus, and we think you will be, too.

Who minds being different... as long as you're better.

Heathkit Computers

FREE
Heathkit Catalog

Read about nearly 400 money-saving, fun-to-build electronic kits.

Use coupon to send for your mail order catalog or bring coupon to a Heathkit Electronic Center for your catalog.

HEATH
Schlumberger

Heath Company, Dept. 334-380
Benton Harbor, MI 49022

Please send me my FREE Catalog. I am not on your mailing list.

Name ______________________

Address ______________________

City ______________________ State ________

CP-138 Zip ________

AVAILABLE LOCALLY AT HEATHKIT ELECTRONIC CENTERS

(Units of Schlumberger Products Corporation) Retail prices on some products may be slightly higher.

ARIZONA: Phoenix, 85017, 2727 W. Indian School Rd., Phone: 602-279-6247; **CALIFORNIA: Anaheim,** 92805, 330 E. Ball Rd., Phone: 714-776-9420; **El Cerrito,** 94530, 6000 Potrero Ave., Phone: 415-236-8870; **Los Angeles,** 90007, 2309 S. Flower St., Phone: 213-749-0261; **Pomona,** 91767, 1555 Orange Grove Ave. N., Phone: 714-623-3543; **Redwood City,** 94063, 2001 Middlefield Rd., Phone: 415-365-8155; **Sacramento,** 95825, 1860 Fulton Ave., Phone: 916-486-1575; **San Diego (La Mesa, 92041),** 8363 Center Dr., Phone: 714-461-0110; **San Jose (Campbell, 95008),** 2350 S. Bascom Ave., Phone: 408-377-8920; **Woodland Hills,** 91364, 22504 Ventura Blvd., Phone: 213-883-0531; **COLORADO: Denver,** 80212, 5940 W. 38th Ave., Phone: 303-422-3408; **CONNECTICUT: Hartford (Avon, 06001),** 395 W. Main St. (Rte. 44), Phone: 203-678-0323; **FLORIDA: Miami (Hialeah, 33012),** 4705 W. 16th Ave., Phone: 305-823-2280; **Tampa,** 33614, 4019 West Hillsborough Ave., Phone: 813-886-2541; **GEORGIA: Atlanta,** 30342, 5285 Roswell Rd., Phone: 404-252-4341; **ILLINOIS: Chicago,** 60645, 3462-66 W. Devon Ave., Phone: 312-583-3920; **Chicago (Downers Grove, 60515),** 224 Ogden Ave., Phone: 312-852-1304; **INDIANA: Indianapolis,** 46220, 2112 E. 62nd St., Phone: 317-257-4321; **KANSAS: Kansas City (Mission, 66202),** 5960 Lamar Ave., Phone: 913-362-4486; **KENTUCKY: Louisville,** 40243, 12401 Shelbyville Rd., Phone: 502-245-7811; **LOUISIANA: New Orleans (Kenner, 70062),** 1900 Veterans Memorial Hwy., Phone: 504-722-6321; **MARYLAND: Baltimore,** 21234, 1713 E. Joppa Rd., Phone: 301-661-4446; **Rockville,** 20852, 5542 Nicholson Lane, Phone: 301-881-5420; **MASSACHUSETTS: Boston (Peabody, 01960),** 242 Andover St., Phone: 617-531-9330; **Boston (Wellesley, 02181),** 165 Worcester Ave. (Rt. 9 just west of Rt. 128), Phone: 617-237-1510; **MICHIGAN: Detroit,** 48219, 18645 W. Eight Mile Rd., Phone: 313-535-6480; **E. Detroit,** 48021, 18149 E. Eight Mile Rd., Phone: 313-772-0416; **MINNESOTA: Minneapolis (Hopkins, 55343),** 101 Shady Oak Rd., Phone: 612-938-6371; **MISSOURI: St. Louis (Bridgeton),** 63044, 3794 McKelvey Rd., Phone: 314-291-1850; **NEBRASKA: Omaha,** 68134, 9207 Maple St., Phone: 402-391-2071; **NEW JERSEY: Fair Lawn,** 07410, 35-07 Broadway (Rte. 4), Phone: 201-791-6935; **Ocean,** 07712, 1013 State Hwy. 35, Phone: 201-775-1231; **NEW YORK: Buffalo (Amherst, 14226),** 3476 Sheridan Dr., Phone: 716-835-3090; **Jericho, Long Island,** 11753, 15 Jericho Turnpike, Phone: 516-334-8181; **Rochester,** 14623, 937 Jefferson Rd., Phone: 716-244-5470; **White Plains (North White Plains, 10603),** 7 Reservoir Rd., Phone: 914-761-7690; **OHIO: Cincinnati (Woodlawn, 45215),** 10133 Springfield Pike, Phone: 513-771-8850; **Cleveland,** 44129, 5444 Pearl Rd., Phone: 216-886-2590; **Columbus,** 43229, 2500 Morse Rd., Phone: 614-475-7200; **Toledo,** 43615, 48 S. Byrne Rd., Phone: 419-537-1887; **PENNSYLVANIA: Philadelphia,** 19149, 6318 Roosevelt Blvd., Phone: 215-288-0180; **Frazer (Chester Co.),** 19355, 630 Lancaster Pike (Rt. 30), Phone: 215-647-5555; **Pittsburgh,** 15235, 3482 Wm. Penn Hwy., Phone: 412-824-3564; **RHODE ISLAND: Providence (Warwick, 02886),** 558 Greenwich Ave., Phone: 401-738-5150; **TEXAS: Dallas,** 75201, 2715 Ross Ave., Phone: 214-826-4053; **Houston,** 77027, 3705 Westheimer, Phone: 713-623-2090; **VIRGINIA: Alexandria,** 22303, 6201 Richmond Hwy., Phone: 703-765-5515; **Norfolk (Virginia Beach, 23455),** 1055 Independence Blvd., Phone: 804-460-0997; **WASHINGTON: Seattle,** 98121, 2221 Third Ave., Phone: 206-682-2172; **WISCONSIN: Milwaukee,** 53216, 5215 W. Fond du Lac, Phone: 414-873-8250.

Circle 54 on inquiry card.

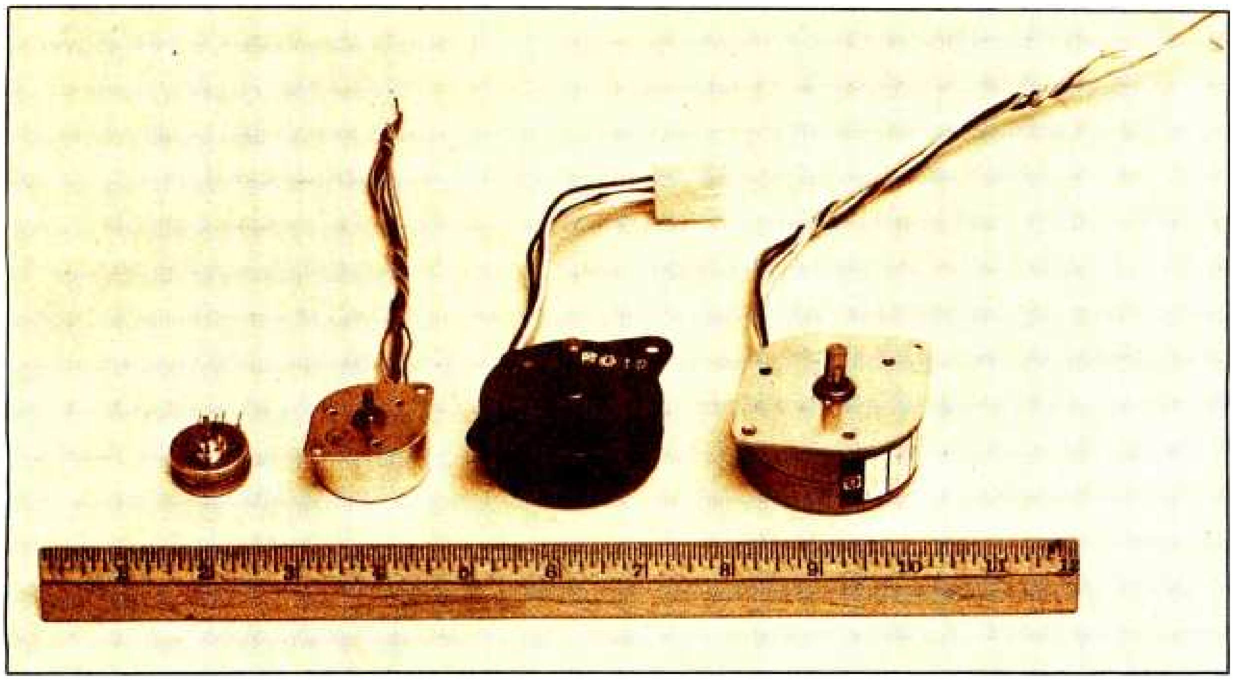


Photo 1: Several typical stepper motors. In order of physical size from smallest to largest, the motors are rated as follows in regard to power consumption and torque: 2 W at 0.5 oz-in, 3 W at 2.0 oz-in, 6.5 W at 9.0 oz-in, and 12 W at 16.0 oz-in.

Taking the First Step

Robert E Bober
449 Pleasant St
Framingham MA 01701

Stepper motors are coming down in cost now to the point where they can be designed into home projects. They are a natural for variable speed or precise angular movement controlled by a microprocessor.

All stepper motors convert electrical pulses into mechanical movements. In this article I will be referring to permanent magnet stepper motors such as those shown in photo 1. This type of motor is classified as either a pulse stepper or a logic stepper. Pulse steppers usually are unidirectional and have one center tapped coil. Logic steppers are multicoil and bidirectional. I will confine my drive controls to the bidirectional logic stepper.

Stepper motors have typical step angles from 3.75° up to 90°. The step angle is determined by the number of coil phases in the motor. For any motor design this is fixed and very accurate. Stepper motors can be readily obtained with operating voltage from 5 V to 48 V, with 12 V and 24 V the most common. The construction of the motor consists of a rotating multipole permanent magnet and stationary multipole electromagnet coils. The rotating magnet is formed by taking a cylinder of ferromagnetic material and magnetizing alternate north and south poles. 24 pole pairs give 7° 30′ steps.

The 4 phase stepper has two center tapped windings. Each winding surrounds one half of the rotor. Soft iron fingers arranged into pairs concentrate the magnetic flux from each winding near the rotor. As a winding is energized, the rotor moves to align its magnetic poles with the poles of the stator. The next coil in line will again shift the stator field and cause another rotor step.

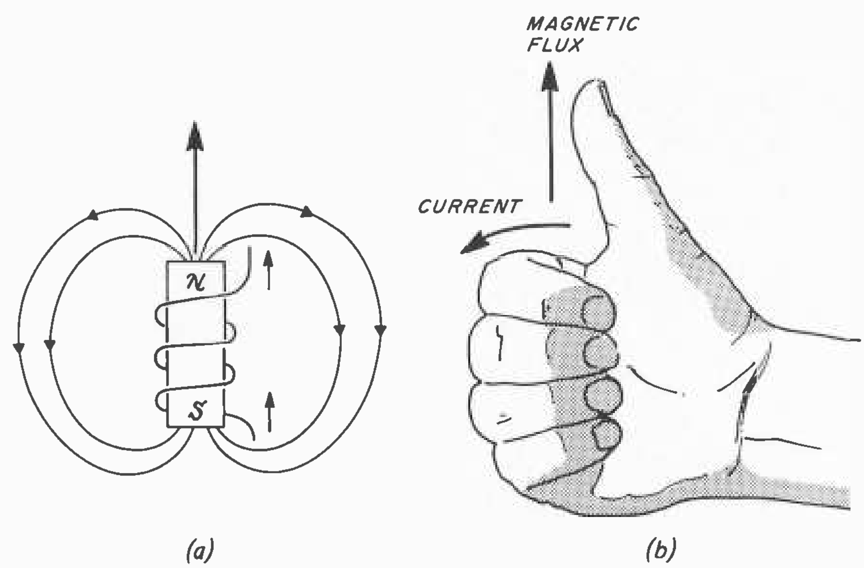


Figure 1: Illustration of the magnetic fields surrounding a current carrying coil. A simple method for determining the direction of the magnetic flux and the north pole of the system is shown in figure 1b. If the fingers of your right hand are wrapped around the coil in the direction of the current, your thumb will be pointing in the direction of the magnetic flux. This is called the right hand rule.

Thus with each pulse, one precise step is made. The trick is to energize the coils in the proper sequence.

Theory of Operation

The basis for all motor rotation lies in magnetics. Like poles repel while unlike poles attract. The tricky part is in creating the magnetic poles and directing the magnetic flux. The magnetic poles can be either permanent magnets or electromagnets. Permanent magnets are made of a variety of materials, and once magnetized by a strong magnetic field will retain their strength. Electromagnets consist of a coil of wire surrounding a soft iron structure. When current flows in the coil, a magnetic field exists. When the current ceases, the field ceases. Figure 1a shows a typical electromagnet. When current flows as indicated, the magnetic flux creates north and south poles as shown. The magnetic flux is related to the direction of current in the coil using the right hand rule illustrated in figure 1b. If the fingers point in the direction of coil current flow, then the thumb points in the direction of magnetic flux and the north pole. If two magnetic systems exist in close proximity, their net magnetic flux is the vector sum of the two individual magnetic contributions at each point in space.

A motor consists of a stationary part called a stator and a rotating part called a rotor. In a stepper motor the stator is an electromagnetic coil and a ferrous magnetic path. The rotor is a permanent magnet,

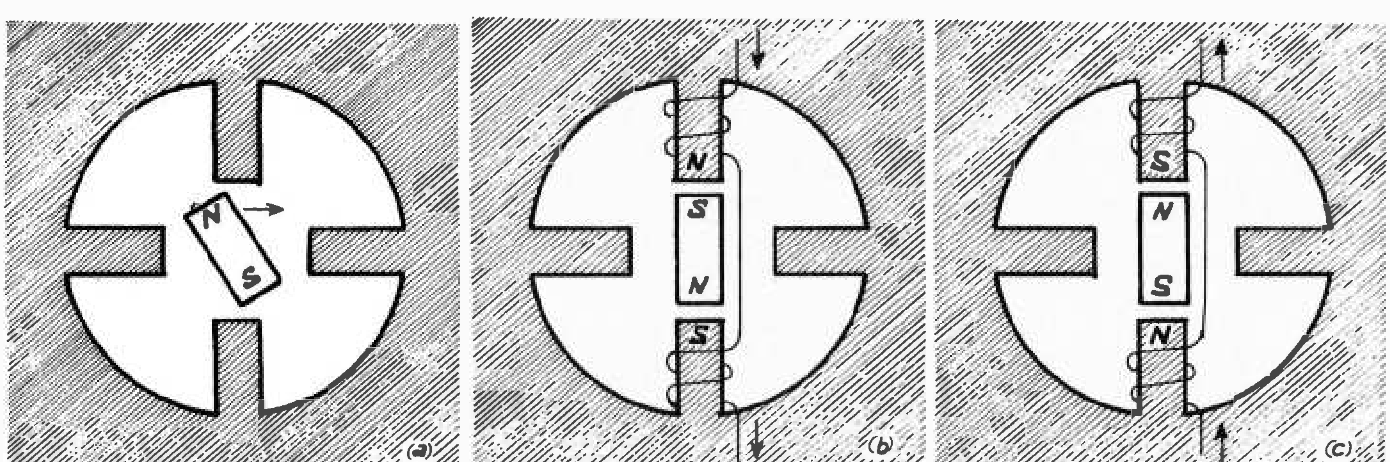


Figure 2: Movement of a rotor in a motor using only two poles. In figure 2a the rotor does not have any outside forces acting on it. In figure 2b one set of coils is energized and the rotor swings to line up opposite poles. When the current in the coils is reversed the rotor poles will switch. However, we cannot tell in which direction they will turn to get there.

Meet The North Star Family

THE NORTH STAR S-100 FAMILY—four high performance products at attractive low prices. Our boards are designed for use in the North Star HORIZON computer and other S-100 bus computers using 8080 or Z80 processors. Visit your computer store for a demonstration, or write for our free color catalog.

16K RAM BOARD

No other S-100 bus memory can match the performance of the North Star 16K RAM at any price. This low-power board has been designed to work at full speed (no wait states), even at 4MHz with both Z80 and 8080 systems. Memory refresh is invisible to the processor, bank switching is provided and addressability is switch selectable in two 8K sections. Best of all, a parity check option is available. Kit: $399. Assembled: $459. Parity Option — kit: $39. Assembled: $59.

MICRO DISK SYSTEM

The North Star MDS is a complete floppy disk system with *all* hardware and software needed to add floppy disk memory and a powerful disk BASIC to S-100 bus computers. The North Star MDS is widely considered one of the best designed and most complete S-100 bus products available. The MDS includes the S-100 interface board with on-board PROM for system startup, Shugart minifloppy disk drive, cabling and connectors, and DOS and BASIC software on diskette. Kit: $699. Assembled: $799. Additional drive — Kit: $400. Assembled: $450. Single Drive Cabinet: $39. Optional Power Supply: $39.

Z80A PROCESSOR BOARD

The North Star ZPB brings the full speed, 4MHz Z80A microprocessor to the S-100 bus. Execution is more than twice the speed of an 8080, and the ZPB operates in systems both with and without front panels. The ZPB includes vectored interrupts, auto-jump startup, and space for 1K of on-board EPROM. Kit: $199. Assembled: $259. EPROM Option — kit: $49. Assembled: $69.

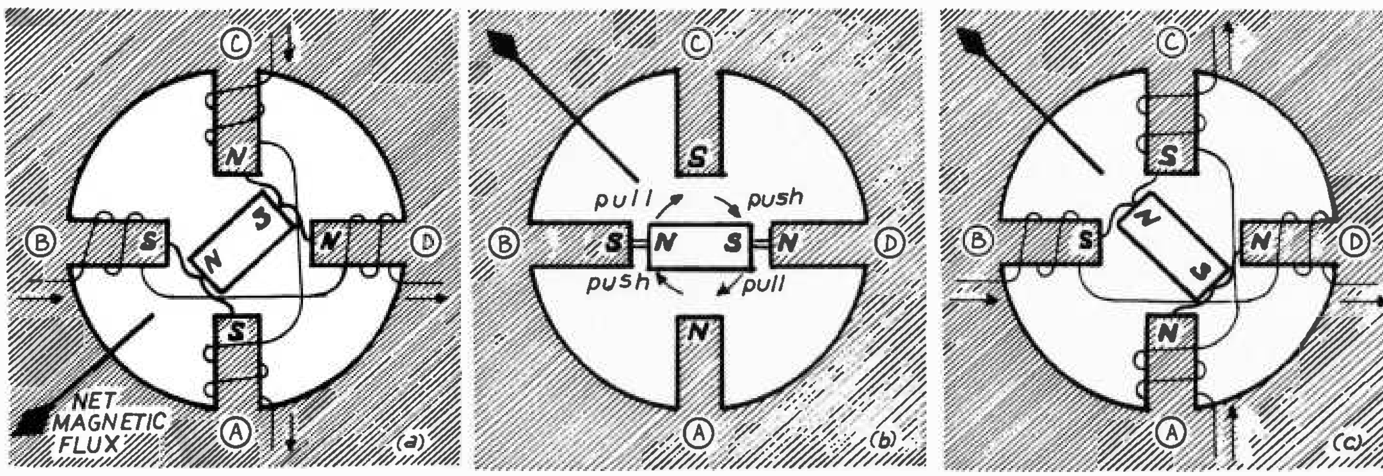
HARDWARE FLOATING POINT BOARD

If you do number crunching, then this board is for you. The FPB performs high-speed floating point add, subtract, multiply and divide with selectable precision up to 14 decimal digits. Arithmetic is up to 50 times faster than 8080 software, and BASIC programs can execute up to 10 times faster. A version of North Star BASIC is included. Kit: $259. Assembled: $359. Prices subject to change.

NORTH STAR ★ COMPUTERS
2465 Fourth Street • Berkeley, California 94710 • (415) 549-0858

Circle 86 on inquiry card.

Figure 3: When an additional pair of stator coils is added to the system it is possible to turn the rotor in a definite direction by a predetermined amount. The net magnetic flux is indicated by the heavy arrow in each case.



usually a hollow cylinder to reduce inertia.

Figure 2a shows a 4 pole motor with none of its poles energized. When power is applied it will snap clockwise to position 2b if the direction of current is as shown. This is because the opposite poles are attracting and the like poles are repelling. However, when the direction of current is reversed it is not possible to predict the direction the rotor will turn in going from 2b to 2c. We must energize another pair of poles as in figure 3 before we gain control of rotation.

In figure 3a the rotor is shown at rest aligned with the net magnetic flux field from the stator coils. All four coils now have current flowing. In figures 3b and 3c the current has been reversed through stator coils A and C. Therefore, the net magnetic flux now adds up as shown. In figure 3b the rotor has started to turn clockwise. This rotation is caused by the repulsion of the like poles as well as the attraction of the opposite poles. When the rotor poles again align with the magnetic flux of the stator, as in figure 3c, the rotor is again at rest. This motor requires four steps per revolution.

In a realizable motor there are more than four poles and they are folded up around the rotor. More poles will result in smoother rotation. The windings, two in these small motors, contribute magnetic flux to a number of poles. These coils may be center tapped to allow reversal of current in the windings with a single polarity power supply. This is illustrated more fully later on. Figure 4 shows an exploded view of a typical small stepper motor. The two coils are enclosed by steel cups which complete the outer magnetic path. The inner magnetic path of the stator is a series of small

Continued on page 102

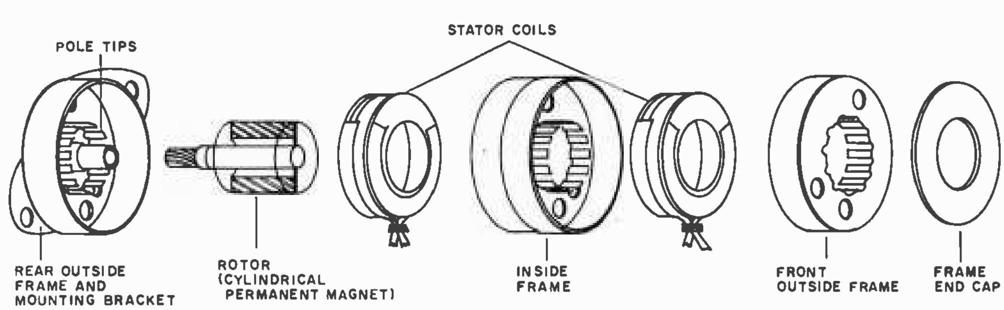


Figure 4: An exploded view of a typical stepper motor showing internal construction.

Ohio Scientific advances the state-of-the-art of small computers.

From our inexpensive 8K BASIC in ROM Challenger IIP to our powerful triple processor Challenger III, Ohio Scientific offers a full range of products that are technologically superior to anything available on the market today.

Challenger IIP

Challenger IIP from Ohio Scientific is our unique personal computer with BASIC in ROM and 4K RAM for programs in BASIC.

Complete with audio cassette interface and a full computer keyboard, Challenger IIP can be connected to a home TV via an RF converter and it's ready to go.

Challenger IIP comes fully assembled and tested for only **$598.00.**

Challenger II

Challenger II from Ohio Scientific is a disk based computer capable of storing up to 500,000 bytes of information on an Ohio Scientific dual drive floppy disk.

Challenger II comes with 16K of RAM (the disk BASIC is automatically loaded into the computer so there is no need for ROM's) and our powerful Disk Operating System (DOS) which allows the computer to perform big computer functions like random access, sequential and index sequential files in BASIC, and I/O distributors which support multiple terminals and industry standard line printers.

And best of all a 16K Challenger II with serial interface, single drive floppy disk, (250,000 bytes) BASIC and DOS costs only **$1,964.00** fully assembled.

Challenger III

Challenger III from Ohio Scientific is the revolutionary, new triple processor computer that allows you to run programs written for the 6502A, 6800 and Z-80 processors.

Incredible as this is, a disk based Challenger III costs only about 10% more than conventional single processor microcomputers. A 32K Challenger III with a serial interface and a dual drive floppy disk assembled and tested costs **$3,481.00.**

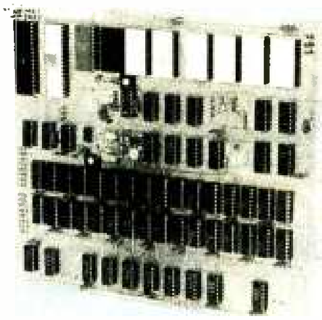
OHIO SCIENTIFIC

11679 Hayden • Hiram, Ohio 44234

For more information send for our Free, short form catalog, or send $1 for our 64 pg. Small Computing Buyers Guide.

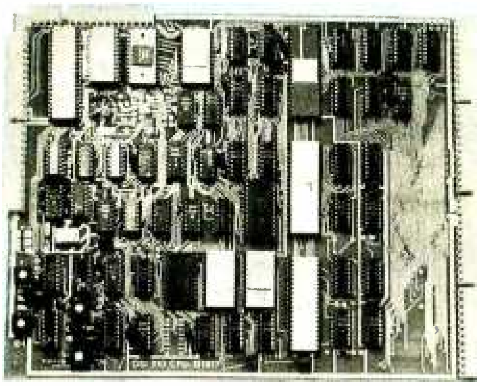
Introducing three boards only Ohio Scientific could build.

Ohio Scientific provides 15 system boards offered in over 40 different versions for Ohio Scientific Computer users. All of the boards are compatible with Ohio Scientific systems and many of them are by far technologically superior to any other microcomputer products on the market. And Ohio Scientific has the technology that made them possible.



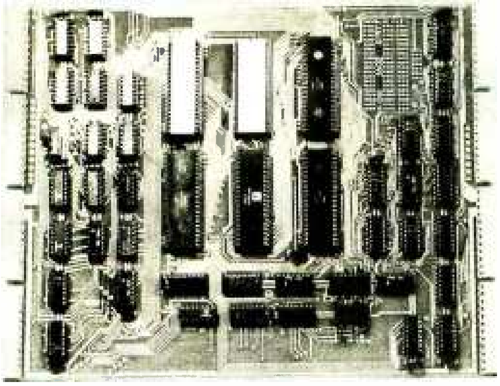
500 CPU Board

This board gives you our ultra-fast 8K BASIC in ROM with plenty of user workspace (4K RAM) for as little as $298.00. Use it as a standalone or as the CPU in a large system. BASIC is there the instant you turn it on. And in the October issue of *Kilobaud Magazine,* our version of 8K BASIC came out the winner in a BASIC timing comparison test of all of our competitors. The 500 is the fastest around!



510 Systems CPU Board

This is our unbelievable triple processor board! Complete with the 6502A, 6800, and Z-80 processors, this board allows you to run virtually all programs published for small computers. Available in the Challenger III, the 510 board is ideal for industrial development and research applications. There isn't another triple processor board like the 510 anywhere, except at Ohio Scientific!



560Z CPU Expander Board

The 560Z board is our multiprocessing board with a Z-80 and 6100 chip. This board allows you to run several processors simultaneously and the 6100 chip lets you run powerful PDP8 software with the 560Z. The 560Z board is the only multiprocessing board available for small computers, and Ohio Scientific makes it!

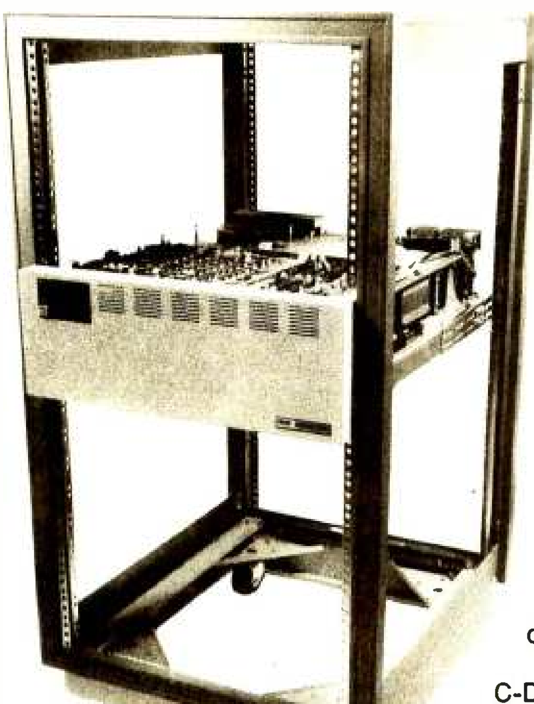
These three state-of-the-art CPUs are only a small part of the picture. Ohio Scientific's advanced technology offers you other unique features such as Multiport Memories, Distributed Processing, Big Disks with up to 300 megabytes on line, and Advanced Software.

OHIO SCIENTIFIC

11679 Hayden • Hiram, Ohio 44234

Announcing the most advanced disk anywhere for $6,000

The 74 megabyte disk from Ohio Scientific



C-D74 from Ohio Scientific is the ultimate storage device for small computers.

The C-D74 is the first Winchester technology disk for small computers making big system technology affordable and reliable for the small system not under maintenance contract.

The disk uses a non-removable sealed chamber drive with a unique rotary positioner to provide the highest performance disk available today.

The Ohio Scientific C-D74 can store all the records of a medium size company for instant access. And the Winchester technology of the C-D74 means that the drive can run 24 hours a day without worry of disk wear.

There are other important C-D74 applications in business computing and research in computing itself. The disk makes small computers practical for much larger jobs than formerly thought feasible, particularly since most business computing is disk bound and not computer bound.

C-D74 provides an unbelievable 35 millisecond average access time to any of 74 million bytes of information. With a 10 millisecond single track seek, the drive has an incredible data transfer rate of 7.3 megabits per second.

Recommended minimum hardware for the C-D74 is a Challenger with 32K RAM and at least 8K on a Dual Port 525 board, and a single or dual-drive floppy disk.

The drive, cable, interface for an Ohio Scientific Challenger and OS-74 operating system software is $6,000 FOB Hiram, OH. Equipment rack shown not included.

OHIO SCIENTIFIC DEALERS

American Microprocessors Equipment & Supply Corp.
20 N. Milwaukee Ave
Prarieview, IL 60069
(312) 634-0076

Century 23
4566 Spring Mountain Rd.
Las Vegas, NV 89102

Computer Business
RR #4
Box 910
Joplin, MO 64801

Computer Hobby Shop
4450 Trinity Avenue
Salt Lake City, UT 84120

Computer Mart of New York
118 Madison Ave.
New York, NY 10010
(212) 686-7923

Computer Power
P.O. Box 28193
San Diego, CA 92128
(714) 746-0064

Delaware Microsystems
92 E. Main St. #1
Newark, DE 19711
(302) 738-3700

Desert Data
Microcomputer Sales
P.O. Box 1334
Tucson, AZ 85702
(602) 623-6502

H/B Computers
217 East Main St.
Charlottesville, VA 22903
(804) 295-1975

Microcomp
P.O. Box 1221
Fond-Du-Lac, WI 54935
(414) 922-2515

Microcomputer Workshop
234 Tennyson Terr.
Williamsville, NY 14221
(716) 634-6844

Omaha Computer Store
4540 S. 84th St.
Omaha, NE 68127
(402) 592-3590

REPRESENTATIVES

Abacus Data
P.O. Box 276
Oil City, PA 16301

Associates Consultants
33 Ogden Ave.
East Williston, NY 11596
(516) 746-1079

BRAG Microcomputers
19 Cambridge St.
Rochester, NY 14607
(716) 442-5861

Computer Business
P.O. Box 171
LaPorte, IN 46350
(219) 362-5812

Johnson Computer
P.O. Box 523
Medina, OH 44256
(216) 725-4560

Omega Computing, Ltd.
Box 220, Station P
Toronto, Ont. M5S 2S7
(416) 424-2174

Pan Atlantic Computer Systems, GmbH
61 Darmstadt
Frankfurterstrasse 78
West Germany
(08102) 3206

Secom Systems
541-1 New Peach Tree Rd.
Chamblee, GA 30341
(404) 934-3272

Spectrum Technology Services
P.O. Box 942
Palos Verdes Estates, CA 90274

Systems Engineering Enterprises
Suite 307
1749 Rockville Pike
Rockville, MD 20852
(301) 468-1823

Tek-Aids, Inc.
1513 Crain St.
Evanston, IL 60202
(312) 328-0110

Yingco, Inc.
2 World Trade Cntr.
Penthouse 107th Floor
New York, NY 10048
(212) 775-1184

The state of the art in small computers.

call 1-216-569-3241

Simulation of Motion

Part 4: Extended Objects, Applications for Boating

Stephen P Smith
POB 841
Parksley VA 23421

Have you ever wondered why the shapes of boat hulls differ so widely? Boating enthusiasts know that certain designs will be best in lakes and rivers, and certain others in open seas. Some boats are much roomier than others; some are safer in rough water; but what penalties in stability and riding comfort might you pay for the extra room or seaworthiness? The motion of a boat depends on its response to the variety of waves it encounters. These motions can be simulated on your personal computer. You can determine how a given design will respond to any sea condition. The basic equations for stepping speed and position into the future will still apply, as they were discussed in the earlier articles of this series; but you'll also need some new techniques. As you implement this simulation, you'll discover that forces in a linear degree of freedom can also produce moments and their resulting motion in an angular degree of freedom. In this article, I'll show how that interaction is handled. I'll also introduce the concept of distributed forces, and a numerical technique to handle them. Although developed for a boating application, these new ways of calculating forces should find use in updating many of our previous simulations.

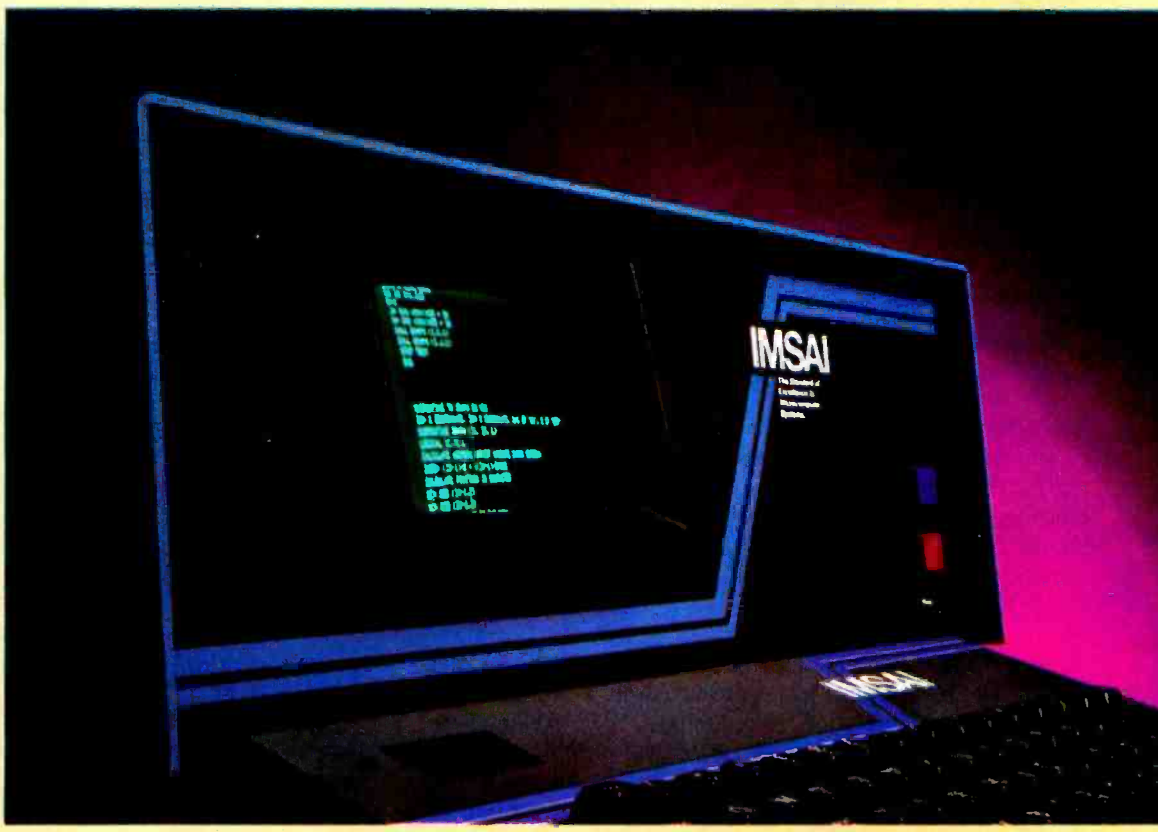
We have already seen quite a variety of ways to calculate forces. Gravity, a force present in every simulation in the last three articles, simply made a constant change in the vertical speed at each step. Thrust, used in rocket and aircraft simulations, came either from a user input or from a table interpolation. Forces in an automobile suspension were found to depend directly on the vertical position (spring force) and the vertical speed (damping force). Aerodynamic forces were computed by multiplying a coefficient (ie: constant) by the sum of the squares of the speeds in each linear degree of freedom. While these examples cover most of the situations you are likely to encounter in simple models, any new simulation might present some unique requirement.

For example, in all the calculations, the forces have had one thing in common. They acted at a single point. We call such forces discrete. In reality, some are not discrete, but act at many points on a body simultaneously. These are referred to as distributed forces. Aerodynamic drag is a typical distributed force. Although we used the drag coefficient to calculate a single force, the retarding action of the air acts all over the body. A coefficient is just one tool used to convert distributed forces into discrete ones. Not all distributed forces can be converted using coefficients, so I'll introduce a more general technique using the boating simulation as an example.

The principle forces on a boat are gravity and buoyancy, the floating power of the hull. Because buoyancy is an upward push provided by the water, it is not difficult to see that it is a distributed force covering the entire area of the boat below the water line. Converting this distributed force to a discrete one will allow us to simulate the vertical motion of the boat.

Perhaps more important to the boat designer or buyer will be the angular, rolling

Price/Performance no one else has put together.



IMSAI Introduces the PCS-80/30 Integrated Video Computer

- ☐ 3 mHz microcomputer
- ☐ 3K ROM monitor
- ☐ 2K RAM
- ☐ 8 expansion slots
- ☐ 14 mHz integrated intelligent CRT (24 x 80)
- ☐ Intelligent Keyboard
- ☐ Parallel interface
- ☐ 2 serial interfaces
- ☐ Graphic and edit CRT functions
- ☐ 28 amp power supply
- ☐ All for **$1499*** assembled and tested.

Software:

ASSEMBLER, DOS, TTY and ACR BASIC, Commercial/Scientific BASIC, FORTRAN IV and more.

Memory Options:

4, 16, 32 and 64K RAM memory boards.

Circle 57 on inquiry card.

Peripherals:

Printers, floppies—mix and match; standard, mini and high performance double density floppy disks on the same system (an industry first), ½" magnetic tape drive, CRT and TV monitors.

Send today for your copy of the 250-page IMSAI PCS-80/30 User Manual. Price $5.00. The cost of the manual may be applied towards your purchase of a PCS-80/30 Integrated Video Computer. Use the coupon provided.

Gentlemen:
Send me the IMSAI PCS-80/30 User Manual, price $5.00. I understand that the $5.00 may be applied towards the purchase of a PCS-80/30 Video computer System.

Check/Money Order enclosed. Amt. $ ________
Charge my ☐ BAC ☐ M/C

\# ________ Exp. Date: ________

Sig. ________

Name ________

Address ________

City ________

State/Zip ________

IMSAI 14860 Wicks Blvd. San Leandro, CA 94577

IMSAI®

The Standard of Excellence in Microcomputer Systems

IMSAI Manufacturing Corporation
14860 Wicks Blvd. San Leandro, CA 94577
(415) 483-2093 TWX 910-366 7287

*U.S. domestic price only. PCS-80/30 kit available. Other PCS-80 kits start at $699. Prices and specifications subject to change without notice.

	Typical Bodies of Water				
Wind Speed	Rivers	Lakes	Inlets	Bays	Open Sea
2 m/sec					
period (sec)	0.6	1.0	1.5	2.0	3.5
length (m)	0.56	1.5	2.3	3.1	20.0
height (m)	0.02	0.06	0.12	0.15	0.5
5 m/sec					
period (sec)	0.8	1.2	2.0	2.4	4.5
length (m)	0.1	2.25	5.0	9.0	30.0
height (m)	0.05	0.08	0.2	0.25	0.75
10 m/sec					
period (sec)	1.25	2.0	3.0	4.25	7.0
length (m)	2.4	6.25	14.0	28.0	80.0
height (m)	0.08	0.15	0.35	0.7	2.0
20 m/sec					
period (sec)	2.5	4.0	6.0	8.5	14.0
length (m)	10.0	25.0	56.0	110.0	300.0
height (m)	0.25	0.65	1.4	2.8	7.5

Table 1: Characteristics of waves. The height, period and length of waves all vary, but for certain conditions, average values have been established. The wave length and period are affected by the depth of the water. The height depends on the wind speed, how long it has been blowing, and the width of the body of water. Readers who want to model real sea conditions should find a good oceanography text, but the above summary should prove adequate for casual use.

and pitching motion of the boat. Angular motion was introduced in a rocket flight simulation (see January 1977 BYTE, page 144). In that case, it was entirely independent of the linear motion. At the end of the same article I suggested that the motion of an automobile body should also be simulated using an angular degree of freedom, but that the angular and linear motions could no longer be considered separately. This is also true in the boating example. The moments used to compute the angular motions will be calculated directly from the linear motions. Because the forces in the automotive example are discrete, we'll develop the technique to handle combined angular and linear motion using the distributed forces of the boat example. In that way, one simulation will serve to demonstrate both of the new concepts. I'll leave the development of a two or four wheel automobile suspension simulation to interested readers.

The motion of a boat is similar in many ways to that of the automobile body. When it is launched, a boat settles into the water in response to gravity. As the hull displaces more water, the buoyant force becomes larger, until at some point, it balances gravity and the boat stops sinking. This point is called equilibrium and is analogous to the equilibrium of an automobile suspension. Unless there is a disturbance, the boat will remain at equilibrium. In the automotive example, disturbances came in the form of a rising or falling road. With boats, we encounter a rising and falling sea, in other words, waves.

Sea waves occur in a variety of shapes. Their length (distance peak to peak), their height (distance peak to trough), and their period (time to rise and fall), all vary apparently at random. In fact, these parameters have fairly well defined relationships. Readers with an interest in modeling sea states should refer to a good marine science text. For this simulation, we'll represent waves with a sine function, and use the data in table 1 to compute their size.

Dealing just with forces for a moment, let's see how a small object is affected by wave motions. Figure 1 shows a bottle, floating in a body of water. We know from our previous simulations that every second, gravity subtracts 9.8 meters per second from the bottle's vertical speed. If the bottle is to remain stationary, the effect of buoyancy (force divided by mass) must be equal and opposite (ie: 9.8 meters per second per second upward). The mass of the bottle should be known. Let it be 0.1 kilograms. The buoyant force is equal to the weight of the water displaced by the bottle. Remember that weight is a force, the effect of gravity acting on a mass. The weight of the water, in newtons, is equal to its mass, in kilograms, times the effect of gravity, 9.8 meters per second per second. Each 1000 cubic centimeters (cc) of water has a mass of 1 kilogram, and thus a weight of 9.8 newtons. Knowing this, and the mass of

extensys corporation

High performance, cost effective products for your microcomputer.

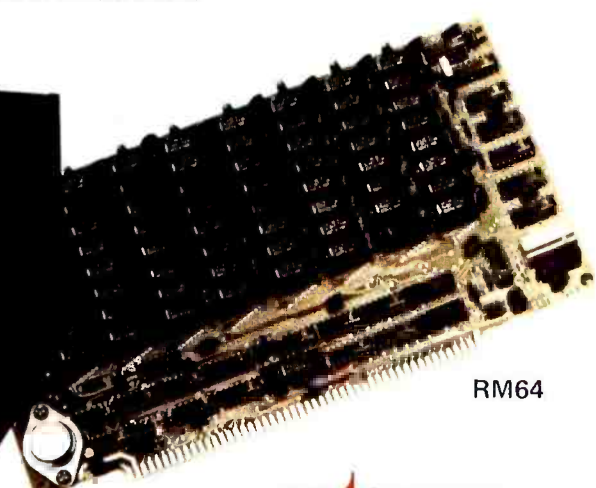
The Extensys FOS100 Floppy Disk System provides a high performance floppy disk system based upon multiprocessors in a multiuser environment. The FOS100 accomplishes this through a completely integrated hardware and software system. The hardware consists of a File I/O board, a PerSci controller board and one or two dual PerSci Model 277 floppy disk drives. The File I/O board has an 8080A microprocessor, 1K of PROM and 8K of RAM on-board to provide a separate I/O processor which will handle all disk I/O functions of the system. The PerSci controller board is encased in an attractive cabinet with either one or two dual PerSci drives.

The software, Extensys Multiprocessor Operating System, EMOS, provides many large computer capabilities, such as multiprocessor, multiuser operation, individual file security based upon user supplied passwords, system security and IBM compatible format. Additionally, the FOS100 has optional Direct Memory Access (DMA) capabilities when used in conjunction with the Extensys MM16 Memory Manager board.

The Extensys RM64 Memory Board provides up to 64K on one board. Designed for S-100 based computers, the RM64 occupies only one bus slot. Four versions of the same board are available: 16K, 32K, 48K and 64K bytes with the less-than-64K versions easily expandable by inserting memory chips into existing sockets. The Board uses dynamic RAMs with 300 ns access time and "invisible" on-board refresh. Other features include bank select logic for PROM/ROM coexistence plus expansion beyond 64K, write protect circuitry, and multilayer ground plane construction for noise immunity.

The RM64 comes completely assembled and tested with a one year warranty. Delivery is STOCK to 15 days upon receipt of order, with shipping and handling prepaid in the continental United States.

FOS100



RM64

Are you planning to produce a high performance, cost effective computer system or expanding the capabilities of your current line? If so contact US, the people who can make it happen for you NOW.

extensys corporation

Extensys Corporation
380 Bernardo Avenue
Mountain View, CA 94040
(415) 969-6100

the bottle, we can calculate the amount of water that the bottle displaces. In other words, we can find the volume, V, of the bottle below the water line at equilibrium. Force divided by mass must equal 9.8 to balance gravity, so the equation 9.8 = 9.8/ 1000 * V / 0.1 can be solved for V to find that 100 cc of the bottle is under water. If the bottle is 4 centimeters in diameter, we will find (from the formula for the volume of a cylinder) about 8 centimeters of its length must be below the surface.

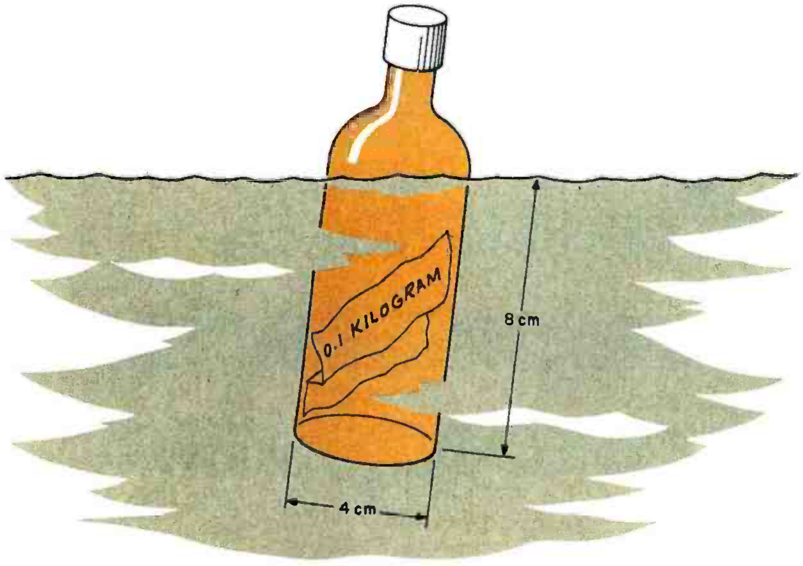


Figure 1: A bottle sinks until it displaces an amount of water equal to its own weight.

Now suppose that the surface of the water rises suddenly. More than 8 cm of the bottle will be underwater, and the buoyant force will exceed gravity. The vertical speed of the bottle will increase and it will rise with the water. When the bottle reaches equilibrium again it will still have a positive vertical speed, so it will pass through that point and continue to rise. Now, however, it is gravity which is the larger force, and the vertical speed will be reduced until the bottle begins to descend. Eventually, these motions will disappear (due to the drag force applied by the water) and the bottle will come to rest at equilibrium. This happens so quickly that the bottle appears to be moving up and down exactly with the waves.

For larger objects, boats for example, the actual motion may be more apparent. We could treat a boat exactly like the bottle and simulate its up and down motion. As I suggested earlier, however, it is the angular rolling and pitching motion of the boat that is of real interest. To simulate these motions,

SCREENSPLITTER

A COMPLETE TV TEXT DISPLAY SYSTEM
from
Micro Diversions, Inc.

40 LINES
86 CHARACTERS/LINE

ONBOARD WINDOW SOFTWARE FOR CONTROLLING UP TO 50 LOGICALLY INDEPENDENT WINDOWS

GENERAL

SCREENSPLITTER is a self-contained TV text display system for your microcomputer. It comes on a single, high-quality S-100 compatible board, complete with its unique Window Package software module. With the Window Package, you can logically segment SCREENSPLITTER's huge 40 x 86 display of upper-lower case characters into up to 50 independent "windows" of various sizes. (You get the idea from our ad!) Each window has its own optional frame, cursor, figure-ground, and optional label, and each window scrolls and automatically formats its text independently of all the others. QUICK! There's some interesting information flashing by in WINDOW 1. Go read it!

The Wind
ow Packa
ge's aut
o-format
ter does
n't care
how ski
nny your
windows
are (it
'll hack
your te
xt down
to one c
olumn if
you can
stand i
t!

WINDOW 1

▪∑∫÷∂~∈≤ı▲¶ Whoops! Our output routine seems to be having problems. Oh well, at least you get to see some of SCREENSPLITTER'S scientific symbols. (You can order a graphics character set optionally.)

▲△▲△▲△ time — And any character may be user-defined as a winking character. How? you ask. Simple: SCREENSPLITTER uses a 2708 reprogrammable memory as its character generator. Turn on the character's "wink" bit in the 2708, and presto!

Hi, and naturally, each of the 3440 characters on the screen may have its figure-ground reversed independently!

Frills, you say? No, thrills! Just take a look in the window up there ↗ to see how SCREENSPLITTER puts these raw materials to work in the onboard 1K Window Package▲ (that back there → is the cursor character).

PARTIAL FUNCTION SUMMARY

INIT()
OPEN(W,X,Y,DX,DY)
CLOSE(W)
FLASH(W)
FRAME(W)
UNFRAME(W)
LABEL(W,STRING)
SCROLL(W,N)
CURSORCHAR(W,CHAR)
CURSORPOS(W,X,Y)
PRINT(W,STRING)
PRINTCH(W,CHAR)
BACKSPACE(W)
CLEARLINE(W)
CLEAR(W)
BLANKLINE(W,COUNT)
COMPLEMENT(W)
PLOT(W,X,Y,CHAR)
MOVEWINDOW(W,X,Y)

POINTS OF INTEREST

- Entire hardware/software system on a single, high-quality S-100 buss compatible board.
- Drives a 10mhz or better TV monitor via standard 75 ohm coaxial cable (supplied).
- 4K static RAM -2114's- display buffer is memory-mapped into your CPU's address space for fast, convenient access if you ever need to bypass the Window Package software.
- User-selectable wait state for operation with 4mhz CPU's.
- 1K onboard 2708 is jumper changeable to a 2K 2716 for user extensions to the Window Package.
- Board presents one TTL load to host, yet drives up to 20 TTL loads via 74367 buffers.
- Provisions for jumpering TV data, sync, blanking off board for external mixing (via 16 pin socket).

WHAT YOU GET

- Complete SCREENSPLITTER Kit, with all IC's, low profile sockets, preprogrammed Window Package EPROM, assembly instructions
- Comprehensive Theory of Operation Manual
- Complete source-code listing, and User's Manual for the Window Package
- 90 day warranty on parts and labor

+
+
+ + +
+ + +
+ + + + +
the PLOT(W, X,Y,"+") thickens

ORDERING INFORMATION

1. Tell us for which 8K boundary you would like your Window Package assembled.
2. Tell us whether you want the scientific symbols, or the graphics characters in ASCII codes 0-31 of your character generator.
3. Send us a personal check, Master Charge or BAC/VISA number and expiration date. Kit price is $295. Assembled, $385. (Virginia residents please add sales tax.)
4. We will send you the SCREENSPLITTER, postpaid (in the continental U.S.) from stock to 60 days.

THE CARE AND FEEDING OF WINDOWS

OK. You have just powered on. Initialize the Window Package and turn on your first window:

```
INIT()
OPEN(1,10,15,20,30)
```

window 1

Now, just to flex your bits, give the user a wake-up flash (a brief figure-ground reversal inside the window):

```
FLASH(1)
```

Now that you have his attention, go ahead and frame the window (you don't have to, of course):

```
FRAME(1)
```

and, while you're at it, label it, and set the scroll line count:

```
LABEL(1,"General I/O")
SCROLL(1,5)          (i.e., when the window
                      fills up, pop it up 5
                      blank lines)
```

Just to keep him interested, switch the cursor character from the default caret to the winking caret:

```
CURSORCHAR(1,▲)
```

Now that he's all excited, eyes bulging from the initial flash, transfixed by the hypnotic winking cursor, hit him with some text through window 1:

```
PRINT(1,"I hate to tell you this, William, but
         last night the kids wired that chair
         you're sitting in with 110 volts AC.")
```

Now (this'll really kill him), open a second window to the right:

```
OPEN(2,10,50,5,20)
FRAME(2)
LABEL(2,"Will's Will")
```

and print out a second message through this new window:

```
PRINT(2,"Please type your last will and testament.")
```

Now, of course, you echo his input through window 2, relying on the default scrolling of 1-line "pop-up" when the window fills up.

And on, and on...

SOME APPLICATIONS

1. You have a BASIC program. Open a number of windows, giving each important subroutine in the program its own window. When your program runs, you get a two-dimensional feel of the flow of the execution - flurries of activity here, brief flashes there. You can have the feeling of being able to converse with each subroutine individually!
2. You have a page-oriented text editor. Pick up a paragraph here, a paragraph there, isolating each in its own window while you rummage through the main text in its own large window. Using the MOVEWINDOW function, you can move blocks of text around to produce a final layout.
3. You have an assembly language debugger. Allocate one window to the real-time clock, another to the run-time clock, and several more to display various registers in your 8080 or Z80. Then, you can keep the debugging information separate from your program's I/O, with the debugging information continually present.
4. You have some fancy games. Give each player his own window and define some "community windows." Let your imagination take over!

PS: Watch for our full graphics board, also with its onboard software... Coming soon!

Micro Diversions, Inc.

7900 Westpark Drive, Suite 308
Mc Lean, Virginia 22101
(703) 827-0888

DEALER INQUIRIES INVITED
EVALUATION KITS AVAILABLE

TIME
06:01:48

Circle 70 on inquiry card.

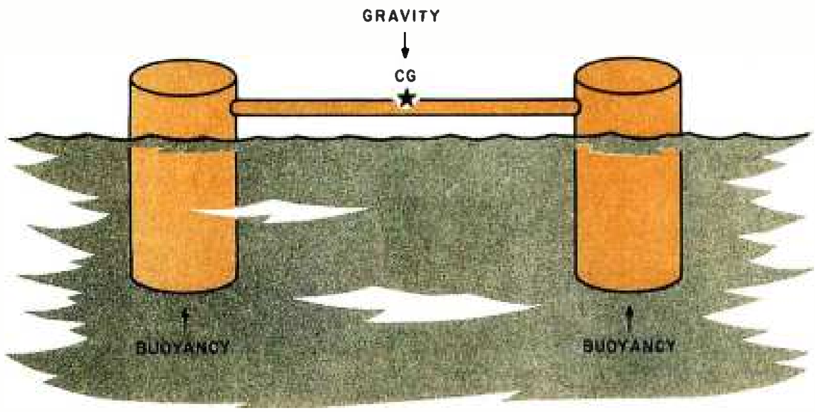


Figure 2: The distribution of the buoyant force determines the angular motion about the center of gravity.

we will need to know not just the total buoyant force, but also how it is distributed over the hull. The device shown in figure 2 will illustrate a general technique for finding the distribution.

You can think of this device as two bottles joined together with a stick or as the two hulls of a catamaran. Just as we calculated the buoyant force on the bottle in figure 1, we can calculate separate forces on each of these two bottles. The sum of the forces can be used to compute the vertical motion of one point on the device. This point is called the center of gravity (CG). The location of the center of gravity is critical. If you were to place the stick on a knife edge and find the point at which it balanced, this would be the center of gravity. It is the point on a body where the effect of gravity appears to be concentrated. Because the mass of an object is distributed throughout its volume, weight is a distributed force. By locating the center of gravity, however, we have a tool that transforms it into a discrete one.

We could also define a center of buoyancy, the point at which the total buoyant force appears to act. Unfortunately, the location of this point can move significantly as the boat rises and sinks in the water. The center of gravity is also subject to some movement, such as when a passenger moves from the back to the front of the boat. Unlike the movement of the center of buoyancy, however, changes in the location of the center of gravity are not tied directly to the results of our simulation, the linear and angular motion of the boat. For this simulation, we will treat the center of gravity as stationary, and try to avoid dealing with the moving center of buoyancy.

Since we cannot deal with buoyancy as simply as we do gravity, we will have to deal with the individual parts of a body more directly to find a general method of handling the distribution. In the case of the "catamaran" in figure 2, this is fairly easy. First, we assume that the bottles are small when compared to the length of the stick. Next, we assume that the center of buoyancy of each bottle is at its center, no matter how it sits in the water. Now, as far as our simulation is concerned, the entire buoyant force on the bottle acts at a point which is at a known distance from the center of gravity. This makes no difference to the vertical degree of freedom, but it is the key which allows us to simulate the angular motion.

Remember from the last article that a moment is the product of a force times a distance. In the current example, each bottle creates a moment equal to the buoyant force times the distance of the bottle from the center of gravity. Note that we define distances to the right as positive, and to the left as negative. Thus an upward force on the righthand bottle creates a positive (counterclockwise) moment. An upward force on the lefthand bottle creates a negative moment. In each simulation step, the moments are summed and then divided by the moment of inertia to find the change in angular speed each second. With this value, we can step the angular degree of freedom into the future, and return to compute new forces and moments.

Now we must determine how the combined angular and linear motion can be used to compute the new buoyant force. The force is proportional to the volume of the bottle below the waterline. For a single bottle, it was computed from the position in the vertical degree of freedom, and the location of the water surface. With the two bottle device, the vertical degree of freedom tells us only the position of the center of gravity. We must use the angular degree to find the relative position of other points on the device. If there is a positive angular position, the device will be turned counterclockwise around the center of gravity. Consequently, the lefthand bottle will be lower than the center of gravity and the right one will be higher. The exact difference is calculated by multiplying the sine of the angular position by the distance of the bottle from the center of gravity. Again, note that points to the left have a negative distance from the center of gravity. Positive angular positions move them down.

Let's illustrate this with an example. Suppose the vertical position of the center of gravity is 0.01 meters, and the angular position is 2 degrees (0.035 radians). A bottle 1.2 meters to the left would be at

$0.01 + SIN (0.035) * (-1.2) = -0.32$ meters

in other words, about 3 centimeters below its equilibrium position. A bottle 1.2 meters to the right would be

$0.01 + SIN (0.035) * 1.2 = 0.052$ meters high.

The vertical position of any other point can be found similarly.

Having found the positions of the bottles, we must now find the positions of the water surface at each bottle. These will come from a sine function modified by a representative wave height, period and length. The argument of the function will be the sum of the current time divided by the period, and the bottle location (distance from the center of gravity) divided by the wave length, all multiplied by two π (to convert to radians). Once evaluated, the function is multiplied by one half the wave height (amplitude) to arrive at a final surface position. Using this scheme the surface varies with both time and location in a good approximation of sea waves.

The data in table 1 can be used to continue the example we began above. Let's place our two bottle catamaran in an inlet with 5 meter per second winds. We have determined that the water surface is given by the following formula.

S = HEIGHT/2 * SIN (6.28318*(TIME/PERIOD + LOCATION/LENGTH))

At TIME = 1.8 seconds, we would find that the water surface at the left bottle is 0.2/2*SIN(6.28318*(1.8/2+(1.2/5))=0.084 meters, just below the equilibrium position. At the right bottle, the surface is at 0.2/2*SIN (6.28318*(1.8/2+1.2/5))=0.077 meters. If we subtract the positions of the bottles from these values and add the 8 centimeter length of the bottles underwater at equilibrium, we will have calculated the length of each bottle below the surface at TIME = 1.8 seconds. For the left bottle this will be (−0.084) − (−0.032) + 0.08 = 0.28 meters. For the right bottle this will be 0.077 − 0.052 + 0.08 = 0.105 meters. If the bottles are 4 centimeters in diameter, then the left one displaces 0.04**2 * 3.14159/4 * 0.028 = 3.45 * 10 −5 cubic meters and has a buoyant force of 9800 * 3.45 * 10 −5 = 0.345 newtons. The moment it produces is 0.345 * (−1.2) = −0.414 newton meters. Similarly, the right bottle displaces 9.67 * 10 −5 cubic meters and produces a force of 0.948 newtons and a moment of 1.14 newton meters. The sum of the forces, 1.293 newtons, is used to update the vertical degree of freedom. The sum of the moments, 0.534 newton meters, is used to update the angular degree of freedom. Now we can compute new positions, forces, moments, etc, and begin the cycle again.

Simulating the motion of a two bottle catamaran may not be very useful, but the technique is easily extended to real boats. Instead of thinking of the hull of your boat as a continuous surface, think of it as a collection of "bottles." Figure 3 shows a boat hull that has been divided in this

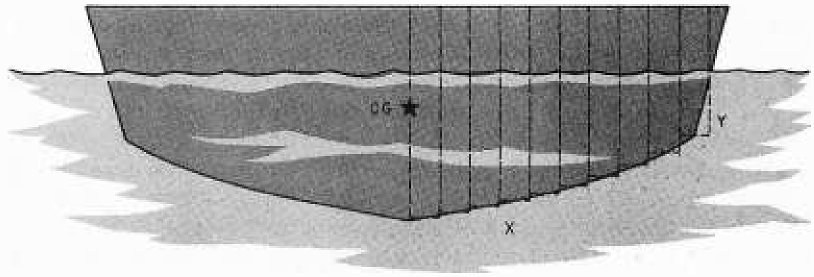


Figure 3: The continuous hull of a boat can be divided into a series of discrete segments or "bottles." X is the distance from the center of gravity to the center of the bottle. Y is the length of the bottle below the water line at equilibrium. Note that the symmetry about the CG enables us to describe the hull while only segmenting half of it.

X(m)	Y(m)
±0.05	0.50
±0.15	0.49
±0.25	0.47
±0.35	0.45
±0.45	0.42
±0.55	0.39
±0.65	0.36
±0.75	0.31
±0.85	0.37
±0.95	0.22

Listing 1: This program simulates the vertical and angular motion of a boat in response to sea waves. Because it involves a lengthy summation, it is inherently slow. I have, therefore, used only the second order predictor corrector formulas, and have employed a large step size. Readers who want more accuracy and who can afford to wait for results should implement the fourth order equations presented in my previous article on automotive applications (December 1977 BYTE, page 112). They should also increase the number of "bottles" used to describe the hull, and decrease the step size.

It should also be noted that the program does not simulate the viscous damping action of the water. As a result, if you are unfortunate enough to specify a resonant frequency of the hull as the wave period, the boat will appear to leap out of the water. While this result is obviously erroneous, it will highlight a design to be avoided.

```
100 REM SHIP MOTION SIMULATION
110 REM DESCIBE HULL CROSS SECTION
111 REM X IS DISTANCE FROM CG TO CENTER OF BOTTLE
112 REM Y IS LENGTH OF BOTTLE BELOW WATER AT EQUILIBRIUM
120 DIM X(10),Y(10)
140 DATA 0.05,0.5,0.15,0.49,0.25,0.47,0.35,0.45,0.45,0.42
150 DATA 0.55,0.39,0.65,0.36,0.75,0.31,0.85.0.27,0.95,0.22
160 FOR J=1 TO 10
170 READ X(J),Y(J)
180 NEXT J
190 REM SET BUOYANCY FACTOR; "BOTTLE AREA"*DENSITY*9.8
200 B=0.01*1.03*9.8
209 REM COMPUTE MASS(M) & MOMENT OF INERTIA OF CROSS SECTION
210 M=0
211 I=0
212 FOR J=1 TO 10
214 M1=B/9.8*Y(J)
216 M=M+M1*2
218 I=I+M1*X(J)*2
220 NEXT J
230 REM SET SEA STATE: HEIGHT(H),LENGTH(L),PERIOD(P)
240 H=0.2
250 L=5
260 P=2
270 REM INITIALIZE INTEGRATION VARIABLES
280 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
290 READ Z,Z1,V,V1,A,A1,R,R1,Q,Q1,C,C1,T
300 REM INITIALIZE STEP SIZE AND PRINT INTERVAL
310 D=0.1
315 K=0
320 K1=0.1/D
321 PRINT "TIME(SEC)    VERTICAL POSITION(M)  ANGULAR POSITION(DEG)"
330 REM SUM FORCES AND MOMENTS ON THE "BOTTLES"
340 GOSUB 600
345 REM PREDICT VERTICAL MOTION
346 REM A,V.Z ARE ACCELERATION.SPEED. AND POSITION
350 A1=F/M-9.8
360 V=V1+D*A1
370 Z=Z1+D*V1
375 REM PREDICT ANGULAR MOTION
376 REM C,Q,P ARE ACCELERATION,SPEED, AND POSITION
380 C1=G/I
390 Q=Q1+D+C1
400 R=R1+D*Q1
410 REM SUM NEW FORCES AND MOMENTS FOR CORRECTOR FORMULAS
420 K=K+1
430 T=T+D
440 GOSUB 600
445 REM CORRECT VERTICAL MOTION
450 A=F/M-9.8
460 V=V1+D/2*(A+A1)
470 Z=Z1+D/2*(V+V1)
475 REM CORRECT ANGULAR MOTION
480 C=G/I
490 Q=Q1+D/2*(C+C1)
500 R=R1+D/2*(Q+Q1)
508 REM PREPARE FOR NEXT STEP
510 V1=V
520 Z1=Z
530 Q1=Q
540 R1=R
550 IF K<K1 THEN 340
560 PRINT T,Z,R*57.296
570 IF T<10 THEN 340
571 STOP
600 REM CALCULATE AND SUM FORCES AND MOMENTS ON "BOTTLES"
630 F=0
640 G=0
660 FOR J=1 TO 10
665 REM POSITIVE HALF OF HULL IS GIVEN
666 REM W IS VERTICAL POSITION OF WATER SURFACE AT BOTTLE J
667 REM W1 IS LENGTH OF BOTTLE BELOW WATER SURFACE
668 W=H/2*SIN(6.28318*(T/P+X(J)/L))
669 W1=Y(J)-Z-SIN(R)*X(J)+W
```

manner. The hull can now be described by a table of Xs and Ys. The Xs represent the distances from the center of gravity to the center of each bottle. The Ys represent the lengths of each bottle below the waterline at equilibrium. Now, instead of making a series of calculations for one or two bottles, we make them for many. Just as before, the sum of the forces influences the vertical motion of the center of gravity, and the sum of the moments influences the angular motion around the center of gravity.

We now have an effective method for dealing with distributed forces. We simply divide the area over which the force acts into small segments. Within each segment, we neglect the distribution and calculate a discrete force and moment. Finally, we sum the force and moments to find the effects on the linear and angular speeds.

I have included a BASIC program with this article to illustrate the technique as applied to our boating example. With the data supplied, it computes the rolling motion of the hull cross section pictured in figure 3. Boating enthusiasts will be able to insert some other hull cross sections (deep vee, trihull, etc) in the data statements and compare the response to the sample sea states. If lateral (side to side) sections are used, the program will simulate rolling motion. If longitudinal (fore and aft) sections are used, the program will compute pitching motions. Interested readers should be able to extend the program to include three dimensional boat models and simulate both angular motions simultaneously.

With the inclusion of techniques for handling distributed forces and combined angular and linear motion, your collection of software tools for simulating motion is fairly complete. When using these tools on a personal computer, you should try above all to limit the scope of your simulations. Determine which motions really interest you and neglect or restrict the others. Divide the simulation into degrees of freedom, preferably three or less if your program is to execute with reasonable speed. Compute each force and moment individually, then apportion and sum them within the degrees of freedom. Finally, step the velocity and the position into the future. Use a small step size in your early runs, 0.01 seconds or less. Increase it to save run time only as long as your results do not change significantly. Following this procedure, and using the BASIC programming examples I have provided as a guide, you should be able to find some interesting new applications for your personal computer.

Listing 1, continued:

```
670 IF W1>0 THEN 672
671 W1=0
672 F1=B*W1
673 G1=X(J)*F1
675 REM MIRROR IMAGE GIVES NEGATIVE HALF
678 W=H/2*SIN(6.28318*(T/P-X(J)/L))
679 W1=Y(J)-Z+SIN(R)*X(J)+W
680 IF W1>0 THEN 682
681 W1=0
682 F2=B*W1
700 G2=-X(J)*F2
710 F=F+F1+F2
720 G=G+G1+G2
730 NEXT J
740 RETURN
750 END
```

```
TIME(SEC)   VERTICAL POSITION(M)   ANGULAR POSITION(DEG)
 0.1              0.00147774275297   0.579343661886
 0.2              0.00834856069988   2.1484430172
 0.3              0.0240789139748    4.23893821533
 0.4              0.0488239037315    6.19033468235
 0.5              0.0789839977127    7.30318099994
 0.6              0.10789765357      7.01051374777
 0.7              0.127464693145     5.02510841752
 0.8              0.130129340759     1.41142156817
 0.9              0.110974303806    -3.40279261698
 1                0.0695560412445   -8.66151908825
 1.1              0.0105695497413   -13.4108553239
 1.2             -0.0568528427092   -16.6916991803
 1.3             -0.120883364889    -17.7238637849
 1.4             -0.169478381978    -16.0583471104
 1.5             -0.193013619278    -11.6821229305
 1.6             -0.18645821109     -5.06270420258
 1.7             -0.150554633294     2.89082493065
 1.8             -0.0917208520104    10.9686349622
 1.9             -0.0206887363075    17.8818895739
 2                0.0503241234783    22.5634234899
```
■

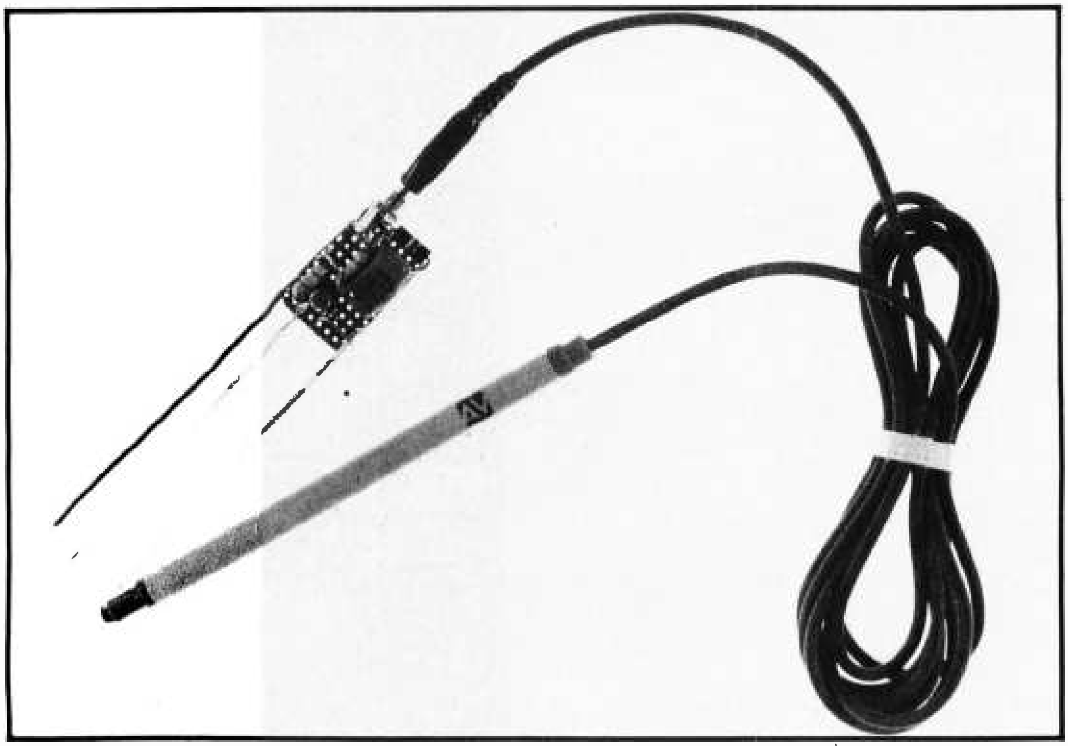


Photo 1: A general view of the authors' light pen and interface circuit. The pen itself was made from a fairly standard marking pen with a photo diode mounted at the tip. A length of shielded cable runs from the diode through the pen body to the miniature phone connector which plugs into the jack on the circuit board.

Add a $3 Light Pen to Your Video Display

John Webster and John Young
University of New Brunswick
Audio Visual Services
Kierstead Hall
Fredricton, New Brunswick
CANADA E3B 5A3

The use of a light pen can greatly facilitate entry of display characters on your video display. The layout of complex game boards, charts and graphs, or character editing can be accomplished more quickly and easily if you have the ability to add or delete characters anywhere on the screen without first having to position the cursor. This article describes the design and construction of a very inexpensive light pen and driver program to accomplish this function with a Processor Technology VDM-1.

The Circuit

Figure 1 shows the light pen circuit that can be constructed for well under three dollars. When used with a VDM-1 it requires no additional IO ports. Component layout of the circuit is noncritical. The authors' prototype was assembled on a small piece of perforated board and attached to the VDM-1 board. It could be mounted anywhere in the computer or keyboard enclosure. A four foot shielded cable connects the photo diode to the other components through an optional jack.

Any discarded ball point or felt tip pen may be used to house the diode. Alternatively it may be attached to the end of the cable with heat shrink tubing. The smaller the diameter of your light pen body, the easier it will be to use.

Any surplus quality photo diode may be used as a pickup. If the diode is of the type

ALL TOGETHER NOW!

The acclaimed Equinox 100™ mainframe kit ($799) is now a complete S-100 system.

Because now there is an Equinox 100™ I/O interface kit ($120) that handles the hard work of interfacing all your peripherals.

And Equinox 100™ 4K memory kits ($109). Assembled 8K memory boards ($188). EQU/ATE™ editor/assembler and BASIC-EQ™ software on cassettes.

It all goes together. It all works together. It's all together now at special system prices.

See The Equinox System™ at your local computer shop. *Call* toll-free to 800-648-5311. BAC/MC accepted. Or *write* Equinox Division, Parasitic Engineering, P.O. Box 6314, Albany California 94706.

THE EQUINOX SYSTEM™ When you put it together, it's really together.

Circle 93 on inquiry card.

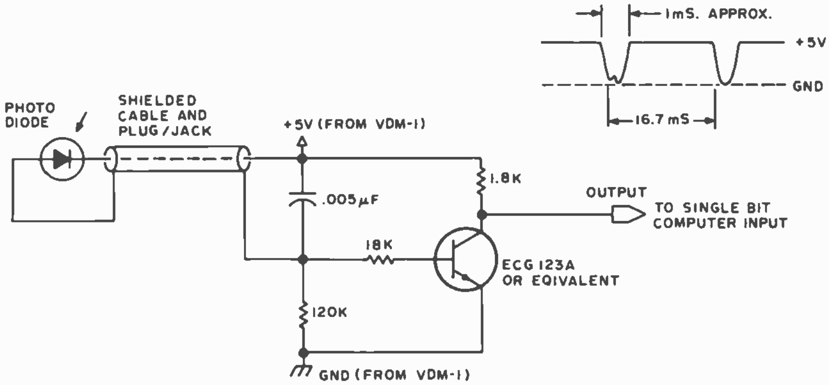


Figure 1: This fairly simple circuit transforms light from the video monitor output into a TTL signal which can be monitored by a program such as that shown in figure 2 and listing 1. A typical oscilloscope waveform of the output is shown in the figure.

mounted in a plastic lens, a fine file and emery cloth may be used to flatten the end and provide a narrower angle of acceptance.

When you have constructed the circuit, use an oscilloscope to monitor the output as you pass the pen across a television screen. A white area should produce an output as shown in figure one. A dark area should produce a 5 VDC level. Sensitivity can be adjusted by using the brightness and contrast controls on your television.

Once you have a satisfactory output, the circuit may be wired to the VDM-1. The output of the circuit is connected to pin 14 of IC39 on the VDM-1 board. This is an input to a spare three state buffer on the status port.

Then connect pin 13 of IC39 to pin 9 of IC39. This hooks the output of the three state buffer to data bit DI7. The output signal from the light pen will now appear on DI7 when an input from status port C8 is performed.

Program Design

The light pen circuit will produce a negative output whenever a white screen area is sensed. This condition may be used by appropriate software to locate the pen's position on the 16 by 64 grid of the VDM-1's display.

First, the top line of screen information is stored away and white cursors are written into all 64 positions of this line. The display is then scrolled so that this first line appears at the bottom of the screen and the rest of the screen is blanked. This is done by outputting hexadecimal F0 to the VDM-1's status port (hexadecimal C8).

The display is then scrolled upward one line at a time until an output from the pen is sensed indicating the proper line. Each time the value output to the status port is modified to scroll up one line, the status port value is also saved on the stack. As soon as the proper line is located, this status word is retrieved and decoded to find the actual unscrolled beginning of line address of the line the light pen is on. This decoding is achieved through a puzzling series of left rotations and additions in the BINGO section of the program. The resultant beginning of line address is then stored at locations hexadecimal ED and EH for future reference.

Black is now written over the white line, position by position, until the output from the light pen disappears indicating the location of the pen. The value in register L now indicates the displacement from the beginning of the line. The values in hexadecimal ED and EE are then recalled and L is added to the low order byte to produce the final H and L values for the light pen position. The information stored from the first line is returned to the screen and data

Now, Artec has an expandable elephant for everyone!

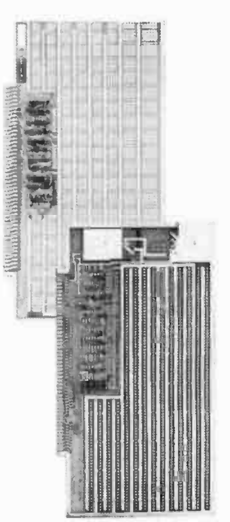
8K-32K of static RAM memory. Fully assembled or in kit form.

No matter what your needs, Artec has a memory board for you. You can start with 8K of TI 4044 memory on a 5.3" x 10" card and work your way up to a full 32K in 8K increments. The access time is only 250ns. The memory is addressable in 4K blocks and is perfect for S100 and battery augmented systems. The Artec 32K Expandable Memory has four regulator positions, bank select and plenty of room for all necessary support hardware. It uses less than 1 amp per 8K of memory (3.9 for 32K), and only +8 volts.

For five years Artec craftsmanship and reliability has been proven in tough industrial use. Now, you too can enjoy breadboards and memories that will work time after time. Send for an Artec board, your order will be sent the same day as received.

Kit: Board &	Fully Assembled Board:
8K of memory—$ 290.00	8K—$ 440.00
8K add on kits—$ 255.00	16K—$ 695.00
Full 32K board—$1,055.00	24K—$1,100.00
	32K—$1,205.00

GP100—$20.00
Maximum design versatility along with standard address decoding and buffering for S100 systems. Room for 32 uncommitted 16 pin IC's, 5 bus buffer & decoding chips, 1 DIP address select switch, a 5 volt regulator and more. High quality FR4 epoxy. All holes plated through. Reflowed solder circuitry.



WW100—$20.00
A wire wrap breadboard, similar to the GP100. Allows wire wrap of all sizes of sockets in any sizes of sockets in any combination. An extra regulator position for multiple voltage applications. Contact finger pads arranged for easy pin insertion.

Buffering Kit—$12.65
All the necessary components to bootstrap any Artec board into your system. Buffering I/O, DIP switch heat sinks and every support chip you need.

TO ORDER: Use your Mastercharge or BankAmericard. Or just send along a money order. We can accept only U.S. currency. Please include $3 handling on all orders. California residents add 6.0% sales tax.

FOR MORE INFORMATION: For more information about these or any of Artec's complete line of circuit boards or for either industrial or personal use, please call or write. A catalog will gladly be sent.

Please send me: (Include Quantity)
____ 32K ____ GP100 ____WW100
☐ I've enclosed a money order.
☐ Mastercharge No. ________ ________ Exp. Date
☐ BankAmericard No. ________ ________ Exp. Date
Name ________
Address ________
City ________ State ______ Zip ______
Calif. Res. add 6% Sales Tax *$3.00 Handling Encl.*

10% discount for students & computer club members.

ARTEC ELECTRONICS, INC.
605 Old County Rd., San Carlos, CA 94070
(415) 592-2740

Circle 6 on inquiry card.

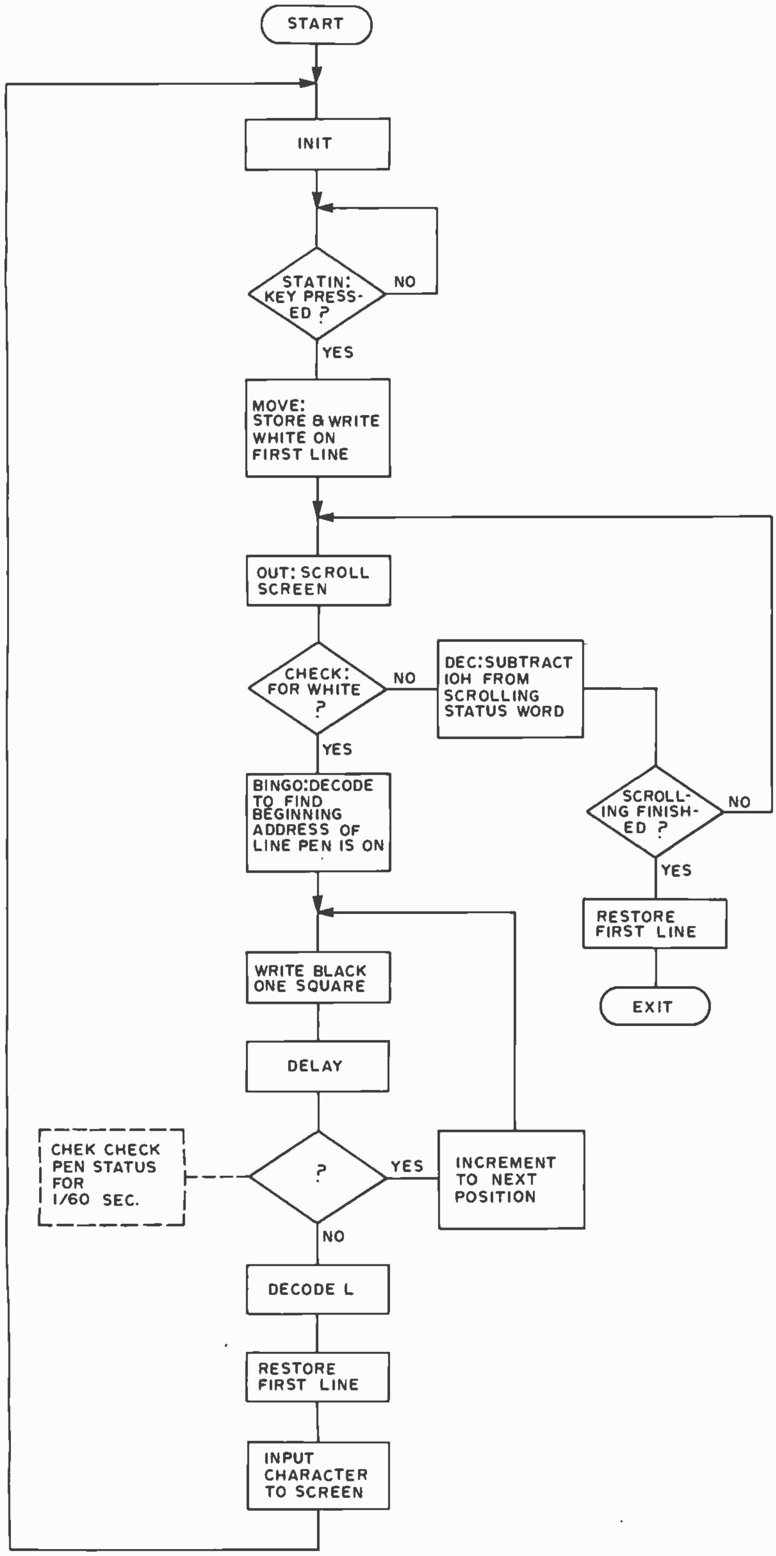


Figure 2: The general design of a cursor control algorithm for use with the light pen. The authors' version was used with the Processor Technology VDM-1, but a similar procedure should be achievable with other video generators.

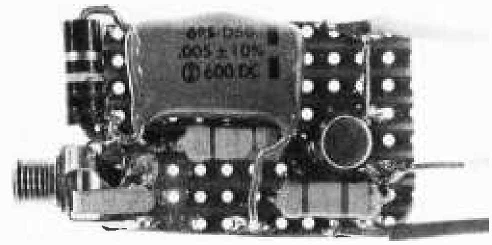


Photo 2: Detail of the light pen interface, as constructed on Vero perforated board (a product which contains pre-etched copper runs to which components can be soldered). The three wires leaving at right go to the processor input, power and ground; the light pen input is through a miniature phone connector at left.

from the keyboard is moved to the light pen position on the screen. The program then returns to the initialization section and waits for a new character.

If speed of operation is more important than program length a more sophisticated binary search procedure may be used once the proper line is found. If half the line is written over with black before the output of the pen is checked then half the line may be eliminated with only one check. Six such checks will cover all 64 possible positions on the line.

Delay loops of at least 1/60 second must be incorporated into each check to insure that the scan lines at the pen location are actually being written to during the check.

Using the Light Pen

With the LIGHT program running, the computer waits for a keyboard entry and then scrolls the screen and begins its search. If it finds that the pen is not on the screen it exits to the location stored at hexadecimal addresses 35 and 36.

If the pen is on the screen its location is found and the data from the original keyboard entry is entered at that location. The program then returns to INIT, restores the screen to its unscrolled format, and waits for another keyboard entry.

In addition to facilitating the arrangement of complex displays or pictures, the light pen may be incorporated into any number of games (like tic-tac-toe) or utility programs. Editing of memory dumps, for example, can easily be accomplished by moving a block of memory to the screen, modifying it with the light pen, and then moving it back.

```
Address  Hex Code   Line Statement            Commentary

0000     31 F6 00   0001 INIT LXI SP,0F6H
0003     21 00 CC   0002  LXI H,0CC00H        H,L to first screen position
0006     01 40 A0   0003  LXI B,0A040H        load B with white cursor,
                                              load C with length of line to be stored
0009     11 AD 00   0004  LXI D,0ADH          first 'store to' address
000C     AF         0005  XRA A               zero accumulator
000D     D3 C8      0006  OUT 0C8H            init screen to unscrolled format
000F     DB 00      0007 STATIN IN 00         }
0011     E6 40      0008  ANI 040H            } wait for key pressed
0013     CA 0F 00   0009  JZ STATIN           }
0016     CD 37 00   0010  CALL MOVE           store and write white
0019     3E F0      0011  MVI A,0F0H          value for scrolling and blanking
001B     4F         0012  MOV C,A
001C     D3 C8      0013 OUT OUT 0C8H
001E     C3 42 00   0014  JMP CHECK
0021     D6 10      0015 DEC SUI 010H         scroll up line by line
0023     B9         0016  CMP C
0024     C2 1C 00   0017  JNZ OUT
0027     DB 01      0018  IN 01H              pen not on screen, reset status port
0029     11 00 CC   0019  LXI D, 0CC00H       }
002C     21 AD 00   0020  LXI H,0ADH          } getting ready to restore line one
002F     0E ED      0021  MVI C,0EDH          }
0031     CD 37 00   0022  CALL MOVE           to restore first line
0034     C3 YY XX   0023  JMP XX YY           exit from LIGHT if line not found
0037     7E         0024 MOVE MOV A,M         store and write white routine
0038     70         0025  MOV M,B
0039     12         0026  STAX D
003A     23         0027  INX H
003B     13         0028  INX D
003C     79         0029  MOV A,C
003D     BD         0030  CMP L
003E     C2 37 00   0031  JNZ MOVE
0041     C9         0032  RET
0042     C5         0033 CHECK PUSH B
0043     01 FF 04   0034  LXI B,04FFH         delay value
0046     F5         0035  PUSH PSW
0047     DB C8      0036 INPUT IN 0C8H        test output from light-pen
0049     07         0037  RLC
004A     D2 58 00   0038  JNC BINGO           found line
004D     AF         0039  XRA A               }
004E     0B         0040  DCX B               } keep checking for 1/60 sec.
004F     B8         0041  CMP B               }
0050     C2 47 00   0042  JNZ INPUT           }
0053     F1         0043  POP PSW
0054     C1         0044  POP B
0055     C3 21 0`   0045  JMP DEC             try next line
0058     21 00 C    0046 BINGO LXI H,0CC00H   decoding routine to find actual
005B     F1         0047  POP PSW             line starting address
005C     E6 C0      0048  ANI 0C0H
005E     07         0049  RLC
005F     07         0050  RLC
0060     84         0051  ADD H
0061     67         0052  MOV H,A
0062     3B         0053  DCX SP
0063     3B         0054  DCX SP
0064     F1         0055  POP PSW
0065     E6 30      0056  ANI 030H
0067     07         0057  RLC
0068     07         0058  RLC
0069     85         0059  ADD L
006A     6F         0060  MOV L,A
006B     22 ED 00   0061  SHLD 0EDH           store beginning of line address
006E     21 00 CC   0062  LXI H,0CC00H
0071     C1         0063  POP B               to balance stack
0072     3E 20      0064 CHEK MVI A,020H      black square
0074     77         0065  MOV M,A             to screen
0075     06 A0      0066  MVI B,0A0H          delay value
0077     AF         0067  XRA A
0078     05         0068 DCL DCR B
0079     B8         0069  CMP B
007A     C2 78 00   0070  JNZ DCL
007D     01 FF 04   0071  LXI B,04FFH         'check delay' value
0080     DB C8      0072 DECL IN 0C8H         check for output from light pen
0082     07         0073  RLC
0083     D2 A9 00   0074  JNC HERE            not this square, try next
0086     AF         0075  XRA A
0087     0B         0076  DCX B
0088     B8         0077  CMP B
0089     C2 80 00   0078  JNZ DECL
008C     7D         0079 DIDO MOV A,L         decode to find actual value of L
008D     2A ED 00   0080  LHLD 0EDH
0090     85         0081  ADD L
0091     6F         0082  MOV L,A
0092     22 ED 00   0083  SHLD 0EDH
0095     11 00 CC   0084  LXI D.0CC00H
0098     21 AD 00   0085  LXI H,0ADH
009B     0E ED      0086  MVI C,0EDH
009D     CD 37 00   0087  CALL MOVE           restore information to first line
00A0     2A ED 00   0088  LHLD 0EDH
00A3     DB 01      0089  IN 01H              input data
00A5     77         0090  MOV M,A             move data to screen
00A6     C3 00 00   0091  JMP INIT
00A9     23         0092 HERE INX H
00AA     C3 72 00   0093  JMP CHEK

                    END
```

Listing 1: An 8080 program, hand assembled, to implement the flow chart of figure 2. Address constants and IO port assignments are given for the authors' system. ■

Rated G

The Best Game in Town.

Welcome to ComputerLand. An incredible adventure into the world of personal computers. A one-of-a-kind shopping experience.

Each ComputerLand store presents everything you ever wanted to know about computers. And then some.

Take our Game Room, for starters. You'll find excitement for the whole family in our endless variety of challenging computer games. You can battle the Klingons in an out-of-this-world game of Star Trek. Create an electronic work of art with a computer controlled TV. Test your skill in a game of computerized hangman.

You can even plot your biorhythm.

But we're more than just fun and games. Each ComputerLand store offers a knowledgeable and personable staff of professionals to serve you.

Plus the greatest available selection of micro components. Whether it's a data processing system for your business or a computer controlled sprinkler system for your home, you'll find whatever you need at ComputerLand.

Read on.

Genuine Service.

We want to supply you with the one system that's right. Rather than a complete system that isn't. Or a limited system that is.

That's why, at ComputerLand, you deal with real professionals who are also real people. People who speak your language . . . in addition to BASIC, COBOL or FORTRAN.

People, in short, who can offer both the novice and the old hand the same expert guidance in selecting the optimum system he or she needs.

Yet, assisting in the purchase is only the beginning of ComputerLand's service. If the kit you bought requires a little more do-it-yourself than you yourself can do, we provide assembly assistance.

If that complex program proves to be just that, we provide programming assistance.

And if your system breaks down, our in-store service department will get you back up and running.

Right now!

Great Selection.

Your first stop at ComputerLand may well be your last stop.

ComputerLand offers the finest quality and largest selection of all the major brand names. Like Apple Computer, Cromemco, DEC, Diablo, Hazeltine, ICOM, IMSAI, Lear Siegler, National Semiconductor, North Star, Texas Instruments, Vector Graphics and more.

Plus a complete inventory of tools, books and accessories.

What's more, at ComputerLand, we deal in product. Not promises. Our inventory is on our own shelves. Rather than the manufacturer's. So you can take delivery on tomorrow's components today.

Which means, simply put, that at ComputerLand, you get exactly what you want.

Exactly when you want it.

Be Our Guest.

Begin with the grand tour of our exhibit areas. "Test-drive" any of our individual systems.

Then tell us your needs. We'll sit down and talk about the system that's right for you. It's as easy as that at ComputerLand.

The great computer store. RATED G.

Call or write for the address of the ComputerLand store nearest you. Franchise opportunities available.

ComputerLand™

1922 Republic Avenue, San Leandro, CA 94577 (415) 895-9363

Circle 24 on inquiry card.

Letter from the Publisher

We would like to thank our readers for the essential part they have played in the growth of BYTE. From its inception in September of 1975, BYTE has experienced colossal growth to a circulation of over 110,000.

And we are still growing. We have been doing everything we could to provide good service to the subscriber and have been suceeding, we believe, quite well. With such rapid growth and expansion, however, there is always the chance that not all things will move as efficiently as they should, and to some degree this has been the case with service to our subscribers.

BYTE first handled its own subscription fulfillment, and it became apparent rather quickly that, with all the other procedures and duties performed to bring the reader a fine monthly publication, to continue handling subscription service in-house exceeded the bounds of practicality. As soon as was possible we employed a fulfillment service bureau to handle the burgeoning roster of subscribers. In retrospect, it seems that no sooner did the new service take over our file than we outgrew them. As the size of the list and volume of orders became too great for us to handle in-house, so did the same happen with our newly adopted service bureau.

Solution?

Yes. BYTE went looking for a fulfillment agency big enough to suit our present needs with the capacity to service efficiently the many more subscribers we project we will have in the future, . .and we found one.

If you are one of the people who have found our service a little less than satisfactory, we offer with this explanation our profound apologies, and feel that we can promise even better service to our subscribers in the future.

Again, we offer our gratitude to our readers for helping to make BYTE what it is.

Va. Peschke Londner

Virginia Peschke Londner
Publisher

SWEETS for KIM

A Low Calorie Text Editor

Dan Fylstra
22 Weitz St #3
Boston MA 02134

```
ENERG   LDY   #10     SET UP FOR 10 MSEC DELAY
        JSR   WAIT    LOOP FOR THAT LONG
        LDY   #0      SEND 0'S TO OUTPUT PORT
        STY   PORT    TO TURN OFF MAGNET CURRENT
        RTS           RETURN TO CALLER
WAIT    LDX   #200    NO. TIMES THRU INNER LOOP
LOOP    DEX           DECREMENT INNER LOOP COUNT
        BNE   LOOP    LOOP UNTIL COUNT IS 0
        DEY           DECREMENT OUTER LOOP COUNT
        BNE   WAIT    LOOP UNTIL COUNT IS 0
        RTS           RETURN TO CALLER
```

Listing 1a: A segment of 6502 assembly language code used to demonstrate SWEETS, a Simple Way to Enter, Edit and Test Software. SWEETS is a small text editor and assembler which operates on hexadecimal code and which is designed to fit in the KIM-1's 1 K byte small memory while leaving room for the user's programs. The key sequence for editing is shown in table 1b.

Keys								LED					
	AD	F	F	0	1	0	0	F	F	0	1	0	0
+	AD	A	0	0	A			A	0	0	A		
+	AD	2	0	0	2	0	0	2	0	0	2	0	0
+	AD	A	0	0	0			A	0	0	0		
+	AD	8	C	0	0	1	7	8	C	0	0	1	7
+	AD	6	0					6	0				
+	AD	F	F	0	2	0	0	F	F	0	2	0	0
+	AD	A	2	C	8			A	2	C	8		
+	AD	F	F	0	3	0	0	F	F	0	3	0	0
+	AD	C	A					C	A				
+	AD	D	0	0	3			D	0	0	3		
+	AD	8	8					8	8				
+	AD	D	0	0	2			D	0	0	2		
+	AD	6	0					6	0				

Table 1a: The sequence of keys used to enter the program in listing 1a when using the SWEETS editor and assembler. The right side of the table shows the resulting LED readout seen at each step. Notice that an entire instruction is entered and displayed at one time.

If you would like to experiment with microcomputers on a limited budget, the MOS Technology KIM-1 is an excellent choice. For $245, it comes preassembled with, among other things, a 6502 microprocessor, a read only memory monitor, an audio cassette interface, 1 K bytes of programmable memory, and its own special peripheral: a 23 key keyboard plus a 6 digit LED display. The monitor lets you load a machine language program byte by byte from the keyboard, and once loaded the program can be saved on tape via the audio cassette interface. The KIM-1 manual shows how you can "hand-translate" an assembly language program into the absolute hexadecimal form required for keyboard entry.

This is fine for very small programs, but the process of hand translation gets rather tedious after you've assembled a few hundred bytes of code. And, worse, once you've painstakingly worked out all the subroutine call addresses and branch displacements and keyed the whole program in, you invariably find that you've forgotten something. Often, instructions must be inserted or deleted in the middle of the program, which throws everything off by a few bytes.

The obvious solution to this problem is to obtain a text editor and assembler program for the 6502. But, alas, such a program probably needs more than the 1 K bytes of memory provided on the KIM-1, and, more seriously, it requires an alphabetic character terminal device such as a Teletype. What if you can't afford the extra peripherals and memory? Are you doomed to spend most of your microcomputing hours keying in the same program over and over again?

Maybe not. Perhaps we can avoid most

of the tedium by concentrating on those features of a text editor and assembler which we really need. Although we'll be limited by the KIM-1 keyboard to hexadecimal instruction entry, perhaps we can provide an automatic way to insert and delete instructions and to fix up all those subroutine call addresses and branch displacements. And perhaps by limiting ourselves to these features, we'll be able to cram the "editor and assembler" into some fraction of the KIM's 1 K of memory.

This is the purpose of SWEETS. SWEETS is an example of a program invented to fit an acronym: It stands for Simple Way to Enter, Edit and Test Software. If you own a KIM-1 and have grown tired of absolute machine language programming, now you can step up to "symbolic hex"! While it's not as convenient as a real text editor and assembler, SWEETS can save you a lot of time and index finger soreness.

SWEETS Functions

Under the control of the KIM-1 monitor, the 6 digit LED display normally shows you the address and data of a single byte of memory. You can enter data using the hexadecimal keys, but this causes the data previously in the displayed byte of memory to be destroyed.

Under the control of SWEETS, however, an entire instruction of one, two or three bytes in length is displayed on the LEDs at any given time. An instruction can be inserted just before the displayed instruction by pressing the AD key followed by from 2 to 6 hexadecimal keys. When this is done, the instruction just entered appears on the display; the old instruction and everything following it in the program area have been moved down to make room. Similarly, pressing the DA key causes the currently displayed instruction to be deleted, and everything following this instruction in the program area is moved up to eliminate the slack space.

Successive instructions can be examined by pressing the + key, which advances to and displays the next complete instruction. And to go back to a previous point, or to find an arbitrary point in the instruction sequence, you can press the GO key followed by a two byte (four hexadecimal digit) search pattern. SWEETS will search for the first instruction(s) whose initial two bytes match the search pattern, and then will display this as the current instruction.

This much of SWEETS can be used by itself; but so far we're still burdened by the need to calculate and adjust subroutine call addresses and branch displacements. To lift this burden, we can use hexadecimal "labels." A label is a 3 byte "pseudo-instruction" with an opcode of hexadecimal FF. The second byte is the "label number," any hexadecimal value, and the third byte is ignored. A label is inserted in the hexadecimal instruction sequence at each point where an alphabetic label appears in a normal assembly listing. When we key in a subroutine call, jump, or relative branch instruction, we enter the destination label number as the second byte of the instruction, in place of a branch displacement or absolute address. As we insert and delete instructions, the "label" pseudo-instructions move up and down in memory along with the rest of the code.

When we're ready for a test run of the edited program, we can use the KIM-1 monitor to execute the SWEETS "assembler." This program removes the label

Keys					Display					
GO	A	0	0	0	A	0	0	0		
DA					8	C	0	0	1	7

Table 1b: The procedure used in SWEETS to locate and delete an instruction, in this case the superfluous instruction LDY #0 (A000 in hexadecimal code). The rest of the program is moved up in memory and the next instruction is then displayed, as shown.

```
0200   A0   0A          ENERG   LDY   #10
0202   20   09    02            JSR   WAIT
0205   8C   00    17            STY   PORT
0208   60                       RTS
0209   A2   C8          WAIT    LDX   #200
020B   CA               LOOP    DEX
020C   D0   FD                  BNE   LOOP
020E   88                       DEY
020F   D0   F8                  BNE   WAIT
0211   60                       RTS
```

Listing 1b: The absolute hexadecimal form of the program segment shown in listing 1a after removal of the LDY #0 instruction (see table 1b) and execution of the SWEETS assembler (shown for purposes of comparison in the format of an ordinary assembler output listing).

The difference is value. Take a look at our low-cost microcomputer's heavy duty cabinet. Even with the cover removed, it maintains its rigid structural integrity. But we don't stop there. Quality built Vector Graphic products, from boards and kits to complete systems — such as our Memorite turn-key text editor shown below — beat *all* competitors in mechanical, thermal and electrical design.

Based on an 8080A microprocessor and the S-100 bus structure, Vector Graphic microcomputers are compatible (with the exception of minor I/O patches) with all of the current 8080A software.

If you're designing small business systems, text editors, control equipment, games or even microprocessor development, it will pay you to look into our low cost microcomputers and interface boards, kits or assembled.

Rush me details today.

Yes, I'm interested in:
☐ Systems ☐ Boards, kits or assembled ☐ Microcomputers

Name________________

Firm________________

Phone ________________

Address ________________

City/State/Zip________________

Vector Graphic Inc.,
790 Hampshire Rd., Westlake Village, CA 91361 (805) 497-6853

VECTOR GRAPHIC INC.

Sold and serviced nationally.

Vector Graphic's microcomputer:

What's in it for you?

Word processing systems for $7,950 — features normally found in systems costing twice as much.

Circle 123 on inquiry card.

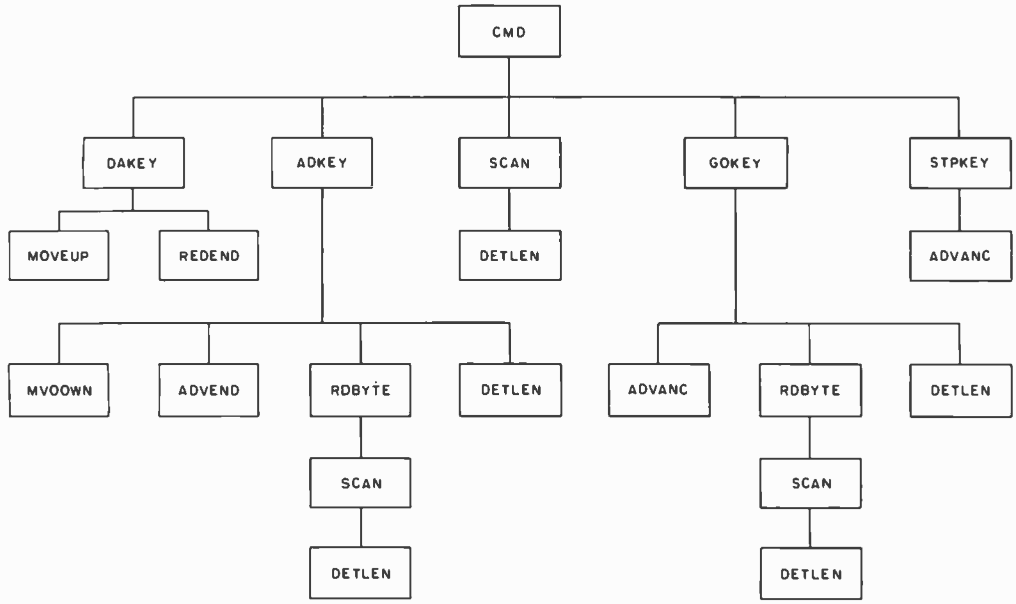


Figure 1: The subroutine calling tree structure of SWEETS. CMD, the control routine, maintains the LED display and scans the keyboard for a command key (by means of SCAN) and transfers to one of the four command processing subroutines, ADKEY, DAKEY, GOKEY or STPKEY. These routines perform the editing functions with the aid of three other subroutines: DETLEN (which determines instruction lengths), MVDOWN, and MOVEUP (which move portions of edited program down and up in memory, respectively).

pseudo-instructions from the instruction sequence, and replaces label references in branch, jump and subroutine call instructions with the proper branch displacements or absolute addresses. Then the edited program is ready for a test execution. (Since the test is likely to fail, leading to further changes in the edited program, we should always dump the program on the audio cassette in "symbolic hexadecimal" form before executing the SWEETS assembler. Then we can reload it later, replacing the program in memory which has been converted to absolute machine language.)

As an example, suppose that you wished to enter the program segment shown in listing 1a, which is taken from an earlier BYTE article of mine (see "Selectric Keyboard Printer Interface," June 1977 BYTE, page 46). Table 1a shows the keys you would press and the resulting instructions displayed on the LEDs by SWEETS. You might then notice that the instruction LDY #0 is superfluous after the call to subroutine WAIT, so you would search for and delete this instruction as shown in table 1b. Finally you would execute the SWEETS assembler, leaving the contents of the program area as shown in listing 1b.

Of course, we will pay some penalty for use of these features of SWEETS, since we will have less memory available for the program to be debugged while SWEETS itself is loaded and running. But larger programs usually can be divided into segments, and loaded, "assembled," and debugged that way. Also, since the SWEETS hexadecimal editor and assembler run separately, we can conserve memory space by loading the assembler from tape whenever we want to use it, overlaying the editor in memory and reloading it from tape in a similar way when we need it again.

Although SWEETS is a useful tool in its present form, you will undoubtedly want to customize it for your own purposes. But to customize SWEETS you've got to understand exactly how it works, so let's take a look at the overall design of SWEETS before puzzling over its realization in 6502 assembly language.

The SWEETS Editor

The subroutine calling tree in figure 1 gives you a quick, "top-down" overall look at the SWEETS editor. CMD, the control routine, maintains the LED display and scans the keyboard for a command key (using SCAN) and then transfers to one of the command processing routines: ADKEY, DAKEY, GOKEY and STPKEY. These routines perform the editing functions with the aid of three critical subroutines: DETLEN, which determines the length of an instruction in bytes based on its opcode; MVDOWN, which moves a portion of the edited program down in memory to make room for an inserted instruction;

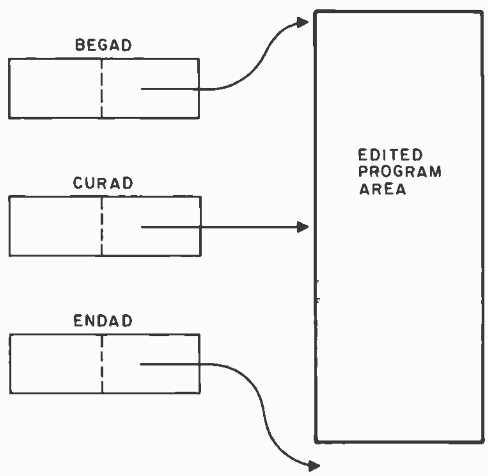


Figure 2: Three 16 bit pointers are used to manage the edited program area. BEGAD points to the beginning of the program area; ENDAD points to the location immediately beyond the end of the program area, and CURAD points to the currently displayed instruction.

```
                        ; SET CURAD = BEGAD
1780   A5   E0   BEGIN     LDA   BEGAD       LOW-ORDER BYTE
1782   85   E4             STA   CURAD
1784   A5   E1             LDA   BEGAD+1     HIGH-ORDER BYTE
1786   85   E5             STA   CURAD+1
1788   60                  RTS               RETURN TO CALLER

                        ; CURAD = CURAD + BYTES, COMPARE TO ENDAD
1789   18        ADVANC    CLC               CLEAR CARRY
178A   A5   E4             LDA   CURAD       LOW-ORDER BYTE
178C   65   E8             ADC   BYTES
178E   85   E4             STA   CURAD
1790   A5   E5             LDA   CURAD+1     HIGH-ORDER BYTE
1792   69   00             ADC   #0
1794   85   E5             STA   CURAD+1
1796   C5   E3             CMP   ENDAD+1     COMPARE HI-ORDER
1798   30   04             BMI   ADRET
179A   A5   E4             LDA   CURAD
179C   C5   E2             CMP   ENDAD       COMPARE LO-ORDER
179E   60        ADRET     RTS               RETURN TO CALLER

                        ; ENDAD = ENDAD + BYTES
179F   18        ADVEND    CLC               CLEAR CARRY
17A0   A5   E2             LDA   ENDAD       LOW-ORDER BYTE
17A2   65   E8             ADC   BYTES
17A4   85   E2             STA   ENDAD
17A6   90   02             BCC   ADRET1      CHECK CARRY
17A8   E6   E3             INC   ENDAD+1     INCREMENT HI-ORDER
17AA   60        ADRET1    RTS               RETURN TO CALLER

                        ; ENDAD = ENDAD - BYTES
17AB   38        REDEND    SEC               SET CARRY
17AC   A5   E2             LDA   ENDAD       LOW-ORDER BYTE
17AE   E5   E8             SBC   BYTES
17B0   85   E2             STA   ENDAD
17B2   B0   02             BCS   REDRET      CHECK CARRY
17B4   C6   E3             DEC   ENDAD+1     DECREMENT HI-ORDER
17B6   60        REDRET    RTS               RETURN TO CALLER
```

Listing 2: Four utility subroutines used by SWEETS to manipulate three 16 bit pointers which point to the beginning of the program area, the location just beyond the end of the program area, and the currently displayed instruction.

and MOVEUP, which moves a portion of the program up in memory to eliminate the empty space created when an instruction is deleted.

The edited program area is managed with the aid of three 16 bit pointers: BEGAD, which points to the beginning of the program area; ENDAD, which points just beyond the end of the program area; and CURAD, which points to the currently displayed instruction. This layout is shown in figure 2. Whenever a new instruction becomes the "current" one, subroutine DETLEN is called to determine its length in bytes, and this value is saved in the variable BYTES.

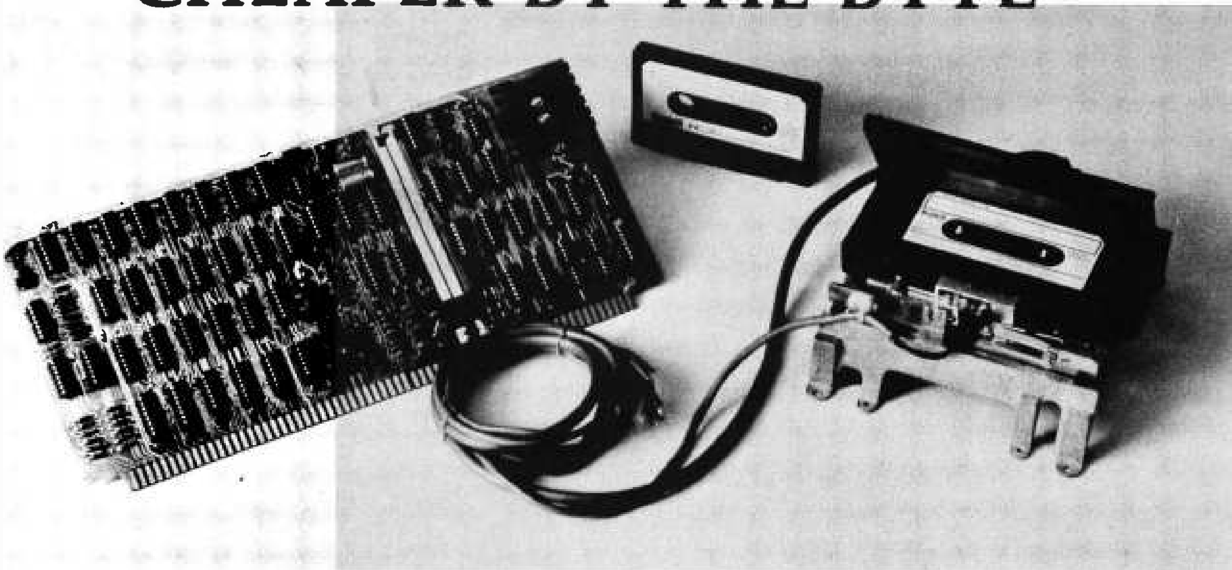
The most basic functions we need in SWEETS are some utility routines to manipulate these 16 bit pointers on an 8 bit machine such as the 6502. The routines we need are shown in listing 2. The most important one is ADVANC, which advances the current instruction pointer CURAD to the next instruction, and tests to see if the end of the program area has been reached. As we shall see later, STPKEY, the command processing routine for the + key, is basically just a call to ADVANC.

Another basic function is the subroutine DETLEN, which we've already mentioned. It is shown in listing 3. The logic of this routine clearly depends on the system of encoding opcodes on the 6502: in most cases (DETLEN tests for the exceptions), the low order hexadecimal digit of the opcode tells us the instruction length. For example, all opcodes of the form x5 represent two byte instructions, while all opcodes of the form xC represent three byte instructions.

The heart of the SWEETS editor lies in the subroutines MOVEUP and MVDOWN, which are shown in listings 4a and 4b. The main concern in these routines is that we must be careful not to move a byte up or down to a location which contains another byte that will be moved later. For MOVEUP, we must move bytes starting at CURAD and proceeding down to ENDAD, while for MVDOWN, we must move bytes in the opposite direction, as shown in figure 3.

So far we haven't faced the issue of how to control our one and only peripheral, the KIM-1 keyboard and LED display.

CHEAPER BY THE BYTE



The Dual Digital Cassette Storage System
Only $585. Only from Peripheral Vision

For the first time ever there's an inexpensive, reliable system for mass data storage. Peripheral Vision introduces the Phi-Deck Cassette Storage System for the S-100 Bus. For only $585.

That's right. $585. That gets you two Phi-Decks, controller card kit and power supply. Which means you get more bytes per buck. Take a look.

The features:

- Controller card supporting mutiple drives
- Computer-controlled operating system — an 8080 version of PhiMon
- Large data capacity:
  254,000 8-bit bytes on C-30 cassettes
  508,000 8-bit bytes on C-60 cassettes
- Software error rate less than 1 bit in 10^8

What our system allows you:

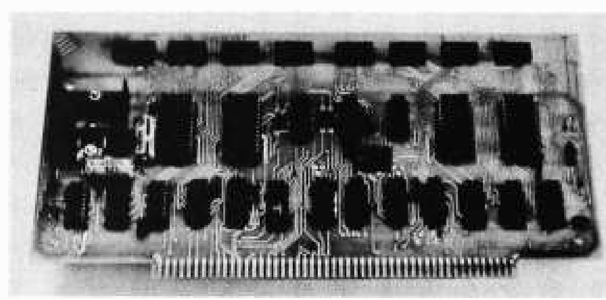
- Indexed software controlled program file
- Indexed random retrieval
- Multipass compilers
- Duplication and backup of important data
- Large data files — lists, names, business accounts
- System residence

What our software operating system offers:

- 8080-based design
- Cyclic redundancy check (CRC) error detection
- System retries after soft errors
- Automatically bypasses hard errors
- Block size from 1 to 256 bytes
- Phi Deck Monitor Operating System

How to get connected:

With Peripheral Vision's new four-port I/O board. It's another one of our bargains. Only $85. To save you bucks.



We've only given you a glimpse of our exciting new cassette storage system and the four-port I/O board. For more details, just fill in the coupon below. For cheap little bytes.

PERIPHERAL VISION INC.

P.O. Box 6267/Denver, Colorado 80206 303/777-4292

O.K., I'll byte! Send me details on your cheaper way to store data and get it connected!

Name ______________________

Address ______________________

City/State/Zip ______________________

Circle 94 on inquiry card.

Listing 3: DETLEN, a subroutine which determines instruction length based on op code.

```
0080  A0  00        DETLEN  LDY   #0
0082  B1  E4                LDA   (CURAD),Y   PICK UP OPCODE
0084  A0  01        DETLN1  LDY   #1          ASSUME LENGTH IS 1
0086  C9  00                CMP   #0          TEST FOR 'BRK'
0088  F0  19                BEQ   DETERM
008A  C9  40                CMP   #$40        TEST FOR 'RTI'
008C  F0  15                BEQ   DETERM
008E  C9  60                CMP   #$60        TEST FOR 'RTS'
0090  F0  11                BEQ   DETERM
0092  A0  03                LDY   #3          ASSUME LENGTH IS 3
0094  C9  20                CMP   #$20        TEST FOR 'JSR'
0096  F0  0B                BEQ   DETERM
0098  29  1F                AND   #$1F        STRIP TO 5 BITS
009A  C9  19                CMP   #$19        TEST FOR ABS, Y
009C  F0  05                BEQ   DETERM
009E  29  0F                AND   #$0F        STRIP TO 4 BITS
00A0  AA                    TAX               TO TABLE INDEX
00A1  B4  A6                LDY   LENTB,X     LENGTH FROM TABLE
00A3  84  E8        DETERM  STY   BYTES       SAVE IN 'BYTES'
00A5  60                    RTS               RETURN TO CALLER
00A6  02  02  02    LENTB   .BYTE 2,2,2,1,2,2,2,1
00A9  01  02  02
00AC  02  01
00AE  01  02  01            .BYTE 1,2,1,1,3,3,3,3
00B1  01  03  03
00B4  03  03
```

```
17B7  A5  E4  MOVEUP  LDA  CURAD       START MOVE FROM
17B9  85  E6          STA  MOVAD       BEGIN OF PROGRAM
17BB  A5  E5          LDA  CURAD+1     SEGMENT (CURAD)
17BD  85  E7          STA  MOVAD+1
17BF  A4  E8  UPLOOP  LDY  BYTES       AMOUNT TO MOVE
17C1  B1  E6          LDA  (MOVAD),Y   FETCH BYTE
17C3  A0  00          LDY  #0
17C5  91  E6          STA  (MOVAD),Y   STORE BYTE
17C7  A5  E6          LDA  MOVAD       CHECK FOR
17C9  A6  E7          LDX  MOVAD+1     END OF MOVE
17CB  C5  E2          CMP  ENDAD       LOW-ORDER BYTE
17CD  D0  04          BNE  INCMOV
17CF  E4  E3          CPX  ENDAD+1     HIGH-ORDER BYTE
17D1  F0  09          BEQ  MVURET
17D3  E6  E6  INCMOV  INC  MOVAD       INCREMENT LO-ORDER
17D5  D0  E8          BNE  UPLOOP
17D7  E6  E7          INC  MOVAD+1     INCREMENT HI-ORDER
17D9  B8              CLV
17DA  50  E3          BVC  UPLOOP      BACK TO MOVE MORE
17DC  60      MVURET  RTS              RETURN TO CALLER
```

```
00B6  A5  E2  MVDOWN  LDA  ENDAD       START MOVE FROM
00B8  85  E6          STA  MOVAD       END OF PROGRAM
00BA  A5  E3          LDA  ENDAD+1     SEGMENT (ENDAD)
00BC  85  E7          STA  MOVAD+1
00BE  A0  00  MVLOOP  LDY  #0
00C0  B1  E6          LDA  (MOVAD),Y   FETCH BYTE
00C2  A4  E8          LDY  BYTES       AMOUNT TO MOVE
00C4  91  E6          STA  (MOVAD),Y   STORE BYTE
00C6  A5  E6          LDA  MOVAD       CHECK FOR
00C8  A6  E7          LDX  MOVAD+1     END OF MOVE
00CA  C5  E4          CMP  CURAD       LOW-ORDER BYTE
00CC  D0  04          BNE  DECMOV
00CE  E4  E5          CPX  CURAD+1     HIGH-ORDER BYTE
00D0  F0  0D          BEQ  MVDRET
00D2  38      DECMOV  SEC              SET CARRY
00D3  E9  01          SBC  #1          DECREMENT LO-ORDER
00D5  85  E6          STA  MOVAD
00D7  8A              TXA
00D8  E9  00          SBC  #0          DECREMENT HI-ORDER
00DA  85  E7          STA  MOVAD+1
00DC  B8              CLV
00DD  50  DF          BVC  MVLOOP      BACK TO MOVE MORE
00DF  60      MVDRET  RTS              RETURN TO CALLER
```

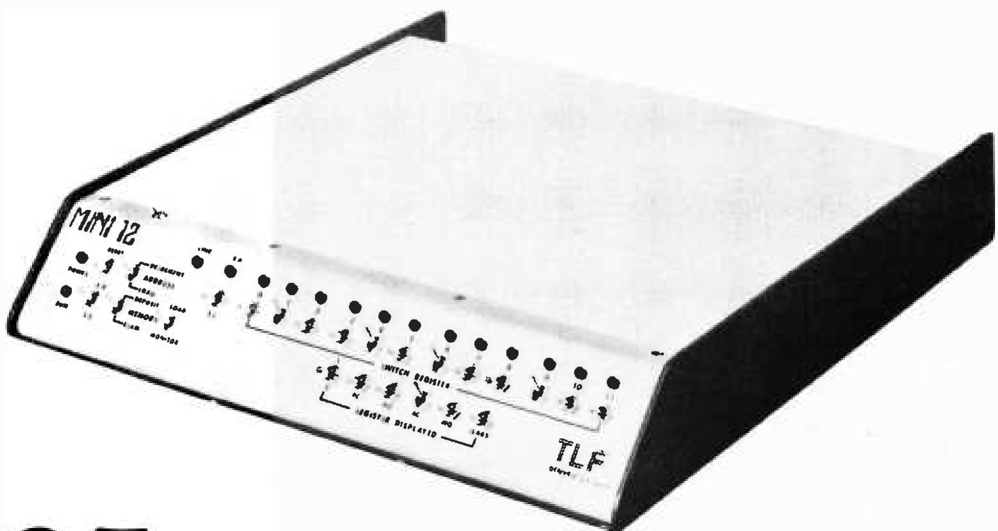
Listings 4a and 4b: Subroutines MOVEUP and MVDOWN, which form the heart of the SWEETS editor. MOVEUP moves a given program segment starting at address CURAD and ending at address ENDAD upward in memory (toward decreasing addresses) by the amount stored in BYTES. MVDOWN performs the same operation downward by the amount stored in BYTES.

Fortunately, several routines are provided for this purpose in the KIM-1 monitor; the source listings for these routines are available on request from MOS Technology. In the SWEETS assembly code listings, we have underlined references to KIM-1 monitor subroutines and variables for easy identification. We will use the KIM-1 subroutine SCAND1, which lights up the LEDs momentarily and checks to see if a key is pressed, and the subroutine GETKEY, which returns a numeric value in the accumulator telling us which particular key has been pressed.

The six LED digits display the contents of three successive bytes in memory, denoted POINTH, POINTL and INH in the KIM-1 monitor. Unfortunately, the order of these bytes is the opposite of the normal order of the bytes in an instruction in memory, so we must reverse the order as the first step of our subroutine SCAN (listing 5). The main additional complication in this routine is the need to "debounce" the keyboard's bare contact switches in software. Since SWEETS performs its operations so quickly relative to a mechanical event, the key from the last operation invariably is *still pressed* when we come back to the keyboard looking for the next command. Also shown in listing 5 is subroutine RDBYTE, which calls SCAN to read two successive hexadecimal digits from the keyboard.

With all of this machinery in place, the top level logic is straightforward. The control routine, CMD routine, and the command processing routines are shown in listings 6a, 6b and 6c. The most complicated of the processing routines is ADKEY. It determines how many bytes to read for the inserted instruction, and displays each byte as it is entered; then it copies (in reverse

THE FIRST
AFFORDABLE
MINICOMPUTER



$895

ASSEMBLED

**READY TO COMPUTE,
CALCULATE, CONTROL,
OR . . . . ? . . . . ? . . .**

STANDARD FEATURES:

Word Length	12 Bits
Instruction Set	PDP-8E Compatible
Memory Size	8K, Expandable to 32K
Extended Memory Control	DEC Compatible
Serial I/O Port	DEC Compatible, current loop
Parallel I/O Port	Compatible with DEC DR-8 - EA
Programmable Real Time Clock	Compatible with DEC DK8 - EP
Full Function Programmer's Front Panel	
Binary Load and Punch Routines in ROM	

TLF CORPORATION P.O. Box 2298, Littleton, Colorado 80161
303-922-6241 Telex 454541

Circle 119 on inquiry card.

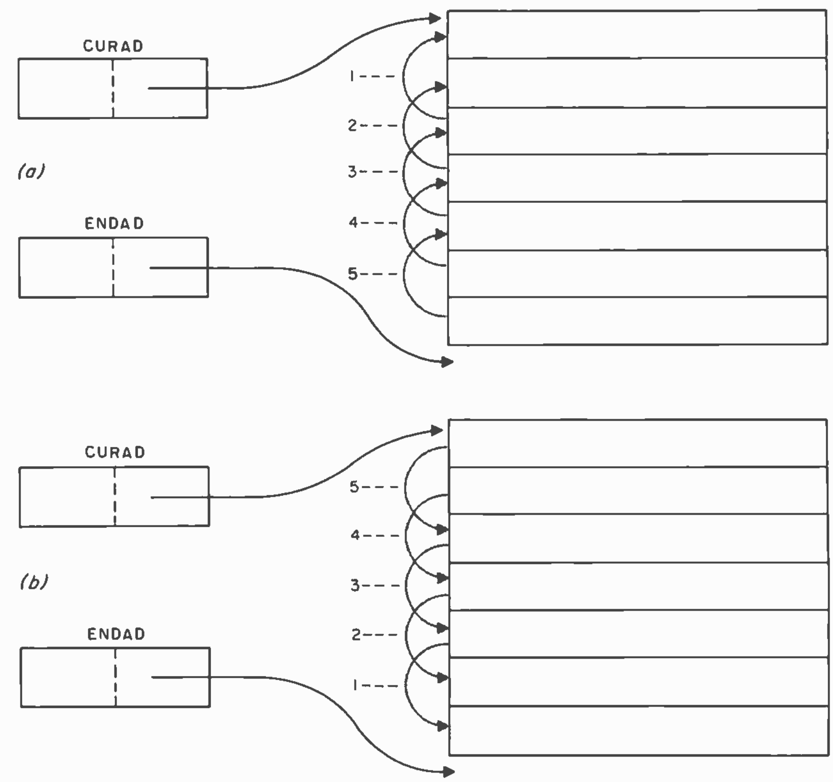


Figure 3: Correct procedures for moving programs in SWEETS. Figure 3a shows that the uppermost location must be moved first when transferring a section of program upward. Otherwise, some locations could be inadvertently destroyed. Figure 3b shows the analogous situation for a downward movement of code.

```
0100  20  80  00     SCAN     JSR   DETLEN      DETERMINE LENGTH
                     ; COPY INSTRUCTION TO DISPLAY AREA,
                     ; REVERSING ORDER OF INSTRUC. BYTES
0103  A0  00                  LDY   #0
0105  A6  E8                  LDX   BYTES
0107  B1  E4         SCOPY    LDA   (CURAD),Y   INSTRUCTION BYTE
0109  95  F8                  STA   INH-1,X     TO DISPLAY AREA
010B  C8                      INY
010C  CA                      DEX
010D  D0  F8                  BNE   SCOPY
                     ; LOGIC TO 'DEBOUNCE' KEYBOARD CONTACT
010F  20  22  01     SCAN1    JSR   SCAN3       WAIT UNTIL LAST
0112  D0  F8                  BNE   SCAN1       KEY IS RELEASED
0114  20  22  01     SCAN2    JSR   SCAN3
0117  F0  FB                  BEQ   SCAN2       WAIT FOR NEW KEY
0119  20  22  01              JSR   SCAN3
011C  F0  F6                  BEQ   SCAN2       BUT REJECT JITTER
011E  20  6A  1F              JSR   GETKEY      GET CODE FOR KEY
0121  60                      RTS               RETURN TO CALLER
                     ; SET UP PARMS AND CALL KIM-1 DISPLAY SCAN
0122  A4  E8         SCAN3    LDY   BYTES
0124  A2  09                  LDX   #9
0126  A9  7F                  LDA   #$7F
0128  8D  41  17              STA   PADD        SET UP DATA DIRECT
012B  20  28  1F              JSR   SCAND1      CALL KIM-1 ROUTINE
012E  60                      RTS               RETURN TO CALLER

                     ; RDBYTE READS TWO HEX DIGITS, RETURNS BYTE
                     ; VALUE IN ACCUMULATOR.  IF A NON-HEX DIGIT
                     ; KEY IS PRESSED, IT RETURNS THE KEY CODE
                     ; IN THE ACCUMULATOR AND N FLAG = 0

012F  20  0F  01     RDBYTE   JSR   SCAN1       GET FIRST KEY
0132  C9  10                  CMP   #$10        IS IT A HEX DIGIT?
0134  10  11                  BPL   RDRET       NO, RETURN
0136  0A                      ASL   A           SHIFT OVER 4 BITS
0137  0A                      ASL   A
0138  0A                      ASL   A
0139  0A                      ASL   A
013A  85  E9                  STA   TEMP        SAVE FIRST DIGIT
013C  20  0F  01              JSR   SCAN1       GET SECOND KEY
013F  C9  10                  CMP   #$10        IS IT A HEX DIGIT?
0141  10  04                  BPL   RDRET       NO, RETURN
0143  05  E9                  ORA   TEMP
0145  A2  FF                  LDX   #$FF        SET N FLAG = 1
0147  60             RDRET    RTS               RETURN TO CALLER
```

Listing 5: Subroutines SCAN and RDBYTE. SCAN displays the instruction at location CURAD, scans the keyboard for a depressed key, and places the code for that key in the accumulator. RDBYTE calls SCAN to read two successive hexadecimal digits from the keyboard.

order) the new instruction bytes from the display to the program area. If you've understood everything so far, you should have little trouble following the code for these top level functions. More important, once you're familiar with the basic SWEETS design, you can easily add customized top level routines of your own.

The SWEETS Assembler

None of the editor routines just discussed were concerned with the processing of the hexadecimal "labels" described earlier as one of the features of SWEETS. This is because, as far as the editor is concerned, a label is just another 3 byte instruction. Labels take on a special meaning only when the SWEETS assembler is invoked.

The assembler operates in two passes over the program area. On the first pass, the assembler searches for "instructions" with an opcode of hexadecimal FF (the labels). When one is found, the second byte of the instruction (the label number) is moved to the end of the program area, and the current instruction address is also deposited there (figure 4a). The label instruction is then deleted using MOVEUP to take up the slack space. This process continues until all of the labels have been removed and stored in the "symbol table" at the end of the program area (figure 4b). Since the labels are (by design) three bytes long, we gain the space for the symbol table when

(a)

```
0148  20  2F  01   GOKEY    JSR   RDBYTE     GET FIRST BYTE
014B  10  28                BPL   GCMD       OF SEARCH PATTERN
014D  85  FB                STA   POINTH     SAVE IN DISPLAY
014F  20  2F  01            JSR   RDBYTE     GET SECOND BYTE
0152  10  21                BPL   GCMD       OF SEARCH PATTERN
0154  85  FA                STA   POINTL     SAVE IN DISPLAY
0156  20  80  17            JSR   BEGIN      CURAD := BEGAD
                   ; LOOP SEARCHING FOR 2-BYTE MATCH
0159  A0  00       GOLOOP   LDY   #0
015B  B1  E4                LDA   (CURAD),Y  COMPARE 1ST BYTE
015D  C5  FB                CMP   POINTH     AGAINST PATTERN
015F  D0  07                BNE   GONEXT
0161  C8                    INY
0162  B1  E4                LDA   (CURAD),Y  COMPARE 2ND BYTE
0164  C5  FA                CMP   POINTL     AGAINST PATTERN
0166  F0  0A                BEQ   CMD        MATCH, NEXT CMD
0168  20  80  00   GONEXT   JSR   DETLEN     DETERMINE LENGTH
016B  20  89  17            JSR   ADVANC     ADVANCE TO NEXT
016E  F0  15                BEQ   ERROR      MATCH NOT FOUND?
0170  D0  E7                BNE   GOLOOP     CONTINUE SEARCH
```

(b)

```
0172  20  00  01   CMD      JSR   SCAN       WAIT FOR A KEY
0175  C9  10       GCMD     CMP   #$10       TEST FOR VARIOUS
0177  F0  2B                BEQ   ADKEY      COMMAND KEY CODES
0179  C9  11                CMP   #$11
017B  F0  1E                BEQ   DAKEY
017D  C9  12                CMP   #$12
017F  F0  13                BEQ   STPKEY
0181  C9  13                CMP   #$13
0183  F0  C3                BEQ   GOKEY
0185  A9  EE       ERROR    LDA   #$EE       OPERATOR ERROR:
0187  85  F9                STA   INH        SET UP HEX 'EE'
0189  85  FA                STA   POINTL     IN DISPLAY AREA
018B  85  FB                STA   POINTH
018D  20  1F  1F   ERR1     JSR   SCANDS     CALL KIM-1 ROUTINE
0190  D0  FB                BNE   ERR1       UNTIL KEY RELEASED
0192  F0  DE                BEQ   CMD

                   ; STPKEY ADVANCES TO THE NEXT INSTRUCTION

0194  20  89  17   STPKEY   JSR   ADVANC     ADVANCE TO NEXT
0197  10  EC                BPL   ERROR      CHECK FOR ADVANCING
0199  30  D7                BMI   CMD        PAST END OF PROGRAM

                   ; DAKEY DELETES THE CURRENT INSTRUCTION

019B  20  B7  17   DAKEY    JSR   MOVEUP     MOVE UP REST OF PROG
019E  20  AB  17            JSR   REDEND     ADJUST ENDAD UPWARDS
01A1  B8                    CLV
01A2  50  CE                BVC   CMD
```

(c)

```
                   ; READ OPCODE, DETERMINE INSTRUCTION LENGTH
01A4  20  2F  01   ADKEY    JSR   RDBYTE     ACCEPT OPCODE UNLESS
01A7  10  CC                BPL   GCMD       NON-HEX KEY PRESSED
01A9  85  FB                STA   POINTH     SAVE IN DISPLAY
01AB  20  84  00            JSR   DETLN1     DETERMINE LENGTH
                   ; READ REST OF INSTRUCTION INTO DISPLAY
01AE  84  EA                STY   COUNT      SAVE LENGTH
01B0  C6  EA                DEC   COUNT
01B2  F0  12                BEQ   ADSET      1-BYTE INSTRUCTION
01B4  20  2F  01            JSR   RDBYTE     READ SECOND BYTE
01B7  10  BC                BPL   GCMD       NON-HEX KEY PRESSED
01B9  85  FA                STA   POINTL
01BB  C6  EA                DEC   COUNT
01BD  F0  07                BEQ   ADSET      2-BYTE INSTRUCTION
01BF  20  2F  01            JSR   RDBYTE     READ THIRD BYTE
01C2  10  B1                BPL   GCMD       NON-HEX KEY PRESSED
01C4  85  F9                STA   INH
                   ; MOVE CODE DOWN TO MAKE ROOM
01C6  20  B6  00   ADSET    JSR   MVDOWN     MOVE CODE DOWNWARD
01C9  20  9F  17            JSR   ADVEND     ADJUST ENDAD DOWN
                   ; INSERT INSTRUCTION INTO NEW SPACE
01CC  A0  00                LDY   #0
01CE  A2  02                LDX   #2
01D0  B5  F9       INSERT   LDA   INH,X      FETCH FROM DISPLAY
01D2  91  E4                STA   (CURAD),Y  STORE INTO PROGRAM
01D4  CA                    DEX
01D5  C8                    INY
01D6  C4  E8                CPY   BYTES      UNTIL ENTIRE INSTRUCTION
01D8  D0  F6                BNE   INSERT     IS INSERTED
01DA  F0  96                BEQ   CMD
```

Listing 6: Processing routines used in the SWEETS editor. Listing 6a shows GOKEY, which searches the program for a given 2 byte pattern and makes this the current instruction. It can also search for labels. The CMD (for "command") routine, listing 6b, waits for a command key to be pressed and transfers to the processing routine for that key. If an invalid key is pressed, "EEEEEE" is displayed. ADKEY (listing 6c) accepts a new instruction, inserts it, and shifts the code following it downward to make room.

we delete the labels from the instruction sequence.

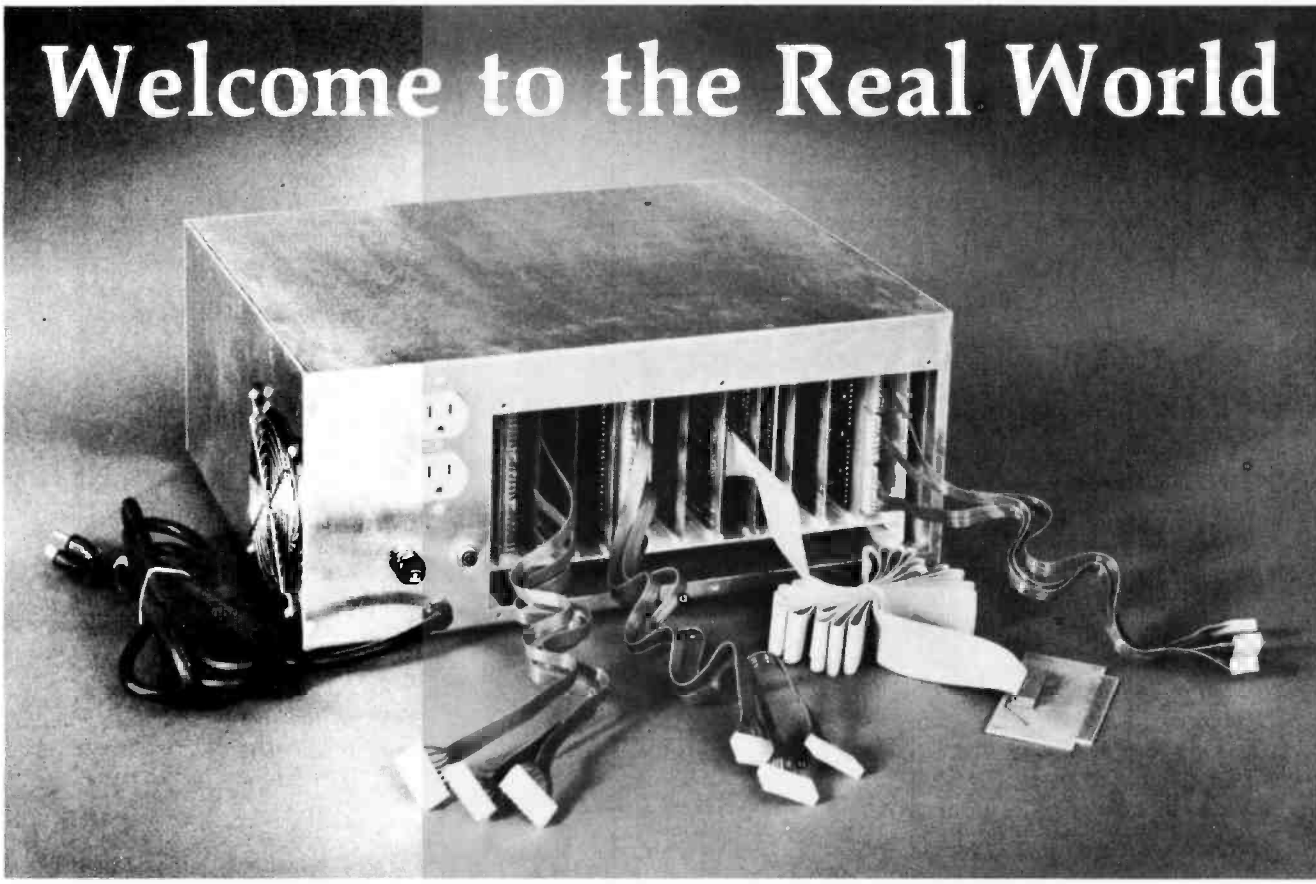
On its second pass through the program area, the assembler searches for subroutine call, jump and relative branch instructions. When one of these instructions is found, its second byte, normally a label number, is used to search for a matching label in the symbol table. Assuming that the label is found in the table, the corresponding actual address is inserted into the second and third instruction bytes for jump or subroutine call instructions, or a branch displacement is calculated and inserted for relative branch instructions (figure 4c). Since at times we may wish to enter instructions with an actual address or displacement rather than a label number, no substitution is made if the label is not found in the symbol table.

The assembly source code for the SWEETS assembler is presented in listings 7a, 7b and 7c. The subroutine FINDLB is used by pass 2 of the assembler to look up labels in the symbol table. Note, too, that the assembler uses some of the editor's subroutines: DETLEN, ADVANC, REDEND, and MOVEUP. The addresses shown in the assembly code listing are designed to allow the assembler to overlay the main part of the editor without destroying those editor subroutines which the assembler must use.

Some Operating Hints

Except for subroutine call addresses, each SWEETS routine is relocatable: it will execute properly no matter where it is loaded in memory. The assembled code shown here is designed to provide the largest possible contiguous area (512 bytes at hexadecimal addresses 200 to 3FF) for editing and assembling programs. This has the disadvantage of breaking up SWEETS into four pieces: one in page zero, two in page one, and one starting at address 1780 (which makes it a bit cumbersome to load piece by piece from audio cassette). The SWEETS routines could be consolidated, however, to provide two or more noncontiguous areas for program editing.

In general, when starting up SWEETS, or after reloading a "symbolic hexadecimal" program from tape, you must store the proper values in BEGAD, CURAD and ENDAD. Then, of course, you merely key in the CMD routine starting address and press GO. The assembler, which can be started up in the same way, automatically returns control to the KIM-1 monitor; the editor can be interrupted at any point by pressing RS (reset). Avoid using the ST



with the Real-World Interface from The Digital Group

A computer should have a purpose. Or as many purposes as you can imagine. Because a computer belongs in the real world.

And now, the Digital Group introduces the Real-World Interface. A system component that's actually a system in itself, and specifically designed to help you get your computer to control all those tasks you know a computer can control so well.

Automate your sprinkler system. Heat and cool your home. Guard against burglars. Shut off lights . . . It's all a part of the Real World, easily controlled with the Digital Group Real-World Interface.

Our Real-World Interface is initially made up of three basic components — motherboard and power supply, parallel CPU interface and cabinet — plus three types of plug-ins: AC controller, DC controller and prototyping card. The recommended software packages are Convers, Assembler or Maxi-Basic, in that order.

Some of the features include:

Motherboard & Power Supply
- 12 slots — 11 control cards, one for the interface card
- +5V DC±5% @ 1A, +12V DC ±5% @ 1A, -12V DC ±5% @ 1A contained on board
- May be free-standing (with care)

Parallel CPU Interface
- All buffering for Data Out (25 TTL loads), Address (25 TTL loads) and Data In (10 TTL loads)
- Includes cable and paddlecard for connection to dual 22 on Digital Group CPU back panel. Two 22-pin edge connectors included
- Requires two output ports and one input port

AC Controller
- Eight output devices (2N6342A-2N6343A, -12 amp Triacs); Each output 240V AC max, 12A max RMS
- Control AC motors, lamps, switches, etc.
- Opto-isolated (MCS-2400 or equivalent)

DC Controller
- Eight output devices (2N6055) each output up to 50V and up to 5A
- Control DC motors, switches, solenoids, etc.
- May use internal +12V DC for load or external DC up to 50V DC

Price
- For the motherboard and power supply, parallel CPU interface and cabinet, our kit price is only $199.50, or $260 assembled. Now *that*'s down to earth.

We've only just begun our Real-World Interface System. There are many more plug-ins and applications coming along soon. So write or call The Digital Group now for complete details.

And welcome to our world.

the digital group INC.

P.O. BOX 6528 DENVER, CO 80206 (303) 777-7133

Circle 39 on inquiry card.

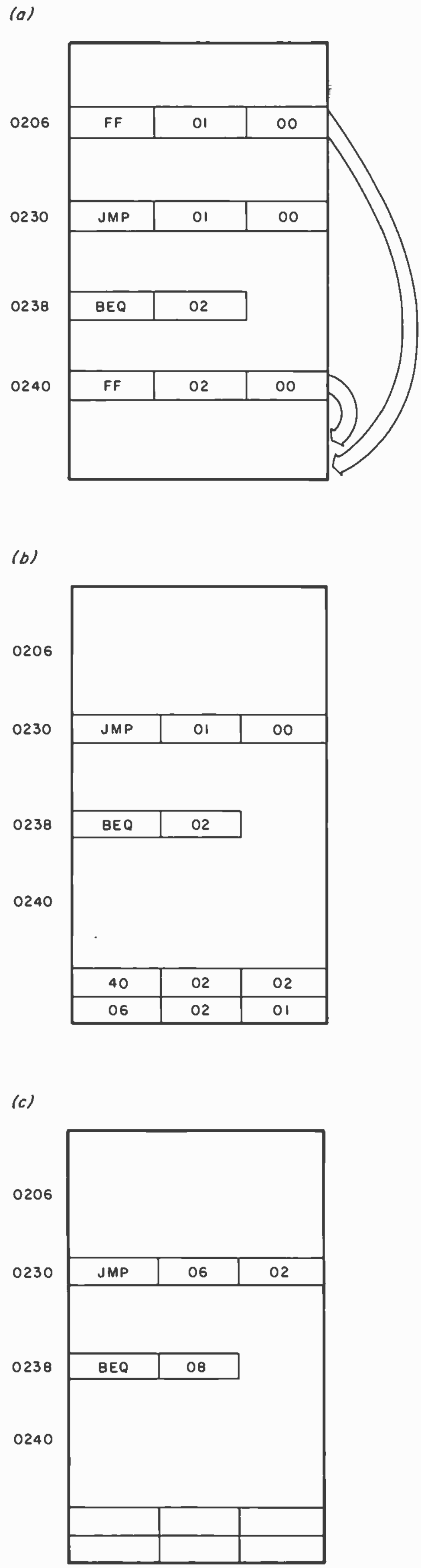


Figure 4: Mechanics of pass 1 of the SWEETS assembler are shown in figure 4a. The assembler first searches for "instructions" having an op code of hexadecimal FF (the labels). When one is found, the second byte of the instruction, which is the label number, is moved to the end of the program area and the current instruction address is also deposited there. The label instruction is then deleted using subroutine MOVEUP. Figure 4b is a continuation of the process shown in figure 4a, showing that all of the labels have been arranged in a symbol table at the end of the program area. A typical result of pass 2 of the SWEETS asasembler is shown in figure 4c. Here a jump instruction has been modified so that the actual address of the destination appears in bytes 2 and 3 of the instruction, and the actual branch displacement has been calculated and inserted for a relative branch instruction. In general, this pass takes care of all jump, subroutine call, and relative branch instructions.

BEGAD	00E0, 00E1
ENDAD	00E2, 00E3
CURAD	00E4, 00E5
CMD	0172
ASSEM	011C

Table 2: Locations of the variables BEGAD, ENDAD, CURAD, CMD and ASSEM. BEGAD, CURAD and ENDAD must be set up by the user to point to the area of memory which will hold the edited program. CMD is the entry point to the SWEETS editor, and ASSEM is the entry point to the SWEETS assembler.

(stop) key repeatedly, since this may cause the stack to grow in length to the point where it could destroy one of the SWEETS routines. The special address information you need is summarized in table 2.

Once you have SWEETS up and running, you can use it to develop improvements to SWEETS itself. In order to do this, you will have to edit code in the program area which is designed to run in another area of memory. One way to facilitate this is to add a 16 bit offset to jump and subroutine call addresses as they are resolved in pass 2 of the assembler. Another addition to SWEETS would be a small routine to save ENDAD at the end of the program area, set up the starting and ending addresses for the KIM-1 audio cassette dump routine, and then transfer control directly to this read only memory routine to carry out the tape dump operation.

One of the peculiarities of SWEETS is that it tends to make itself obsolete. This is because of our insatiable desire to do more with our personal computers. As soon as you find that writing a 512 byte program isn't so tedious anymore, you'll immediately want to write a 1024 byte program (at least), and then you'll be stretching the capabilities of SWEETS and the KIM-1. In a sense, SWEETS, as its name suggests, is an enticement: It helps develop the market for assemblers. But why not give it a try? It's a lot sweeter than absolute hex.

(a)

```
0100  B1  E4      FINDLB  LDA  (CURAD),Y   PICK UP LABEL
0102  A0  FF              LDY  #$FF        SYMBOL TABLE INDEX
0104  C4  EB      FDLOOP  CPY  LABELS
0106  F0  0D              BEQ  FDRET       NO LABELS IN TABLE
0108  D1  EC              CMP  (TABLE),Y   DOES LABEL MATCH?
010A  D0  0A              BNE  FDNEXT
010C  88                  DEY              WE HAVE A MATCH
010D  B1  EC              LDA  (TABLE),Y   GET HI-ORDER ADDR
010F  AA                  TAX              INTO X REGISTER
0110  88                  DEY
0111  B1  EC              LDA  (TABLE),Y   GET LO-ORDER ADDR
0113  A0  01              LDY  #1          INTO A REG., Y=1
0115  60          FDRET   RTS              RETURN TO CALLER
0116  88          FDNEXT  DEY
0117  88                  DEY              ADVANCE TO NEXT
0118  88                  DEY              SYMBOL TABLE ENTRY
0119  D0  E9              BNE  FDLOOP
011B  60                  RTS              UNLESS END OF TBL
```

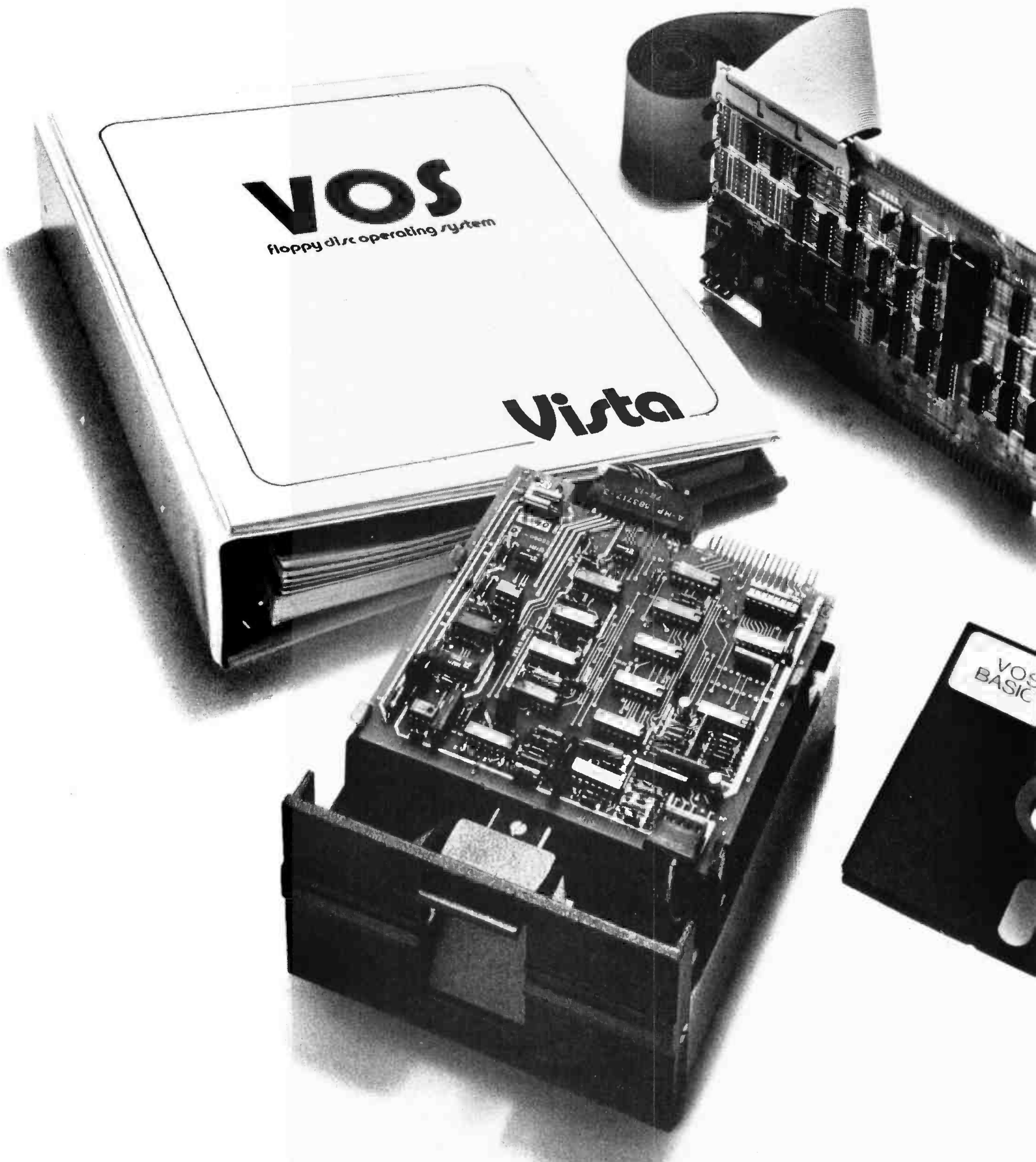
(b)

```
011C  20  80  17  ASSEM   JSR  BEGIN       CURAD := BEGAD
011F  18                  CLC
0120  A5  E2              LDA  ENDAD       ENDAD + 6 IS JUST
0122  69  06              ADC  #6          BEYOND UPPERMOST
0124  85  EC              STA  TABLE       LABEL IN TABLE
0126  A9  FF              LDA  #$FF
0128  85  EB              STA  LABELS      BEGINNING TBL INDEX
012A  65  E3              ADC  ENDAD+1     ADJUST TABLE DOWN BY
012C  85  ED              STA  TABLE+1     256 FOR INDEX BASE
012E  20  80  00  ASLOOP  JSR  DETLEN      DETERMINE LENGTH
0131  A0  00              LDY  #0
0133  B1  E4              LDA  (CURAD),Y   PICK UP OPCODE
0135  C9  FF              CMP  #$FF        IS IT A LABEL?
0137  D0  1D              BNE  ASNEXT
0139  C8                  INY
013A  B1  E4              LDA  (CURAD),Y   YES, GET LABEL NO
013C  A4  EB              LDY  LABELS      GET TABLE INDEX
013E  91  EC              STA  (TABLE),Y   DEPOSIT LABEL IN TBL
0140  88                  DEY
0141  A5  E5              LDA  CURAD+1     HI-ORDER ADDRESS
0143  91  EC              STA  (TABLE),Y   DEPOSIT IN TABLE
0145  88                  DEY
0146  A5  E4              LDA  CURAD       LO-ORDER ADDRESS
0148  91  EC              STA  (TABLE),Y   DEPOSIT IN TABLE
014A  88                  DEY
014B  84  EB              STY  LABELS      SAVE NEW TBL INDEX
014D  20  B7  17          JSR  MOVEUP      MOVE UP PROGRAM
0150  20  AB  17          JSR  REDEND      ADJUST ENDAD UPWARD
0153  B8                  CLV
0154  50  D8              BVC  ASLOOP      BACK FOR NEW LABEL
0156  20  89  17  ASNEXT  JSR  ADVANC      TO NEXT INSTRUCTION
0159  30  D3              BMI  ASLOOP      UNTIL ENDAD REACHED
```

(c)

```
015B  20  80  17          JSR  BEGIN       CURAD := BEGAD
015E  20  80  00  RSLOOP  JSR  DETLEN      DETERMINE LENGTH
0161  A0  00              LDY  #0
0163  B1  E4              LDA  (CURAD),Y   PICK UP OPCODE
0165  C9  20              CMP  #$20        JSR INSTRUCTION?
0167  F0  04              BEQ  JMPJSR
0169  C9  4C              CMP  #$4C        JMP INSTRUCTION?
016B  D0  0E              BNE  CHKBR
016D  C8          JMPJSR  INY              ADVANCE TO LABEL
016E  20  00  01          JSR  FINDLB      LOOKUP IN TABLE
0171  F0  1C              BEQ  RSNEXT      LABEL NOT FOUND
0173  91  E4              STA  (CURAD),Y   LO-ORDER ADDRESS
0175  8A                  TXA
0176  C8                  INY
0177  91  E4              STA  (CURAD),Y   HI-ORDER ADDRESS
0179  D0  14              BNE  RSNEXT      TO NEXT INSTRUC
017B  29  1F      CHKBR   AND  #$1F
017D  C9  10              CMP  #$10        BRANCH INSTRUC?
017F  D0  0E              BNE  RSNEXT
0181  C8                  INY              ADVANCE TO LABEL
0182  20  00  01          JSR  FINDLB      LOOKUP IN TABLE
0185  F0  08              BEQ  RSNEXT      LABEL NOT FOUND
0187  38                  SEC
0188  E5  E4              SBC  CURAD       DEST. – SOURCE
018A  38                  SEC
018B  E9  02              SBC  #2          DEST. – SOURCE – 2
018D  91  E4              STA  (CURAD),Y   = DISPLACEMENT
018F  20  89  17  RSNEXT  JSR  ADVANC      TO NEXT INSTRUC
0192  30  CA              BMI  RSLOOP      BACK TO EXAMINE IT
0194  4C  4F  1C          JMP  START       TO KIM-1 MONITOR ■
```

Listing 7: The assembly source code for SWEETS. Subroutine FINDLB (listing 7a) is used during pass 2 of the assembler to look up labels in the symbol table. FINDLB looks up the label at CURAD, Y and returns with Y=1, X=the high order part of the address, A = the lower part of the address, and Z=0. Z is set equal to 1 if the label is not found. Listing 7b shows pass 1 of the assembler during which labels are collected and stored with their addresses at the end of the program. Listing 7c is pass 2. During this pass, the operands of the branch, jump and JSR instructions are converted from label references to displacements or actual addresses. Note that jump indirect operands are not converted.


VOS
floppy disc operating system
Vista

Personal Computers in a Distributed

Jeff Steinwedel W3FY
715 Reseda Dr, Apt 2
Sunnyvale CA 94087

The combination of many small processors with some way to communicate from one to another clearly has much potential.

Since the first microprocessors became available I have been convinced that their most dramatic applications would be in connection with a large-scale communications network. The economics of a distributed network would suit the individualized structure of personal computing. The combination of many small processors with some way to communicate from one to another clearly has much potential. Already, the processor technology has arrived; but it seems that a simple, inexpensive communications system is not forthcoming. Both the telephone system and cable TV could be technically workable, but require centralized expenditures of large amounts of capital, as well as a political commitment to the application. My argument is that there is an economic and technological short cut to a *distributed* network through use of the radio spectrum for communications.

This article is speculative, in that new radio spectrum rules would have to come into effect for this network to exist. However, I think the idea is technically feasible, and the political aspects perhaps provide a raison d'etre for a national personal computing organization. Further, the FCC has already validated some of the principles involved.

If participating individuals were to construct computer controlled VHF transceivers around a common set of guidelines, and if these radio stations were designed to transmit and receive data over a number of predefined channels for extended periods without operator intervention, such a communications network could be achieved. In many ways this type of system would parallel the 2 meter FM amateur radio repeater system, except that data and control would be computer oriented.

Standardization would necessarily be defined in a number of areas: frequency selection, routing algorithms, communication mode encoding, data and communication types, character codes and data rates, etc. One of the very desirable features (for the FCC) would be that the system could easily be made to be self-logging and self-monitoring. Ideally, the system could also be self-policing so that any "Citizens' Computer Radio Service" could be a model for efficient spectrum usage with minimum government interaction.

Why Build a Network?

What would be the characteristics and advantages of such a system? The actual mechanics of radio transmission should be transparent to the user. The most common type of communications would be station-to-station relayed data transfers. For example, if I were to initiate a data transfer (message) from my station, I would just create the message, define the destination, and let the operating system take over. My computer would then find a similar station suitable for relaying the message,

Communications Network

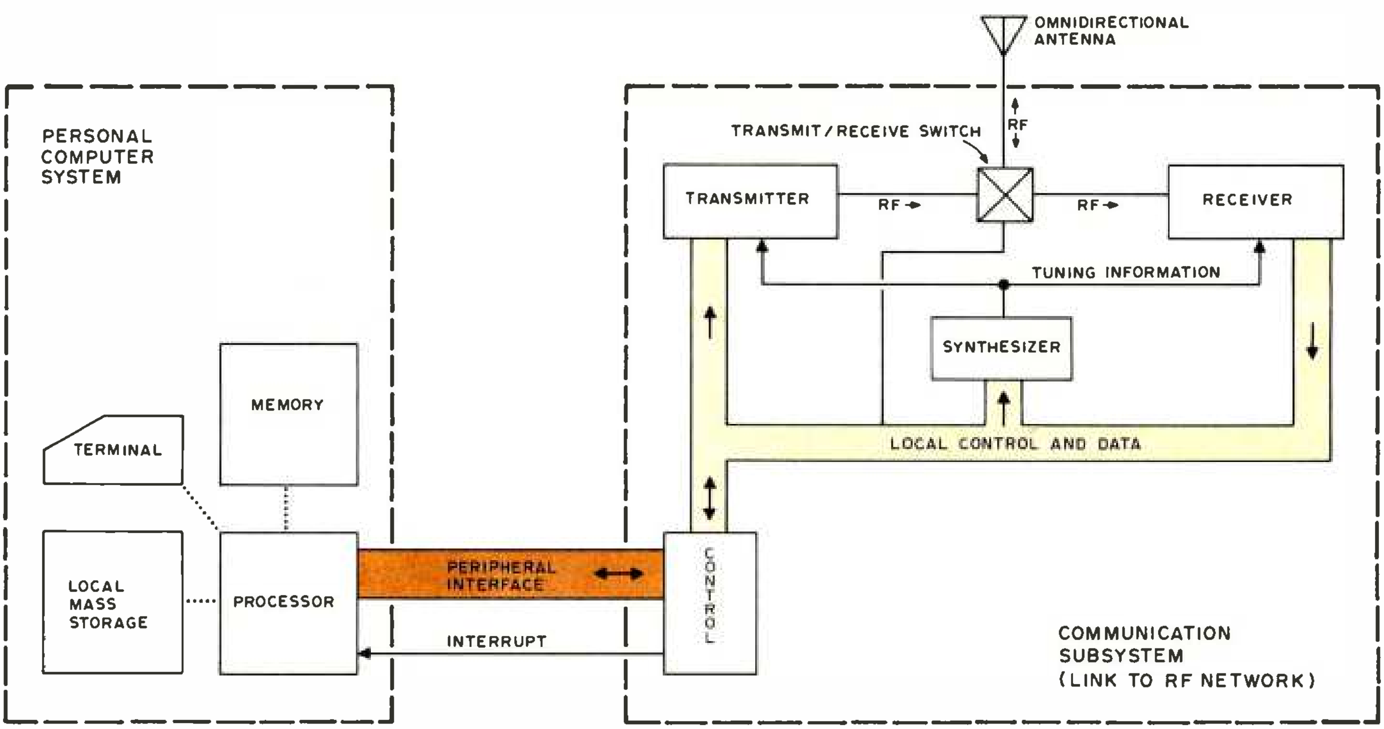


Figure 1: Conceptual outline of the communications subsystem as a peripheral of the typical personal computer system. The system components assumed of the computer are some memory, mass storage which is completely computer controlled (this excludes manually manipulated audio cassettes), a terminal and of course, a typical microprocessor. The communications subsystem consists of a frequency synthesizer which sets the communications channel used, a transmitter, receiver, antenna switch and control logic to interact with the computer. The control logic design can be simple or complex, depending upon how much of the "smartness" of the network terminal is incorporated into the personal computing system's programs as opposed to the logic of the communication subsystem's controller. The details of the software protocols are well understood in the computing field, and examples of radio data communications networks funded by ARPA have been demonstrated quite successfully and can be used as inspiration for this endeavor.

and (optionally) return the data path information to me. Full redundant error checking could be employed to insure data reliability, a necessity for exchanging software through a number of relays.

Hopefully, regulations affecting these communications would not have the restrictions of the amateur radio service regarding commercial interest and entertainment content. Thus, the network could become a truly democratic marketplace with wide distribution of a large range of intellectual products. Some data categories would require special transmission techniques. For example, data could be defined in such a way to make it easily segmentable as a function of dynamically available buffers, optimum transmission rates, or communication time windows.

Hopefully, any "Citizens Radio Service" could become a model for efficient spectrum usage and a synergistic interaction of individuals across the country via computer controlled relays.

Another possible use of such a communications network would be that individual stations could maintain data to be accessible by the network. For example, suppose an individual has a floppy disk or video disk with a library of Star Trek games that are public information. Standardized file access software would allow any network user to access these programs directly or make his or her own contributions to the library. Obviously, such data is not necessarily limited to computer programs.

Essentially, a communications network of this sort, if defined with maximum generality, would be a multiprocessor system of a unique sort. Advanced individuals would undoubtedly give the network artificial intelligence attributes, and the system might even become evolutionary like Conway's LIFE. What is necessary now is discussion of the viability of the idea and the creation of any optimal functional specification. This is an opportunity for small processor hackers to cooperatively produce a new and unique entity that would certainly have long-term cultural ramifications, considering the acceleration of technology.

Hardware Requirements

A reasonable first step towards implementing this scheme would be to develop a useful subset within the present structure of radio frequency allocations. It would be difficult to have a totally new communications service gain regulatory approval and user acceptance from a zero start. Probably the easiest way to begin would be by using amateur radio as an initial vehicle of experimentation. Obtaining an amateur license for VHF privileges is not difficult; Morse code proficiency of only five words per minute is required along with a basic theory test. For the sake of demonstrating the maximum potential for the idea, let's assume a fairly elaborate structure for this feasibility model. However, it is probably more realistic to assume that local groups will put together small networks that would suit specific needs, later expanding into something closer to what will be explored here.

The hardware could be structured as follows: Some spectrum should be dedicated to this application. Within amateur radio, this amounts to a gentleman's agreement, which in the amateur environment has generally been a very successful mechanism. A portion of the 144 to 148 MHz or 220 to 225 MHz band would be a good choice. Because the higher frequencies presently enjoy less use, let's postulate that 224 to 225 MHz be set aside for personal computing. This band could easily be split into 99 channels at 10 kHz separation, from 224.005 to 224.995. This channel spacing should allow data rates up to at least 1200 bps. A good modulation scheme would be audio modulated FM. FM is easy to synthesize and detect, and audio modulation would allow compatibility with conventional modems. Frequency shift keying, while potentially narrower than FM, would require greater frequency precision to receive accurately.

Frequency determination should be by digital frequency synthesis so that the computer would have direct control over channel selection. Because of advances in phase locked loops and other integrated circuit technology, synthesizers are becoming the preferred method of discrete frequency generation, even in radios with manual control. The next few years will see the introduction of complete LSI synthesizer systems, many intended for the Citizens' Band market.

The modem and synthesizer are two elements of the communications subsystem that would perform as a peripheral device of the personal computer system. This device, while basically a VHF transceiver, must be organized to interact directly with the controlling software. For example, it could be structured in a way very similar to a UART (universal asynchronous receiver transmitter) device, with control and data registers accessible to the system bus. A first in first out data file would be useful to relieve some of the data load from the processor, although this certainly would not be a necessity. The simplicity of a character oriented system would have large appeal.

The communications subsystem probably should operate in an interrupt driven mode with the processor, again, in much the same way as a conventional UART can be wired.

Continued on page 94

The Brains of Men and Machines

With this second article on the brain's output control system, we begin a more detailed look at the mechanisms by which the brain accomplishes some of the functions which robot systems will also be called upon to perform. (A number of the terms which are used in this article were defined and discussed in the first part which began on page 11 last month.) As we reach a more concrete level of description of the brain's operation, we will encounter many points which are not yet entirely resolved, and many questions which are subjects of dispute between competing theories. Since it would seem that the present reader is more likely interested in potential applications of brain architecture than in the exact nature of the debate on fine points of physiology, I will simply present the position which seems to me to be most strongly supported at the present time. I will also make some simplifications where they seem warranted by the intended purpose of these articles. (To atone for these sins, I will also offer a list of references for the reader who is interested in pursuing the subject in greater depth.)

It seems likely that any robotics system will require some kind of output controller concerned with the generation and execution of patterns of movement in space, and the required control systems may be expected to range from very simple to very complex. The evolution of the biological brain of course has also had to solve this problem, and it has accomplished it with a set of capabilities for control which are probably as complex as any that we will be likely to encounter for a long time to come. The jointed limb scheme which has been employed as the chief means of locomotion and manipulation in terrestrial animals requires a very complex control system. It is true that a robot, which is free of such restrictions as an uninterrupted blood flow to all of its parts, has other options; wheels and treads for example. These devices might permit simpler control systems, but I would like to suggest that for a system capable of operation in a generalized environment, the jointed limb scheme may be superior. Try to picture a wheeled or treaded robot scaling a cliff or climbing a tree, or even using a stool to dust the bookshelves. Since a motion control system which can handle the jointed limb scheme can also handle simpler systems, it may be most appropriate to plan for the future by starting with this basic scheme in early designs.

The Motor Control System

With regard to the actual mechanisms which are to be controlled, it is interesting to note that they are of only two basic types. The only two things that you are capable of doing are contracting a muscle and releasing glandular secretions, period. Everything else is only some combination of these two. Muscles and glands are the only devices to which the brain interfaces. In the present discussions we will concern ourselves exclusively with the muscles and the system which controls them, usually called the "motor control system."

There are two fundamental principles employed in the brain's motor control system. The first is to buffer each level of command with subprocessors which interpret the commands from higher levels as objectives; and compute appropriate outputs for achieving the objectives, while taking into account local feedback inputs and environmental information. A whole series of such steps is employed, with the "objectives" becoming more concrete at each stage. In this fashion, a pyramid of processors is defined which can accept very general directives and execute them in a reflex fashion with quite considerable flexibility in the face of varying loads, stresses and

Part 2: How the Brain Controls Outputs

Ernest W Kent, Associate Professor
Dept of Psychology
The University of Illinois at Chicago Circle
Chicago IL 60680

other perturbations. This system by itself is quite capable of things such as bipedal locomotion with maintenance of balance on uneven terrain. It cannot, however, operate in a goal directed fashion.

The second principle of the motor control system involves the operation of higher level systems which generate output strategies in relation to behavioral goals. This principle is the division of output tasks on the basis of their relation to input information rather than type of motion required. We shall examine some specific examples which illustrate each of these ideas.

Kinesthesis

The operation of the motor control command chain depends heavily on certain sensory inputs which provide feedback and status information for moment to moment operations, and it is appropriate to begin our investigation of output with a look at these inputs. Perform this small experiment. Close your eyes and put one hand somewhere out in front of you, then touch it with your other hand. Most people have no difficulty doing this quite accurately. The question is how, with your eyes closed, could you guide your hands to the right spatial locations? The answer is that we have a number of special sensory systems of which most of us are not even aware. These senses have the primary purpose of informing the brain's output control processors of things such as the relative positions of the limbs, the tensions of the muscles, the acceleration of the body in different directions, etc. Most people are unaware of these senses because they do not have a conscious content or "experience" associated with them, as do senses such as vision and smell. Nonetheless, they are among the most extensive and intricate sensory systems of the brain, and when they are damaged, the results are immediately apparent. With damage to the systems which report limb position, some people are unable to carry out the small experiment you just performed. In fact, such people are generally unable to execute any muscular action correctly without constantly watching what they are doing.

The sensory system which reports on the status of the limbs is called kinesthetic sense, or kinesthesis, and it handles three sorts of information. These are joint angle, degree of load on a muscle, and degree of stretch or extension of the muscle. These three types of input information are used at various levels of the motor system to control sequencing and provide feedback information. This is another instance where place coding specifies the particular unit and type of quantity in question, and frequency coding carries the intensity information. The transducers which translate these quantities into neural impulse streams need not be discussed in detail since adequate mechanical counterparts are readily available.

Vestibular Sensory Inputs

The other sensory system which is strongly related to the brain's output control is the vestibular sensory system. This is the system responsible for the "sense of balance" among other things. Specifically, it provides continuous readout of the inclination of the head with respect to gravity, and the acceleration of the head in three perpendicular planes. This sensory system is located in a single set of transducers on either side of the head near the middle ear, rather than a multitude of transducers distributed through the body as is the case with the kinesthetic sense. Although the output therefore only refers to the head, the position of the head with regard to all other parts of the body can be computed from the information provided by kinesthetic inputs. Accordingly, the output

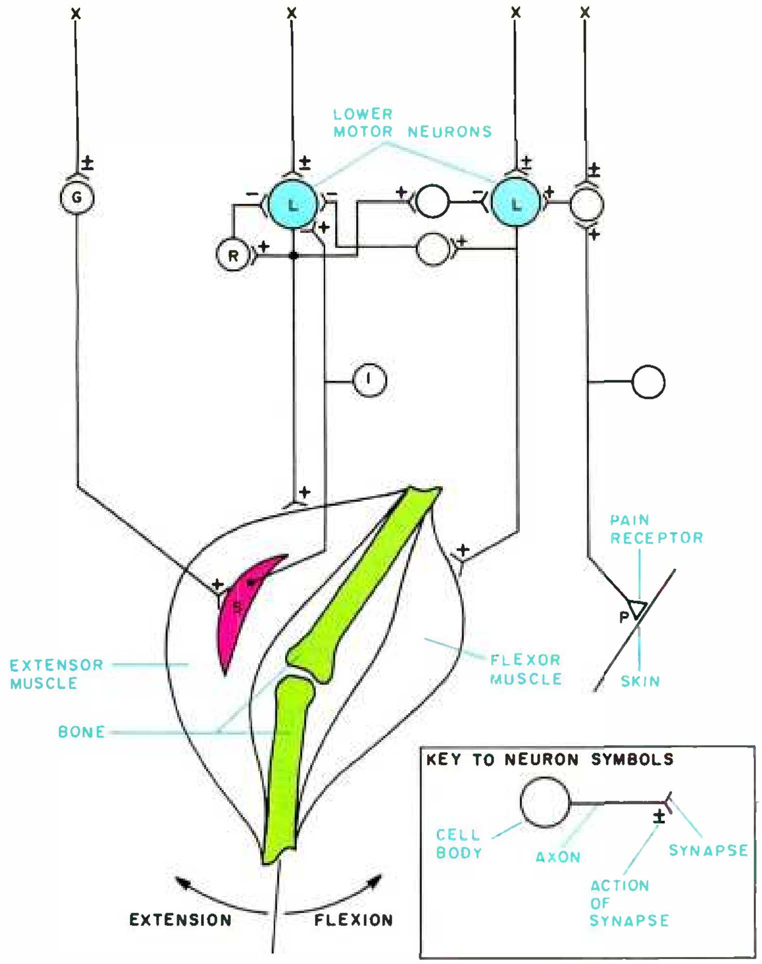


Figure 1: Some important parts of the lower motor neuron (LMN) circuitry which has final control over muscle contractions. See text for an extended discussion of this low level closed loop feedback system.

of the vestibular transducers is made widely available throughout the system as input to most of the high and low level motor processors. In this case too, the existence of easily available transducers for such quantities makes it unnecessary to discuss them in detail. Any device capable of reading out inclinations and accelerations will do when designing our robots.

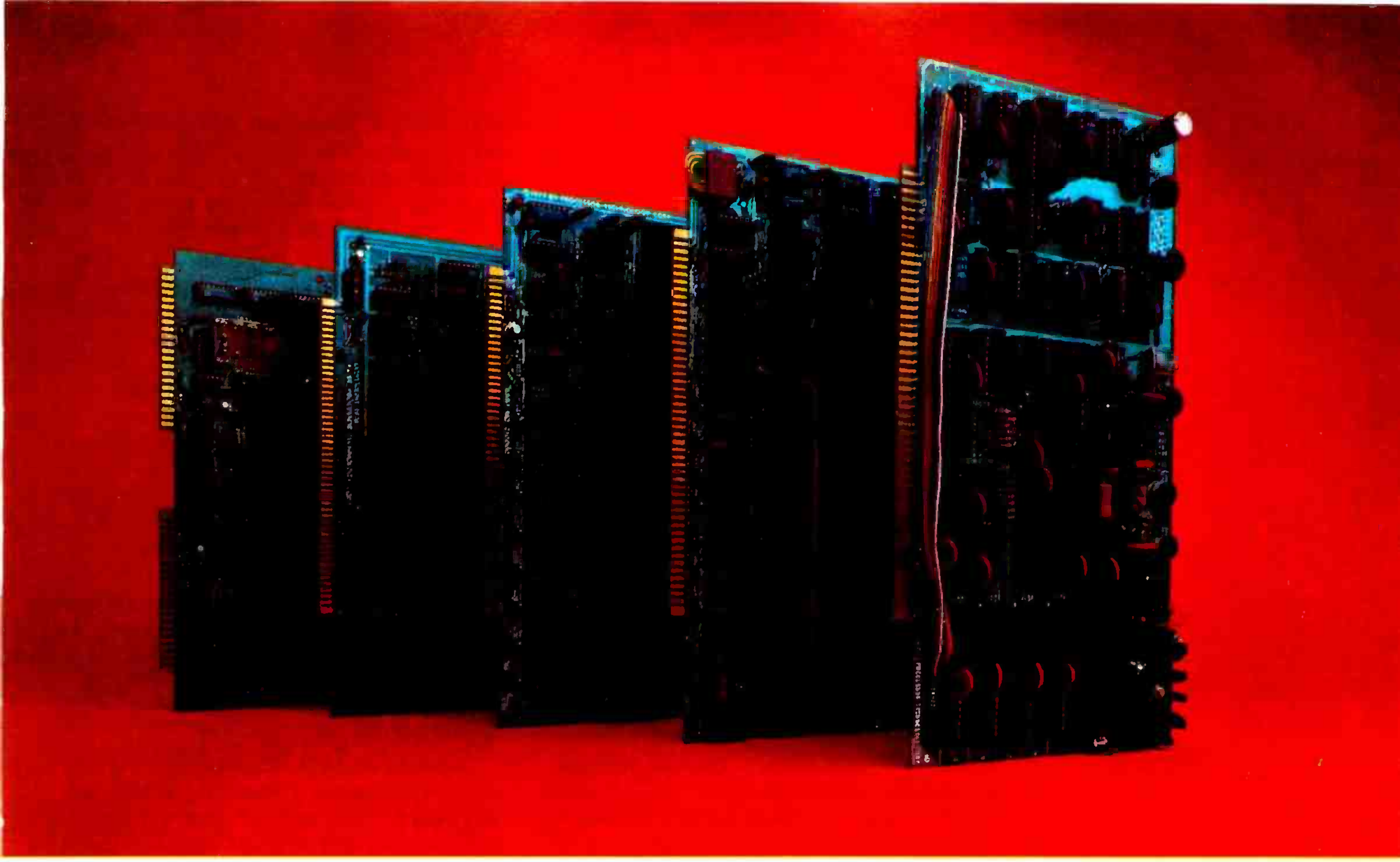
The Typical Joint: a Control System

In most cases, muscles work in opposing pairs, one to open or extend a joint and one to flex or close it. This is necessitated by the fact that muscles can only exert force in one direction (contraction). Figure 1 demonstrates the arrangement for a typical joint. This diagram also shows some of the neural elements which control the contraction of these muscles. The principal neuron of this system, the one which provides input to most muscle fibers, is called a lower motor neuron, and is labeled L in figure 1. This type of neuron (and the other neurons associated with it) is located in the spinal cord, and is the final processing stage before output to the actuator. This little system is a good place to illustrate some of the principles of the brain's motor organization. We shall refer to the lower motor neuron and its associated elements as an "LMN system." Basically, LMN systems must accept commands from a multitude of other systems which desire access to the muscle in question, attend to them according to their priority, modify them according to inputs from kinesthetic and vestibular systems as well as status information from related LMN systems, provide an appropriate output to the muscle, and make their own status information available to other systems. There are a great many LMN systems in the spinal cord. Every muscle is composed of thousands to millions of fibers, and in the case of muscles used for precise operations, there may be an LMN system for each individual fiber. In other cases, a single LMN system may control many fibers of a muscle.

In a practical robotics application, I see no reason why a single servo actuator and "LMN" processor for each joint would not suffice. There are reasons why a single processor for many joints is less practical, but before addressing this issue, let us examine the LMN system to see what sorts of things it does.

In figure 1, for clarity, we show only a single LMN driving each muscle. The degree of contraction of the muscle is proportional to the output pulse frequency of the LMN; the higher the frequency, the stronger the contraction. The circuit shown on the right illustrates the simplest type of protective spinal reflex; a pain receptor in the skin (P) fires a neuron in the LMN system which fires the LMN driving the flexor muscle. This simple high priority operation quickly removes the limb from danger. Inhibitory cross connections of the LMNs driving the two muscles insure that they do not act antagonistically; one relaxes as the other contracts. This reciprocal circuitry is generally active in all LMN operations unless specifically overridden. Not shown are outputs which inform higher centers of this action to allow for the necessary corrective action of other muscles and limbs which must take up the redistribution of weight, counteract shifts in center of gravity, etc.

Inputs to the LMN system from higher centers may request a variety of actions, such as holding a particular position, moving to a specified position, moving with a particular velocity, etc. The LMN attached to the extensor muscle on the left in figure 1 is shown with some of the associated neurons which are involved in the process of carrying out these instructions while compensating



Subsystem B

Each board is a standout. Together, they're a powerhouse.

In the beginning there were boards, thousands of them.

That's how we started in the business. Making memories and interfaces for other people's computers...and making them better.

Now that our own Sol has become the number one small computer, you might think we're putting less emphasis on our board business.

Not so.

We're just doing more creative things with them.

One neat package gets your computer on the air.

For example, we've built Subsystem B, which ties together five Processor Technology modules into a completely integrated system that makes other S-100 Bus computers work almost as well as our Sol.

Subsystem B includes a memory module, three input/output modules, a general purpose memory, and appropriate software.

A specialized software program called CUTER knits together your computer and its peripherals to create an integrated, smoothly working system.

It's the fastest, cleanest way to get on line, and it costs less than if you bought each module separately.

You get your choice of two low power, reliable memory modules in 8K or 16K capacity.

Our VDM-1 video display module (still $199 in kit) is the industry standard display device with over 6,000 in use.

Our CUTS high speed, low cost ($110 in kit) audio cassette interface is the most reliable on the market and is supported by our broad line of cassette software including Extended BASIC, FORTRAN,* PILOT,* FOCAL and numerous others.

And our 3P+S input/output module offers a low cost way to handle virtually all the I/O needs of any S-100 Bus compatible computer system. There are close to 10,000 in the field. Price is just $149 in kit.

Yes, we may have become the maker of the Number 1 small computer — the Sol. But we haven't neglected the quality of our board business. We can't afford to... because we use many of them in our own computers.

For our most recent literature and price list see your dealer or write Processor Technology Corporation, Box B, 7100 Johnson Industrial Drive, Pleasanton, CA 94566. (415) 829-2600.

Processor Technology

*Available soon.

Circle 98 on inquiry card.

for external loads. Note that there is a special muscle fiber (S) which receives its input from the small motor neuron (G) rather than from the LMN driving the other fibers in the surrounding extensor muscle. This special fiber is part of the transducer system for a kinesthetic monitor of muscle stretch. There is a sensory neuron (I) which has an input attached to the S fiber, and this neuron is fired when the S fiber is subjected to stretch, at a rate proportional to the degree of stretch. Since the S fiber is mechanically attached to the rest of the muscle, it is stretched or relaxed by inputs or forces which extend or contract the main muscle, as well as by its own private input signals from neuron G. The axon of the I neuron makes an excitatory synapse on the LMN, thus increasing its drive when the S fiber is stretched. Since increased output by the LMN tends to contract the main muscle and relieve the stress on fiber S, we have a negative feedback loop.

Suppose that the higher centers in the system wish the LMN system to maintain a particular angle on the joint. This is specified by a set of constant inputs from above (X) to the LMN, and to neuron G. Now suppose that a stress such as increased load in the hand is suddenly applied to the joint. This will tend to flex the joint further, causing the extensor muscle to be stretched beyond the specified degree of contraction. This in turn stretches the S fiber and increases the output of neuron I, and thereby, the output of the LMN. The resulting increase in contractile force of the muscle compensates for the increased load. This allows the system which requested the maintenance of joint angle to remain ignorant of loading conditions and fluctuations.

On the other hand, a new input to neuron G can cause the S fiber to contract independently of the drive to the main extensor muscle, thereby increasing the output of the I fiber for the same degree of extension of the main muscle. This defines a new "set point" for the system. (Hence the need for a separate joint angle kinesthetic system for output to higher systems which don't want to untangle the effects of inputs to G on outputs from I.)

From this point, it is clear that the normal considerations of control theory are applicable, and it does not matter whether the system is neural or electronic. For example, in this system the mechanical response time of the muscle and joint, which are in the feedback loop, may be slow compared to the response time of the neural elements. In this as in any other system, that means that instability and oscillation may result if the system gain does not roll off at higher frequencies. This roll off is accomplished by the small neuron R which produces a fast self-inhibitory action on the LMN with each LMN output pulse. At low input pulse rates from higher systems, the weightings of the synaptic contacts (as described in last month's article) is such that the pull down from firing threshold in the L cell produced by the R cell's input has substantially decayed away before the next positive input arrives, and thus has no effect on it. At higher input frequencies however, the positive input pulse will encounter increasingly greater antagonism from the recurrent negative input produced via R by the preceding output pulse, and will thus be less effective in bringing the axon hillock above threshold. This effectively reduces the gain of the system progressively as higher frequencies are approached.

Fitting Lower Motor Neurons into a Larger Context

Looking at the LMN system in the context of the whole hierarchical motor output system, it is apparent that the brain is using a "temporal byte" of frequency coded analog information to specify information about degree or quantity of action. In addition, the set of all of the input lines to the numerous LMN systems constitutes a "spatial byte," or place code, which is essentially digital in character, and in which the selected lines (bits) select the set of LMN systems which are addressed and thereby determine the nature of the movement to be performed, but not its speed, force, etc.

At first glance, it would seem reasonable to try to model the behavior of the LMN system with an analog device such as an op amp with a feedback loop. In practice, such an analog device might be quite tricky since the LMN system must integrate inputs from a wide variety of sources with different priorities. A real LMN has about 10,000 synaptic inputs. There is also the difficulty of encoding the analog information from other systems. Given that we will have many fewer LMN type units to worry about, it may be more practical to do both addressing and value transfer with digital techniques. This would suggest a digital processor of some simple type to replace the LMN unit rather than the op amp, and it may be that this would in the long run be the easiest way of dealing with the interactions of the various inputs to the system.

The next question that arises is, why not use one processor at high speed to run all the joints? There are several considerations. One that is immediately obvious is reliability. If one LMN system is lost, the others

The end of Kit-Kits.

The end of bad solder joints, heat damaged components and sick IC's. Introducing the Semikit. Item 1, a 16KRA Memory Board, $369.

Let's face it. Loading and soldering PC Boards is not much fun for the kit builder. Even more important, it's the place where most of the trouble gets introduced. The real fun and education comes in running and testing boards.

Now the Semikit with fully tested IC's.

At the price of a kit, Processor Technology Corporation introduces the Semikit. It's a fully stuffed, assembled and wave soldered PC Board loaded with IC's that have gone through Q.C. and final check-out (a first in the industry).

We leave you the fun of testing with our fully documented set of instructions. We do the production tasks of loading, wave soldering and inspecting the boards. You do the more interesting and time consuming chore of testing and burning-in the boards.

The result is one sweet deal for both of us. You get a board where the primary causes of damage (poor solder joints, excess solder and bad IC's) are virtually eliminated. You get a board of highest professional quality. And we get the business!

The 16KRA Memory Board's at your dealer now.

Your Processor Technology dealer has the first Semikit, a 16KRA Memory Board, in stock and ready to go right now. You can take it home tonight for $369 as a Semikit or for $399 fully assembled, tested and burned-in.

You'll have a 16,384 byte memory with a better price performance ratio than anything on the market today. Now you can afford to add quality, high density memory to your system for remarkably little. And you can add enough to solve complex computing problems right in the main frame.

The memory features invisible refresh. There's no waiting while the CPU is running. Worst case access time is 400 nsec. Each 4,096 word block is independently addressable for maximum system flexibility. Power is typically 5 watts, the same as most single 4K memory modules. Back-up power connection is built-in.

Other Semi's are coming your way.

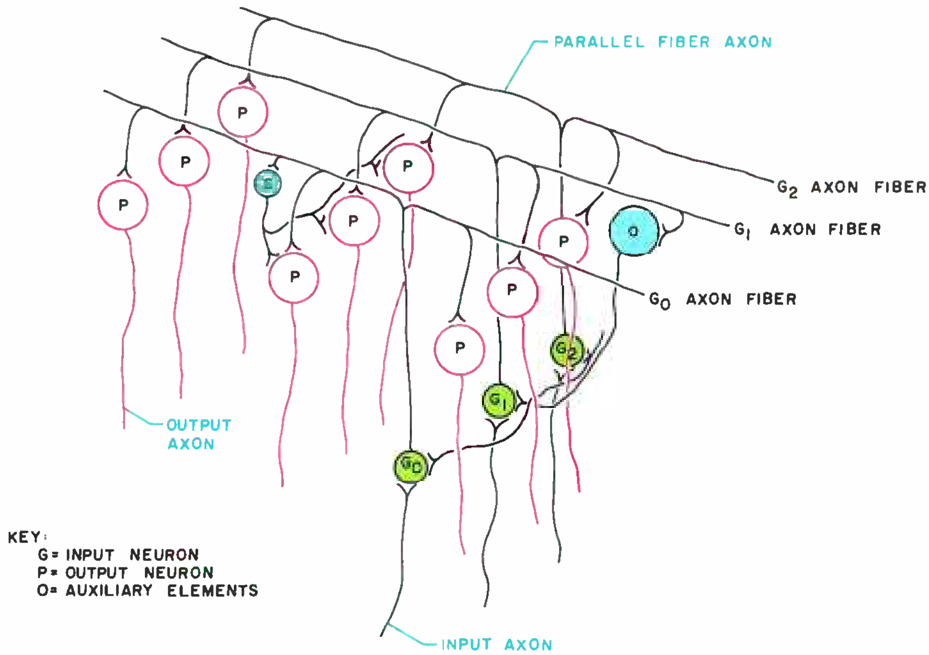
The 16KRA Memory is Processor's first step in adding more fun, capability and reliability to your computer system at lower cost. Other modules are on the way to your dealer now. Come on down today.

Or you may contact us directly. Please address Processor Technology Corporation, Box B, 7100 Johnson Industrial Drive, Pleasanton, California 94566. Phone (415) 829-2600.

Processor Technology

Circle 98 on inquiry card.

Figure 2: Principal elements of the cerebellar cortex. The output cells (P) are fired in sequence by pulses traveling down the parallel fiber axons of the input cells (G). Each input axon selects a set of output cells for activation, and time delays in the parallel fiber axons help establish sequences of outputs. Other cells (O) handle interactions between elements of this cortex.



can take compensatory measures almost automatically. Second, since the output of each LMN system is a factor in the output of each of the others, and since the LMN system is a part of several otherwise distinct feedback loops, a single central processor system would have to be quite complex. Essentially it would face the solution of a number of simultaneous differential equations, or else have to deal with each component motion in sequence. This sort of sequential operation would produce a slow, jerky "movie robot," because each action would have to be completed to obtain the results as input data for computing the next action. A processor with sufficient speed, sophistication and core to handle the differential equations might well be more complex and costly than the multiple simple parallel processor approach. At the other extreme, which the brain has apparently found to be the best approach, programming would be a very simple test-operate-test-exit sequence, in which the actions of other units performing other actions simultaneously are entered as data each time around the loop. The moment we break out of this sequence to handle several "simultaneous" operations with a serial set of such sequences, things get more complex. However, at processor speeds it should certainly be possible to do some of this without doing much more than adding a little scratch pad memory to the simplest robot system's ROM. The best compromise for a robot remains to be demonstrated. Finally, a hierarchical system with interactive parallel units at the bottom frees the upper levels of the system to engage in coordinating the actions of the lower parts into complex actions of the entire organism or device. This function by itself may require substantial processing power and time without the added burden of those jobs which the brain delegates to the LMN systems and their immediate superiors.

Reflex Automatons

This organization of LMN units and their "supervisors" forms a reflex machine capable of quite elaborate motion control and generation (although it does not initiate motion except in response to high level commands, or as a predetermined response to specified sensory inputs). It is essentially an automaton, but a very complex one. The organization of the hierarchy is quite conventional, and similar to a military command chain. The processing elements which have the responsibility for coordinating the movements of different limbs, for example, output control commands to the LMN units at the local level, rather than to the muscles directly, and leave the LMN units to handle the details. They in turn receive orders from, and report to, processing units that are concerned with coordination of whole body actions, the maintenance of posture and balance, and so on. Its major departure from a "command chain" model is the existence of elaborate lateral information transfer between processing elements at the same level in the hierarchy. The operational principles at each level are quite similar to those we have examined in detail in the LMN units which form the lowest rank in the system.

In the brain, this hierarchical system is

Continued on page 146

Seven points to consider before you buy your small computer.

In this magazine, alone, there are probably a dozen ads for small computers. New companies are breaking ground like spring flowers.

How, then, do you determine which computer offers the features you need most... at the price you can afford?

We'd like to propose seven basic questions to help you make an intelligent decision.

1. How complete is the computer system?

Many buyers of small computers are in for a rude awakening when they have to spend additional money for interfaces.

The Sol-20 Terminal Computer was the first *complete* small computer system. Everything you need to make it work is included in the basic package.

2. Is powerful system software available?

It won't do if your system is "tongue-tied."

Processor Technology Corporation has devoted more effort to the development of software than any other small computer maker. Our latest offering is the first fully implemented disk operating system for a small computer: PTDOS. It contains over 40 major commands, several languages and numerous utilities. Our high level languages include Extended BASIC, Assembler, FORTRAN,* FOCAL and PILOT.*

3. Is the system easy to expand?

More and more computer owners are expanding their small computers to handle business and other specialized requirements.

The largest Sol system can handle 64K bytes of RAM memory and operate with a three megabyte on-line disk memory. Sol systems use the S-100 Bus. So you can use a wide variety of hardware.

4. Is the computer well-engineered?

Our Sol systems are the most conservatively rated and ruggedly built in the industry, period. In addition we designed them with you, the user, in mind; Sols are easy to build and a joy to operate.

5. Does it have proven reliability?

What is the track record? There are over 5,000 Sol systems in the field. Our track record for reliable performance is unparalleled in the small computer field.

6. Does it have good factory support?

A computer is a complex piece of hardware. So you want to be sure it is backed up with complete manuals, drawings and a factory support team that cares.

Processor Technology offers the most extensive documentation of any small computer manufacturer. And we maintain a patient, competent telephone staff to answer your questions.

7. Are maintenance and service people accessible?

Where are they located?

Processor Technology has maintenance and service people in over 50 cities around the U.S.

As you continue turning the pages, see how we stack up to the other computers in this magazine. If we've succeeded in whetting your appetite, see your Sol dealer or write for information on the complete family of Sol computers.

Processor Technology Corporation, Box B, 7100 Johnson Industrial Drive, Pleasanton, CA 94566. (415) 829-2600.

*Available soon.

ProcessorTechnology

Circle 98 on inquiry card.

BYTE's Bugs

Entomological Archives

We like to set the record straight about bugs whenever we can, even the old variety. With this in mind, we point out that there is a bug in figure 3 of the June 1976 article, "Building an M6800 Microcomputer" (see page 45). The Mikbug PIA (IC11) is shown with pins 2 and 9 reversed; pin 2 (PAO) should be the output to pin 2 of IC16; and pin 9 (PA7) should be the input from pin 8 of IC15. Our thanks to author Bob Abbott for this information. Bob sent it to us over a year ago, but it got lost in the limbo of our files.■

Random Errors

John D Leasia PE
2005 N Wilson Av
Royal Oak MI 48073

Unfortunately, the pseudorandrom number generator shown in page 218 of the November 1977 BYTE will not generate a complete set of numbers from 00 to FF as stated. The error lies in the programming, not in the method. Numbers ending in 2, 3, 6, 7, A, B, E and F cannot be obtained.

As programmed, the seed is multiplied by 11, not by 13. In the 6800 program, if the opcodes at addresses 0005 and 0006 are interchanged, the program will correctly compute all 256 numbers without a repeat. Interestingly enough, as programmed, exactly half of the possible numbers are generated with no repeats. The missing 1s end in the digits shown above, which group in 2s.

I found it necessary, on my KIM-1, to clear the carry before each add operation. Otherwise the program would repeat before all 256 numbers were generated.

My program requires additional bytes due to the addition restrictions of the 6502:

Address	Hexadecimal Code	Op Code	Commentary
0000	D8	CLD	; Clear decimal
0001	A5 12	LDA RND	; Load N
0003	0A 18	ASL CLC	; Multiply by 2
0005	65 12	ADC RND	; Add N
0007	0A 0A	ASL ASL	; Multiply by 4
0009	18	CLC	;
000A	65 12	ADC RND	; Add N = 13N
000C	18	CLC	;
000D	69 01	ADC# 01	; Add 1 = 13N+1
000F	85 12	STA RND	; Store in RND
0011	60	RTS	; Return
0012	XX	RND	; Seed location■

Bugged Tidbit

In your October 1977 issue, the programming tidbit on page 174 to substitute for the absolute value function will not detect the condition when (A–B) is negative and within the interval specified by a positive delta. To correct your instruction you will need another constant: NDELTA = - DELTA to test (A–B) when it is negative. A shorter alternative for the whole instruction would be:

```
If    ((A-B) < DELTA
and    (B-A) < DELTA)
then   . . .
```

If you look long and hard at your instruction, you will notice two missing right-hand parentheses. I'm sure you know only too well how such slips inspire an old-maidish compiler to nag, nitpick and fuss.

Victor Kincannon
720 Coolidge St
Fennimore WI 53809■

A Slightly Sour SWEET 16.

John Feagans from Commodore Business Machines Inc has detected a slight bug in the program listing of the SWEET16 interpreter (see "SWEET16: The 6502 Dream Machine" by Stephen Wozniak, BYTE November 1977, page 151). The program, which normally starts at location F700 in hexadecimal on the Apple computer, was reassembled to start at location 0800 for the listing in the article. But the symbol S16PAG, which defines the high order byte of the address pushed on the stack for the RTS instruction as described on page 152 of the article, should have been changed from hexadecimal F7 to 08.

Ed Voightman, Dept of Chemisty at the University of Florida, also spotted the bug.■

Continued from page 82

Also, there should be some hard wired functions implicit in the communications subsystem to ease the burden on the processor. Indeed, control of the subsystem might be a function to be delegated to a dedicated subprocessor. The communications subsystem controller must: operate the transmitter and receiver, communicate with the main personal system processor, direct data to the synthesizer, recognize special signals, send control signals to the modem, detect busy channels, respond to "home" and "scan" modes, allow manual interaction, derive status information, and maintain data and control registers or buffers. There are probably many other functions that could be allocated to the subsystem depending on the intelligence and complexity the designer desires or can afford. Using a dedicated microprocessor in the subsystem design would have the traditional advantage of easy expansion of functional capability. However, most of these functions could also be carried out in main processor's software, which would make the communications subsystem a simpler peripheral.

Software Considerations

So much for the basic hardware. Even if the communication subsystem has its own dedicated microprocessor, most of the network intelligence will be communications operating system software. It is this software that will determine what to do when the operator creates a communications module. Without resorting to complicated notational devices, a communications module is simply a command or a message; and the message is command(s) and data together. Commands could be oriented toward data transfer, such as, "Send the following data to _____." or "Do you have data named _____?". Commands could control the current operating status; eg: "Do not accept data for relay; monitor broadcast data only." or could reflect manual control, "Go to channel 22." Commands would be segregated into two types: internal and external. Internal commands would be intended for one's own system only, although standardization would certainly occur and be useful. External commands, on the other hand, would require standardization because they would be transmitted as part of a message and would control the handling of the data by other stations.

A reasonable first step towards implementing this scheme would be to develop a useful subset within the present structure of radio frequency allocations. It would be difficult to have a totally new communications service gain regulator approval and user acceptance from a zero start.

Besides handling explicit commands, the operating system must have other intelligence. A primary consideration is that each system should know its physical and logical location in the network so that appropriate relays can be worked out. If the initial experiments are carried out over a limited area so that everyone can directly communicate with everyone else, the physical map can be ignored. But eventually the participants will become spread out enough to require the software to determine the best direction in which to initiate a relay. One aspect of having logical and physical maps imbedded in the operating system is that each system will have some sort of address associated with it so that it can be accessed through the network. The address could be an encoding of the actual location (physical or logical) within the system, analogous to a phone number or mailing address; or it could be an entry point to a table of relevant data, such as an amateur radio call sign. Since the logical structure of the network would be some sort of tree, an explicit address code could be a sequential list of tree branches formatted into a numerical code.

The software will define how the channels in the allocated spectrum are used. One technique that could be very successful is that of defining a special frequency for establishing initial contact between two stations. After this communication has been completed, the stations involved in the particular transfer can use other frequencies. Having such a standardly defined "monitor channel" greatly simplifies the logical structure of the network because it allows for open-ended participation by interested systems. If such a channel was not used, the continuity of the network might very well depend on stations meeting on particular channels by prearrangement or assignment. Thus, all systems not actively using the network would configure the communications subsystem to monitor a single channel and interrupt the central processor when the channel became active. Response delays could be assigned or determined dynamically so that not all monitoring systems would simultaneously reply to a network request signal.

Once contact has been established a software algorithm must determine the next action. Although present law requires the presence and control of a person operating this radio station, the ultimate usefulness of this sort of distributed network will depend on demonstrating the feasibility of a completely computer controlled system. At this point in the control flow it would be useful to discuss filtering, a type of algorithm in which information about a message is used to control the transfer of the message. One type of filtering has already been discussed, ie: routing information. If there were other alternatives, a message would not be relayed through a station in the opposite direction from the

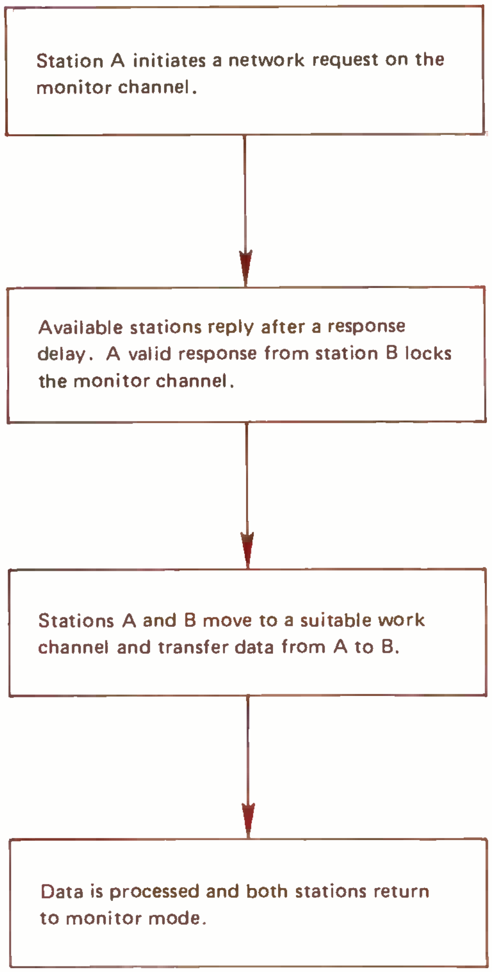


Figure 2: Control flow of a basic relayed transfer. This is an oversimplification of a two party transaction. Station A wants to send a packet of data to station B. A more complex situation exists in the case where A is sending data to some station Z which has no direct contact possible; then B might be the first link in a multiple station relay of the data.

destination. There are many other "filterable" parameters that are dependent on message content. A specification of the universe of possible destinations would allow some transfers to be designed as data for a specific addressee, and thus of interest only to that individual or to relaying systems. On the other hand, data could also be declared to be of general interest, which would be an invitation for all those interested to monitor the data transfer even if it was necessary for only one system to assume responsibility for relaying the information.

Such "addressed" and "broadcast" message types are at the extremes of the filtering spectrum. As more and more data is transferred on the network, it will become more desirable to be selective about how the information is handled. At first, it will be very attractive to accomplish the filtering on the monitor channel, but this would be very sensitive to the mean wait of time of this frequency. Thus, as network use increases, a hierarchy of filtering will develop. The monitor channel would support filtering based on the ability and desire of answering stations to handle the type and quantity of the transfer involved, as well as selectivity based on the priority of the network request. This latter parameter would allow emergency messages and certain types of technical diagnostics (a shutdown command from the FCC, for example) to receive maximum attention. Conversely, distribution and interest codes would probably be best filtered off the monitor channel.

Interference Problems

Another problem that will develop as communications density increases is interference between stations. This is not a trivial problem because the control algorithms will not be nearly as flexible in working around interference or interpreting garbled data as human operators. However, several approaches do seem feasible. The most basic method is simply to search and wait, with a very sensitive channel busy detector that would eliminate any possibility of interference once a clear channel is located. Another scheme would involve time multiplexing so that stations being inadvertently jammed would have a specific time to complain to their neighbors. A third possibility is to employ split frequency modes where each station transmits on a channel that seems to be clear to it.

Thus far, aspects of the operating system have been described that enable systems to establish contact and operate in a one-to-one or one-to-many transfer mode. For a basic architecture this capability is adequate, with all systems involved in the transfer returning to a monitor mode when a particular interaction has been completed. It is possible that this methodology would give good performance, even with a very busy network, because of the potentially low overhead of changing modes and reinitializing communications through programmed control. However, it may also be found to be very desirable to integrate and concatenate network operations so that many data transfers can be achieved when stations establish contact. This is an area where empirical results would be helpful in evaluating alternate approaches.

Limitations of Amateur Radio

All of what has been described thus far can be done within the constraints of amateur radio; however, such an implementation would impose limitations that would only be eliminated by a broad redefinition of the regulations. The most desirable situation would involve spectrum dedicated to the network with a set of rules appropriate to the application. One of the most basic requirements of this scheme is that multiple dedicated frequencies be available exclusively

Go tell it to your computer for $189.

You can have low cost voice entry to your computer with SpeechLab.

Heuristics' original Model 50 SpeechLab™ at $299 was a great success. So, not content to rest on our laurels, we went back to our design bench and came out with the Model 20. It sells for $189 completely assembled and tested for use with any S-100 Bus computer such as Sol, IMSAI, Altair or Cromemco and as the 20A, the Apple.

Now obviously, we had to do a few things to bring the price down. The Model 20 has a somewhat shorter manual. Technically, it's not quite as complex. Don't kid yourself though, the Model 20 does a real job. You can control your computer to do such tasks as data entry, quality control and everything else where you want a "hands-off" operation. Elsewhere in the ad we've illustrated some examples of the kinds of things you can do.

About the Model 20

Heuristics' new Model 20 handles up to 32 words. Software is ROM based. The Model 20 uses 64 bytes per spoken word which is callable from BASIC.

Hardware includes 2 band pass filters with 2 bit amplitude, 2 zero crossing detectors and a linear amplifier.

The combined hardware/software/lab manual includes 10 experiments.

The ROM based speech recognition program occupies 2K bytes of ROM.

The Model 50, still a great buy at $299.

Here's an S-100 compatible system (8080 or Z-80) capable of handling a 64 word vocabulary using 64 bytes per spoken word. You'll get high speech recognition.

Software includes SpeechBasic BASIC programming language in source and paper tape, assembly language speech recognition program in source and paper tape, hardware and self-test program in source and paper tape. SpeechBasic plot, correlation, recognition and advanced recognition programs are offered in source.

The assembly language program uses less than 4K bytes for a 32 word vocabulary. It loads at 100H. A high core version is available. SpeechBasic BASIC interpreter loads at 0 and requires 8K. A diagnostic program is included.

Documentation includes a 275 page lab manual with 35 experiments and a 100 page hardware manual. All manuals are now available as separate items. The big manual is ideal for schools and colleges as an introduction to speech and pattern recognition.

Hardware includes 3 bandpass filters (6 bits amplitude), 1 zero crossing detector, a linear amplifier, compression amplifier, 6 bit A/D converter and a beeper. The raw waveform is available for advanced experiments. A new noise cancelling microphone is also offered.

Prices (f.o.b. Los Altos, CA)

Model 20A (Apple)	$189
Model 20S (S-100 Bus)	$189
Model 20 manual	$ 15
Model 50	$299
Model 50 lab manual	$ 25
Model 50 hardware manual (both manuals $35)	$ 15
Model NC-1 noise cancelling microphone	$ 85

More information.

For more information visit your nearest computer store. Most dealers now carry the Heuristics line. Or if more convenient write for literature. Please address Heuristics, Inc., Box B, 900 N. San Antonio Road, Los Altos, CA 94022. Phone (415) 948-2542.

Heuristics INC.

Even if the communications subsystem (the network node of your home computer) has its own dedicated microprocessor, most of the network intelligence will be in the communications operating system software.

for network use. As has been demonstrated by 2 meter FM repeater usage, this is a definite possibility if many enthusiastic individuals seek to dedicate an underused spectrum segment to a specific activity. To assume that this can happen again may be too optimistic. Portions of the 220 MHz band have already been proposed for a new hobbyist type application, sort of midway between present Citizens' Band and amateur types of communication.

Another difficulty with amateur radio constraints is the requirement that transmissions be under the direct control of the operator. One of the reasons that VHF was suggested is because the propagation characteristics are relatively constant and could allow 24 hour operation. An optimum scenario would involve minimum interaction by the system operator. There would be a short period of operator activity in the evening (or morning or whenever) to see what data had been transferred during the previous day or two, evaluate new data acquisitions, and initiate messages. Since the system would be designed to support data without an explicit address, full-time operation would allow individual systems to interact with the network to find new data according to program. I would not expect that most of the data transfers would be initiated manually when the network reached maturity. This would be the major unique characteristic of the entire system. In a nondistributed network, costs would accrue on a per transfer basis, so it would be unlikely that individuals would pay to have their computers talk to each other all day and all night. In this distributed system, the ongoing costs would be those required to run the computer system and a small radio, and would not be large, even if run intermittently 24 hours a day. System use with the constraint of manual operation would probably not result in a synergistic multiprocessor environment either; watching a computer can become boring quickly. Compared to a timesharing system or other conventional data networks, the response time of this distributed system will be very slow, which would justify a longer time to get results. The slow speed of this network is not really a disadvantage because the application is quite different than timesharing, for example. The existence of the distributed network assumes each node includes a local computer to handle real time applications. It is the extent of this local processing capability that will give the network its unique characteristics. Therefore, it is essential that the network be organized so as to maximize these characteristics.

The hardware and software that has already been described would not have to be substantially modified to support a dedicated spectrum version of this network outside amateur radio's province. The major changes would be organizational and political with technical enhancements. The hardware model that has been designated the communications subsystem would remain relatively fixed although there would be greater functional standardization, and more installations would include more highly evolved hardware. The commercial manufacture of peripheral communications hardware could certainly be expected at this point. The software would undergo more changes, although it should be a clear objective from the beginning to design the system, and particularly the software, so that it is modular and easily expandable. New software features must be implemented and shown to be reliable to allow the individual systems to do useful work without operator intervention. Automatic logging and remote control would be two of these features. More effort than is now obvious would probably have to be put into completing decision trees, that is, ensuring a reasonable machine solution given any possible set of input conditions. The initial forms of many algorithms in the amateur radio context would probably have an escape, such as, "After N operations, or after T seconds, ring bell and wait for operator command." Obviously, structures like this will have to be different outside amateur radio in another band. Hopefully, the evolution of the software will happen within the network itself, much more so than the hardware. The communications network is the ideal medium for individuals to define problems, and develop and distribute the solutions.

Regulatory Aspects

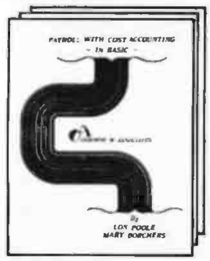
To distribute the solutions involves a regulatory change that would have significant effect. While I am hopeful that individuals would freely distribute some software that is created within this network, particularly software designed to enhance the operation of the network, I also hope that there will be a way to allow economics. Because the resource of radio spectrum time is limited, the economic characteristics must be regulated to preserve the values of distributed communications. On the other hand, a commercial influence could have beneficial aspects if it were properly applied. The goal of introducing economics is positive: Individuals working at home, using their own equipment, could create and distribute products within a free market. The market, like the network, would have as its primary

OSBORNE & ASSOCIATES, INC.

The World Leaders In Microprocessor Books

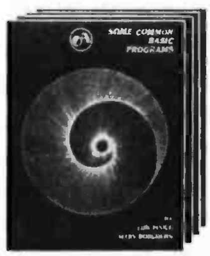
If you want information on microprocessors, begin with the Osborne books.

PROGRAM BOOKS WRITTEN IN BASIC



Payroll With Cost Accounting
Accounts Payable And Accounts Receivable
General Ledger

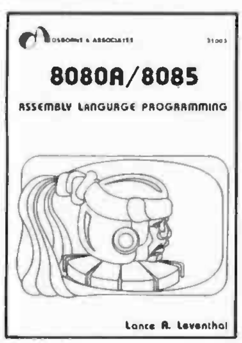
These books may be used independently, or implemented together as a complete accounting system. Each contains program listings, user's manual and thorough documentation. Written in an extended version of BASIC.
#22002 (400 pages), #23002*, #24002*



Some Common BASIC Programs

76 short practical programs, most of which can be used on any microcomputer with any version of BASIC. Complete with program descriptions, listings, remarks and examples.
#21002 (200 pages)

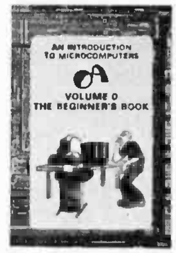
ASSEMBLY LANGUAGE PROGRAMMING



6800 Assembly Language Programming
8080A/8085 Assembly Language Programming

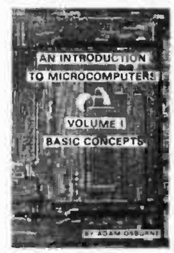
Assembly language primers in the classical sense, these books treat assembly language as a means of programming a microcomputer. Full of simple programming examples.
#31003 (400 pages), 32003*

AN INTRODUCTION TO MICROCOMPUTERS



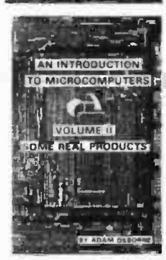
Volume 0 - The Beginner's Book

If you know nothing about computers, then this is the book with which to start. It tells what computers are all about and describes their component parts. When you've read this, you'll be ready for Volume I.
#6001 (300 pages)



Volume I — Basic Concepts

The world's best selling computer text describes those characteristics common to all microprocessors yet specific to none. All hardware and programming concepts you will ever need to know are explained at an elementary level.
#2001 (350 pages)



Volume II — Some Real Products (revised June 1977)

Every common microprocessor and all support devices are described. Only data sheets are copied from manufacturers. Major chip slice products are also discussed.
#3001A (1250 pages)

PROGRAMMING FOR LOGIC DESIGN

8080 Programming For Logic Design
6800 Programming For Logic Design
Z80 Programming For Logic Design

These books describe the meeting ground of programmers and logic designers; written for both, they provide detailed examples to illustrate effctive usage of microprocessors in traditional digital applications.
#4001, #5001, #7001 (300 pages each)

OSBORNE & ASSOCIATES, INC. • P.O. Box 2036 • Berkeley, California 94702 • DEPT. 15

		PRICE	QTY	AMT
6001	Volume 0 — The Beginner's Book	$ 7.50		
2001	Volume I — Basic Concepts	7.50		
3001A	Volume II — Some Real Products	15.00		
4001	8080 Programming For Logic Design	7.50		
5001	6800 Programming For Logic Design	7.50		
7001	Z80 Programming For Logic Design	7.50		
31003	8080A/8085 Assembly Language Programming	7.50		
21002	Some Common BASIC Programs	7.50		
22002	Payroll With Cost Accounting	12.50		
	TOTAL			
	Sales Tax (Calif. residents only)			
	Shipping Charges			
	TOTAL AMOUNT OF PURCHASE			

- 6-1/2%. SF Bay Area residents only
- 6%. California residents outside SF Bay Area
- Payment must be enclosed for orders of 10 books or less.
- I have enclosed:

☐ check ☐ money order

NAME

ADDRESS

CITY STATE ZIP PHONE

SHIPPING CHARGES Shipping charges for bulk orders to be arranged.

☐ 4th class (no charge, allow 3-4 weeks within USA. not applicable to discounted orders)
☐ $.50 per book, UPS (allow 10 days) in the U.S.
☐ $1.50 per book, special rush shipment by air in the U.S.
☐ $3.00 per book, foreign air mail

*This book is not yet available.
Please notify me when it is published:
☐ 23002 Accounts Payable and Accounts Receivable
☐ 24002 General Ledger
☐ 32003 6800 Assembly Language Programming

Please send information on:
☐ consignment
☐ dealer discount
☐ foreign distributors

F3

Circle 90 on inquiry card.

All of what is described here can be accomplished within the constraints of amateur radio; however, such an implementation would impose limitations which could only be eliminated by a broad redefinition of the regulations and operating practices.

attribute its distributed characteristics. Centralized capital would be minimized, and individual creativity maximized.

A subsidiary regulatory characteristic that would necessarily be modified to allow the implementation of economics is data security. Presently, there is no allowance for secret communications on frequencies accessible to the public, presumably to prevent spies from using walkie-talkies for "cloak and dagger" schemes. The desirability of secure communications and the impossibility of enforcing nonencoded computer conversations will force a change in this rule.

It should be pointed out that the Federal Communications Commission has not been dogmatic about maintaining the regulatory status quo. Significant changes have occurred when it could be clearly demonstrated that the largest public interest would be served by modifying the administration of the radio spectrum. The potential utility and benefit of distributed communications make the changes much more plausible than they would otherwise seem. Also, the network can only be considered a potential reality in conjunction with very recent technical innovations. Such a network was not a viable possibility in 1970; now *is* the proper time to begin its implementation.

What sort of applications for the network might occur if it achieved special regulatory status? While there are many possibilities, a general application I call "library building" provides a useful illustration. The goal of this mode of operation is that systems with mass storage capability would interact so that each participant would share a subset of the file structure with the other participants. Because of the relatively large amounts of data involved, such exchanges would probably not occur on a relayed basis; rather, individual files could be relayed later by specific command. In order to participate in library building, systems would maintain directories of several types of files, eg: those maintained and available, those files desired, keys to file types (for example, "games" or "8080 code"), both desired and not desired, and specific files that are not desired. This activity would most frequently be dyadic (that is to say, they would be initialized when a pair of stations determined mutual interest in the activity). Library work represents a network activity that could best be carried on with little operator intervention. A typical command would be the equivalent of "get everything new and share anything except files A, B and C." In fact, this could be a standard background command to be executed when operator initialized transfers become null. The operator would interact with this function by requesting a regular summary of files acquired and dispersed. A prerequisite for this sort of file oriented activity is that standards be developed for file management within personal computing, so that transfers can be made with both processor independence and device independence.

The unique characteristic of any computer is the ease with which it is given new capability by feeding it new software. Thus, the distribution of software through this interactive network could rapidly result in an explosion of new functions. Once the system has been bootstrapped, growth could be faster and more meaningful in terms of legitimate achievement than that experienced in any other medium. Possible future scenarios may give more perspective to the implications of the network.

The system should be interfaced to other networks. Common carriers and cable television are present possibilities, and local laser links and direct communications through satellites are likely to occur in the future. Nondistributed data networks will be a major feature of the cultural technology of the 1990s, providing many of the services already discussed on the scale of television today. Amateur computing in distributed networks could set trends and establish precedents for the revolution to follow.

The hardware definitions for the network could evolve to allow the establishment of new categories of node stations with special functions. One such function could be the data concentrator, a large, fast processor with several wide channels assigned to it. Large amounts of data could be burst transmitted over longer distances to condense much relay work. Other specialties, such as computational batch processor, game playing adversary, etc, will evolve as the applications do.

Blue Sky

New hardware should have a profound impact on the network, especially when that new hardware is a data oriented version of the video disk. Since the video disk is a highly cost effective way to reliably transfer large amounts of data (on the order of 10^{10} bits), it would be impractical to replicate this sort of transfer over a communications link. Further, nondigital data would require extra hardware, long transfer times and prohibitive bandwidths for even VHF radio. However, if we can assume the existence of another commercially oriented system for the economical creation and distribution of physical disks, even at very low volumes, then there is a definite place for a communications network to interact with these disks. To assume such a support system is not unreasonable because of the extreme potential for commercial application. However, the growth of such a support system probably would be accelerated by demand from the computer enthusiast market. The interaction of video disks with the network would occur as an interface to digitally controlled video disk drives and disk program material with imbedded software. If two communicating systems were using identical or similar disks, control information could be exchanged through the network to access the common data. With the huge amount of analog storage available, organized as video, video stills or audio, the imbedded software and transferred control would provide much flexibility applied, for example, as educational or creative utilities.

This aspect of video technology used in conjunction with the network would be helpful in supporting various sorts of synergistic multiple processor functions. In this mode, a number of systems would share a channel or channels via time multiplexing. A useful application, which has already undergone experimentation via timesharing, is the computerized conversation, an ongoing round table discussion that occurs outside the constraints of real time and space. Eventually, as the systems become more sophisticated, this mode could support multiprocessor creative activities, such as music or video synthesis, as well as the creation and use of educational materials. Each communicating processor would use similar creative software, and the individuals would supply data to produce a sort of computer symphony.

One of the most interesting applications of the network capabilities under discussion will be computer gaming on a very large scale. Games could be highly complex, involve months of real time, and have teams of dozens or hundreds of systems. The network will be interfaced to the specialized large systems that will be the amusements of the next decades, a development made more plausible by the many predictions of greater leisure time in the future. The games will evolve to the level where individuals may be more concerned with the construction of an optimum game playing system, rather than playing the games directly. This level of sophistication approaches practical artificial intelligence.

So What?

What I have attempted in this article is to demonstrate the implications of using existing technology to construct a new type of communications network that would radically effect much within personal computing. There are two difficulties: Radio spectrum must be allocated to the activity, and standardization of the technical details must be achieved. I realize that this article has not gone too far with specific technicalities, and that is partially because I felt that the intelligence of a distributed system should be determined by a number of individuals and not predefined. Also, many aspects have been characterized that will involve a substantial amount of technical detail, and I felt it would be more useful at this point to describe alternatives, possibilities and general approaches rather than specific algorithms.

Always in the past, new technology has been greeted with a "so what" attitude. The ultimate implication of the proposals in this article is that the home computer can be part of a dynamic ongoing process rather than an expensive toy that plays the latest tapes. This is not a negative comment about "play" applications; rather, it is a statement that no technology is appropriate in the wrong environment. It is my contention that as a tool the computer is pretty interesting, but no tool is of ultimate interest without real work. It is my hope that others will add to these ideas to make the personal computer a necessary component of a useful system.■

Continued from page 38

interlaced parallel fingers. The magnetic field strength is concentrated in the tiny air gap between these fingers, near the surface of the rotor. The rotor itself is a permanent magnet which has a series of poles magnetized around its periphery. The number of rotor poles equals the number of stator air gaps. It is the attraction and repulsion between these poles of the rotor and the stepping magnetic field of the stator that cause the motor to rotate.

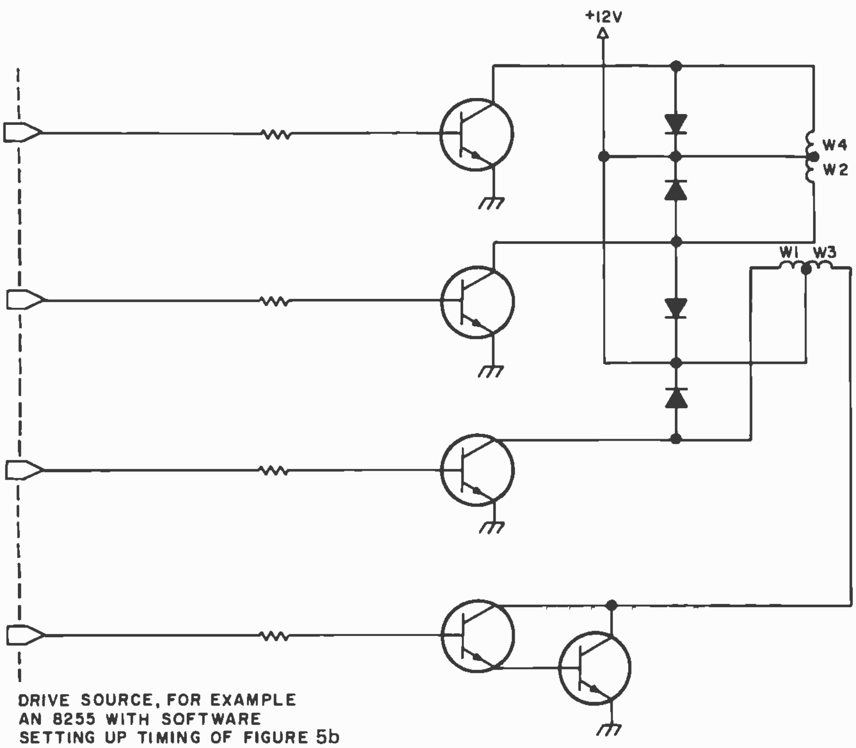


Figure 5a: Simple method for driving 4 phase stepper motors utilizing NPN transistors and a positive power supply. For larger motors two transistors connected in parallel, a Darlington amplifier, may be required as shown for winding 3 in this drawing. The values of the resistors and diodes will depend on the stepper motor being used and the drive source for each phase.

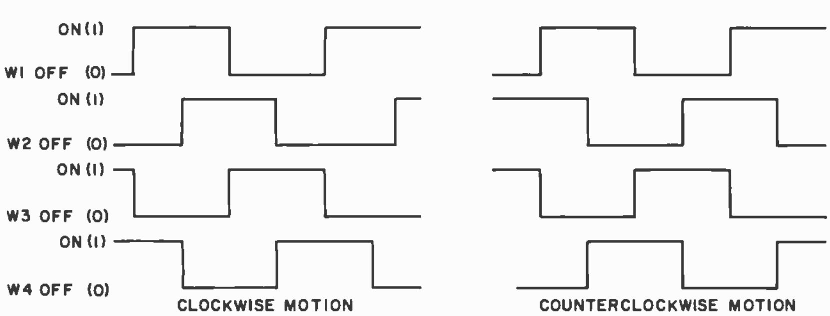


Figure 5b: Timing diagram for a 4 phase stepper motor has one winding being energized and one being de-energized at a time. One side of each winding is conducting current at any time. The energizing pattern is reversed to reverse the motor rotation.

Hardware Solution

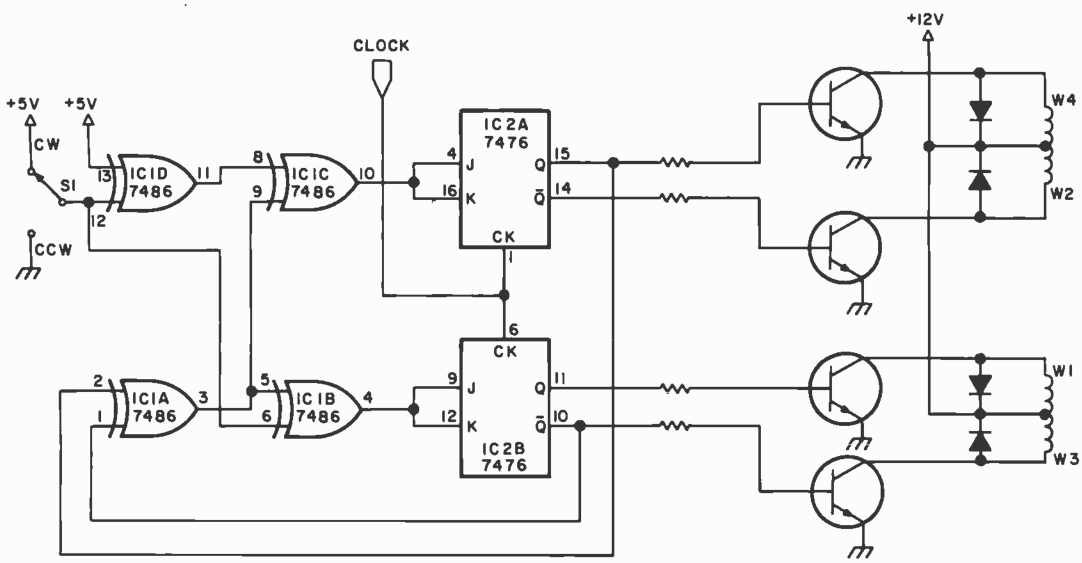
An easy way to drive the coils is with NPN transistors as shown in figure 5. Two transistors at a time are switched to ground to cause current to flow in the required direction. Note that windings 2 and 4 (w2 and w4) cause current to flow in opposite directions of the same coil. A high power motor may require a Darlington drive, as shown in winding 3. This is just two low power transistors driving a high power unit to insure that your IO port will be able to drive the motor coil safely. A 1.0 mA output from an 8255 IO port drives 20 mA with a single transistor and 400 mA with a Darlington. Each transistor multiplies the current by a factor of 20. Two is the limit though, because the guaranteed output voltage of the IO port is 1.5 V. Each transistor requires 0.7 V, so two of them require a total of 1.4 V. The windings must be energized in the sequence shown in figure 5b. Notice that at any given time one half of each coil is energized.

Let's take a quick look at a circuit to produce this coil driving sequence. The circuit shown in figure 6 provides the proper sequence for a reversible drive. Speed is controlled by the frequency of the clock input. For coarse control the clock can be generated by a 555 type oscillator. For very accurate control this clock can be generated by a crystal oscillator. Switch S1, which could be an IO line, controls the direction of rotation. The frequency is more difficult to obtain directly. A digitally controlled oscillator whose setting is controlled by a digital to analog converter would provide very precise and accurate speed control. No processor timing would be required. A typical example of such an oscillator is shown in figure 7. The number of input bits used (in this case eight) determines the number of speed selections.

If the number of steps is more important than precise speed, the circuit of figure 8 can be added to control the clock. The thumb switch inputs could be IO port lines. The LOAD line transfers the selected count into the counter. The START line sets the gate to a transmission mode. When the counters count back down to zero, a pulse is emitted from the borrow line and resets the gate to a blocking condition. The selected number of pulses has been counted out to the motor drive. The motor speed is still controlled by the oscillator frequency.

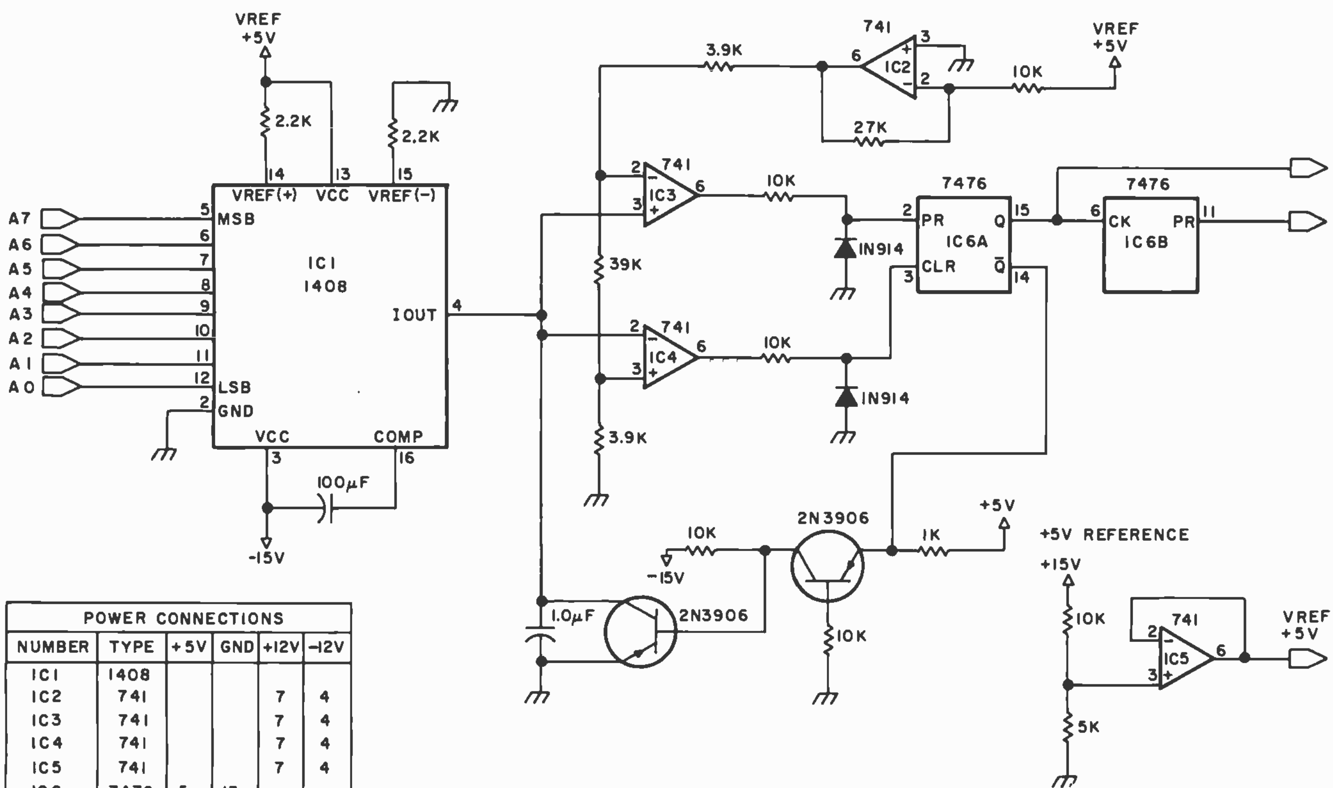
Software Solution

If your only chore is to drive a single motor, then a microprocessor is probably not necessary. But if several motions are



POWER CONNECTIONS		
NUMBER	+5V	GND
IC1	14	7
IC2	13	5

Figure 6: A pair of flip flops provide the memory and exclusive OR gates provide the steering to generate the drive patterns in a hardware solution to the stepper motor drive problem.



POWER CONNECTIONS					
NUMBER	TYPE	+5V	GND	+12V	−12V
IC1	1408				
IC2	741			7	4
IC3	741			7	4
IC4	741			7	4
IC5	741			7	4
IC6	7476	5	13		

Figure 7: This digitally controlled oscillator generates a frequency proportional to the integer output to the digital to analog converter. The frequency can be used as the clock input of figure 6, providing a variable motor speed from a hardware driver.

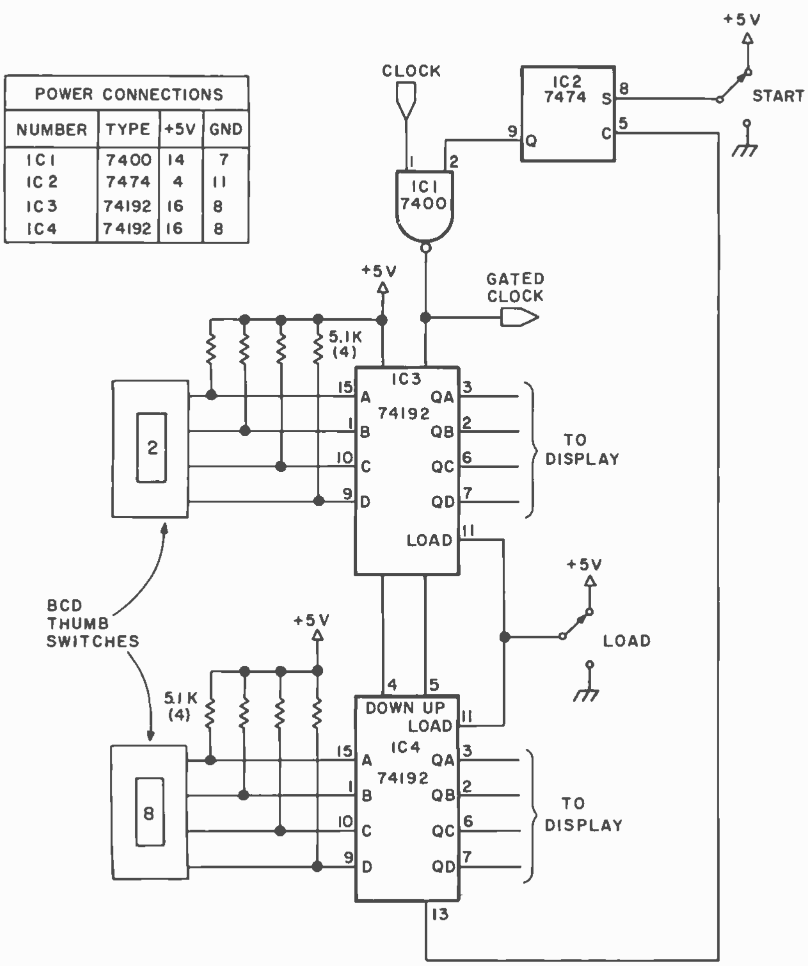
Generated Pattern
(one line per clock state)

	W1	W2	W3	W4	
Clockwise ↓	1	0	0	1	↑ Counterclockwise
	1	0	1	0	
	0	1	1	0	
	0	1	0	1	

Table 1: The pattern generated by the circuit of figure 6. The number 1 represents current flowing in a winding. Reversing the drive pattern will reverse the motor direction.

	W3	W4	W1	W2	
Clockwise ↓	0	1	1	0	↑ Counterclockwise
	0	0	1	1	
	1	0	0	1	
	1	1	0	0	

Table 2: If the drive patterns of table 1 are rearranged as shown, a pair of rotating 1s becomes apparent. This simplifies the generation of these patterns through software.



POWER CONNECTIONS			
NUMBER	TYPE	+5V	GND
IC1	7400	14	7
IC2	7474	4	11
IC3	74192	16	8
IC4	74192	16	8

Figure 8: This circuit will generate a selected number of pulses when the start button is pushed. The thumb switches and the two push buttons could be replaced with signals from an IO port. If desired, a display may be added to the circuit to indicate the number of counts left.

required and the speed and position of each must be controlled, then the microprocessor saves considerable hardware. The entire job can be done inside the processor, with only the Darlington power drive transistors outside. The problem of generating and keeping track of the pulse trains becomes a software task. Let's first look at a routine to drive a single motor.

If the winding drive pattern of table 1 is rearranged as in table 2, a rotating pattern of 1s becomes apparent. Now direction of rotation is controlled by the direction that the 1s are shifted (left or right). Speed is controlled by the rate at which these 1s are shifted and transferred to the motor. Internal counting can be used or external interrupt driven timing can be used. Since I needed other time events, I chose to use the Texas Instruments TMS 5501 for my experiments. This versatile chip provides five separate timers, eight input and eight output lines, an external sense line and a bidirectional serial link. All these priorities are taken care of, too. I will deal only with one timer and merely assume its interrupt has been vectored to my subroutine properly.

The basic scheme is to set the timer for an interval of from 64 μs to 16.32 ms and count off the desired number of intervals. When the desired total time has elapsed, the motor drive pattern is rotated and output. The timing begins again. For higher speeds, above 60 steps per second, only one timer interval is required between each step. By choosing the time interval and the number of intervals, a wide range of motor speeds may be selected. A flowchart is shown in figure 9, and the code is in listing 1. The motor outputs are the four low order bits of one port on an 8255 output port. Now let's look at some of the details of the code.

First, of course, the status must be saved and the interrupt re-enabled. Next check the number of elapsed intervals. If it's down to zero then look at the number of steps requested. If there's more to go then decrement the steps counter and update memory location STEPS. Next update the number of timer intervals (CONTR) because you'll have to count them off again.

The timer is started and set by writing a word, in the range of decimal 0 to 255, to memory.

In the hardware motor drive, each coil drive pattern is determined from the previous pattern by the feedback from two outputs. In this software version it is not possible to read the latched output so the

INTERFACE AGE™ MAGAZINE PRESENTS

MICRO BUSINESS '78™ CONSUMER SHOW

DATE: MARCH 17, 18, 19, 1978

PLACE: PASADENA CONFERENCE CENTER

PASADENA, CALIFORNIA

MICRO BUSINESS '78™ will provide a series of marketing forums and exhibits to introduce the small independent businessman to the new low-cost, high-power business microcomputer that will reduce his company's costs, place him in a more flexible marketplace and provide timely data information.

Emphasis will be on the small budget requirements for purchase of an in-house computer. The show will demonstrate the latest systems, exhibiting complete hardware and software from small hand-held programmable calculators to full turn-key computers.

- Latest in Word Processors
- Newly-Released Business Software
- Low-Cost Text Editing Typewriters
- Modularized Computers

THE LOW COST, dependability, simplicity of operation, and cost savings advantages of microcomputers will be discussed in a series of lectures to remove the many misconceptions the average businessman may have about the microcomputer technology. Lectures by such companies as IBM, Commodore Business Machines and Radio Shack will present the businessman with the latest information about application, service and investment.

Author Adam Osborne will discuss business software.

OTHER LECTURES on the program include:

- Small Business Computing Systems
- Evaluating Your Business Computer Needs
- Software Companies
- The Mainframe Companies & The Small Computer
- The Small Business Computer Company
- Computer Stores and the Small Business System
- Retail Mass Marketing of Microcomputers

Sponsored by: INTERFACE AGE Magazine

EXHIBITORS: PLACE YOUR RESERVATION NOW!

Produced & Managed by:
Show Company International
8687 Melrose Avenue
Los Angeles, California 90069
(213) 659-2050
Ed Tavetian

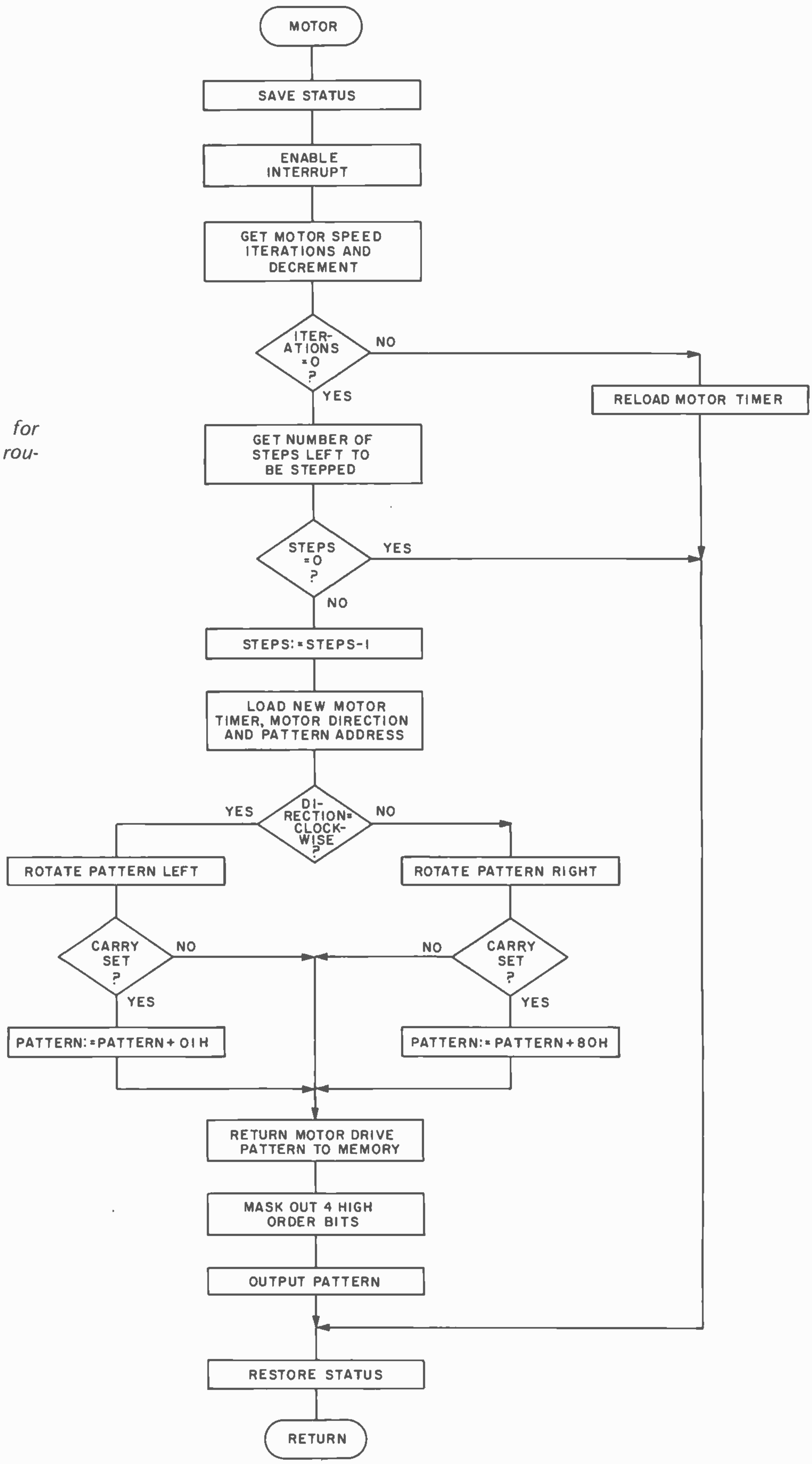


Figure 9: Flowchart for the single motor drive routine of listing 1.

last pattern is maintained in memory location PHASE. I chose to use a full eight bits to store the 4 bit drive pattern and repeated the four bits in the high nybble. Then the overflow on rotate could be sensed in the carry bit. I used RAL and RAR for rotating, sensed overflow in the carry bit, and added a correction of either 01H 80H respectively. I learned later that if I used RRC and RLC, rotate right or left with carry, then this was taken care of by the processor. After rotation the new drive pattern must be stored back in memory to be available during the next cycle. The high nybble is then masked out. After the four high order bits are masked out the motor drive pattern can be output via the 8255 output port. All that remains is to return the status and registers to their former condition and then return to the main line program to await another interrupt. Four of the TMS 5501 output lines could have been used, eliminating the need for another output chip.

A word about stepper motor speed is in order here. With a light load, the motor will respond and follow commands at speeds of up to 250 to 300 steps per second. With a frictional load this maximum speed will reduce linearly in proportion to the friction. An inertial load, such as a flywheel, will not reduce the maximum speed. However, with an inertial load the speed must be programmed over several steps from slow to fast. If the maximum speed is not approached gradually, the electrical stepping moves faster than the load can. The motor will just sit and stutter. A check with an oscilloscope would show the proper pattern sequence occurring. Use of a more complex program with actual motor speed feeding back into it via hardware could help solve this problem.

```
        ;MOTOR COIL DRIVE, INTERRUPT DRIVEN, INTERVAL 3, RST3
MOTOR:  PUSH PSW        ;SAVE STACK
        PUSH B
        PUSH H
        EI
        LXI H,CONTR
        MOV A,M         ;GET COUNT REMAINING
        DCR A           ;DECREMENT COUNT
        MOV M,A
        JZ MDCR         ;CHECK STEPS IF CONTR ZERO
        LXI H,INTVL     ;IF NOT
        MOV A,M         ;GET INTERVAL
        STA TIMER       ;AND RESET INTERVAL
        JMP MDONE
MDCR:   LXI H,STEPS
        MOV A,M         ;GET STEPS
        ANA A           ;SET FLAGS
        JZ MDONE        ;EXIT IF DONE
        DCR A           ;DECREMENT STEPS
        MOV M,A         ;STORE NEW STEPS
        LXI H,COUNT
        MOV A,M
        LXI H,CONTR
        MOV M,A         ;RESTORE ORIGINAL CONTR
        LXI H,INTVL
        MOV A,M
        LXI H,TIMR      ;RESET INTERVAL TIMER
        MOV M,A
        LXI H,DIR
        MOV A,M         ;GET DIRECTION
        LXI H,PHASE
        STC
        CMC             ;CLEAR CARRY
        ANA A           ;SET FLAGS
        JP CCW
CW:     MOV A,M         ;GET MOTOR DRIVE PATTERN
        RAL
        JNC MDR
        ORI 00000001B   ;CORRECT FOR OVERFLOW
        JMP MDR
CCW:    MOV A,M
        RAR
        JNC MDR
        ORI 10000000B   ;CORRECT FOR OVERFLOW
MDR:    MOV M,A
        ANI 00001111B
        OUT OUTMDR      ;OUTPUT SIGNAL TO MOTOR
MDONE:  POP H
        POP B
        POP PSW
        RET

COUNT     NUMBER OF TIMER INTERVALS TOTAL
CONTR     NUMBER OF TIMER INTERVALS REMAINING
INTVL     TIMER INTERVALS 0-255
STEPS     NUMBER OF STEPS TO BE MOVED
DIR       MOTOR DIRECTION
PHASE     MOTOR DRIVE PATTERN
TIMR      TIMER ADDRESS IN MEMORY SPACE
OUTMDR    MOTOR OUTPUT ADDRESS IN IO SPACE
```

Listing 1: An 8080 assembly language program for driving a single motor.

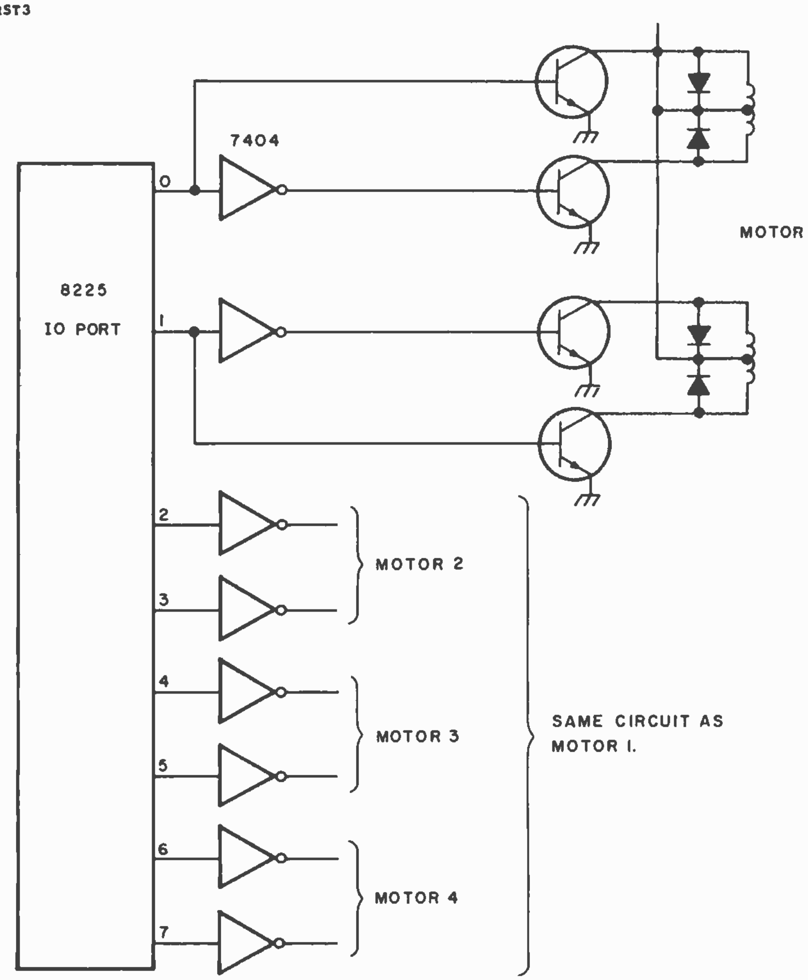


Figure 10: By adding a pair of inverters to each motor drive, four motors can be driven from one 8 bit output port.

(3a)

BINARY COUNT	BINARY COUNT COMPLEMENT
0 0	1 1
0 1	1 0
1 0	0 1
1 1	0 0

(3b)

DESIRED SEQUENCE	DESIRED SEQUENCE COMPLEMENT
0 1	1 0
0 0	1 1
1 0	0 1
1 1	0 0

Table 3: A mismatch exists between the desired pattern of table 3b and the generated binary count of table 3a.

```
HOME1:  LXI H,OUT1
        MOV C,M          ;GET OUTPUT
        MVI B,0B0H       ;LOAD CW, 100 STEPS
HSTRT:  LXI H,PHAS1      ;SELECT MOTOR M1
        CALL MOTOR2      ;MOTOR STEP SUBROUTINE
        JZ LOAD1         ;EXIT IF DONE STEPPING
        MVI A,0FFH       ;ELSE
        STA TIMRS        ;RESTART MOTOR TIMER
LOOP1:  LXI H,HSTRT
        JMP LOOP1        ;WAIT FOR MOTOR
```

Listing 2: An 8080 assembly language mainline program for driving several stepper motors.

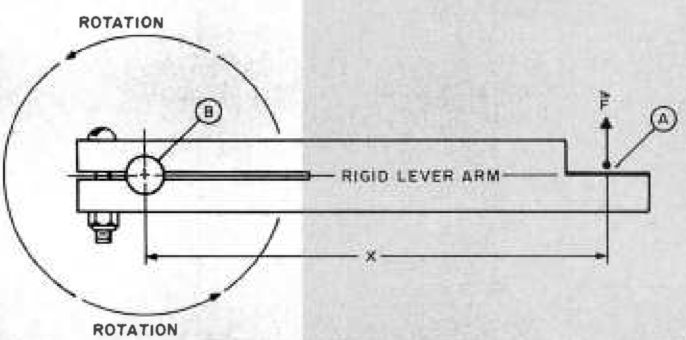
After getting a motor to turn on command, the next challenge is addressing several motors while trying to conserve IO lines. I did this by changing the method of generating the drive pattern. In the first program the entire pattern was stored, rotated and output to the motor. In the second program a different pattern was stored, one that would allow two output wires to control each motor. It was necessary to add a pair of 7404 inverters outside the processor as in figure 10. At first it appeared that two bits could be incremented inside the processor to generate the output signals, but the sequence was wrong as illustrated in table 3b. The problem was solved by storing the four pairs of bits in a register.

Two memory locations are reserved for each motor. The first, PHAS1, stores the rotating bit pattern required to drive the motor. Each time a motor is to be stepped, its pattern will be rotated and output. This register always indicates the last pattern

FUNDAMENTALS OF ROBOTS AND MECHANISMS

What is Torque?

*Torque is the force which produces, or tends to produce, motion about an axis of rotation in a mechanical system. It is a measure of the power of a mechanical output device such as a stepper motor or other device. A specification of the available torque from such an actuator under various circumstances is an essential part of designing it into a mechanical system, just as current ratings are crucial to power semiconductor drive design. A torque is specified in units which reflect its origin as a magnitude of force available at some distance from an axis of rotation. In the English system of measurement, a typical unit is "ounce-inch;" 1 oz-in of torque means that at a distance 1 inch from the axis, a force of 1 ounce would be exerted. In the metric system, a typical unit might be dyne-centimeter (CGS) or newton-meter (MKS) where the metric units of force (dyne or newton) and distance (centimeter or meter) are used. . .*CH



If force F is measured at point A (a distance x from center of rotation B), the torque is given by

$$T = F.x$$

Similarly, if a stepper motor has a rated torque of T, it can exert a force of

$$F = T/x$$

at distance x from its center of rotation.

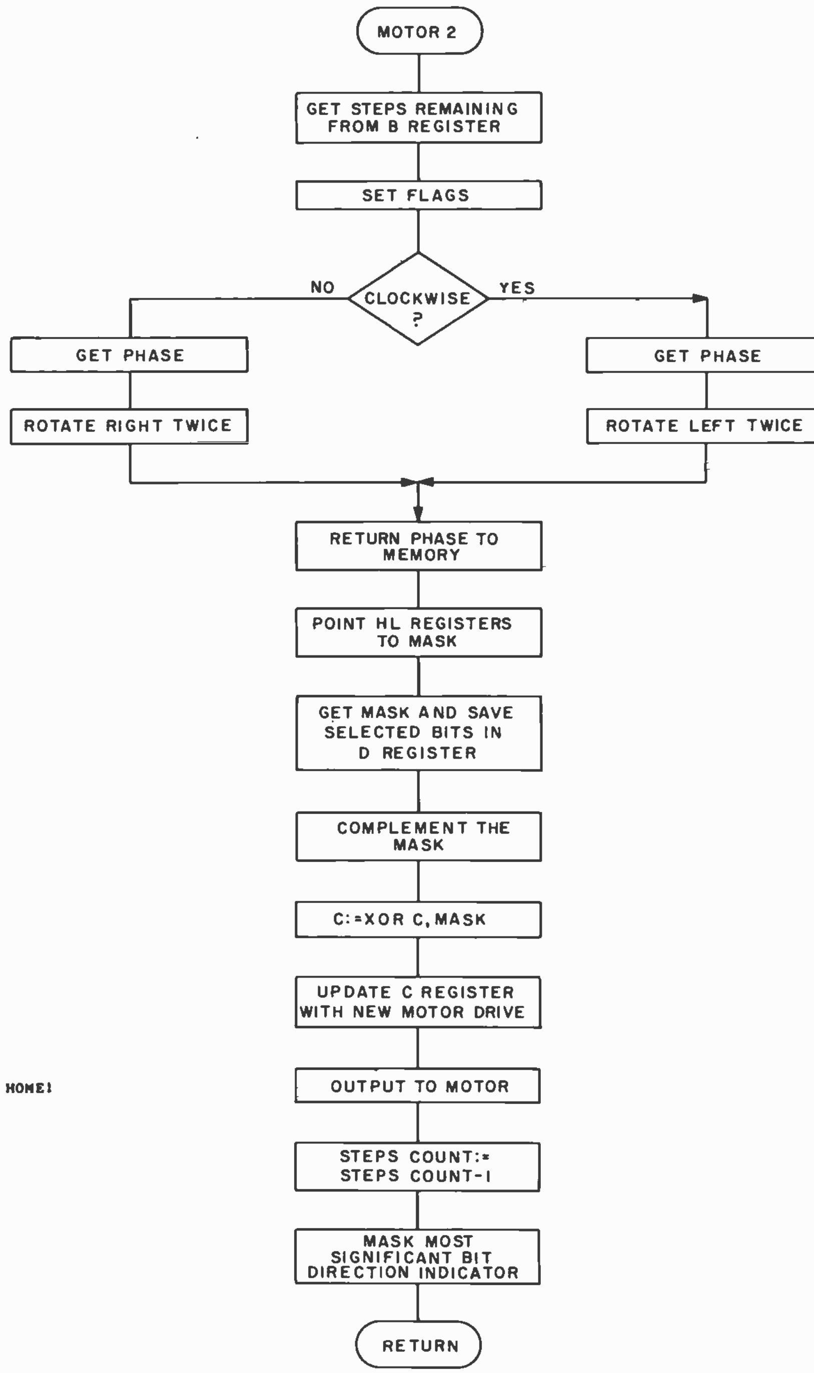
output. The second register, MASK1, contains a mask. For example: mask 0C hexadecimal (00001100 binary) is for the motor attached to output wires 2 and 3, motor 2 in figure 10.

The subroutine must be entered with certain preliminaries taken care of. The HL register pair points to the drive pattern of the motor to be driven. Register B contains the direction of rotation. If the most significant bit is 0 the rotation is counterclockwise; if the most significant bit is 1 the rotation is clockwise. Register B also contains the number of steps to be made. The TMS 5501 was again used as a timer. One motor must complete its steps before another starts.

The flowchart for this second motor drive routine (listing 3) is shown in figure 11. The registers and two memory locations are shown in figure 12 as they look before and after a pass through the subroutine. Remember that bits 0 and 1 of register C are driving the motor of interest. Register D is a temporary storage location and bits 0 and 1 of register D reflect the change in drive pattern. The most significant bit of register B is a 1 indicating clockwise rotation. The lower order bits are the steps count, and are decremented by 1 each pass. The most significant bit is masked out in the last operation. This sets up a zero test and jump upon return to the mainline program, if the count has been completed.

Now let's look at how the subroutine is called. Mainline program HOME1, shown in

Figure 11: The flowchart for the motor drive routine of listing 3.




```
                        ;STEPPER MOTOR DRIVE ROUTINE
                        ;CALLED FROM MAIN LINE PROGRAM HOME1
MOTOR2  MOV A,B
        ANA A           ;SET FLAGS
        JM  CW          ;DETERMINE DIRECTION
        MOV A,M         ;GET PHASE
        RRC
        RRC             ;ROTATE
        JMP MTRJP
CW:     MOV A,M         ;GET PHASE
        RLC
        RLC             ;ROTATE
MTRJP:  MOV M,A         ;PUT BACK
        INX H           ;POINT TO MOTOR MASK
        ANA M           ;MASK OFF MOTOR
        MOV D,A         ;TEMPORARY STORAGE
        MOV A,M         ;GET MASK
        CMA             ;COMPLEMENT MASK
        ANA C           ;GET OUTPUT
        ORA D           ;COMBINE NEW AND OLD
        MOV C,A         ;REPLACE IN MEMORY
        LXI H,OUT1
        MOV M,A
        STA MTROT       ;START MOTOR
        DCR B           ;DECREMENT COUNTER
        MOV A,B
        ANI 7FH         ;MASK OUT DIRECTION
        RET             ;RETURN TO MAIN PROGRAM
```

Listing 3: This is the 8080 assembly listing of a second motor routine. This one is called by mainline routine HOME1. Note that the zero flag is carried back to mainline program to indicate steps done.

Before

	H	L
	Phase 1	Address
D / E	00 00 00 11	
B / C	10 00 01 11	01 01 10 11

Phase 1	10 00 01 11
Mask 1	00 00 00 11

After

	H	L
	Out 1	Address
D / E	00 00 00 10	
B / C	10 00 01 10	01 01 10 10

Phase 1	00 01 11 10
Mask 1	00 00 00 11

Figure 12: The status of the HL, DE and BC register pairs before and after a call to the MOTOR routine.

listing 2, addresses OUT1, the motor drive storage, and puts it in register C. Literal value 0B0 hexadecimal is moved to register B indicating 100 steps clockwise. The HL register pair is pointed to PHAS1 which selects motor 1, and then MOTOR2 is called. When the MOTOR2 routine returns a jump on zero test, J2 jumps over the looping to continue the next part of the main program. If 100 steps have not occurred, the zero flag will not be set in MOTOR2, and the program falls through the test. The motor timer is then restarted. The HL register pair is set to HSTRT and the program loops, waiting for time-out. When time-out occurs, a PCHL instruction vectors the program to HSTRT and another MOTOR2 call.

If you had the time, some extra bytes and several stepper motors, what might you accomplish? The first rotation on command is pretty exciting, but not to anyone but you. There are some useful applications right next to your computer. An XY plotter might be useful in your graphics work. If you aren't handy with mechanics you might modify the paper drive of an analog recorder to provide bidirectional stepper control. An analog to digital converter to drive the pen and a relay to lift the pen between points will give you a reasonable alternative to an expensive plotter.

Other hobbies just cry out with applications. The model railroad buff can control the neatest turntable. A stepper motor in a yard engine would give ultra low speed. A machinist might combine a couple of beefy stepper motors and a lathe to create a simple numerically controlled machine tool. One of the most interesting applications, robots, takes on added dimensions with precise control possible. A variation of this might be to program the strings of a puppet. (That one is going to need lots of program storage.) The radio amateur can automate his or her existing receiver by coupling to the turning knob. Any place where several motions have to be synchronized, stepper motors can be the solution.■

BIBLIOGRAPHY

1. Intel Corporation, *8080 Microcomputer Systems User's Manual,* 1975.
2. Bober, Robert E, "Stepperdrive Circuit Simplifies Printer Control," *EDN,* April 5 1976, page 114.
3. Helfrick, Albert, "8 bit Frequency Source Suited for Microprocessor Control," *EDN,* September 20 1976, page 116.
4. Webster, John and Simpson, William, *Software Design for Microprocessors,* Texas Instruments Learning Center, 1976.
5. Baker, Robert, "Put the 'Do Everything' Chip in Your Next Design," *BYTE*, July 1976, page 40.
6. Shogren, Carl, "The AC/PM Motor: A Drive Whose Time Has Come," *Machine Design*, June 27 1974, page 83.
7. Hollis, Ralph, "Newt: A Mobile Cognitive Robot," *BYTE*, June 1977, page 36 (stepper motor discussion).
8. Guilder, J, "Focus on Stepping Motors," *Electronic Design,* October 25 1977, page 48 (an excellent tutorial).

Where the little guy can get a stepper motor:

Sigma Instruments Inc
Braintree MA 02184

18-1408 $22
18-2013 $23

Minimum billing is $35.

North American Philips Controls Corporation
Cheshire Industrial Park
Cheshire CT 06410

Empire Electrical Company
54 Mystic Av
Medford MA 02155

K82201 $17
K82401 $17.85

American Design Components
39 Lispenard St
New York NY 10013

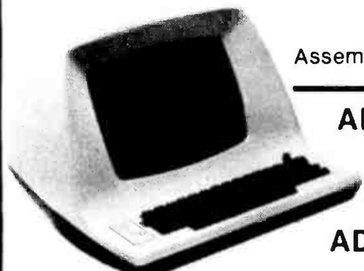
Berger-Lahr Corporation
Peterborough Rd
Jaffrey NH 03452

SYNCHRO-SOUND
ENTERPRISES, INC.

Everything you need in small computer systems!

LEAR SIEGLER ADM 3A TERMINAL

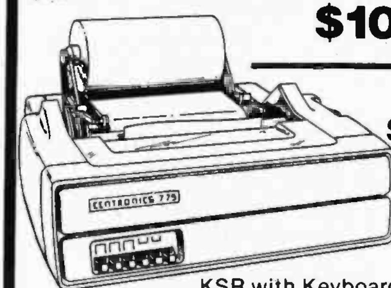
Kit **$739.95**
Assembled **849.95**



ADM 1A TERMINAL
Assembled
$1398.00

ADM 2A TERMINAL
Assembled **$1895.00**

NEW! CENTRONICS 779 PRINTER
$1099.00



703 SERIAL PRINTER
$2395.00

761 PRINTER
KSR with Keyboard **$1695.00**

HAZELTINE 1500 VIDEO TERMINAL
Assembled **$1149.00**
Kit also available

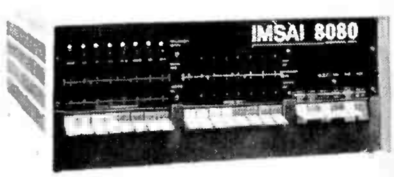
MODULAR 1 INTELLIGENT TERMINAL
Assembled **$1659.00**

OKIDATA MODEL 110 LINE PRINTER

Friction Feed
$1149.00
Tractor Feed
$1279.00
RS 232C Serial Interface
$260.00

MODEL 22 LINE PRINTER
Tractor Feed **$2249.00**

IMSAI 8080 MICROCOMPUTER

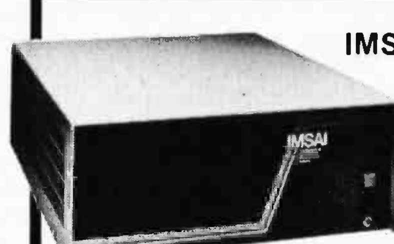


With 22 slot
$649.95

TELETYPE MODEL 43 PRINTER
$1299.00

SPECIAL BUYS

IMSAI Double Density Persci Drive	$2295.00
Sorac IQ 120 Video Terminal	959.00
DEC LA 180 Printer	2769.00
Compucolor 8001 Color Computer	2595.00
ICOM Microfloppy System	989.00
North Star Microfloppy Disk Kit	599.00
Cromemco Z-2 Kit	565.25
Javelin 9" Video Monitor	159.95
Micropolis Model 1053MOD 2	1799.00
IMSAI AP44-44 Col. Printer Kit	329.00
TDL Xitan Alpha 1 Computer Kit	699.00
ICOM Model FD 3712	2795.00



IMSAI PC S-80/15 8085 COMPUTER
Kit
$749.00
Assembled
$929.00

We carry a full line of the following: TDL, Centronics, Seals, Hazeltine, Micropolis, Hayden, IMSAI, Cromemco, Compucolor, Icom, Lear Siegler, Okidata, DEC, Javelin, North Star, Peripheral Vision. Same day delivery and shipping on most items. Full modern repair facilities on premises for complete servicing of everything we sell.

SYNCHRO-SOUND ENTERPRISES, INC.
The Computer People
193-25 Jamaica Avenue,
Jamaica, New York 11423
212/468-7067 TWX: 710-582-5886
Hours 9-4 daily and Saturday
Visit our new showroom
Working units on display
Dept. BBB BankAmericard • Master Charge

Circle 95 on inquiry card.

A Minifloppy Interface

David Allen
Electronics Consultant
1317 Central Av
Kansas City KS 66102

Floppy disk drives have been around for some years now; the basic technology of such drives is well proven and the drive designs seem reliable as mass storage for small systems. Having used the standard size floppy for some time, my first reaction to the introduction of the "minifloppy" was to view it as a cute gimmick, since the minifloppy's price had not then dropped in proportion to its performance.

However, after using the minifloppy for a while and having seen the inevitable dropping in price as production expands, I am becoming much more enthusiastic. The reduction in bit rate will make it easier to interface, and the reduction in bit density should make it slightly more reliable in small user environments. Its performance, while reduced, is quite adequate for many applications, especially when its price is taken into account. [*One personal computing manufacturer, for instance, markets a dual drive peripheral for their systems at a total of $1000, which is hard to find in a dual drive standard size floppy disk...* CH] Its small size and relaxed specifications allow room for more cost cutting than the full-size design. Competitive technologies like bubble memories are perhaps several years away from equivalent costs per bit. It is thus quite appropriate to give serious consideration to the small floppy.

Shugart was first to arrive on the market with a small floppy, but is no longer alone there. Wangco has a competing drive in production. BASF has a drive which was displayed at the 1977 NCC show in Dallas; it is reportedly just entering production as of fourth quarter 1977. Micro Peripherals Inc has a minifloppy drive scheduled for production at this time. A notable feature of the drive is its use of a band driven head positioning mechanism designed to improve track-to-track access time by a factor of five or six. Pertec has also announced a drive, and Radio Shack is reported to be working on a low cost version for use with their Z-80 based microcomputer system.

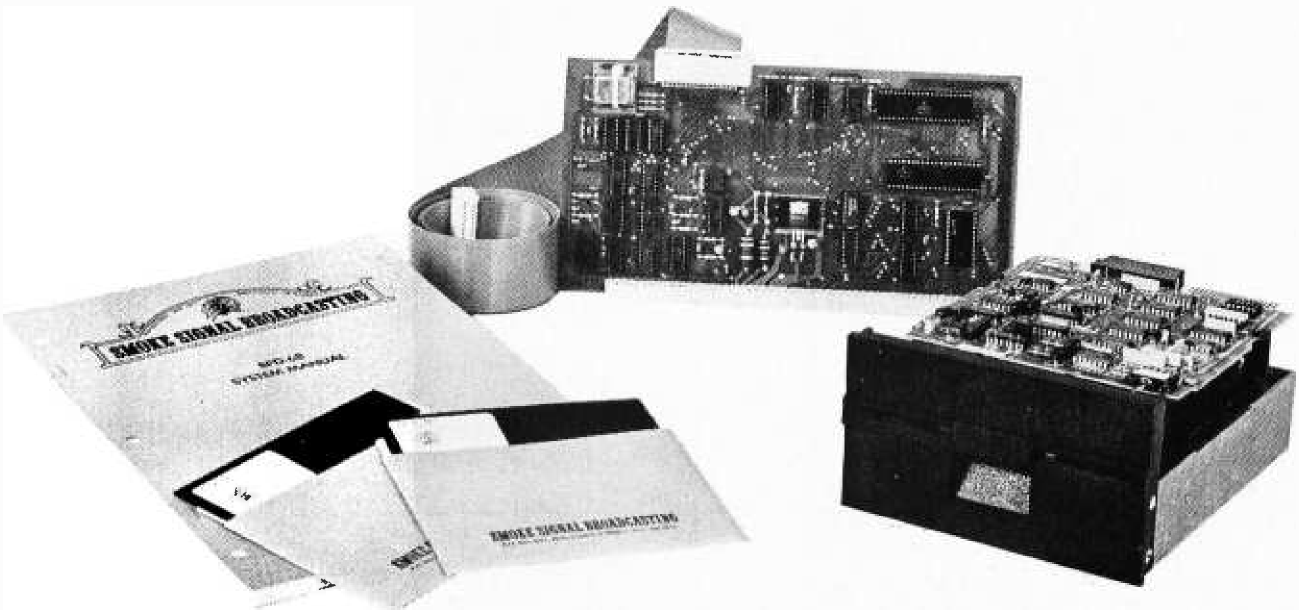
Fortunately, Shugart's interface configuration has been copied in the Wangco and BASF drives right down to the use of the same connectors and pin assignments. This makes Shugart's interface a defacto standard, as well as a great improvement over the diversity experienced in full-size floppies. I hope the present plug compatibility will continue.

Wangco and others are advocating an ANSI (American National Standards Institute) standard specification that includes 40 tracks (as opposed to Shugart's current 35 track maximum). Both the Wangco and BASF drives write on 40 tracks. The first 35 tracks are positioned in the same way as the tracks in Shugart's drive; the last five tracks were located closer to the center of the diskette than Shugart apparently thought safe. If the 40 track approach does get ANSI acceptance, though, it is reasonable to expect Shugart to make a 40 track version.

How Small Floppies Differ from Full-size Units

Several significant differences exist in the interfacing required for the small floppies versus the large floppies. One profound difference is that the former are all powered by DC motors. (AC motors are being investigated as an option by one manufacturer.) This allows the motor to be powered down during long periods of nonusage. The power saving is such that battery powered operation is realistic; the fully powered Shugart drive uses only 15 W total (18 in Wangco's), and a sector can be read in less than two seconds of motor operation. This suggests usage in data logging applications, traditionally the province of cassette drives. More noticeable to a large floppy user is the fact that the Shugart minifloppy stays almost stone cold during operation. (My Memorex 651 got so hot recently due to the combined heating of the step motor and hysteresis synchronous drive motor that the pressure pad adhesive decomposed into a sticky goo, allowing the pressure pad to slide off center and causing the first hard errors since the 651 large floppy was interfaced to my

THE SSB $150 FLOPPY DISCOUNT



Affordable

The tribe at Smoke Signal Broadcasting took our BFD-68 disk system and scalped the price, but not the features to create the ABFD-68 (Affordable Basic Floppy Disk). We appreciate the fact that the computer hobbyist gave us our start and we haven't forgotten you.

$649 Assembled

Compare Price. Our SS-50 bus compatible disk system is $150 less than the assembled price of the leading S-100 disk system. And you can at least double that savings when you buy one of the computers manufactured by MSI or SWTPC that use the superior 6800 microprocessor.

Programmable

The BFD-68 is well known for its fine software. The system comes with the best disk operating system available and we offer a multitude of other compatible software products. These include a BASIC interpreter with disk file handling capability. By the way, our DOS now easily handles true random access files as well as sequential. Also, we have a super fast BASIC compiler for business applications. In addition, a Text Editor, 2 Assemblers, a Trace Disassembler useful for program debugging and an Object to Source Code Generator are all stock items available for immediate delivery.
A word processor will be available very soon.

Reliable

We delivered our first mini-floppy disk system a year ago — 6 months ahead of any other 6800 based mini system. Thus, we've had twice the experience in building reliability into the system. Our NEW disk controller was designed using all we have learned in the past year about system reliability.

The ABFD-68 contains all the built in reliability of our regular BFD-68 plus you save money by supplying your own cabinet and power supply for the disk.

Available

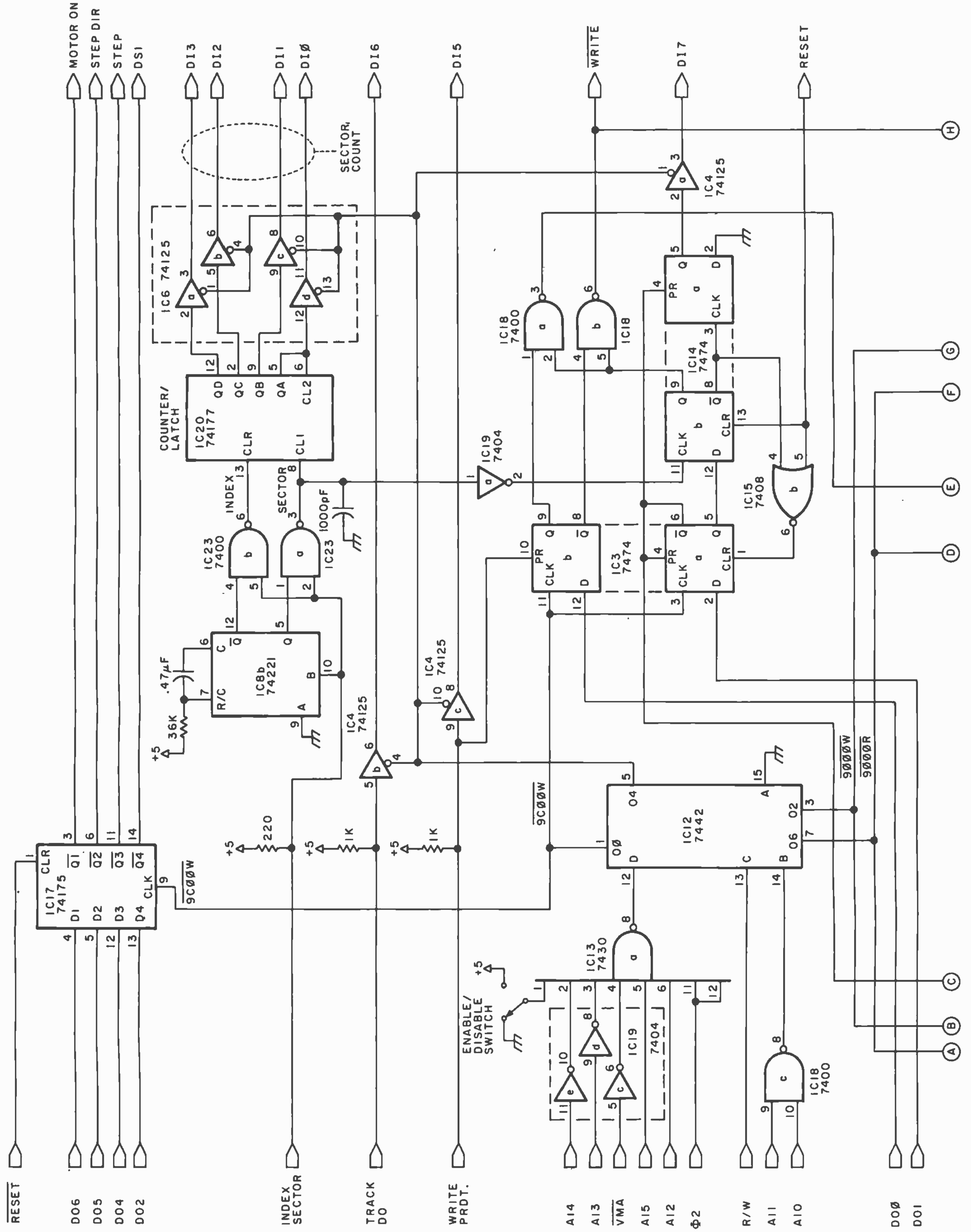
We've shipped literally tons of our BFD-68 disk system in the past year and have learned to keep our production up with demand. Give us a call and chances are we'll be able to ship you the new ABFD-68 from stock and charge it to your Master Charge or Visa card. Better yet, ask us for the name of the computer store nearest you that carries our complete line of computer products.

SMOKE SIGNAL BROADCASTING

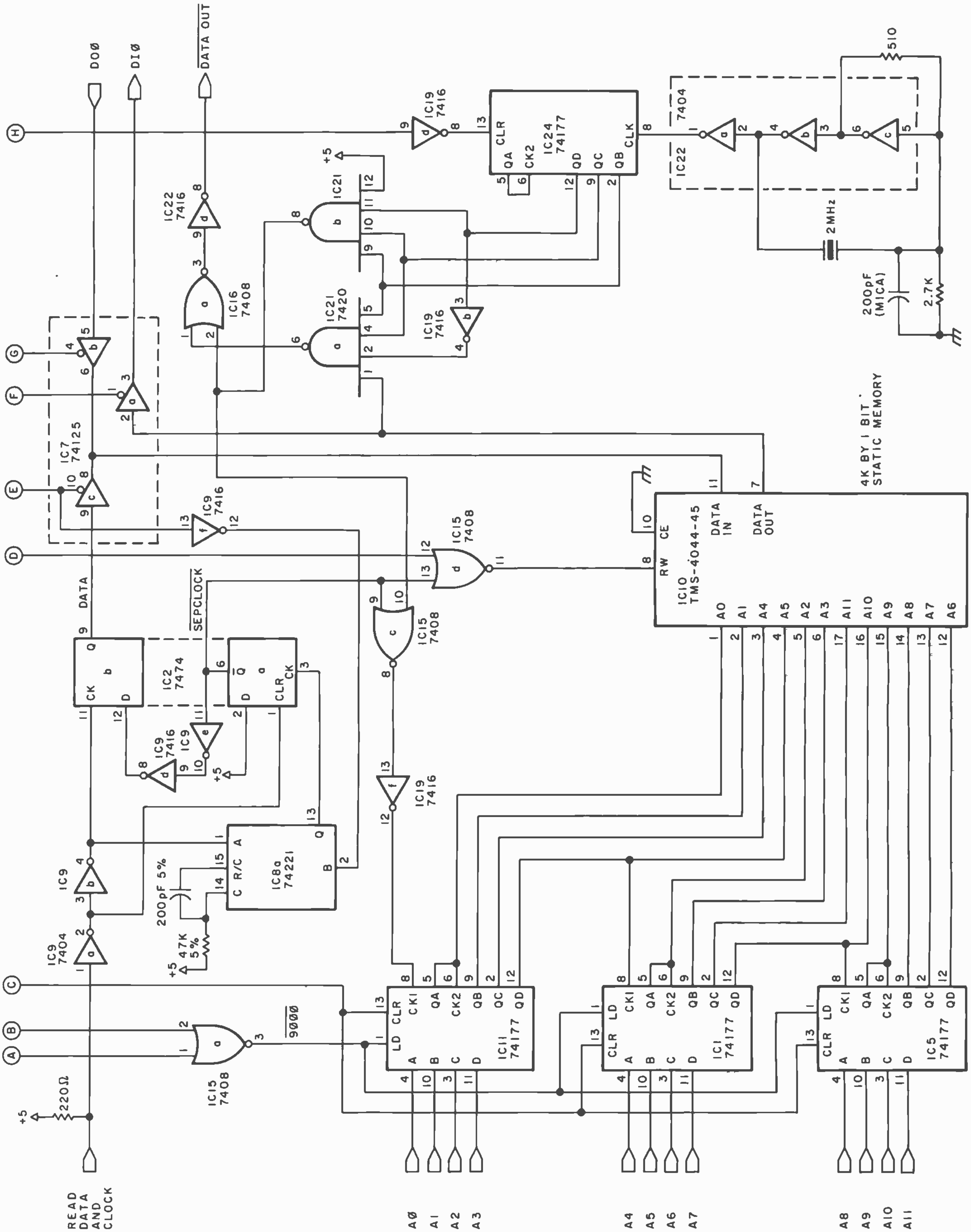
P.O. Box 2017, Hollywood, CA 90028 • (213) 462-5652

Circle 105 on inquiry card.

Figure 1: The author's minifloppy interface circuit. This circuit uses a 4 K bit buffer memory, IC10, as a key element of its operation. For output, this memory is addressed one bit at a time by writing to addresses 9000 to 9FFF of memory address space thru counters IC1, IC5 and IC11. Bit 0 (low order) of the word written to memory is stored in the appropriate bit. For



input, data from the disk drive is serially stored into the memory with the clock derived from the input signal advancing the address counter. Then after a physical read operation has defined the contents of the 4 K memory part, it can be read by loading and testing bit 1 of each location from address 9000 to 9FFF.



6800 developmental system.) The Wangco 82 warms noticeably during continuous operation (probably due to its slightly larger step motor), but this is not objectionable. The lack of heat should contribute significantly to long-term reliability in the small floppies.

Fortunately, the DC motor chosen for use by both Shugart and Wangco is not a cheap cassette recorder motor. A tachometer, integral to the motor, provides feedback to hold the speed constant. Increasing the motor's load (eg: thumb on flywheel) causes no appreciable change in motor speed as perceived on a strobe disk, and no data errors. The service life of the motor is quoted as 8000 hours, an order of magnitude better than most cassette recorder types. At two seconds or less per data transfer, the motor should last almost indefinitely.

The motor is turned on by introducing a TTL zero (low) logic level on one of the interface lines. Shugart and Wangco both recommend a one second delay for the motor to come up to speed following turn on, and Shugart suggests a one second oneshot to signal readiness for data movement. I provide this delay in software where necessary. Since many of my operations are manual (ie: keyboard entered disk operating system commands), inherent delay in usage masks the motor's startup latency.

Power for the unit is simplified by the lack of any negative voltage. Only +12 VDC at approximately 1 A and +5 VDC at approximately 0.5 A are required.

Spiral Steps

The Shugart and BASF drives differ from full-size floppy units by not using a lead screw to drive the head carriage assembly. Shugart uses a disk with a spiral groove, rotated by a step motor; the head carriage rides in the groove on a single ball bearing. This can be the source of confusing problems during improper operation, since no stops are provided at the track 00 and track 35 points. The head carriage is free to run out of the groove somewhat like the needle on an old record player when the song is over. Once the groove is lost, numerous step pulses must be issued until the groove is again found and track 00 accessed. During proper operation, however, this confusion should never arise.

The small floppies use a track 00 switch to indicate head position over the outermost track, as do the full-size floppies. Except for the "lost groove" problem described above and timing of the step pulses, hardware and software techniques used on full-size floppies are directly applicable to the small floppies.

Recognition of Sectors

In the Memorex units, the INDEX pulse is already separated from the SECTOR pulse. This is not the case with small floppies. It is a simple matter to add a oneshot and logic gates to extract the INDEX pulse from the combined pulse train, but this circuit must be provided by the user.

The number of sectors for hard sectored systems may be either ten per track or 18 per track and is solely a function of the diskette used.

Speed

The minifloppy data transfer rate is half that of most large floppies. This is due to the reduced bit density recorded on the media, and to the reduced diskette rotation speed (300 rpm). The extra delay due to increased rotational latency is quite noticeable to the user.

Extracing the Data

Similar to the SECTOR and INDEX pulse situation above, the user of the small floppy must also separate the data which is read from the clock pulses. The Memorex and other full-size drives provide these two signals individually, but the small floppies don't.

The circuit which the user must provide to accomplish this separation may be elegant or simple, depending on the user's needs and tastes. Shugart complicates the data separation problem by discussing bit position shifts (in time) due to adjacent ones or zeros. They recommend different time windows depending on whether the previous bit cell was 1 or 0. This technique is intended to provide greater reliability. I did not use either of Shugart's recommended circuits; I also ignored the bit shift problem. The circuit used (see figure 2) has a single 6 μs oneshot and defines the data window somewhat differently. (More on this later.) The reliability of this circuit is such that the drive never makes hard errors and rarely (less than one per day) makes soft errors. In double density operation the more elegant approach to the problem of data separation would undoubtedly be needed. Fortunately, the low bit density of the small floppy allows me to take a relatively lax approach.

Shugart versus Wangco

Both the Shugart and Wangco minifloppies pass data with excellent reliability.

Note: *Since this article was written, author Allen notes that both Shugart and Wangco have redesigned portions of the printed circuit boards and mounting hardware. The descriptions on this page are based on early models of the designs.*

Significant differences do exist between the units, but they are largely mechanical; it is difficult to fault either unit for its electrical performance, in single density operation at least.

The most obvious difference is Wangco's use of a conventional lead screw for head positioning, as noted earlier. This makes the Wangco drive look like a miniaturized version of a standard drive.

On Wangco's drive, all parts, including the motor controller, are mounted on a single printed circuit board. Wangco uses the Fairchild 7391 motor controller integrated circuit, while Shugart employs its own motor controller using a 741 op amp plus discrete components on a separate printed circuit board. Both Wangco and Shugart use the same Buehler DC motor and tachometer.

Wangco has a second set of index and write protect LEDs already installed in the drive that allow recording on the flip side of standard diskettes. Shugart uses a microswitch to sense write protection, and there appears to be no provision for addition of either a second switch or second sector LED.

Wangco's printed circuit board is anchored by four standard 4-40 machine screws. The body of Wangco's power connector is mounted parallel to the plane of the printed circuit board facing rearward and anchored to the board with a nylon tie wrap. On the Shugart drive, the main board is gripped loosely at the edges by four spring clips which are held in place by Tinnerman style nuts over plastic posts. The board is susceptible to being pulled loose from the frame. Shugart's power connector is mounted at right angles to the board with the lower rear corner facing inward. If the power cable is given a good

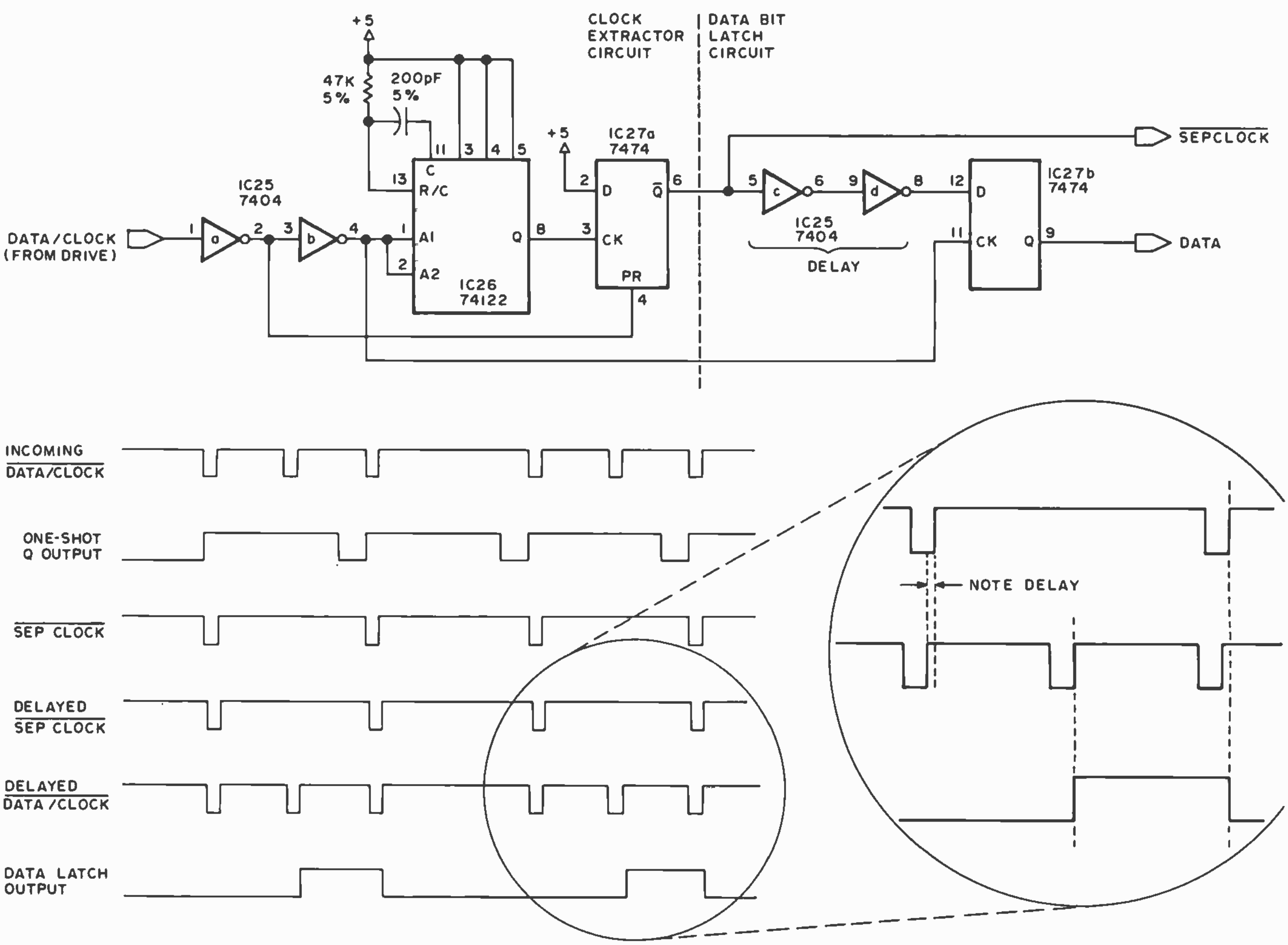


Figure 2: A circuit for separating data pulses from clock pulses in the author's system. Data and clock pulses arrive mixed together at the input to IC25a. IC26, a oneshot, provides an output pulse to the data bit latch, IC27a, in such a manner that only clock pulses are outputted at $\overline{\text{SEPCLOCK}}$. The timing diagram shows the effect of intentional propagation delays (exaggerated here for purposes of illustration). The delayed $\overline{\text{DATA/CLOCK}}$ signal clocks data flip flop IC27b off during clock pulses because the clocking occurs during the time of the delayed $\overline{\text{SEPCLOCK}}$ low level.

Table 1: Power wiring table for figures 1 and 2.

IC Number	Type	+5 V Pin	Gnd Pin
IC1	74177	14	7
IC2	7474	14	7
IC3	7474	14	7
IC4	74125	14	7
IC5	74177	14	7
IC6	74125	14	7
IC7	74125	14	7
IC8	74221	16	8
IC9	7404	14	7
IC10	TMS-4044-45	18	9
IC11	74177	14	7
IC12	7442	16	8
IC13	7430	14	7
IC14	7474	14	7
IC15	7408	14	7
IC16	7408	14	7
IC17	74175	16	8
IC18	7400	14	7
IC19	7404	14	7
IC20	74177	14	7
IC21	7420	14	7
IC22	7404	14	7
IC23	7400	14	7
IC24	74177	14	7
IC25	7404	14	7
IC26	74122	14	7
IC27	7474	14	7

jerk, the board could be dismounted and the corner of the board possibly broken off at the power connector.

The two units are indeed plug compatible. The controller described below, initially designed for the Shugart drive, passed data to and from the Wangco unit with no errors and no wiring changes. One minor difference in interfacing is that Wangco uses pin 2 of the interface connector as a fourth drive-select line; pin 2 is unused on the Shugart unit and only three Shugart units can be paralleled without making cabling changes.

The Wangco unit's head would not load when first powered up in my demo unit. It moved freely with finger pressure when power was off, and even pulled in with power on when given a little assistance. Minor adjustment might have been in order, but I thought that perhaps gravity could provide the needed assistance. This was verified when the unit was turned 180° on its side and it began loading and working flawlessly. Head load solenoids have been a weak point of floppy disk drives at least since the Memorex 651.

The Prototype Controller

In order to get the minifloppy working with the least fuss, I adapted the original controller designed for the Memorex 651 to the minifloppy (see figure 1). This permitted use of all existing disk-based software with very few changes (see "A Floppy Disk Interface," page 58, January 1977 BYTE).

The controller uses the same 256 byte hard sectors, buffered in the interface card, as before. Today, 256 bytes is unnecessarily small. Programmable memory prices have dropped sufficiently to warrant the use of a 512 or 1024 byte buffer, which could significantly increase the apparent speed of the disk. Logistics, not economics, dictated the continued use of the 256 byte sector. The hardware and software can be easily changed to incorporate a larger buffer.

The similarity of the small floppy controller to its parent will be readily apparent if the schematics are compared. A reduction in chip count results from the smaller number of sectors which saves a counter and bus driver, and from the use of a single 4 K static programmable memory in place of the separate 2102s of the parent design.

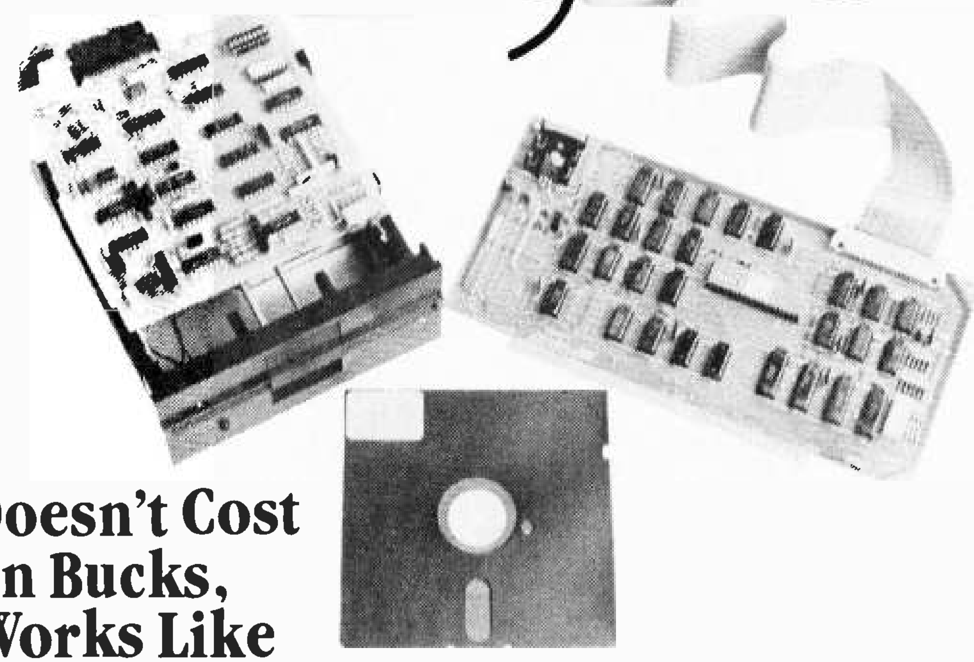
Two separator circuits are built around a 74221 dual oneshot: one for sector and index pulses and the other for the received data and clock pulses. The data and clock pulse separator circuit is really just a variation of the sector and index circuit. In each, a missing pulse (either an index or data pulse, depending on which circuit) is being sought.

Sector and Index Separator

Sector pulses are consistently present, occurring at regular intervals in time. Since the anticipated index pulse will occur approximately midway between adjacent sector pulses, the sector pulses are used to define the position of a sampling window. The oneshot is used as a gating signal to strip the index pulses out of the sector pulse train.

In the sector and index pulse separator, the window begins at the trailing edge of each input pulse. The presence of the window (ie: oneshot fired but not yet run out) enables a gate which will then pass any

$1,000,000



No, It Doesn't Cost a Million Bucks, It Just Works Like It Did!

The MSDD-100 Floppy Disc System offers you an inexpensive, modern way to get real data processing power from your S-100 Bus Computer System. Disc storage is a must in every microcomputer. With the MSDD-100 system, this power doesn't cost much more than cassette, and is far faster.

For business applications, disc storage is an absolute must. The MSDD-100 has enough bulk storage capacity to handle hundreds of inventory entries, yet has enough speed to get you to any of these records in just a second or so. That's power!

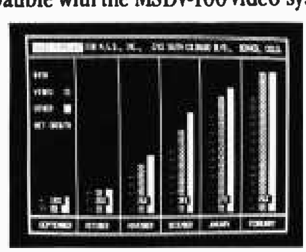
Computer programmers know the advantage gained with disc storage. Program development is speeded up greatly. With a disc system, you spend your time writing software, not waiting for slow peripherals such as paper tape. For example, paper tape can be handled at about ten characters per second. Cassette tape runs at maybe thirty characters per second or so. By comparison, the transfer rate with the MSDD-100 Disc System is 15,000 characters per second. That's fast!

You can easily put over 90,000 characters of data on a single diskette, about 5" in diameter. The MSDD-100 controller will allow you to hook up to three separate drives onto your system. That's over a quarter of a megabyte on line and accessable in a second or two. That's storage.

For $599, you get the unique, modern MSDD-100 Disc Controller Card, the industry standard Shugart SA-400 Floppy Disc Drive, a comprehensive manual, system utilities software and patches into altair basic, and a preassembled cable. We offer fully integrated 8K and extended basic languages at nominal extra cost. That's MSD!

MSD Gets Down to Basics.

Microsoft, the company that created the popular, widely used altair basic, wrote MSD Disc Basic, 8K and extended versions. Microsoft basic is well know for its power and speed. MSD bings you this language for the MSDD-100 Floppy Disc System. MSD offers the first really compact disc basic, our 8K version, and extended basic, with powerful features such as double precision number processing, formatted data output, multiple data terminals, and most efficient memory usage. This fine software is now available to every MSDD-100 user. MSD basic is also fully compatible with the MSDV-100 video system.



The MSDV-100 video display system is a powerful, S-100 compatible video output device. With a high resolution 80 x 24 format. The small system user can now have a sophisticated video display system comparable to the most expensive of video terminals. You don't have to settle for less anymore!

The MSDV-100 Video Display System is designed around a custom ACSII character set. Vertical lines, horizontal lines, and line intersections are part of its well thought-out repertoire. The upper and lower case alphanumeric characters are of the highest quality. The MSDV-100 is a memory mapped, two board device that can be fed into a standard video monitor. Thoroughly developed software is included, with a wide selection of drivers supplied in both basic and assembly language. With MSDV-100 driving software, the MSDV-100 becomes an elegant, sophisticated video terminal with scrolling and cursor control. Full compatability with MSD basic, of course!

MICRO SYSTEMS DEVELOPMENT
MSD

2765 So. Colorado Blvd. Suite 110 Denver, CO 80222 (303) 758-7411

Sanyo Monitor (VM4209)	$150
Micro-Floppy Disc System	$499*
(Assembled)	$599*
Video Display System	$285
(Assembled)	$385
Additional Drivers	$350 ea.
Diskettes	$4.25 ea.

*Power Supply not included.

Micro-Floppy Disc System	$599*
(Assembled)	$699

Price Increases Effective Feb. 1

To place Order, send check, money order or BA or MC Card # with exp. date and signature. Uncertified checks require 6 weeks processing. Phone orders accepted.

Please Send me the following: AMOUNT

TOTAL: ____

Name ____

Address ____

City, State, Zip ____

☐ Send me more information

Circle 80 on inquiry card.

pulse so long as the window is still open. The window closes before the next input pulse, so only an index pulse, if present, can pass through to the index output. The complement output of the oneshot is also used: if the window is open for index pulses it is thus closed to sector pulses, and vice versa. Thus, the same window which keeps sector pulses out of the index pulse line also keeps index pulses out of the sector pulse line.

Double Density Operation

Much of the interest surrounding floppy disks, large or small, centers on double density operation. Various schemes exist for recording twice the usual amount of data on each sector. This possibility stems from the fact that the FM encoding used on the original floppies is inefficient in the number of flux changes used per data bit. By switching to a more efficient encoding technique (eg: MFM, M^2FM, Modified-Miller, GCR; see *IEEE Spectrum*, July 1977) twice the amount of data can be recorded on each sector with little or no increase in the number of flux changes. The new encoding techniques are a mixed blessing, however, since their bandwidth requirements are different from FM, their tolerance of the "bit-shift" phenomenon is different, and they require a more complex data separator and decoder. FM encoding is still the easiest, cheapest, and most reliable technique.

Of the alternative codes used to achieve double density, GCR (Group-Coded Recording) looks quite attractive. Micro Peripherals Inc has implemented double density using GCR in a full size floppy disk and controller system currently being marketed. (For an alluring, albeit incomplete synopsis of GCR, see *Computer Design*, December 1976 or *Perkin-Elmer Data Systems News*, June 14 1977.) GCR is nothing more than the old standby NRZ with its attendant advantages, but, since ordinary NRZ has no clocking information and a potentially high DC content during long strings of ones or zeros, the data is reformatted to eliminate the long strings. The reformatting converts each four bit group of original data into five bits of group coded data; the five bits in the encoded version will *always* have a mix of ones and zeros, even if the real data is all in one state. Reformatting in GCR can be accomplished in software, as opposed to MFM, etc, which almost unavoidably must be encoded and decoded in hardware. Thus, GCR has good possibilities as a low cost, high reliability scheme for achieving double density.

Data and Clock Separator:

The FM data and clock separator used in this controller is considerably simpler than Shugart's recommended circuit. It evolved from an understanding of two basic functions which must be provided by any such separator:

1. extraction of clocking information
2. latching data and holding it long enough for transmission to the using system

Shugart's use of the oneshot mixed these two functions together, and complicated a simple task. The oneshot should be used only for the purpose of clock extraction; use of the oneshot to provide a window for data taking will result in reduced tolerance to bit shifts. The circuit of figure 1 shows the clock extractor and data bit latch separately.

The clock extractor uses the oneshot to strip any data pulses out of the data and clock pulse train. The oneshot's time interval extends from the leading edge of the clock pulse past the trailing edge of any data pulse which might appear within the bit cell. The oneshot will then be triggered only by clock pulses, and will likewise set the clock flip flop at each clock pulse's leading edge. The clock flip flop will be reset promptly at the trailing edge of the incoming pulse. The inverter, IC25b, provides propagation delay to help insure that the clock flip flop can be set by the oneshot. The output of the clock flip flop is a train of clock pulses (no data pulses) which are synchronous with and slightly delayed (by a few nanoseconds) from the incoming clock pulses.

These derived clock pulses are subsequently used in the data bit recovery process. The window during which a data bit might appear is ideally described as the interval between the trailing edge of one clock pulse and the leading edge of the following clock pulse. The data bit latch, IC27a, is therefore set by the trailing edge of any pulse other than the clock pulse. Although both data and clock pulses are present at the clocking input to the data flip flop, it discriminates against clock pulses because the derived clock pulses are present at its data input. The dual inverters, IC25c and IC25d, provide propagation delay which facilitates discrimination against clock pulses. Thus, any pulse that is not a clock pulse will set the data bit latch flip flop and be held until the trailing edge of the next clock pulse.

Other Functions

Since the derived clock pulses are approximately 1 μs wide, they can be used directly as write pulses to store the data bits into the programmable memory. The propagation delay mentioned above also provides a slight data hold time which insures that the data will be stable at the programmable memory's input throughout the duration of the write pulse.

ICs 3a and 3b serve, as in the previous design, to synchronize the reading and writing operations with the leading edge of the sector pulses. As before, it is up to the host processor to request a read or write transaction one sector in advance to allow the controller to take control at the appropriate time; ie: if the host processor wants to write to sector 3, it must request this sometime during the reading of sector 2 so that the controller will be set up and ready when sector 3 rolls around.

Software

I have modified the software, which was previously developed for use with the Memorex 651 large floppy, for use with the small floppies. The most notable change is in the number of sectors per track, which is now ten for the small floppy with the SA-107 type media. Since the sector size of 256 bytes is unchanged, no radical changes were necessary in the original software. The software still fits into a 1 K byte read only memory when used in conjunction with a Motorola MIKBUG system. When used on nonMIKBUG systems, an overhead of 100 or so bytes will be incurred to support the character printing and receiving routines.

Summary

A small floppy in conjunction with the controller of figure 1 represents perhaps the cheapest and easiest way to add a floppy disk to a small system. 22 common TTL integrated circuits and one MOS integrated circuit memory (which is second sourced and should be readily available) are used. The controller requires no adjustments providing that suitable quality components are used. Sector buffering on board is again used to facilitate independence of any particular processor or system configuration and permit concurrent interrupt handling where desired.■

Copies of the software for the interface are available from the author (complete assembly listing only, no object tapes) for $10. Persons interested in a printed circuit board or complete kit, single or double density, are invited to inquire about Altair (S-100) and SwTPC bus compatible versions of this controller. Please address all correspondence to me at the address shown at the beginning of this article.

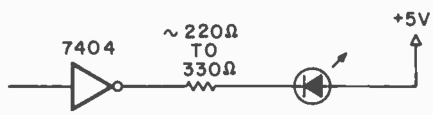
Ask BYTE

TTL TO DRIVE LEDs?

Are TTL integrated circuit devices able to drive LED display devices with just a current limiting resistor?

M V Amiot
3R Marcel Doret
92140 Clamart FRANCE

Yes, it is done quite frequently both in production and in homebrewers' circuits. The following is a typical configuration using a 7404.



Note that the critical specification to consider is the low level output current, noted in the Texas Instruments TTL Data Book for Design Engineers *as* I_{OL}. *Most TTL integrated circuits can sink a current of 16 mA, corresponding to a fanout of 10 unit loads at the inputs of other TTL gates. Using one LED with dropping resistor and a 16 mA current would be within nominal specifications over the entire temperature range of the part; in fact, however, ratings are conservative and an upper bound on the actual LED drive current possible is the short circuit output current of the TTL gate, typically about 50 mA. (Manufacturers do not recommend shorting more than one output of a package at a single time.) What this means is that by using something less than a short circuit level of current it is possible to drive perhaps 20 to 30 mA and get a brighter display than that provided by the nominal 16 mA.*■

WHAT DOES REFRESHING FROM MEMORY MEAN?

Since APL uses scads of memory, and the most drastic reduction in memory price ($1495 for 64 K from Extensys Corporation) is for a dynamic memory, a crucial problem for APL would be its use with a nonstatic memory.

Allen Atwood's article in August 1977 BYTE (page 108) says "One would not want to refresh a display from memory using APL." Why wouldn't I? At these new low prices, I am very much interested. Is this memory refreshing difficulty just for the 8080, which Mr Atwood's article is about, or for other microprocessors also?

Zilog's technical manual for the Z-80 states that the Z-80's Memory Refresh Register "enables dynamic memories to be used with the same ease as static memories." Extensys says that its memory board has "complete dynamic refresh logic." My question is, can APL be properly implemented on the Z-80, using the much cheaper dynamic memory?

Henry Williams
4323 Gleneste-Withamsville Rd
Cincinnati OH 45245

Your question is due to two different uses of the concept of "refreshing" something. In the context of dynamic memory systems, refreshing refers to the technique of assuring that every memory region of the chips is referenced repeatedly with a certain minimum frequency of reference. This dynamic memory refreshing requirement is invariably satisfied in hardware, whether on the memory board itself as in many of the available memory boards or by logic built into the processor design such as the refresh algorithm of the Z-80.

Allen Atwood was referring to a different concept, namely refresh of visual displays, and in particular the classes of displays which require explicit programming to generate their data on a continuing basis. Typical classes of displays which require continual programmed refresh include vector displays and point displays. (For an example of vector displays see Steve Ciarcia's article on page 78 of November 1976 BYTE; another example is provided by "The Beer Budget Graphics Interface" of Peter Nelson, seen on page 26 of November 1976 BYTE, and used for the output of Dave Kruglinski's program described in "How To Implement Space War" in October 1977 BYTE, page 86.) In all such programmed display refresh techniques, assembly language on the typical processor is barely fast enough for flicker free images of reasonably complex pictures. Use of any high level language interpreter (including APL, BASIC, etc) is more than likely to be too slow. But as in the case of dynamic memories, many display techniques involve hardware refresh, in which case the speed of programmed refresh is not a question.

In short, since dynamic memory refreshing is a hardware task, there is no memory refreshing difficulty with respect to APL, or any other language. An important point related to this is the fact that as far as APL or any other high level language is concerned, there is not one bit of difference between a static and a dynamic volatile programmable memory technology: both forms of memory lose all data when power is removed; both forms act as a main memory resource to the computer. It is true that dynamic memory must use "hidden refresh" schemes to be truly equivalent to static memory in all aspects including access time and wait *states; but this difference only has significance if you are trying to use programmed timing loops and the processor's crystal clock to make measurements of time. It is true that static memory has a higher parts count (32 chips for 16 K bytes static versus 8 chips for 16 K bytes excluding refresh logic for 16 K bytes dynamic). It is also true that static memory consumes perhaps twice as much power as the equivalent dynamic memory. But these factors are not major ones in many user oriented situations: it is memory capacity which counts most in the choice of the product, not how the memory is implemented.*

In choosing a personal computer system, the choice between static and dynamic has all the functional distinctions for the user of the choice between square headlights and round headlights in a car: both work, both perform their functions. . .CH■

BYTE's Bits

The Future of Personal Computing at COMPCON 78

The personal computer industry is just about three years old; in fact, the first personal computer was introduced in January 1975, by MITS Inc, Albuquerque NM. Subsequently, an entirely new industry has appeared, including hardware manufacturers, software specialists, retail stores, trade publications and computer trade shows. Personal computers have already had a profound impact on hobbyists and industrial users. With the development of new application software, microcomputers are rapidly being adapted for business, professional and educational uses. In just three years, personal computing has been placed within the reach of every consumer with as yet undetermined and possibly far reaching consequences.

With these points in mind, COMPCON 78, Jack Tar Hotel, San Francisco, February 28 thru March 2 1978, will present a look at the phenomenon of personal computing. Four panel sessions have been arranged with experts who will be discussing various aspects of the computer revolution. These panel sessions start at 7 PM and cover topics such as "Women's Contributions in Innovative Computer Applications" on Monday, "Robotics and Bionics" on Tuesday, "Editors of Computer Magazines" on Wednesday, and "Computer Art and Music" on Thursday. Each session is arranged to provide a broad spectrum of end users with the opportunity to hear about and discuss the latest advances in each of these areas. Panelists include experts in computer based bio-feedback, members of the US Robotics Society, individuals involved in educational representatives from several of the major manufacturers of microcomputer equipment, and experts in computer art and music.

Hopefully these discussions could help shape the future of personal computing. Special exhibits will also be available for hands-on learning: from 5 PM to 10 PM Monday thru Wednesday, attendees will be able to get firsthand experience with a broad range of equipment including speech synthesizers, video terminals, disk systems, code graphics, and a wide range of microcomputer main frames and peripherals. The conference registration fee covers attendance at all personal computing sessions and exhibits. There is a registration fee of $5 for individuals wishing to only attend the personal computing sessions and exhibits.

Organizers Alice Ahlgren, marketing manager, Cromemco Inc, Mountain View CA 94040, (415) 964-7400; and Robert Albrecht, Author, Dragon, and Friend of Children, Menlo Park CA 94025, (415) 323-6117, will be glad to supply more information.■

A Course

A 2 week course in the fundamentals of digital electronics and microcomputer interfacing will be held at Virginia Military Institute from July 17 thru July 29 1978.

For information and registration forms write to Dr Philip B Peters, Dept of Physics, Virginia Military Institute, Lexington VA 24450.■

A Note for Robot Experimenters

The September 1977 issue of *Dr Dobb's Journal*, volume 2, number 8, arrived here recently. In it readers will find a contribution entitled "An Interactive Programming Language for Control of Robots" by Lichen Wang. The item includes a description of the language, as well as 8080 code for the interpretive language, assembled beginning at location 0 in address space and assuming peripherals in the form of a Processor Technology VDM-1 at locations CC00 to CFFF and an ASCII keyboard input port. The robots envisioned by this software are represented as simulations on a graphic display, a useful first step towards debugging and implementation of motion programs and strategies. *Dr Dobb's Journal* can be reached at POB E, Menlo Park CA 94025. Back issues are available while they last at $1.50.■

Clubs and Newsletters

Conducted by David Wozmak

Apple I Library

A software and hardware library for Apple I users has been started, and all material will be available to interested people at cost. If you wish more information, write to Joe Torzewski, 51625 Chestnut Rd, Granger IN 46530.

Hobby Computer Club

Located in Belgium and the Netherlands, the Hobby Computer Club is a Dutch-speaking club with more than 230 members.

Their first central meeting, held on October 31 1977, was a big success, with more than 140 members attending. Ten different types of computers were present, and the spirit of the members was high.

At the present time about one third of the club's members have computers up and running, and another 60 percent are going to be using the 6800 processor for their planned systems.

For more information, write to HCC, Delftsekade 12, 2266 AJ Leidschendam NETHERLANDS.

Cosmac Elf Club

A user group for the Cosmac Elf has been started in Houston TX. The group sends out a monthly newsletter with interesting articles pertinent to the Cosmac Elf, such as an ASCII keyboard interface for the Elf.

For more information and a complimentary copy of the newsletter, write to Charles E Manry, 2012 Williamsburg Ct S, League City TX 77573.

TI-59 Newsletter

The main emphasis of this newsletter is on the exchange of TI-59 calculator programs related to gambling (especially horseracing). It also deals with business and practical applications of the calculator.

The club, masterminded by Hal Davis, has over 1000 members. The membership fee is $24 per year, which includes the newsletter, a program catalog, toll-free information numbers, a marketplace, and more.

The money goes toward paying for articles and programs published in the newsletter. To find out more about this organization, write to Hal Davis, Mathematical Applications Service, POB 149, New York NY 10956.

Boston Computer Society

The Boston Computer Society (BCS) is an association of computer hobbyists, professionals and people interested in the computer field. Because BCS serves the constantly changing needs of its members, the program varies at each meeting. In general, the group functions as a resource center and information exchange for those involved or interested in the computer field.

The monthly gatherings are divided into three parts: a guest speaker, an "Answers to Questions" session and an open period. The guest speaker is generally a professional who presents an interesting computer application or related skill. An attempt is made to appeal to nearly all levels of interest. The "Answers to Questions" period allows attendees to use the combined knowledge of the society to solve problems such as product searches, computer access, technical problems, etc. The open period allows members to meet one another, read the current computer magazines, try out one or more of the available computers, or participate in various activities.

At the present time, meetings are held at 7:30 PM on the fourth Wednesday of each month, except July, at the Commonwealth School, 151 Commonwealth Av, Boston. The school is located on the corner of Dartmouth St in Boston's Back Bay, one block from the Boston Public Library and the Copley MBTA stop.

Admission to the first meeting is free. Dues are $5 per year. Membership includes admission to all Boston Computer Society sponsored activities, and notification of each meeting by mail. Write to the Boston Computer Society, 17 Chestnut St, Boston MA 02108.

BASIC

Bridgeport Area Computer Club will now be known as BASIC (Bridgeport Area Society for Involved Computerists). The society meets on the first Wednesday of each month at Trumbull Town Library, located about four miles north of exit 48 of the Merrit Parkway. The Society publishes a monthly newsletter called *MicroFlash*. Membership is open to all interested in computers. Annual dues are $8 for regular membership and $6 for student membership. For further information, write to BASIC, 12 Wildwood Dr, Trumbull CT 06611, attn: Al Song.

Sci-Fi Letter Network

A science fact and science fiction letter network, called *AEC Transfer*, is being formed. Its purpose is to allow people interested in computers and science fiction to find other people in their fields with whom to share this interest through correspondence. Send a SASE to Bill Callahan, *AEC Transfer*, Computer Division, 8 Gedney Way, Newburgh NY 12550.

PACC

Pittsburgh Area Computer Club is part of the Midwest Affiliation of Computer Clubs, and holds meetings at different times and places every month. The February meeting will be on the 19th; contact PACC, 400 Smithfield St, Pittsburgh PA 15222, for location and time. The club also publishes a monthly newsletter containing news of local events and articles by members.

TCHG-NT

The Computer Hobbyist Group of Northern Texas is one of the oldest and largest computer clubs in the US.

They currently run two meetings per month, one in Arlington, at the UTA University Hall, in room 108 on the third Saturday of every month at 1 PM, and the other in Richardson at the UTD Green Center, room 2.530 at 1 PM on the first Saturday of each month.

Their newsletter, *The Printed Circuit*, is a worthwhile source of comment. Each issue includes extensive meeting summaries from both Arlington and Richardson, product and publication reviews, the "It's Obvious" column of the not necessarily obvious, technical articles, and so on.

The mailing address is TCHG-NT, POB 1344, Grand Prairie TX 75051. Their information number, which is toll-free from either Dallas or Ft Worth, is (817) 265-9054 or (214) 265-9054.

Amateur Computer Group of New Jersey

The ACGNJ is an excellent club led by Sol Libes, Marty Nichols and a host of others. Club activities include users' groups for the 6800, 650X, 8080, Z-80, Cosmac, and SOL. Extensive 8080, Z-80, 6800, 650X software libraries are also available.

Meetings feature presentations by various computer manufacturers and knowledgeable guest speakers. Locations and times vary.

The *ACGNJ News*, edited by Russel Gorr, is one of the best monthly newsletters I've seen. It has club information, technical data, a "Rumor Page" written by Sol Libes, bits of information from members, classified ads, software, and the "System of the Month." Membership, which includes the newsletter, is $6 per year.

Prospective new members can get information by writing to Amateur Computer Group of New Jersey, UCTI, 1776 Raritan Rd, Scotch Plains NJ 07076.

Minnesota Computer Society

The Minnesota Computer Society meets the first Monday of each month at 7:30 PM at Brown Institute, 3123 E Lake St, Minneapolis MN (unless announced otherwise).

The December 7 1977 meeting featured Dick Finstad and John Ballenthin describing and demonstrating the system they designed and assembled.

Earl Joseph, staff scientist with UNIVAC, will present his talk "Future Smart Computers and Other Future Things" at the February 6th meeting. For further information contact Minnesota Computer Society, c/o Jean Rice, POB 35317, Minneapolis MN 55435.

Microcomputer Investors Association

An association has been formed for the purpose of facilitating the exchange of data and information relating to microcomputers and investing, in the areas of stocks, bonds, warrants, stock options and commodities. The Microcomputer Investors Association is professional and nonprofit in nature.

In order to benefit from the experience of others, there is a basic requirement that, at least once each year, each member submit an article for publication in the association's journal. The initial issue of the journal, *The Microcomputer Investor*, has been mailed to members. Article titles included "A Proposed Format for Information Exchange," "Market Prediction Using Fourier Analysis and Synthesis," "Point and Figure Analysis (A Computer Program)," "Evaluation of Stock Options," "Investment Strategies: A Linear Programming Analysis." For more information, send a SASE to Jack Williams, The Microcomputer Investors Association, 2415 Ansdel Ct, Reston VA 22091, or send $30 and an article for publication (ten pages or less, original, typewritten, double-spaced copy, on a subject in some way related to utilizing microcomputers for investment purposes).

Toronto Region Association of Computer Enthusiasts

TRACE began in February 1976 as an informal meeting of ten people interested in personal computing and quickly attracted followers. Since that time it has grown to approximately 100 active members and about as many casual followers. Approximately half of the members have personal systems of some form.

The ideas behind TRACE are numerous. The main purpose is to foster communication and resource sharing among people, both hobbyists and professionals, interested in personal computing. The meeting format usually includes a system demonstration and one or two talks on topics related to microcomputers. In addition to the meetings, the club has a monthly newsletter, group purchasing, and a library of product literature, books and periodicals in the field of small computers. Other activities include flea markets, exhibitions and a software library.

The club meets at the north campus of Humber College at 8 PM on the fourth Friday of the month, and at the Ontario Science Centre at 2 PM on the second Sunday of the month. For more information contact TRACE, POB 545, Streetsville, Ontario CANADA L5M 2C1.■

Languages Forum

Comments on APL Character Generators

Olav Naess
Welhavensgt 65
Bergen NORWAY

Some writers of letters to BYTE have asked for character generator read only memories with APL symbols for video displays.

I don't think a read only memory is a good solution. I have counted 33 APL symbols which don't belong to the ASCII set. If they replace the lower case letters, the computer is rather useless for text processing. Then comes the problem with superimposing symbols. A video scan cannot backspace or rewrite a line like a typewriter, so each composite symbol would have to be represented in the read only memory as another symbol. (Displays with random access could indicate the composite symbols by changing the constituent symbols between each frame scan, but I don't think this is a good and practical procedure.) That would mean 17 characters extra, and still 26 in addition if underlined letters are to be written. So a 256 character read only memory would be required, which I think is rather impractical, particularly if ordered by hobbyists. Besides, future APL versions might introduce new symbol combinations.

The solution to the problem should be to use a video system with programmable characters, as used in "The Detailer," an Altair (S-100) card made by Micro-Graphics. (I think the same principle is used by the Micromind and Noval computers.) The Detailer, which displays 16 64 character lines, has a 1 K byte directly accessible memory whose contents in the usual way determine which symbols are to be displayed. But it also has another 1 K byte programmable memory block whose contents determine the appearance of the symbols. By inverting a bit in the symbol selecting byte one gets the symbol description information from the programmable memory instead of from an ordinary character generator. Each symbol position is then described as a dot matrix, 8 dots wide and 12 dots high, which extends out to all the neighboring matrices. Each of the 64 software determined symbol patterns which are simultaneously available is described by a 12 byte vector in the on card memory, and 50 of them are required for the APL symbols. (Underlined letters may be replaced by brightness inverted letters through inverting a bit in their symbol selection bytes.) Replacing the APL symbol set (to obtain lower case letters or graphics) involves just moving 64 x 12 = 768 bytes in the computer's memory.

It's nice to have the computer really programmable.■

Baking Baker

We received the following letter in reply to Roxton Baker's letter (July 1977 BYTE, page 11) which referred to P M Lashley's "useless, self-serving, supercilious, unnecessary attack on another man's efforts." This is a reference to Mr Lashley's original letter (February 1977 BYTE, page 77) advocating structured programming.

Shal Farley
Caltech 1-53
Pasadena CA 91126

This being my first letter to the editor, I at first hesitated to get involved. But the utter idiocy of "Lashing Lashley," was the proverbial last straw. Let me address Mr Baker's adjectives individually:

1. Useless. The only thing that could make Mr Lashley's efforts useless are the truly ignorant who refuse to learn.
2. Self-serving. I doubt that Lashley has any economic interest in whatever form of slow torture hobbyists choose for themselves. If he does, I don't.
3. Supercilious. (*Webster's Second Edition:* "adjective: lofty with pride; haughtily contemptous.") In his efforts to be emphatic, Mr Lashley has apparently come across as snobbish to the likes of Mr Baker. Rather, I found his letter refreshing amidst the Pong-Trek-Toe morass. Choosing

ALGOL or PASCAL over FORTRASH is not a matter of reinventing the wheel. All three languages have been implemented for well over a decade. Rather it is a matter of choosing radial ply or steel belted tires over wagon wheels.

4. Unnecessary. In view of the current state of micro software, Mr Lashley's comments were among the most useful things I'd read. The thrust of his article was that history need not repeat itself. Start with the more usable languages.

5. Is Mr Baker going to throw away the decade of work put into structured programming just to conform to the noun?

Structured programming languages are not the toys of wild eyed dreamers, nor are they an intellectual curiosity. To put it in perspective, they are the natural outgrowth of progress in computer science. Programmers' frustrations with machine code led to the first assemblers. Frustration with the limitations of assembly language, the need for defined control and data structures (eg: IF - THEN - ELSE, DO, COMPLEX declarations, ARRAY declarations), led to the development of FORTRAN. Then the experience gained with FORTRAN's limited control structures, and the knowledge of the most common programming errors made while using FORTRAN led to the development of structured programming languages. FORTRAN is painful in part because of numeric labels that lead to confusion and misguided GO TO's. The lack of block structure leads to fragmented code with GO TOs weaving thru the text. When writing programs of great length, block structure helps to divide and conquer as a strong ally of subroutines. By coding and testing the low level blocks and routines first, one may be sure of their operation when testing the higher levels. Thus the location of bugs is circumscribed to as small a portion of code as possible. While it is perfectly possible to write well structured code in FORTRAN, it is a big pain to do so. Also there are several types of data structures that cannot be implemented in FORTRAN that turn out to be quite useful (for example, dynamic array bounds at run time).

By starting with a structured language microcomputer software could leapfrog ahead and save all that effort. If only hobbyists will learn from the past. I learned to program in FORTRAN on an IBM 370/158, and the number of sleepless nights I spent resubmitting my prog after getting rid of yet another bug. . . well, it's just painful to think about. Since then I've been running on a DECsystem 10. Their FORTRAN is much better, but I still found myself tracking down GO TO wrong numbers and correcting random syntax errors. A friend suggested structured languages and the difference is dramatic, from days of debugging FORTRAN, to hours of debugging ALGOL-60. The coding time is also reduced as the language is more "natural" to program in. PASCAL is a derivative of ALGOL-68 and although I've not used it the reports from those who have are very favorable.

Lest you have any fear, it is not a black art. It is in fact much less so than assembly language or FORTRAN. It is part of the continuing process of learning to control the computer in the most convenient manner possible. That's what higher level languages are for. That's the way of the future. As programmers' time becomes more valuable it becomes economic to shape the software and hardware to the needs of the programmer. This is already true of the hobby computer; each user adds the memory and the peripherals of his or her choice. It would be advantageous for the software to be equally facile.

I didn't mean this to be a tutorial, but I feel we stand at a crossroads. The hobbyists can jump for the manufacturers' first and easiest product and be stuck on the same compatibility treadmill as the mainframes, or you may start with a better product and go from there. There are some historic parallels: The Europeans didn't adopt a television standard until many years after the US. As a result they now have a much better product. I would strongly recommend that hobbyists demand, thru their purchasing power, that structured languages be implemented as cross or resident compilers for their systems. The time, effort and frustration saved while developing software will be worth it.■

Languages Forum is a feature which is intended as an interactive dialog about the design and implementation of languages for personal computing. Statements and opinions submitted to this forum can be on any subject relevant to its purpose of fostering discussion and communication among BYTE readers on the subject of languages. We ask that all correspondents supply their full names and addresses to be printed with their commentaries. We also ask that correspondents supply their telephone numbers, which will be printed unless we are explicitly asked to omit them.

Continued from page 12

are pretty well settled, though, and so this series closes with a comparison of the two designs, to the extent that they can be compared. That comparison should illustrate the wide range of solutions that the problems admit and thus the importance of identifying the problems in a design effort like this.

This design is a plausibility argument. . .it is complete (though skeletal) and provides a basis for discussion.

Why You Want to Read This Article

Since another network design will follow this one very soon, you could rightly wonder why this one should be published at all. There are several reasons. At the very least, this design is a plausibility argument intended to counter the skepticism with which some people view the prospect of a network for personal users of information processing. It is also a learning experience: it is a complete (though skeletal) design that can be compared with the PCNET (and other personal computer net designs when they are published) in order to learn about the design issues involved and the stances one can take on them. It can also be implemented; that would get people into communication quite soon, learning from the experience as well as filling the immediate communication needs.

In an age of throwaway material goods, a throwaway design or a throwaway implementation should not seem altogether inappropriate. In fact, they are especially appropriate where the problem at hand is a novel one or has novel constraints, as is the case here. You can speculate all you wish and design as carefully as you wish, but it remains speculative until you try it out. The motivation for building a throwaway net is not entirely intellectual: even if you consume it in the very act of building it, you will leave behind a very valuable thing: a useful communication facility. It is more likely, though, that the first few personal computer nets will persist for a while and that they will serve as media for sharing the experience and insights needed for building their successors. This very persistence will also force people to face an important problem often ignored in fledgling nets, that of interfacing to other nets.

You can speculate all you wish and design as carefully as you wish, but it remains speculative until you try it out. . .

What Is a Net?

Let us pause, before plunging into detailed considerations, to gain a broad perspective on the task at hand. A computer network typically consists of some hardware and some software. The hardware includes all the physical facilities used by the net: phone lines, radio links, computers, etc. Since hardware costs money and must be maintained, the design should minimize the required hardware and distribute it so that its cost can be recovered gracefully. The software includes the necessary agreements governing the use of the hardware, together with the procedures and computer programs implementing the agreements. The agreements include the rendezvous conventions by which conversations are established, the language in which conversations take place, and the rules of behavior under which conversations are conducted. Under the linguistic heading, I mean to include everything from message formats to the representations of characters and bits, etc.

Network specialists have come to use the term "protocol" to refer to these agreements, ranging from all of them down to just the rules of conduct; in this series of articles, I will take "protocol" to mean the extreme of all agreements. Protocols are most easily formulated in layers, with each layer using the one just below it and otherwise almost completely independent of the others. (In this regard, the PCNET and mine resemble one another quite strongly. For example, each has a protocol to transmit bits from one network node to the next and to detect erroneous transmissions; each then uses that layer to provide an errorless, correctly sequenced stream of bits (thence characters) to the higher layers.)

This design exercise concentrates almost exclusively on protocols. Since hardware is relatively expensive, the design requires only a modest amount of hardware, and that of quite readily available kinds. The PCNET, however, looks as if it will trade a modest amount of inconvenience to reduce the hardware requirement even further, but it may also optionally reduce the inconvenience with some highly specialized hardware if some thorny problems can be solved. Actually, though, the distinction between my design and the PCNET alternative with minimum hardware is more one of emphasis than one of substance; either design can be easily adapted to the hardware requirements of the other.

The Experience of the ARPA Net

The ARPA Net embodies an enormous body of experience that can be brought to bear on the design of a personal computer network. However, the ARPA Net experience should be used with caution: both designs have a lot in common but have several important differences. The most important difference is that, while the ARPA Net is heavily subsidized (The ARPA Net is a research tool developed for the US Department of Defense under the auspices of the Advanced Research Projects Agency. It ties together a multitude of large proces-

The LATEST in Tape Systems

MODEL CC-8 $175.00

4800 BAUD CASSETTE RECORDER

An ASYNCHRONOUS NRZ type Recorder with remote motor start/stop. Error rate 10^8 at 4800 BAUD. Can be used from 110 to 4800 BAUD into a UART or "Bit Banger PIA" –no clocking required. This is not an audio recorder. It takes RS232 or TTL signals from the terminal or computer and gives back the same signals. No audio interface is used. Motor start/stop is manual or through TTL or RS232 signals.

Tape speed is 3.2"/second nominal; 1.6"/sec. optional. 110 volt, 60 Hz, 5 watts. (220 Volts on special order). Can use high quality audio cassettes (Philips Type) or certified data cassettes. Can be used in remote locations from a 12 Volt battery.

Recommended for DATA LOGGING, WORD PROCESSING, COMPUTER PROGRAM RELOADING and DATA STORAGE. Especially recommended for 6800 systems, 6502 systems, 1800 systems and beginners with the 8080 systems. Manual control except for motor start/stop. 6800 or 8080 software for file or record searching available on request with order. Used by major computer manufacturers, Bell Telephone and U.S. Government for program reloading and field servicing.

AVAILABILITY – Off the shelf.

6800 CONTROLLER for SWTP

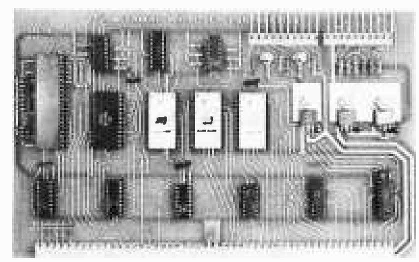
PROVIDES MONITOR AND TAPE SOFTWARE in EPROM. EXPANDS MIKBUG with 1 K of ADDITIONAL ROM PROGRAM.

This is a complete tape controller for the SWTP 6800 system. Has 3 K of EPROM space for your own programs. A 1 K ROM (2708) is provided with all tape and monitor functions. The ROM program is identical to our extensive 8080 ROM program.

Has one ACIA for one or two tape drives, one USART for an additional Serial port and a 4 bit parallel port for motor control. Will control one or two CC-8 or 3M3A drives with the software provided. Can be used with other tape drives controllable with 4 TTL bits if appropriate software changes are made.

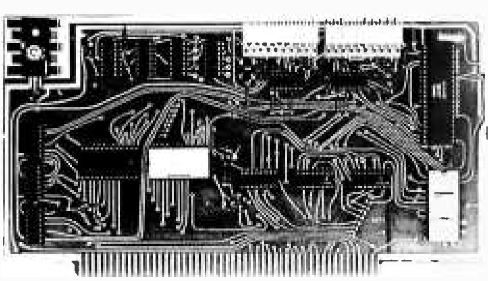
Extra serial port is provided for your use with a second terminal or printer. (RS232, TTL or 20 ma)

The ROM program supplements the MIKBUG program and is entered automatically on reset.



AVAILABILITY–Off the Shelf.

$190.00, Tested & Assmb. ($160.00, Kit)



2SIO (R) CONTROLLER
$190.00 ($160.00 Kit)

PROVIDES MONITOR AND TAPE SOFTWARE in ROM TERMINAL and TAPE PORTS on SAME BOARD CONTROLS ONE or TWO TAPE UNITS (CC-8 or 3M3A)

This is a complete 8080, 8085, or Z80 system controller. It provides the terminal I/O (RS232, 20 mA, or TTL) and the data cartridge I/O, plus the motor controlling parallel I/O latches. Two kilobytes of on board ROM provide turn on and go control of your Altair or Imsai. NO MORE BOOTSTRAPPING. Loads and Dumps memory in hex on the terminal, formats tape cartridge files, has word processing and paper tape routines. Best of all, it has the search routines to locate files and records by means of six, five, and four letter strings. Just type in the file name and the recorder and software do the rest. Can be used in the BiSync (IBM), BiPhase (Phase encoded) or NRZ modes with suitable recorders and interfaces.

This is Revision 7 of this controller. This version features 2708 type EPROM's so that you can write your own software or relocate it as desired. One 2708 preprogrammed is supplied with the board. A socket is available for the second ROM allowing up to a full 2K of monitor programs.

Fits all S100 bus computers using 8080 or Z80 MPU's. Requires 2 MHz clock from bus. Cannot be used with audio cassettes without an interface. Cassette or cartridge inputs are RS232 level.

AVAILABILITY – Off the shelf.

Z 80 BOARD for SWTP COMPUTER: Now you can use the 8080/Z80 software programs in your SWTP 6800 machine. Replaces your MPU board with a Z80 and ROM so that you are up and running with your present SWTP memory and MPC card. $200 assembled and tested. ($160 kit)

AVAILABLE–November '77.

OVERSEAS: Export Version 220 volt 50 hz. Write factory or: Megatron-Datameg, 8011 Putzbrunn, Munchen, Germany; Nippon Automation 5-16-7 Shiba, Minato-Ku, Tokyo, Japan; Hobbydata, FACK 20012, Malmo, Sweden; G. Ashbee, 172 Ifield Road, London SW 10-9ag: Trintronics, Ltd., 186 Queen Street W., Toronto, Ontario, Canada; EBASA, Enrique Barges 17, Barcelona 14, Spain; ARIES, 7, rue Saint Phillipe du Roule, 75008 Paris; Microlem 20131, Milano, Italy; Eagle Electric, Capetown, S. Africa.

For U.P.S. delivery, add $3.00 Overseas and air shipments charges collect. N.J. Residents add 5% Sales Tax. WRITE or CALL for further information. Phone Orders on Master Charge and BankAmericard accepted.

National Multiplex Corporation

3474 Rand Avenue, South Plainfield NJ 07080 Box 288 Phone (201) 561-3600 TWX 710-997-9530

Circle 82 on inquiry card.

sors nationwide at universities and around the world using satellite links.), a network for individuals should (and can I believe) support itself right from the start. That consideration has a most important ramification in my design: the inclusion of "Community Information Exchanges" (CIEs) that serve as focal points and buffers between the other nodes of the network. Further, the ARPA Net design includes the notion (inapplicable to a personal computing network) that most network nodes will be connected to the net most of the time. Thus, its protocols provide a general interprocess communication facility and include useful services as special cases of it. A personally oriented network by contrast should be optimized toward the two facilities that are most immediately applicable to the personal computer community (and that have, incidentally, proved most useful in the ARPA Net): sending mail and sending files of programs. Finally, I think a personal computer network should be oriented to an overall architecture in which internode connections are sporadic, fleeting and relatively infrequent.

An Overview of The CIE Net: The Basic Idea

A Community Information Exchange (CIE) Net should be designed from the start to gracefully accommodate any foreseeable circumstances to which it might eventually need to adapt. Thus, the suggestions presented here allow for a large number of nodes to eventually become attached to the network. The protocols have room to expand into, and they identify themselves so that several incompatible sets of protocols can be accommodated at a given time. Addressing is defined for the United States, but the addresses can be expanded to cover other countries as well. A separate mechanism is included to let this net gracefully interface to other networks serving the same territory.

The suggestions included here amount to a partial design. They are quite definite so as to form a firm basis for discussion. Some suggestions were arbitrarily chosen just to make the design more definite, while other suggestions embody principles that are important to the overall concept.

The overall design is formulated in such a way as to enable it to be implemented cheaply and at maximum convenience to the implementers. Its backbone is a network of Community Information Exchanges (CIEs) that serve as buffers for network traffic. The overall network is designed to work even when the exchanges are connected only fleetingly and at only loosely scheduled intervals. One party to each connection must be capable of timing out the other party and taking corrective action when necessary; otherwise the protocols are designed so that they can be implemented in string BASIC on computers that do not allow access to lower levels of programming, such as direct binary manipulations or input and output. The required hardware support is minimized to equipment that is readily available to homebrewers.

The Four Kinds of Stations

A CIE Net would be composed of four kinds of nodes, with communication lines between them. By default, they will communicate by telephone lines in a universal language discussed in another section, but any two nodes are free to use any other mutually agreeable medium when they talk to one another. The four kinds of nodes are the Community Information Exchange (CIE) that buffers network traffic; the subscriber, a person wishing to take part in a CIE Net; the relay station that moves messages from one CIE to another; and the gateway connecting a CIE Net to other networks. In the rest of this series, I will use the term "station" for these four kinds of nodes in order to avoid confusion with anything that is of no interest here but might be called a node in more general network terminology. The various stations can be thought to inhabit different computers, and the design uses that model. However, any particular computer can be host to several stations of possibly different types, and the design specifically allows them to optimize their own intercommunication using any facilities provided by their host.

A CIE serves as a communication buffer among subscribers and relay stations. That is, it must have file storage that a program can access fairly expeditiously, and it should have some complement of answer-mode modems which subscribers and relay stations can call at their convenience during the night when phone rates are lowest. While different realizations of file storage have different requirements and capabilities in their technological, financial and operational aspects, there are a few functions that a CIE will require of its file storage facilities. It must be able to read or allocate and write multiple thousands of characters worth of buffers which it may need to hold for several hours or several days. It should be able to gain access to almost any part of this storage within a few seconds, and it should be able to read or write succeeding character positions at very nearly the transmission rate of the lines on which it will receive calls from subscribers or relay

The CIE Net is designed with four classes of nodes in mind: the community information exchanges with mass storage resources, individual subscriber nodes, relay nodes, and gateways to other networks. . .

A network is a systematic combination of hardware and software. . .hardware of computers, phone lines, etc, and software of protocol agreements and computer programs implementing the agreements. . .

stations. The CIE can encrypt or compress data as it sees fit; the protocols are structured in such a way that no intervening stations need give any special treatment to an encrypted or compressed message transmitted from one CIE to another. Finally, its subscribers may wish it to provide them with long-term storage, but it should also be prepared to gracefully recover temporary storage that is no longer needed by the network.

A subscriber is thought of as a station possessing an originate-mode modem, some local processing power and file storage and a terminal or other interface to a human being. (Of course, the subscriber may be realized as a set of programs cohabiting a host computer with a CIE and reachable by a person having only a terminal and an originate-mode modem; the implementation details are irrelevant, and the functional distinction is still useful.) The subscriber station need not be able to rapidly position its file storage because the person operating a subscriber can easily anticipate the requirements and because subscribers are involved only at the end points of a message's transit through the net. The subscriber, not the CIE, has charge of all programs that handle files of messages, selective display and rapid retrieval of messages and whatever negotiations with its human are necessary to generate a message in the format described in these protocols. Although the subscriber is extremely important to the CIE Net, the protocols by which a CIE and its subscribers might communicate are idiosyncratic to the stations' realizations, the subject of a large body of literature, and irrelevant to the process of transmitting messages between endpoint CIEs; therefore this article makes no attempt to enlarge the current stock of such suggestions.

A relay station, like a subscriber, can be modeled as having an originate-mode modem, a terminal, some processing power and some file storage. A relay station, however, need have no file storage, but can instead have a second originate-mode modem and thus connect two CIEs quite directly with only fleeting use of buffer storage. A relay station can be thought of as providing a communication link between two CIEs: it would first phone one and collect messages headed in the general direction of the other; then it would phone the second and both unload those messages and collect messages headed in the general direction of the first, which it would then call back.

Finally, a gateway is conceptually an answer-mode modem and a connection to some other network, along with whatever processing power or file storage is required to connect the CIE Net to the other net. Depending on the desires and resources of the people connecting the gateway to the CIE Net, it could be capable of transmitting traffic in either direction or both directions between the two nets, or it could be connected to any arbitrary number of networks. These protocols provide a frame to hold messages headed through a CIE Net to an outward facing gateway; the content of a message inside the frame is completely unspecified and can reflect any idiosyncrasies of the gateway and the other net. An inward facing gateway is treated exactly like a CIE by its neighboring relay stations, but it does not serve subscribers the way a real CIE does. These protocols completely ignore the content of a message headed into a gateway and require messages coming out of a gateway to conform to the same rules as any other CIE Net messages. The motivation is that the network on the other side of a gateway might cover territory that doesn't overlap that served by the CIE Net, or it might thinly cover the same territory, say by providing high speed links between major cities. This opacity also frees the CIE Net from any necessity to commit itself to any of the possible internetwork message formats currently under discussion in the International Network Working Group. These protocols make no attempt to handle the problem of choosing a gateway through which to route messages that should be routed through some gateway; instead, they presume that the source CIE can choose the proper gateway or can send the message to some other CIE that agrees to make such a decision and then forward the message appropriately.

These comments describe functional nuclei of the conceptual CIE network; they make only mild commitments to the implementation details by which the four kinds of stations are realized. For example, it is easy to imagine a gateway cohabiting a host computer with a CIE or a relay station. Also, a computer on the ARPA Net typically contains a CIE, a relay station and a large number of subscriber stations sharing programs among themselves and each sharing file storage with the CIE and relay station. These cohabitation possibilities serve to emphasize the freedom with which stations can agree to speak any language other than the universal language specified here. Finally, don't forget that the commitments in this proposed design are structured in this way to specifically allow a CIE to be put onto any dial-in time sharing computer without its operators needing to cooperate

Simulation Games and Networks

In the realm of possibility, but hardly a real world phenomenon until a network such as PCNET or CIE Net is implemented, is the idea of multiple player simulation games. A simulation game is a game of considerable flexibility and considerable complexity which forms a model of some aspect of the real world and explores the parameters of the model. Crude examples of such games are the familiar Star Trek, Kingdom, Wumpus, etc: games of personal computing as it stands today. The extension of game ideas which can happen over a network is the multiple user game, for example, a trading game involving 100 or so active players, a battle simulation game involving several hundred active members of various teams, or wild fantasy games in mythical worlds with mythical creatures and dangers. The network concept provides a vehicle for interpersonal computer aided intellectual sports whose players could grow into a vigorous subset of the users of this communication channel.

The US Postal Service has provided an invaluable service in the form of the ZIP code of mailing addresses, which is highly correlated with the geographic location of the individual mailing address. By incorporating the ZIP code into each individual's CIE Net address, message routing information is inherently made a part of the address. . .

with (or even suspect) the effort; presumably, a CIE handling any moderate volume of traffic could be connected to a relay station operated by the staff of its host computer to their mutual benefit.

Addresses

The addressing scheme proposed here has several interesting properties. It refers to a geographic location in a transparent enough fashion that messages can be routed to any place in the US by stations having a minimum amount of specific routing information. Conversely, when a new station must choose an address, the range of choice is limited in such a way that only local negotiations are necessary. On the other hand, the limits are so generous that those local negotiations should quickly terminate in universal satisfaction. This scheme also ensures that, in a direct interstation conversation, each station can easily mark transmissions to distinguish echoes of its own transmissions from those sent by the other station.

The basis of this addressing scheme is the US Postal Service ZIP code. As a perusal of the first few pages of the ZIP code directory will show, the first digit designates one of ten major areas of the country, the next two digits designate a sectional center within the national area, and the last two digits specify a nearby post office (or branch). Thus, any network making a routing decision for a message can compare the intended destination address with its own address. If they differ in the first digit, the routing station can consult a table of nine entries to decide the appropriate general direction. If they agree in the first digit position but differ in the next two, then the right direction can be found in a second table of 99 entries. Otherwise, a third table of no more than 99 entries will show the proper direction in which to forward the message. Thus, routing to the proper ZIP code area can be accomplished by using tables with no more than 207 entries in a given computer; actual programs will probably use far fewer table entries and capitalize on the high degree of regularity with which ZIP codes have been assigned.

The power of the addressing scheme proposed here is extended in two directions from the ZIP code basis just outlined. In the upward direction, this scheme anticipates a possible future requirement for network addresses outside the US by specifying the optional prefix USA for addresses that the ZIP code can reach. Thus addresses become partially self-identifying: an address that doesn't start with a digit or the letters USA cannot possibly be mistaken for an address specified by these protocols. In the downward direction, the precision of the ZIP code is refined by the addition of a suffix both for the sake of the interstation protocol and in anticipation that several alternative (or competing CIEs) might be set up in a given ZIP code area, perhaps even on the same host computer. The suffixes are three digit numbers; blocks of suffix values are preassigned, and assignment of a specific suffix within a block is a matter of local negotiation. The preassigned blocks are:

> 0-9: (reserved for testing interstation protocol)
> 10-99: forwarding centers (see below)
> 100-899: CIEs and gateways
> 900-999: relay stations.

Since any one ZIP code seems to serve a maximum of about 20,000 people, the block preassigned to CIEs is big enough to allow a CIE for each 25 people; that ought to be generous enough for almost any eventuality.

The block assigned to relay stations has a deceptive appearance that merits further elaboration. The interstation transmission blocks need to be marked with the identity of their sender. Since a transmission block is quite short and since any interstation transmission has only two parties, the sender's identity can be reduced to a single bit derived from a comparison of the two network addresses. That comparison is possible if a CIE also serves as the relay station, but the relay station needs its own address if it is not also a CIE. If the relay stations in an area come from a large, highly variable, loosely organized pool of volunteers, it may not be practical to permanently assign network addresses to them. The problem can be neatly solved, however, by allowing relay stations to arbitrarily pick network addresses as the need arises. Since the block from which they pick an address cannot include a CIE address, then there is no danger of the addresses coinciding, and so a short address can always be found.

This scheme has the disadvantages, because it is entirely numeric, of being prone to human error and of being somewhat opaque to people. The situation can be partially alleviated by the addition of symbolic addresses (eg: San Francisco) and forwarding centers. The general idea is that people might be willing to set up forwarding centers (probably in conjunction with CIEs) that would be willing to interpret symbolic addresses, reformat the messages with proper network addresses, and forward them in the proper direction. The details are less than clear to me at the moment. Forwarding centers could also solve the problem of mobile subscribers: a subscriber, before leaving a CIE, could tell a nearby forwarding center where messages should be redirected. When messages are subsequently routed through the forwarding center, it can send them on (or discard them) appropriately. When a forwarding center is not acting as a blind mail drop, it should also tell the sender of the message where future messages should be sent.

An additional disadvantage of this scheme is that, since the address space can potentially name 80 million CIEs, there is no practical way to broadcast a message to all CIEs. Should that prove desirable, a few suffixes (eg: 890-899) could be given special meanings, such as, "Deliver copies of this message to yourself and to all CIEs with higher addresses."■

This completes an introduction to the concept of the Community Information Exchange Network (CIE Net). In the second part of this three part series, the discussion next month turns to the description of the CIE Net protocol details. In the final installment of this series, several issues of a technical and legal point of view will be considered in more detail, as well as comparison of CIE Net with the PCNET design in more detail.

Continued from page 90

capable of receiving and executing commands to perform such high level reflex actions as running, carrying, etc, without further attention. Beyond this point, we find several more specialized systems which may issue commands to this "motor automaton," or reach around it and access the LMN systems directly, or enter the automaton at any level. To understand the division of labor among these systems, we need to focus on the way in which the execution of the output is related to the data which directs it. There are basically two systems which can be used, and both have been used in robot systems. The first is the "dead reckoning" approach, in which the details of the required action are computed in advance, and then executed without regard to their results. (An interesting example of this in a robot system is described in Ralph Hollis' article on NEWT in the June 1977 BYTE, page 30). The other approach of course is to continually monitor the results of the movement and apply corrections as required. Both of these systems have their uses, advantages, and weaknesses, and the brain employs both systems, usually cooperatively in the same actions, although "pure" examples of each can be found.

One of these systems is associated with the part of the brain called the cerebellum. The cerebellum is not an instigator of action, nor is any conscious experience associated with its activities. It plays an important role however in the expression of actions, of both reflexive and voluntary types, which are generated elsewhere. Among the functions which the cerebellum performs are the translation of parallel to serial output, and the control of feedforward correction in open loop control circuits.

Before describing these functions further, it will help to examine the circuitry of the cerebellum. This structure, which lies above the pons, consists of two parts, an overlying cortex and a set of nuclei. The neurons of the cerebellar cortex are arranged in a distinctive pattern which is endlessly repeated over the surface of the structure. A few elements from this pattern are shown in figure 2. Simplified to the bare essentials, this consists of an input element (G) which has an axon that runs for some distance, spatially parallel to the axons of all of the other input elements, and which in the course of its passage activates a row of output elements (P). Firing an input element thus selects a particular set of outputs. Since pulses may travel rather slowly in small diameter axons such as those of the input elements, the time of arrival of the select pulse at successive output elements may be long compared to the duration (or transmission time) of their outputs. Thus the cerebellar cortex may act as a tapped delay line, as well as a decoder. If the final output elements are switched to other input elements, elemental sequences may be serially cascaded to form larger patterns. There are a number of auxiliary elements associated with the G and P types, and these are lumped as O elements in our diagram. They are capable of performing such functions as selectively inhibiting individual output elements, and controlling interactions between adjacent parallel row systems. Thus, these

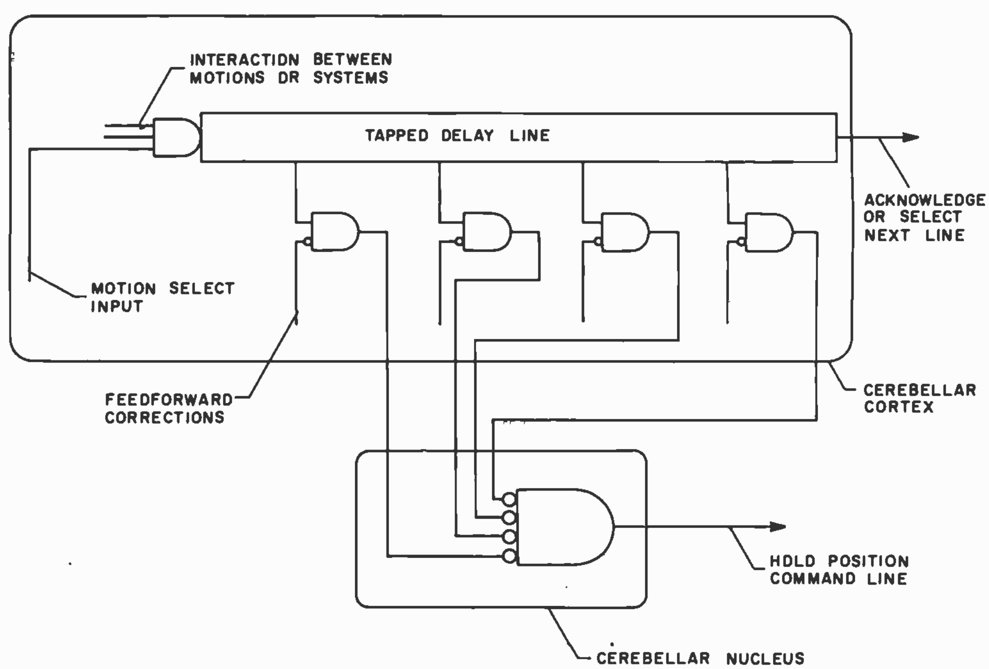


Figure 3: A functionally equivalent logic circuit for one row of a cerebellar cortex fiber system. The tapped delay line is the logical equivalent of the parallel fiber axon's propagation characteristics.

SURPLUS ELECTRONICS

IBM SELECTRIC®
BASED I/O TERMINAL
(USED) $695.00

- Tape Drives • Cable
- Cassette Drives • Wire
- Power Supplies 12V15A, 12V25A, 5V35A Others, • Displays
- Cabinets • XFMRS • Heat Sinks • Printers • Components

Many other items
Write for free catalog
WORLDWIDE ELECT. INC.
10 FLAGSTONE DRIVE
HUDSON, N.H. 03051
Phone orders accepted using VISA or MC. Toll Free 1-800-258-1036
In N.H. 603-885-3705

Circle 126 on inquiry card.

RADIO SHACK AND COMMODORE USERS

Low Cost Software on Compatible Cassette Tapes for your TRS-80 and PET computers.
— Games
— Small Business
— Investment and
— Scientific Programs.

Tapes also available for OSI, APPLE II, SWTP 6800, and Processor Technology.

SEND FOR FREE CATALOG

Silver State Enterprises
P. O. Box 27111
Lakewood, CO 80227

Circle 104 on inquiry card.

JENSEN

FREE CATALOG OF HARD-TO-FIND PRECISION TOOLS

Jensen's new 152-page catalog is your single source for hard-to-find precision tools used by electronic technicians, scientists, engineers, instrument mechanics. Also contains 10 pages of useful "Tool Tips" to aid in tool selection. Send for your free copy today!

JENSEN TOOLS & ALLOYS
1230 SOUTH PRIEST DRIVE · TEMPE, AZ. 85281

Circle 64 on inquiry card.

?

WHAT IS
a really good way to protect all your valuable back issues of BYTE that will keep them handy, clean, and organized?

Send to: BYTE, POB 5120, Philadelphia PA 19141
for information on binders and files.

Software Professionals

Can You Qualify?!

One of our clients is staffing a software group to maintain their leading edge in state-of-the-art technology.

Growth positions in the 15-40K range are available.

Some of the areas include: *Compiler Design* (string handling, character manipulation, syntax processing) *Data Base Design — DOS* (index sequential file structure, etc) and *Real Time Operating Systems* (async & bysync, interrupt processor, resource allocation).

We would like to tell you more about these and other New England opportunities. All inquiries will be held in strict confidence.

Call us collect 603-888-5500 or send resume.

Touchstone Associates
Personnel Consultants
104 D.W. Highway
Nashua, N.H. 03060

Touchstone

Circle 120 on inquiry card.

BYTE
Article Index

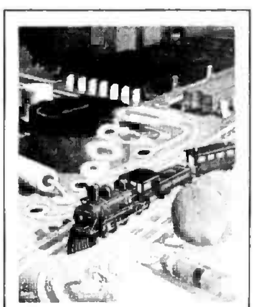
Complete listing of all feature articles appearing in Volume I of BYTE — September 1975 thru December 1976. Indexed for easy reference. Includes all errata.

FREE

To get yours, send a $.24 stamped self-addressed envelope to:

BYTE Index
70 Main Street
Peterborough NH 03458

STILL A FEW LEFT...



The limited edition full color computer engineering poster, **$3.50.**

BITS
70 Main St
Peterborough NH 03458

Circle 9 on inquiry card.

HP 9825A COMPUTER

I am selling my personal HP 9825A Desktop Computer. Here is a chance to own a professional quality computer for little more than a hobby-grade system.

FEATURES:

- 6 MHz, 16 bit CPU
- High speed, 250 K byte tape cartridge
- Upper/lower case display and keyboard
- 8 K bytes RAM
- Operating system and drivers in ROM
- Interactive editor in ROM
- HPL language in ROM
- Built-in 16 column printer

ROM's for I/O, MATRIX, STRING VARIABLES, ADVANCED PROGRAMMING are also included as well as IEEE-488 interface card. Full documentation. Price: $4700.

Roger Cox
4327 N Chestnut St
Apt 18
Colorado Springs CO 80907
(303) 599-9274

Circle 32 on inquiry card.

CALIFORNIA APPLIED TECHNOLOGY

32K-MEMORY CARD-$410
S-100 STATIC KIT

2708E-PROM-$14

2708 PROGRAMMED-$20
WITH 8080 SYSTEMS MONITOR

80 MAJOR COMPUTER PRODUCT LINES AND AN EXHAUSTIVE LINE OF ELECTRONIC COMPONENTS

COMPUTERS
HARDWARE AND SOFTWARE
CUSTOM HARDWARE AND SOFTWARE
RESEARCH AND DEVELOPMENT

P.O. BOX 1239
TWENTYNINE PALMS, CA. 92277
714-367-6996

FREE BASIC PROGRAM AND CATALOG

'ALEX IS COMING'

Circle 12 on inquiry card.

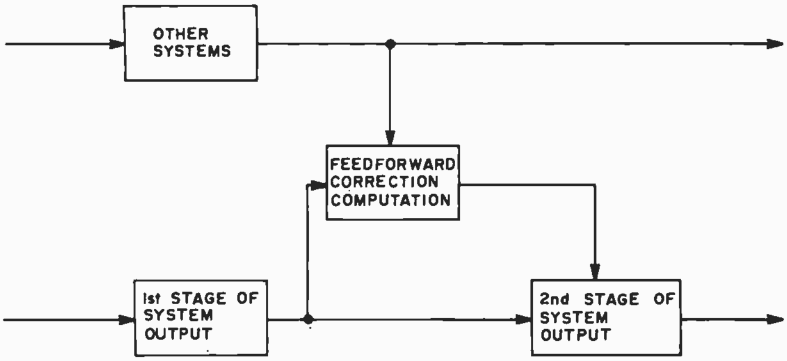


Figure 4: Feedforward as a means of correction in systems where feedback is too slow. This technique involves an open loop correction based on other systems' input data.

elements may impose modifications on output sequences, or call on adjacent systems (which control similar muscle functions) for assistance. Some of these functions have actually been simulated on large digital machines in experimental motion control systems. A schematic of a circuit modeling the essential features is shown in figure 3.

The outputs of the cerebellar cortex fall on the neurons of the cerebellar nuclei, which relay them widely throughout the brain. Inputs to the cerebellum likewise originate in many portions of the system. There is evidence in fact that different motor system functions may time-share the device! A major function of the cerebellum however is to allow for interaction between different command systems.

Feedforward Controls: Coordination

To illustrate this point, let us see how it is applied to feedforward modification of output. In any system which is not amenable to feedback control, such as one involving actions that are more rapid than the loop time that would be required to control them, or ones that would require very extensive processing of feedback input, it is nonetheless possible to achieve considerable correction for moment to moment conditions by passing the basic output command to both the next level of the output system and to a controller which computes the necessary deviations from the basic command and forwards these to the lower echelons of the output system. The concept is diagrammed in figure 4. Thus, a reflex motor loop which performs some function such as walking sequences may need to be modified from its basic pattern by information about head tilt from the vestibular system, while at the same time the reflex vestibular motor systems which keep the head level may require information about what the stepping generator is about to do, in order to allow for impending body tilt. The whole sequence needs to take place *before* any muscle action occurs which could generate feedback information if we wish to move swiftly and still avoid a fall.

The process is popularly called "coordination," and the quality of yours is dependent on the excellence of your cerebellum. What happens in this process is as follows: sequences of motor actions generated at any level of the hierarchical reflex "automaton" system, or at any high level system which inputs to it, are also sent to the cerebellum, either as inputs to the parallel fiber decoding systems or as inputs to the "other" elements which control interactions across parallel systems and gate individual output elements. Thus, the waves of parallel fiber activity generated by different command systems can interact in the cerebellum and modify one another in predetermined fashions. The resulting modified command is sent forward as a set of corrections to the basic command, and the two interact at lower echelons to produce a corrected action. (Yes, they can get there at the same time. We've got control of transmission speed, remember.) One clear advantage is the provision of a common site of interaction for systems which are functionally related, but do not possess physical elements in common.

Now that we've got it taking care of interactions and corrections, how do we get "dead reckoning" of movement parameters? This process relies on a parallel to serial conversion which uses time as an analog of position. A basic function of the cerebellar nuclei is holding or maintaining positions by appropriate outputs to the biasing elements in systems such as the LMN system. The output elements of the cerebellar cortex however act to inhibit the cerebellar nuclei. Thus, damage to the cerebellar nuclei results in tremor, oscillation, and similar signs of excess activity. Damage to the cerebellar cortex on the other hand results in deficits related to underactivity, motions that fall short of the target or fail to initiate. In the case of a pure example of the "dead reckoning" type of motion (frequently referred to as saccadic motion), such as the motion of the eyes in fixing on a new point of focus, the motion itself is of constant velocity. (More accurately, it is driven by a constant input, it clearly can't accelerate and decelerate instantaneously.) Given this, it follows that the extent of the motion is determined solely by the duration of the driving signal. If the motion generating "automaton" circuits are held in check by the cerebellar nuclei, then action of the cerebellar cortex which inhibits the cerebellar nuclei disinhibits the motion generators and the movement begins. If the outputs of a group of

output elements from the cerebellar cortex which are fired in sequence by the same parallel input fiber fall on the same group of neurons in a cerebellar nucleus, they will keep that group inhibited, and the associated motion in progress, for as long as the sequential firing of output elements is maintained in the cerebellar cortex. This will then determine the extent of the motion. The cerebellar nuclear cells are OR gating the output sequence of the cerebellar cortex.

It follows then that if some high level command system computes the type, direction and extent of a required motion, it can pass this information to the cerebellum in a parallel form as a select request for a particular set of input elements, and perhaps a set of gating and switching elements as well. This request will set in motion a time sequence of activity in the cerebellum, which will be appropriately modified by interaction with other current activity in the cerebellum, and output as a motion in space with a particular duration of action and spatial extent. Meanwhile, the requesting device is free to go about its business.

There are many kinds of activity which rely heavily on this type of control, and many of them are learned activities. A good example is playing the piano. This is clearly a learned sequence of movements, but once learned, the action is too rapid for guidance by feedback from ear or eye. It has been suggested that the learning of such motor sequences may proceed through the formation of new functional connections in the cerebellum, so that the end elements of one sequence become select inputs for the next sequence. In any event, we certainly could do it this way.

The action of the cerebellum involves a large analog component, and although this could be, indeed has been, modeled with a fast processor and an array of digital words to represent the states of output elements, this may not be the best approach. A device which offers great promise for a very close analog to cerebellar operation is the surface acoustic wave (SAW) device which transforms electrical signals into surface waves on a piezoelectric medium, manipulates them in unique ways related to their travel time, and regenerates electrical signals at the outputs. A similar result can be achieved with charge transfer devices. Tapped delay lines are easily made, and many such in parallel on a chip have been used for such tasks as electronic focusing of imaging systems. This technology would seem to offer a splendid opportunity for developing a "cerebellar chip." An excellent review of these devices can be found in Brodersen and White's article in the March 18 1977 issue of *Science*.

Higher Motor Systems

Turning now to the motor structures of the higher brain regions, we find two which stand out as particularly important, the basal ganglia and the motor cortex. These structures operate in an interactive fashion in a supersystem which also involves parts of the thalamus, and which has important inputs from systems whose principle functions are best regarded as cognitive and emotional rather than motor. At this level of motor organization, the distinction between concept, desire, and action begins to blur, and these "motor" systems may also be involved in at least certain motor oriented aspects of other functions. It is somewhat misleading, but probably necessary, to discuss separate functions for the higher motor systems. The fact that they are parts of a functional supersystem should be borne in mind.

The motor cortex, more accurately called the somato-motor portion of the cortex, was once thought to be the highest level of motor integration in the brain because of the late evolutionary development of the cortex. It now appears however that it is more properly viewed as a specialized parallel processor system which has been developed to refine and increase the resolution and processing speed of functions which are directed from older structures. A notable feature of the somato-motor cortex is a massive projection of large fast axons which run all the way down the spinal cord and end directly on the LMNs. Along the way, these axons give off many branches to higher level motor centers of the medulla, pons, cerebellum, etc. It appears that this direct communication from highest to lowest levels of the system allows high level command systems to reach around the motor automaton hierarchy for direct intervention. It is obvious that this type of control must be available to a system which is to have a behavioral repertoire that is not built solely of stereotyped action patterns. This is particularly true if the system is to have the capability of constructing novel behavior patterns, either to meet a particular problem, or to serve as a basis for learning new behavioral repertoire items.

A Sense of Touch

Although systems such as the cerebellum and the basal ganglia have direct communication with the hierarchical motor system to control the many motor stereotypies which it automatically generates and regulates, they also both access the somato-motor cortex, and apparently provide most of its direction and control. The somato-motor cortex then may be viewed largely as an extensive decoder for cerebellar and basal gang-

lia initiated actions. There is one situation, however, in which the somato-motor cortex is itself the originating device for motor function. That situation is the control of action based on feedback information from the sense of touch. The reason we refer to it as the "somato-motor" cortex is that this region not only contains the neurons which give rise to the axons controlling the lower motor systems, but also the neurons which receive the input from the touch receptors in the skin. The sense of touch is technically referred to as "somesthesis." The special relation of the sense of touch to this system is explained by the fact that a great deal of fine motor control is under feedback control derived from the various transducers for pressure and other sensations which comprise the sense of touch. This is especially true of organisms such as human beings which place so much behavioral emphasis on the control of precise manipulative movements. While a great many movements which are under feedback control may initially be under visual guidance in reaching the general area, the fine control of the later stages is generally under the control of feedback from touch receptors. When you pick up an object with your hand, it is not your eyes which tell you how hard to squeeze, or just how to grasp. (Have you ever used a keyboard that didn't provide tactile feedback?) The somato-sensory function of the somato-motor cortex involves elaborate encoding schemes which are similar to those which we will consider later with the other cortical sensory systems. For now, suffice it to say that this information may act directly on the motor output aspects of this region to initiate motor activity in those cases where touch information is the appropriate controlling input. In other cases, this information may be used to provide correction to outputs of the somato-motor region which are being initiated and controlled from other structures.

The somato-motor cortex receives its principal control inputs from a group of nuclei in the thalamus, which in turn receive the major share of their input from the cerebellum and the basal ganglia. These thalamic nuclei thus serve as preprocessors which synthesize directives for the sensorimotor cortex out of requests from several systems.

Homing in on a Stimulus

The final portion of the higher motor system which we shall consider in detail is the collection of nuclei known as the basal ganglia. I shall use this term to include some nuclei of the mesencephalon and diencephalon as well which function largely in conjunction with the basal ganglia.

Just as the cerebellum is heavily involved in the operation of feedforward and dead reckoning kinds of control, the basal ganglia are primarily involved in graded, feedback controlled movements, particularly those of a learned nature, or those under direct conscious control. It can probably be regarded as the highest level in the command system which has a primarily motor oriented function.

The structure of the basal ganglia at the neuronal level is entirely different from that of the cerebellum. There is no obvious pattern of spatial arrangement to its neurons, although both local and output elements can be identified. The local elements are much more numerous than the output elements, and form an extensively branched system within the basal ganglia. It appears that most of these have an inhibitory action, so that neighboring elements are quickly turned off by any activity. Some of these connections are recurrent, so that input driven elements, too, tend not to remain active beyond an initial response to input. This is in sharp contrast to the situation in the cerebellum where the entire principle of operation is based on a propagated response in a neuronal network, initiated by a single input. The action of the basal ganglia is of a sort called "self-quenching." That is, an input will initiate a burst of activity, but unless the input is maintained, or augmented by another input, it will rapidly inhibit itself. This is true not only because of the local recurrent inhibitory neurons of the basal ganglia, but also because of negative feedback loops from the basal ganglia to its inputs which tend to damp their initial activity. Notice the similarity of basal ganglia action to that of a differentiator. If one could consider the space coded byte of the active input elements to the basal ganglia as encoding some static scheme of output for motor behavior, the temporal output byte of the basal ganglia might be thought of as having properties similar to the first time derivative of the behavior specified. This output would then be decoded into commands to the motor cortex, cerebellum, and reflex motor system. By outputting this time decaying command, it is ensured that the behavior will not continue unless (1) the command is sustained by some other means, or (2) a new command set is tried, producing a new set of

self-quenching output pulses. This feature is essential if the continuation of a behavior is to be made contingent on its consequences. The basal ganglia in fact have sets of inputs which are precisely configured to achieve this contingency.

The outputs of the basal ganglia run principally to: the thalamus and thence to the motor cortex; to the motor nuclei of the mesencephalon and thence to the subsystems of the reflex motor apparatus; and to the motor nuclei of the pons and thence to the cerebellum. The basal ganglia are thus in a position to transmit information and commands to all aspects of the motor system. There is little here that is not understandable in terms of principles we have already dealt with, and it requires no elaboration.

The inputs to the basal ganglia, on the other hand, are the key to understanding its function. There are three major components of the input. First, the entire cortex projects fibers into these nuclei. These fibers, and those of the second component which arises from the thalamus (a structure which organizes the activity of the cortex, and processes IO for it), tend to make contact with a few specific neural elements in the basal ganglia. These two input groups may be thought of as specifying discrete patterns of activation which are encoded by an action like a series of cascaded AND gates into a pattern of activity, or potential activity, on the output lines of the basal ganglia. If output continuously, these outputs could be decoded by lower motor structures into specific movements. It appears however that these inputs alone are insufficient to sustain much activity in the face of the strong local inhibition generated by their own action.

The third input component to the basal ganglia arises from a group of nuclei which are related to other brain systems that detect the rewarding or punishing quality of the stimulus pattern being decoded by the sensory systems. This input component has a very different distribution; it branches widely within the basal ganglia, each axon making synapses with tens of thousands of neurons. As a result, it cannot specify any very specific pattern of activation in the basal ganglia. Its action is diffuse, and principally temporarily coded. On the other hand, it can exert a widespread gating action on all ongoing basal ganglia activity. Thus, an input containing information about the intensity of the organism's emotional response to the results of ongoing behavior is capable of sustaining or inhibiting the next phase of the behavior. Given the self-quenching nature of activity in the basal ganglia, it is easy to envision a process by which a behavior "suggested" by the cortical and thalamic inputs is only sustained if the initial input results produce a sustaining input which strengthens the initial activation pattern, perhaps by summing with it to overcome the self-inhibition. The third input component is of course ideally situated for such a function.

In its most primitive form, this scheme results in a sort of "homing device" which will cause an organism to follow an increasingly intense stimulus, such as odor, to its source, such as food. That is, as the searching and locomotor patterns generated by the animal result in increases or decreases in the intensity of the pleasurable stimulus, they are appropriately facilitated or eliminated. Out of this simple feedback guidance mechanism, a host of more elaborate behaviors are developed, by evolution and learning, with the aid of the immense processing power of the cortex to provide detailed analysis of the environment and to generate more complex patterns of behavior for trial.

At the present time, we cannot precisely specify the pattern of detailed connections in the basal ganglia which results in these actions. The nature of its operation is inferred indirectly from evidence derived by stimulating its inputs or disabling its outputs. This evidence seems to establish that normal operation of the basal ganglia is essential to orientation and approach to stimuli, and initiation of voluntary behavior and complex learned behavior, particularly that involving anticipatory actions. The convergence in the basal ganglia of processed sensory information from many areas of the cortex provides a source of feedback information which can interact with and modify basic action plans generated by other cortical areas. Damage to the basal ganglia causes a loss of the ability to modify complex actions and judgements on the basis of sensory feedback. (This sort of feedback modification is distinct from the nonspecific *sustaining* action of feedback from the reward detector circuitry.) Finally, as predicted by the model outlined above, damage to the diffusely connected third input component results in failure to initiate behavior or orient to and approach stimuli, while stimulation of this component results in continuation of the immediately preceding behavior.

There is also a growing body of evidence to indicate that the type of learning called "operant conditioning" (see the preceding article in this series) may depend on, or even occur in the basal ganglia. This type of learning essentially involves an increase in the future ability of a behavior pattern to compete with other potential behaviors if it

is followed by activation of the reward system. To achieve this, all that would be required in addition to the basal ganglia model we have described here would be provision for activity in the diffuse input from the reward system to lower the firing threshold of neurons which were active at the time of this input. No such mechanism is presently known, although it is suspected, but in our robots it would be easily contrived.

An electronic analog of the model of basal ganglia action described here is shown in figure 5. (This model does not include the learning function just described.) The essential features are: the provision of a set of gates to encode the simultaneous inputs from the many cortical regions which contribute to the design of the behavior; a circuit which shuts off the encoded output after a brief delay; and an enabling bus representing the input from the reward system which inhibits the shut off circuit on active gates. This model is only illustrative, and better ones could be designed to mimic basal ganglia function. For example, the intensity of activity in the enabling bus should be employed to modulate the intensity of the output.

In practice, considering the very large number of gates required, and the fact that operation of the system is slow since it requires direction from physical results of actions, it will probably be best to simulate much of the gating and modulation in software on a fast processor. A few relevant principles are worth noting here. The ratio of input to output lines in the basal ganglia is very high. It receives input fibers from the entire cerebral cortex, which is by far the largest structure in the human brain. Output neurons on the other hand comprise less than five percent of the neural complement of the basal ganglia. Clearly a great deal of

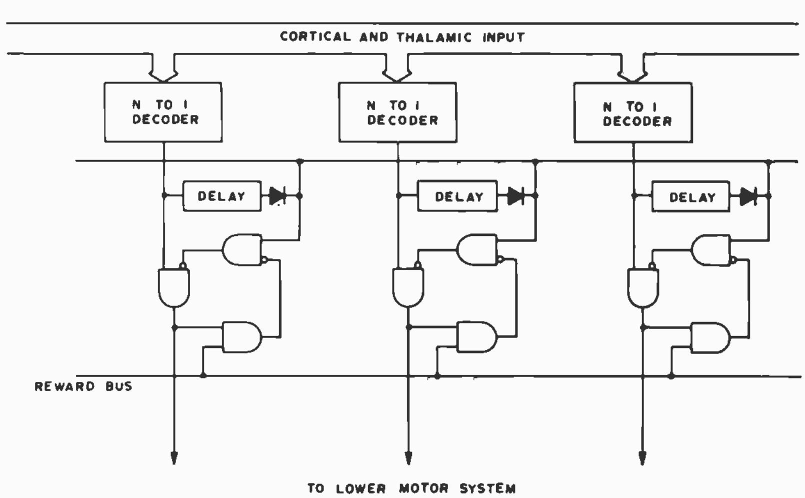


Figure 5: An equivalent block diagram approximating some of the basic relations of the brain's basal ganglia (striatum).

encoding takes place here; output line permutations are selected by gating an enormous number of inputs. Consistent with this, the outputs undergo an equally enormous decoding and fan out into the entire downstream motor system, ultimately specifying the actions of billions of LMN units. The basal ganglia outputs thus represent a "narrow spot" in the system, through which most of the organism's complex goal directed behavior passes. Similarly, the reward system which provides the gating or modulating input to this information flow represents the ultimate distillation of analysis of the entire sensory world of the organism as it pertains to reward. The amount of processing going on at higher levels to generate behavior patterns, and the amount required to evaluate their effectiveness is awe inspiring. Yet, the closing of this most complex feedback loop of all time is carried out relatively easily thanks to interaction at the "narrow points" of the two systems in a simple decision to keep going or quit doing what you're doing. The need for specific feedback to the behavior generating elements is thus eliminated. They simply try something else which they derive from established hierarchies or generate from similarities with past situations.

If we are to provide the capacity for robot behavioral systems to modify large scale behavioral strategies on the basis of evaluation of their effects, or if we wish to provide an operant conditioning capability, it will be necessary to gate or modify massive amounts of information. The most hardware conservative approach may well be to emulate the basal ganglia system by allowing a simple statement of the evaluative system's reaction to perform a "more or less" modulation of the output of the behavior generators at a highly encoded "narrow spot," and leave the behavior generators to try again according to trial and error algorithms, rather than trying to correct them directly. Specific feedback information of a nonevaluative sort, such as corrections to intended position from visual observation of the limb, become part of the command pattern prior to modulation by the evaluative system, simply by being part of the input pattern to be processed in generating the next attempted output patterns. These inputs could be handled by a software gating system, given processor speed, and the intensity of the evaluative function could be digitally coded and applied by software arithmetic rather than by mimicking the brain's analog system.

Having looked at the detailed operation of some of the important components of the brain's motor output system, let us finish

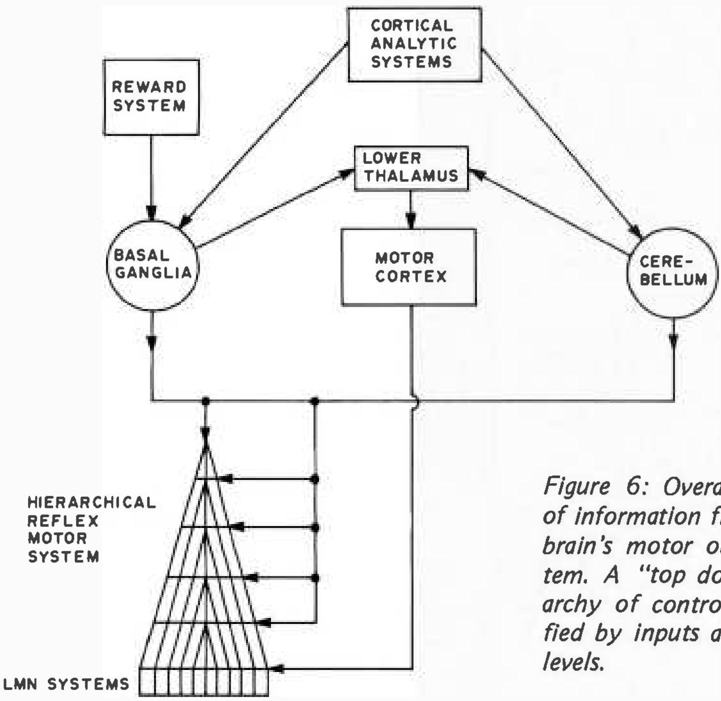


Figure 6: Overall pattern of information flow in the brain's motor output system. A "top down" hierarchy of control is modified by inputs at multiple levels.

with it by looking at a schematic summary which emphasizes the interactions of the different parts of the system. Figure 6 shows the main routes of information flow in the system, together with the major controlling inputs. Some of the "black boxes" such as "reward system" will be covered in future articles.

One of the outstanding features of the system taken as a whole is that it does function in an organized and integrated way, despite the fact that its parts are in many ways autonomous, and certainly not synchronized in their operation. A key to this capability is the provision of status information to each unit of the system by each of its neighors, and the ability of each to employ this information in an intelligent way in formulating its own output. A further refinement is the provision of a structure such as the cerebellum where status information from diverse systems can interact to generate correction information which returns into the main line of the relevant systems. Wide scale availability of information from special movement relevant sensory input systems is another unifying feature.

If we leave out the "behavior generating system," which is properly a decision making system to be considered later, not a system for execution, we can discern four major portions of the motor system (although some structures service more than one portion). The first is a system which handles most of the routine traffic according to established rules, and provides automatic elaboration according to established rules when given high level commands. The second is a system which converts parallel statements of action patterns into serially executed instructions to the first system. The third system provides a highly intelligent output terminal which can access the final output elements directly in the service of any of the higher systems on request. The essential feature here is that it is a parallel control for refined special purpose control, and is not necessary for most routines. Finally, a fourth system provides for interaction of the high level decision making systems with elaborately processed feedback information to generate complex instructions to the other systems, after screening them for effectiveness.

In this constellation of functions, we find the capability to deal with rapid emergency movements, automatic compensation for externally imposed deviations, fine graded control under the direction of any sensory input, and the execution of arbitrary novel patterns. The organizing principle which seems to best define the system is its emphasis on successively more abstract command functions at higher levels in the system, and a corresponding increase in "situation free" statements. That is, a high level element can issue a "walk" command without being concerned about the nature of the terrain. It has distinct analogies to high level programming languages. We shall see a similar organization in reverse in the sensory systems, where detailed information at the receptor level is gradually reduced to powerful statements of object recognition, independent of details of the sensation as the information ascends in the system.

Even with all of this elaborate apparatus to direct and coordinate body motion, the problem of movement in the generalized environment remains a challenging one. Despite the massive investment in processing power that the brain has devoted to the problem, we still fall down sometimes. Producing a robot system that even approaches the brain's abilities will be a great challenge.■

BIBLIOGRAPHY

Elementary

Llinas, R, "The Cortex of the Cerebellum," *Scientific American*, January 1975.

Advanced

Kornhuber, H, "Cerebral Cortex, Cerebellum, and Basal Ganglia: an Introduction to Their Motor Functions," *Neurosciences, Third Study Program*, Schmitt, F, and Worden, F, eds, MIT Press, Cambridge, 1974.

Languages Forum

Reactions to Previous Comments

Leigh Janes
23B Robbins Ln
Rocky Hill CT 06067

I'd like to comment on some of the items in your November 1977 Languages Forum, page 190.

I think Glen Taylor's idea for a "personal computer language development society" is great! I only hope he gets people whose minds are open enough to be able to borrow the strengths of any existing language. Alas, I am not experienced in microcomputers nor language design; the only thing I could offer is enthusiasm and a little experience in using some of the current languages (although I never did discover how to actually use LISP).

If Jeffrey Kenton can't offer Peter Skye anything other than a prophecy of failure, he should have saved his time and stamp. ("The proposed PL/Skye will make no one happy." You never know, Jeff, it might be ecstatic.) RPG on a micro? Super idea! (The premise is that anything which helps make any computer easier to use is a good idea.) Terrible early experiences with PL/I? That's no reason to quit; surely we can learn from that and do better in the next attempt.

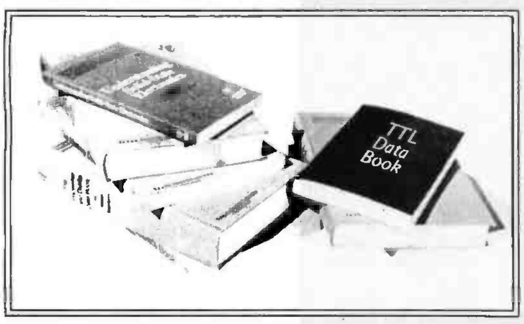
My only contact with PASCAL has been via *A Primer on PASCAL* by Conway, Gries and Zimmerman which leaves out a lot of stuff (because it is really a primer on programming which merely uses PASCAL for its examples). My objections to what I've seen of PASCAL are: the apparent necessity of "declaring" *every* symbol in the program before using any of them; the apparent requirement of *numbering* a statement to go to (I want to name it); and the clumsiness of character string handling.■

A BIT Of The Best

____**Build Your Own Working Robot** by David L Heiserman, published by Tab Books. This book will not tell you how to build Robbie, the robot of **Forbidden Planet**, or a classical android of science fiction. What it will introduce you to is the problems of making a robot mobile device called Buster III, using pre-microprocessor TTL integrated circuits for all logic functions. It is a must book for background reading, but much of the logic can be extremely simplified using today's microprocessor technology. Use this book as a first look at these problems from which you can build further and more elaborate solutions. Softbound, $5.95.

____**Getting Involved With Your Own Computer** by Leslie Solomon and Stanley Veit answers the questions: "What can small computers do? Which is best for my purposes?" Whether your interest is business applications, word processing, education, security, etc., this lucid text will bring you in touch with an exciting new world destined to affect us all. $5.95.

The Texas Instrument Reference Library



____Fingertip Math, $2.95
____Software Design for Microprocessors, $12.95
____The Great International Math on Keys Book, $4.95
____TTL Data Book, $4.95
____Semiconductor Memory Data Book for Design Engineers, $2.95
____Transistor and Diode Data Book, $4.95
____Power Semiconductor Handbook for Design Engineers, $3.95
____The Optoelectronics Data Book for Design Engineers, $2.95
____Understanding Solid State Electronics, $2.95
____Designing With TTL Integrated Circuits, $28.50
____Bipolar Microcomputer Components Data Book, $2.95

NEW!

____**Computational Analysis on the Pocket Calculator** contains some 35 programs written for a specific programmable pocket calculator, the HP-25. These programs implement algorithms in number theory, equation solving, algebraic stability theory, calculus of power series, numerical integration as well as algorithms for the evaluation of special higher transcendental functions, such as the Gamma function, various Bessel functions, the error integral, and the Riemann zeta function.

A unified format has been chosen for the preparation of the programs. By means of the flow diagrams and the detailed descriptions that are provided, the programs are easily adapted to run on any calculator of comparable capacity. $11.50.

____**The First West Coast Computer Faire Conference Proceedings.** Here's a big package of fascinating information and practical guidelines from the symposia held at the 1977 West Coast Computer Faire. **The First West Coast Computer Faire Conference Proceedings** contain 336 pages covering: tutorials for the computer novice; human aspects of system design; robots (including the text of a talk by science fiction writer Fred Pohl); computers for the physically disabled; legal aspects of personal computers; education; electronic mail; music with computers; hardware; software— the list goes on. This compendium will make a useful addition to your reference shelf, since it contains many hard-to-get items. You can't miss for $12!

____William Ralph Bennett Jr's **Scientific and Engineering Problem Solving With The Computer** is one of the most exciting books we've seen in years. Besides teaching BASIC (which it does admirably), this lively, lucid book presents a wealth of imaginative and unusual applications programs taken from many disciplines (A sample exercise: "Using the algorithm in the text with the pair-correlation matrix from Hamlet, compute the most probable diagram path which starts with the letter T"). The exercises run the gamut from random processes to the dynamics of motion, from entropy in language to the Watergate problem. You'll discover BASIC applications in lasers and in the Fourier series and the law (!). In its diversity and elegant style, it ranks with Donald Knuth's works as a milestone in the art of computing. Hardcover, $13.95.

____Scelbi "6800" Software Gourmet Guide & Cookbook

____Scelbi "8080" Software Gourmet Guide & Cookbook, both by Robert Findley. Have you tried cooking up a program lately on your 6800 or 8080 processor? Have you needed a dash of ideas on how to add spice to a program? Then the **Scelbi "6800" Software Gourmet Guide & Cookbook** and the **Scelbi "8080" Software Gourmet Guide & Cookbook** may prove to be quite useful additions to your library.

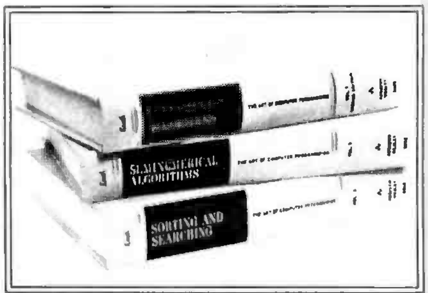
Both books contain a wealth of information on the 6800 and 8080 instruction sets, plus general programming techniques applied to the 8080 and 6800, conversion routines, floating point routines, decimal arithmetic routines, and much more. Order yours today and get a taste of what your processor can do. Both books are priced at $9.95 each.

Old Favorites

____**Basic BASIC** by James S Coan. If you're not already familiar with BASIC, James Coan's **Basic BASIC** is one of the best ways to learn about this popular computer language. BASIC (which stands for Beginner's All-purpose Symbolic Instruction Code) is easy to learn and easy to apply to many problems. **Basic BASIC** gives you step-by-step instructions for using a terminal, writing programs, using loops and lists, solving mathematical problems, understanding matrices and more. The book contains a wealth of illustrations and example programs, and is suitable for beginners at many different levels. It makes a fine reference for the experienced programmer, too. $7.95.

____**Advanced BASIC** by James S Coan. Advanced **BASIC** is the companion volume to James Coan's **Basic BASIC.** In this book you'll learn about some of the more advanced techniques for programming in BASIC, including string manipulation, the use of files, plotting on a terminal, simulation and games, advanced mathematical applications and more. Many useful algorithms are covered, including some clever sorting techniques designed to reduce program execution time. As with **Basic BASIC**, there are many illustrative example programs included. BASIC doesn't have to be basic with **Advanced BASIC**! $6.95.

The Art of Computer Programming



Praised by many critics as the best books in their field, **The Art of Computer Programming, Volumes I, II and III**, are part of a projected seven volume omnibus survey of computer science now being completed by Donald E Knuth.

____Volume I, **Fundamental Algorithms**, begins with a thorough discussion of the mathematics used in computer programming, followed by a treatment of information structures, stacks, arrays, linked lists, dynamic storage allocation, and trees. 634 pp; $20.95.

____Volume II, **Seminumerical Algorithms**, is concerned with random numbers, statistical tests, random sequences, as well as arithmetic (floating point and multiple precision), polynomials, and rational arithmetic. 624 pp; $20.95.

____Volume III, deals with **Searching and Sorting**, and as the name implies, the emphasis is on algorithms for sorting, including combinatorial properties of permutations, internal sorting, optimum sorting, and external sorting. Also included is a section on sequential searching, hashing, digital searching, and more. 722 pp; $20.95.

A hypothetical assembly language called MIX has been developed by the author to illustrate programming examples throughout the series. MIX is easily convertible to other assembly languages.

Prof Knuth writes with style and wit (among many memorable quotes is one from *McCall's Cookbook*!). This classic work belongs on the reference shelf of everyone seriously interested in computer science.

Order By Telephone

WATS up? The BITS **toll free hotline** is now available for customer orders. Dial your bank card orders Monday through Friday between 9:00 a.m. and 5:00 p.m. Eastern Standard Time. **1-800-258-5477.** In New Hampshire, call 924-3355. Before you dial, please have ready your BankAmericard or Master Charge credit card number and expiration date, the titles of the books you wish, and your shipping address.
To call about an order—Please dial (603) 924-3355.

DIAL YOUR ORDERS ON THE BITS TOLL FREE HOT LINE: 1-800-258-5477.

In New Hampshire, call collect: 924-3355

Send to:
BITS, Inc.
70 Main Street
Peterborough NH 03458

Check Payment method:
____My check is enclosed
____Bill my MC No.____________ Exp. date ________
____Bill my BAC No.____________ Exp. date ________

Name ____________________

Address ____________________

City ________ State ________ Zip Code ________

Signature ____________________

Total for all books checked $________

Postage, 50 cents per book for ______ books $________

Overseas, 75 cents per book for ______ books $________

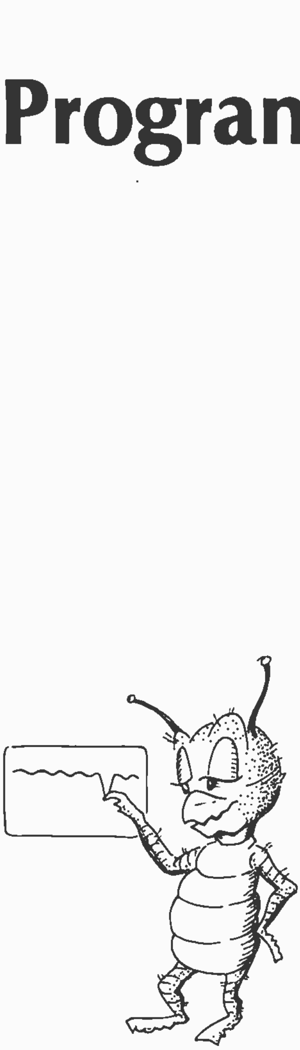
Grand Total $________

Prices shown are subject to change without notice.
All orders must be prepaid.
In unusual cases, processing may exceed 30 days.

You may photocopy this page if you wish to leave your BYTE intact.

Programming Entomology

Gary McGath
7 Silver Dr
Nashua NH 03060



Clobbered Value Bug: Your program changes the value of a variable at a time and place which is unintended. The detection difficulty ranges from the obvious (after if is found) to the subtle (before it is found).

An entomologist is a bug expert. When he sees an insect, it isn't just a bug to him (in fact, he will vociferously protest that not all insects are bugs); it has a particular habitat, lifespan, favorite food, and breeding pattern. Nor is his knowledge just academic; he can tell you how to protect yourself from a harmful one by killing it or keeping it away.

The same sort of knowledge is necessary for programming. The skilled programmer knows what kinds of bugs may attack a program, how to track them down, and how to keep them from getting there in the first place. He knows the ways to get at particular bugs, as well as the general treatments which are effective against all of them.

The first thing to realize about bugs is that they don't appear by spontaneous generation. They have a creator, and their creator is the programmer. (Throughout this article, I am speaking only of user program bugs; hardware bugs are an entirely different breed, subject to different laws, and systems software may be beyond your control. No matter how outrageously the program is acting, it's only following orders. So what you have to ask about a bug in your program is: how did *you* put it there? What kinds of mistakes are you prone to make? If you caught a certain bug in one part of the program, might you have put the same kind of bug elsewhere as well? "Thou art God" . . .and thou must take care of thy creation.

But the fact that each programmer creates his own bugs doesn't mean there aren't species of bugs found in everyone's programs. Knowing about these species can be a great timesaver, especially when the species can be identified by the effects.

One of the most common bugs is the Clobbered Value, found where the programmer assumes the content of a register or the value of a variable is the same as before, but it isn't. Take this attempt to exchange the values of two variables:

```
10 LET X = Y
20 LET Y = X
```

This fails because when statement 20 is executed, the value of X has already been clobbered by the previous statement, with the result that Y never gets changed at all.

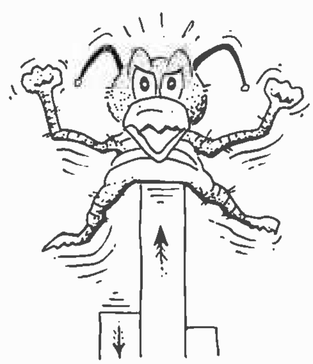
Clobbered Values are frequently found on subroutine exits. It's easy to write a harmless looking CALL or GOSUB (possibly to a routine you haven't written yet) and assume everything will remain the same. But strange things can happen if the subroutine unexpectedly changes some values.

A not too distant relative of the Clobbered Value is the Zapped Stack, found only in machine and assembly code. It appears most often by pushing items onto the program's stack at the start of a subroutine, then failing to pop them, or popping too many things at the end. Another way to invite this bug is to use the stack pointer for some other purpose during the course of a subroutine.

Subroutines are also the habitat of the Botched Call. A certain protocol is needed to call any particular subroutine. If, when you write a call to a subroutine, you expect a value to be returned in the wrong place, or you assume the subroutine will do something which it actually won't (or vice versa), this bug will have gained a foothold. The difference between a Clobbered Value and a Botched Call is that when you have the latter, the subroutine is doing the right thing; the calling program is just mistaken in its expectations.

Another species of bug lurks in jumps, branches, and GOTOs. The Branch Bug

Zapped Stack Bug: Stack oriented machines and software are both very egalitarian with respect to pushes and pops. They like to have the same number of items pushed as are later popped, or else they'll transform themselves from tranquil and placid programs into memory zapping monsters.



is so difficult to fight that serious attempts have been made to wipe out its habitat; languages and programming styles (structured programming) have been developed that use no jumps. The Branch Bug comes in two varieties: jumping to the wrong place, and jumping to the right place with inadequate preparation. The first of these is easy to produce in languages where statement labels have to be numbers (eg: BASIC and FORTRAN, especially BASIC, where every statement has to be numbered whether it's ever going to be a jump destination or not). The jump with inadequate preparation is similar to the Botched Call, but it can often be harder to figure out if the program has a complex flow pattern.

A few special methods are applicable to fighting the Branch Bug. One of these is program flow analysis. A look at the possible paths a program can take will often reveal some of these bugs. Is there a part of the program that can never be reached? Are there traps in the program, loops that can never terminate? Are there jumps which will result in variables being used without having been set to a value?

In languages like BASIC, where every statement is labeled, it's helpful to set off statements that can be reached by jumps either by using special statement numbers or by pointing them out in comment statements. In any language, the statements that can be reached by jumps should be logical breaking points in some sense, places where a new unit of work begins. Except in desperate situations where economy is all-important, jumps should be used to satisfy the logic of the program, not to save a few instructions.

If a subroutine call can be used instead of a jump, it probably should be used. A subroutine will send you back where you came from, so figuring out the flow of the program is easier. For many purposes, you can treat a subroutine as a unit when studying the program; as a single instruction that happens to do complicated things. You can't do this with the instructions reached by a jump.

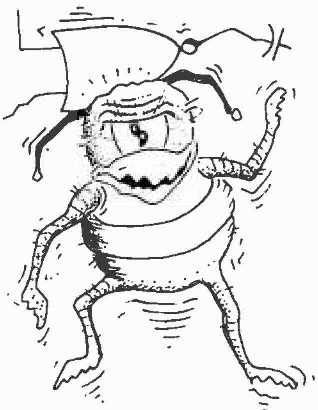
The next bug in our survey feeds on apples and oranges. More generally speaking, the Mismatched Unit is found where the units or dimensions of the quantities being used in a program aren't the ones actually needed. Take the program statement LET V = D * T, where D is a distance in miles, T is the time traveled in hours, and V is intended to be the traveler's average velocity in miles per hour. By using simple algebra on the units, you can see that the result obtained will be units of miles *times* hours, not miles per (ie: divided by) hour.

Bugs of this type are harder to spot when the mismatched variables are further apart in the program, but consistency will keep them from occurring. Simply be sure you know in advance what units each variable has to come in.

Assembly and machine language programming allow an especially messy type of Mismatched Unit to show up: mismatches between addresses and data, or between absolute addresses and relative addresses (values to be added to a base address). To avoid this bug, watch out for the different addressing modes of different instructions.

Another bug with a specialized habitat is the Fencepost Bug, named for its tendency to rest in problems like this one: "If you are putting up a wire fence 100 feet long, supported by posts every 10 feet, how many posts do you need?" Another name for this bug is the Boundary Condition Bug; it's always found in connection with the start or end of some sequence, where

Botched Call Bug: The Botched Call Bug is like the proverbial square peg in a round hole: Unless the peg or the edge of the hole yields, sparks will fly.



Branch Bug: Jumping blindly about in memory, the Branch Bug is always on a collision course with valid execution of a program.

special treatment is needed. One form manifests itself in confusion over whether the first element of a group is number 0 or number 1. Another is found in the attempt to relate each element of an array to the next, as in this statement:

```
IF T(I) <T(I+1) GO TO 100
```

Try this one setting I equal to the dimension of T.

Finally, we come to the most insidious of all bugs, the Timing Bug. The characteristic that makes this bug so fearsome is that a program infested by one may run correctly once but not the next time; it may even run correctly 99 times but fail on the hundredth, using exactly the same data each time. To make matters worse, running programs in single step mode will usually drive Timing Bugs into undetectable hiding.

As the name suggests, the Timing Bug is one that shows up depending on the order in which asynchronous events (events that have an unpredictable relationship in time) occur. Systems that have interrupt facilities are especially prone to being attacked by Timing Bugs, since an interrupt routine may be executed at a different point in the program each time it's run. An interrupt routine may, for instance, set up certain variables to be used by the main program. If another interrupt of the same kind can occur before the variables have been processed by the main program, and if that interrupt changes those variables, unpredictable results can occur. Yet most of the time, interrupts may not occur that close together, so the bad result is said to be nonrepeatable. This means that repeated runs of the program can't be used to systematically close in on the bug.

A Timing Bug can also live on direct memory access (DMA). Some mass storage devices can read or write data in bulk without the intervention of the processor, using those memory access cycles which the processor doesn't use.

Mismatched Unit Bug: A result of inadequate analysis of a calculation, the Mismatched Unit Bug results in strange elixirs. When both apples and oranges are thrown into the analytical engine, what is the nature of the juice which flows out?

The length of time a DMA transfer will take is, at best, very difficult to predict; so a Timing Bug can strike if memory which is accessed by DMA can be accessed or modified by the processor.

Since Timing Bugs are so hard to hunt down, extra efforts should be made to avoid giving them a foothold. Be extra careful in writing interrupt handlers or DMA commands. Watch for places where interrupts need to be disabled. As for the indentification of Timing Bugs, the following rule is useful: if you can prove, in a precise instruction by instruction study, that what happened couldn't possibly have happened

The Timing Bug: This most subtle of all bugs spends most of its time relaxing, and suddenly taking a swipe at apparently random times.

from the execution of those instructions, suspect a Timing Bug; something else was happening during the execution of those instructions.

Incidentally, it's possible to encounter bugs much like Timing Bugs even without interrupts or DMA. An input or output device, such as a keyboard, is asynchronous with the program; the exact behavior of the program will depend on the behavior of these devices. For instance, a program which accepts keyboard input and accumulates it in a buffer may work fine for you, yet a faster typist may make it fail because no provision was made for the chance of exceeding the buffer's capacity. But in a situation like this, it's at least possible to look at every call to an input routine and tell what its effects might be.

This completes our survey of important species of bugs (I have nothing useful to say about the Common Typo, though it does have to be fought). Others will no doubt discover voracious breeds which I have overlooked, and perhaps they will improve on some of the classifications I have mentioned. But knowing about the species which are listed here will hopefully be a help in identifying and killing the bugs in your own programs.

This doesn't mean that classifying bugs is all there is to entomology, neither the biological kind nor the kind being discussed here. Entomology wouldn't be a science if it couldn't say things that are true of all bugs, regardless of species. What I have discussed so far is differentiation; but integration is equally important.

The basic fact that unifies all bugs is the one which I mentioned at the beginning of this article: they're all creations of the programmer. And this fact allows the use of a broad-spectrum killer against all bugs: DDT, standing for Design, Documentation, and Testing. Let's take them in order:

- Design. The best way to stay bug-free is to write programs without bugs. This may sound like superfluous advice, but programmers (myself included) are often tempted into writing programs quickly, rather than writing them well. The attempt usually fails, since such programs will usually cost more in debugging time than the time saved in writing them.

An error born of pragmatism is to suppose that it doesn't matter how you design a program, as long as it works. There are two problems with this idea. The first is that if you use any method that appears to do the job, without regard for well organized design, it will be a lot harder to ever make the program work. The second problem is that even if the program works for its immediate purpose, it will be harder to make changes to meet new needs, since a particular ad hoc solution may not be generalizable.

The first step in designing a program is to lay out a complete plan of attack before writing it. Decide what data structures you will need, and what method you will use. Data structures are often the key to the whole program. First plan the program in a few large steps; then decide what each step will consist of in more specific terms; then repeat the procedure until you're down to the level of your chosen programming language. This is the principle of structured programming, and also of mental unit-economy: avoid having to think about more things at once than your mind can handle. If you can keep everything relevant to a particular operation in your head, you're not likely to put bugs into its implementation.

Flowcharting is often recommended for program design, but it's cumbersome and doesn't lend itself to representing a hierarchical design. Another approach is to use a well designed programming language, such as ALGOL or APL, to write the design. Since you aren't actually going to run the program in that language, you can assume any features that would make the job easier. The point of this is to have a representation of the program that you can understand without strain, so that you don't lose sight of your overall plan while chasing down details of implementation. If you do have bugs after doing this, at least they won't be part of the whole design of the program.

- Documentation. The main reason for writing up the way a program works isn't to explain it to someone else; it's to make sure you understand it yourself. Documentation shouldn't be an afterthought; it should begin with the design of the program (when you write what it is going to do), and continue with comments written along with the instructions.

Good documentation isn't found in sheer number of comments (though there should be a lot); it's found in comments that explain the operation of the program. Comments are especially needed for data, subroutines, and points reachable by jumps. Variables and constants should be explained so that the reader will see how they can be used; this allows us to spot threats to them, such as Mismatched Units and Clobbered Values. If the language allows, give constants names rather than using their numeric values throughout the program; this makes updating easier and renders the Common Typo's attacks more conspicuous. Subroutines should be prefaced with a description of how they are called, what inputs are needed, what values are returned, and what information may be destroyed in the process. Jump points should have an explanation of the conditions under which they are reached.

To make a program at least partly self-documenting, the name of a routine or variable should indicate its use. One of the major weaknesses of BASIC is that it doesn't allow this to be done very much; this is a reason for having a lot of comment statements to explain what BASIC variables and subroutines are used for.

Just as a sample, here's a preface to a hypothetical 8080 assembly language subroutine (see box). The comments explicitly define linkage conventions.

```
;   COMPUTE PROBABILITY OF WIDGET BREAKAGE
;   INPUT – MASS OF WIDGET (GRAMS) IN REGISTER PAIR BC
;           AGE OF WIDGET (DAYS) IN REGISTER PAIR DE
;   OUTPUT – PROBABILITY OF BREAKAGE (PERCENT) IN REGISTER PAIR BC
;   ALL OTHER REGISTERS ARE CLOBBERED
```

The protection provided against Botched Calls should be obvious.

• Testing. If you follow the approach outlined so far, you'll have a better chance of getting your program to work, but you may still have planted a few bugs inadvertently. So you have to test the program before declaring it bug-free. Testing should begin with a simple version of the program, if possible; but it should begin only after the program has been written with enough care so that there's a chance of not finding any bugs.

Use whatever debugging tools are available. High-level languages will usually provide useful information when the program goes wrong. Versions of BASIC that allow single statements to be executed make it possible to find something about the conditions under which an error occurred.

When working in machine language, a debugging program will ease discovery of bugs. Such a program allows the user to put breakpoints into the program being tested (returning control to the debugger when the program counter reaches a certain address) and to examine and modify registers and memory. These programs range from simple 1 K monitors to powerful symbolic debuggers like Digital Equipment Corporation's DDT (Dynamic Debugging Tool, no relation to the name as used here). Having one of these in ROM can be a tremendous help.

If the program works the first time, try it again with different data to make sure. Check out simple cases. Sometimes a program will work in complicated cases, but be bitten by the Fencepost Bug in simple ones. Check out more complicated cases. If possible, use a random number table as a source of test data, along with handpicked cases.

If the program *doesn't* work the first time, try it again with different data. Aim for the simplest case possible. If you can get the program to do something right, that will cut down the number of places where bugs may be lurking.

When a program is being tested, the work is easiest if execution comes to a screeching halt as soon as something goes wrong. A program may be able to run a while after crucial damage has occurred, only to clobber all of memory before stopping. If this happens, it can be almost impossible to localize the source of the disaster. But if the program makes periodic checks for error conditions (such as impossible values or invalid relationships) and reports them, there's a better chance of discovering just where things went wrong. For instance, a routine that fills a block of memory between two addresses might check to make sure that the low address is really lower than the high address. Redundant tests may slow down the program, but they can be taken out when all the bugs are known to be dead.

The overriding consideration to remember in the use of this Design, Document and Test technique is that it's open-ended. It will, in principle, kill any kind of bug; but a new approach to design, a better scheme of documentation, or a novel test may be needed for subtle species. Approaching bugs scientifically means thinking about them. It means recognizing that any bug will have important similarities to previously encountered bugs; and that it may have equally important differences. So when you find yourself struggling to discover what's wrong with a program whose behavior is incomprehensible, you can console yourself with the thought that you may be about to make an exciting entomological discovery that you can use repeatedly.■

Technical Forum

Measuring Program Size

Stefan T Dobrowolski
4419 N Pershing Dr, #1
Arlington VA 22203

In March 1977 BYTE (page 106) David Price presents a Star Trek program which is apparently written in Hewlett-Packard's BASIC, and he states that the program is 9382 bytes long. This figure can be misleading in that, unlike many computers, HP does not store programs in source code, but rather in an internally coded form. Thus the length of a program is dependent on factors different from ones which would be relevant on another system.

For example, a constant uses two words of memory any time it occurs in the program, but a variable uses no more memory if it occurs 50 times than if it occurs only once. Hence a program can often be significantly shortened by setting variables equal to commonly used constants at the beginning of the program, and thereafter using these variables instead of the corresponding constants.

On the other hand, while eliminating nonessential spaces from a program can decrease its length in many other BASICs (eg: Honeywell's), it will have no effect on an HP BASIC program. When a line of source code is entered on an HP, the computer translates it into the above mentioned internal code, ignoring irrelevant spaces, leading zeros in line numbers, etc. Then if the line is listed, the computer "uncompiles" this code and prints the line in a standard format. This is why there are always two blanks after the line number, and a blank preceding and following any keyword (except REM and implied LET) or multicharacter operator.

It is natural for any computer language, even if initially standardized, to evolve into a collection of dialects. Every system has different requirements and resources. Unfortunately the hobbyist can run across programs written in many different dialects, often without having any easy way of judging how they need to be modified to run on his or her system. A possible (partial) solution might be to have an article or series of articles in BYTE comparing the better known versions of BASIC, both those used on microcomputers and the ones found on the more popular timesharing systems. In addition to providing hobbyists with information to help them convert programs written in foreign BASICs, it would also provide some insight on how different systems handle the execution of programs.■

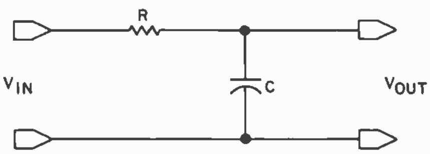


Figure 1: Simple resistor-capacitor (RC) low pass filter.

A Simple Digital Filter

Robert Grappel
148 Wood St
Lexington MA 02173

At first glance, using a computer to build an analog filter seems like the height of overkill. Imagine an analog to digital converter, a microprocessor with memory and peripheral interfaces, a digital to analog converter, and more, just to do the job of a capacitor, a resistor and a coil of wire! People have been building analog filters for years without computers. How can an analog filter be constructed with a digital device like a microcomputer? Digital filtering may be the answer.

All right, you say, we must have some analog to digital conversion stage in there. True enough. A digital filter uses periodic samples of an analog waveform as input. These samples are digitally manipulated in a computer of some sort and then converted back to analog form by a digital to analog converter. One input sample is converted to one output voltage. By integrating this output sequence, the digital process appears to have a continuous analog output.

The rest of this article describes the manipulations we must perform on the digital samples in order to simulate the performance of simple filters. The mathematics is quite complex and is available in many texts. We will try to concentrate here on the practical implementation of such a filter.

Let us take for example the simple RC (resistor-capacitor) low pass filter shown in figure 1. This circuit passes frequencies in the input voltage that are lower than some critical frequency (called the cutoff frequency) determined by the resistor and capacitor values, while severely attenuating any higher frequencies.

The output voltage V_{out}, then, is simply a selectively reduced version of V_{in} (input voltage). The resistor drops the output voltage, as does the voltage that goes into charging C. At first, $V_{out} = a \times V_{in}$ (where the constant a is the amount of attenuation caused by R and C). It can be shown that a=1/RC (where R is in ohms and C is in farads).

The voltage change with time across a capacitor is an exponential function. If the voltage at time zero is V, then the voltage at time t is given by Ve^{-at} (a is the same constant as above).

Now, let us consider that the input is a series of samples, where samples occur every t seconds. The output voltage at any given time is just the algebraic sum of the attenuated input voltage and the voltage on the capacitor at that time. In other words:

t=0	output=aV
t=1	output=aV +aV e^{-at}
t=2	output=aV +aV e^{-at} + aVe^{-2at}

After a number of samples, the output voltage becomes the sum of a number of

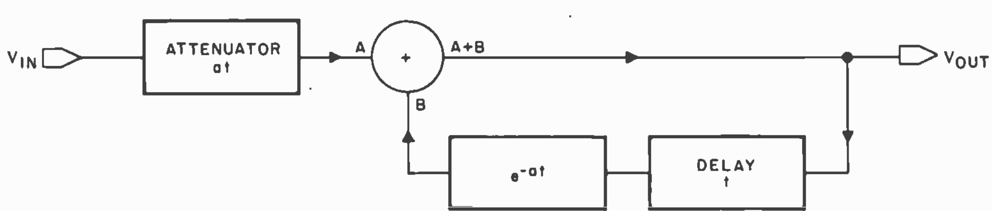


Figure 2: Filtering process performed by the circuit of figure 1 in block form.

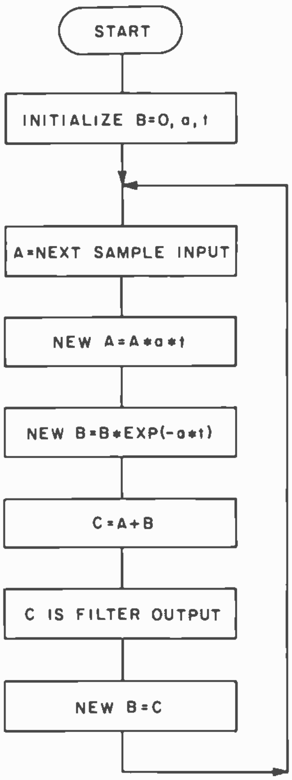


Figure 3: Flowchart for performing a low pass filter function.

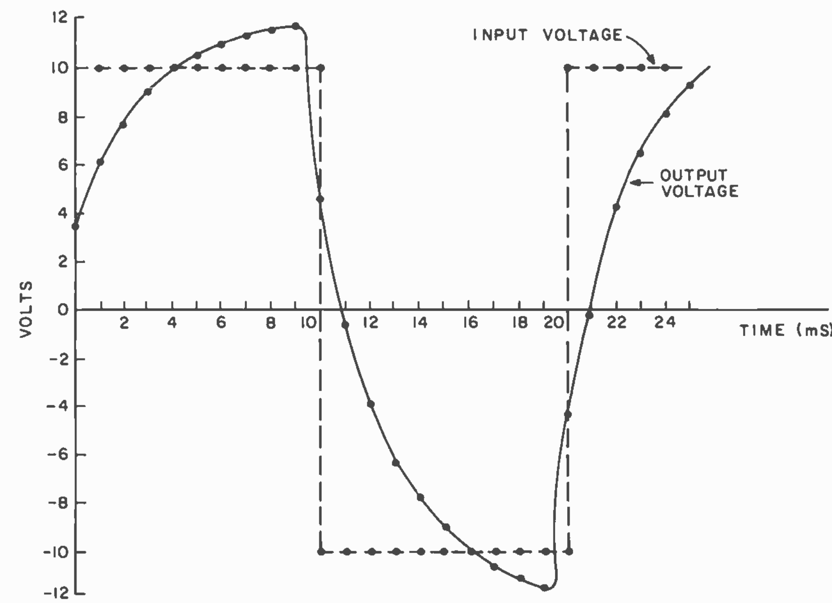


Figure 4: Result of passing a 50 Hz square wave through the digital low pass filter program.

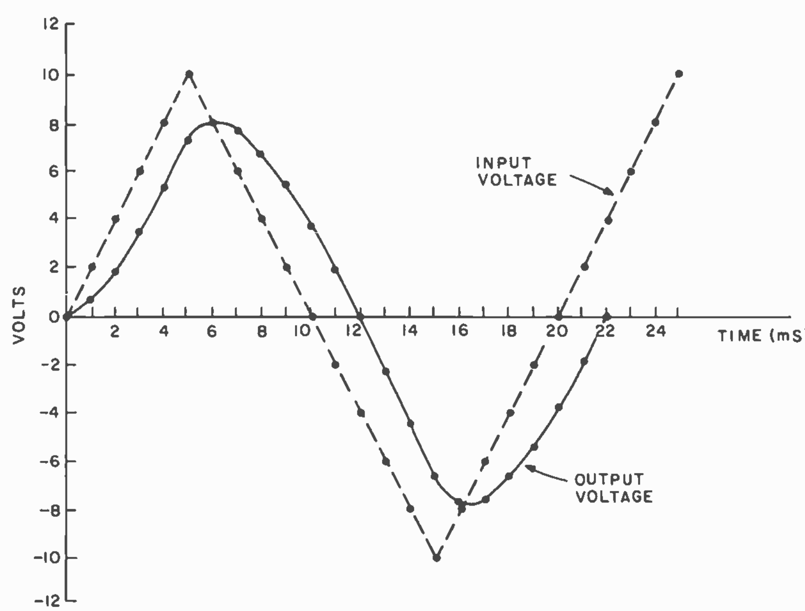


Figure 5: Result of passing a 50 Hz triangular wave through the low pass digital filter program.

such terms. Since the terms are similar, and $e^{-nat} = (e^{-at})^n$, we need to compute the exponential term only once, and multiply it by itself repeatedly to get the extra terms of the series. Figure 2 shows the equivalent circuit of the RC low pass filter in figure 1, shown in block diagram form. It consists of an attenuator (factor a), an exponential and time delay (the capacitor), and a summer. We them jump to figure 3, which shows a flowchart of this process. The only change is that the attenuation factor is given as at, where t is the period between samples. Much of the input to the filter is lost because its frequency is outside the filter range. This does not change the shape of the output (the important factor), but only the magnitude of the output (like adding gain to the filter). If you have accepted that last bit of sleight of hand, you can see that figure 3 is a block diagram of a program to perform low pass filtering.

Figure 4 shows the output of such a digital filter program when the input is a square wave with maximum and minimum values of +10 V and -10 V, respectively, and a frequency of 50 Hz. The sampling rate is 20 samples per cycle, thus t=1 ms. The RC constant a is arbitrarily set equal to 360. The removal of the high frequency components

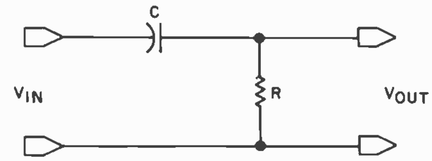


Figure 6: Simple resistor-capacitor (RC) high pass filter.

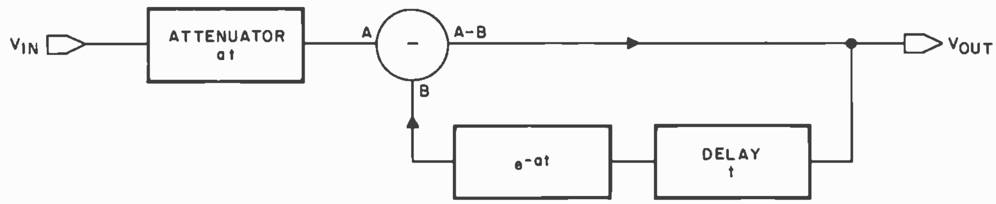


Figure 7: Filtering process performed by the circuit of figure 6 in block form. By changing a sum to a difference we convert the block diagram from a low to a high pass filter.

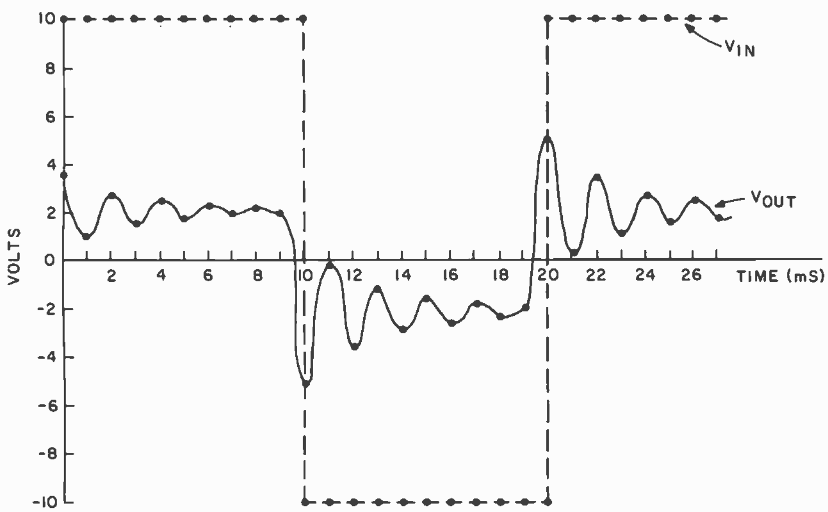


Figure 8: Result of passing the 50 Hz square wave used in figure 4 through a high pass digital filter program.

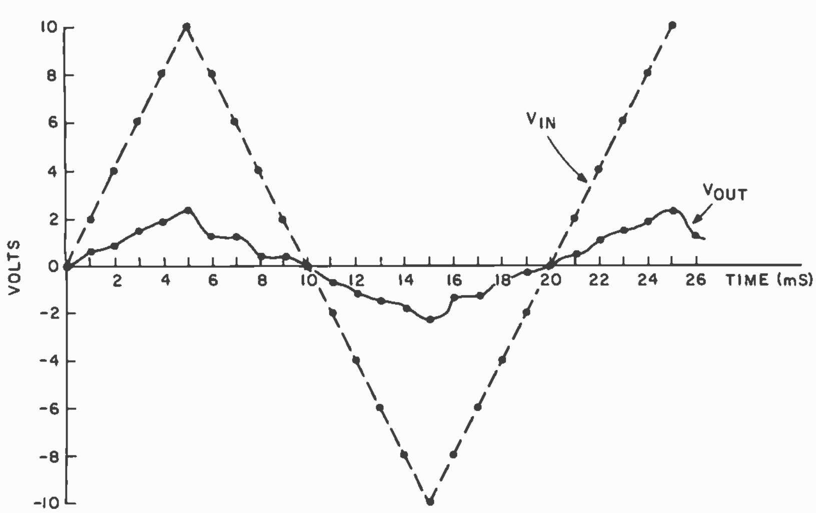


Figure 9: Result of passing the 50 Hz triangular wave used in figure 5 through a high pass digital filter.

of the square wave is evident. Figure 5 shows the filter response to a triangular wave at the same frequency. The output is nearly all at the fundamental frequency of the input, as one would expect of a low pass filter.

Now, how do we simulate high pass filters like the one in figure 6? Here the output voltage is the difference of the attenuated input and the capacitor voltage (since the capacitor resists rapid changes in its charge). Thus, figure 7 is a simulation of a high pass filter. It is identical to the low pass filter except that A+B becomes A−B.

Figures 8 and 9 show the response of a high pass filter program to the same inputs as those in figures 4 and 5. The constants are the same, but the change in output is striking.

One of the main features of software instead of hardware implementation is the ease with which software can be modified. A low pass filter changes to a high pass filter by changing an add to a subtract! Filter constants can be easily modifed, amplification added, etc. More complicated filters, such as band pass types, can be simulated by combining appropriate high and low pass feedback loops with adders and subtracters. The filters can be dynamic, adapting to the input. They can be programmed. This would seem to suggest uses in computer generated music systems, audio processing, removing noise from signals, etc. If one has to convert analog signals to digital form at some point in a design, then it may be useful to do much of the filtering at the digital level.

How can you perform digital filtering in real time without an exponential routine in your computer? The answer is that the exponential function is a constant. It can be programmed in, or provided by a table. The critical points are the two multiplications: one by at and the other by e^{-at}. Both of these values are usually less than 1 for a real filter. (In the examples, at=.360 and e^{-at}=.697.) What we can use is a routine called a *fractional multiply*. This is a routine that multiplies two values, one treated as an integer, and the other treated as a binary fraction.

One such routine, reproduced in listing 1, was written by Ira Chayut. For a detailed discussion of how it works see Programming Quickies, page 124, September 1976 BYTE. This 16 byte subroutine forms the heart of a digital filter program for the 6800. In the program of listing 2 this subroutine is given the name FRACMUL. An analog to digital converter and sampler is assumed to provide 7 bit samples available at the location SAMPLE. The routine FILTER (see listing 2) is set up as an interrupt handler. A periodic interrupt is provided which initiates the sampling and causes a branch to the routine. Since the Motorola processor takes a minimum of 12 μs to respond to an interrupt, a fast analog to digital converter should be ready with a new sample by the time the program needs it. Thus, no delay loops or tests are performed.

FILTER assumes that the analog to digital converter is ready when it is. It is also assumed that the location BVAL is zeroed initially. BVAL is the output of the filter. The constants in the listing are those of the examples (92≅.360 x 256, and 179≅.697 x 256). FILTER should execute in about 300 to 400 cycles on each interrupt. Assuming a 1 ms sampling period, FILTER will consume about 33% of the time of a 1 MHz 6800 processor. The memory occupied is negligible. FILTER is a practical program for use with audio frequencies. Listing 3 shows how easy it is to make FILTER either a high or low pass filter.

Of course, more complex filters will be harder to design and will take more processing time. For really interesting filter applications, an external hardware multiplier will probably be needed, but such circuits are available reasonably, and they can be used for other applications in the computer system when not filtering.

Now, get in there and attack the math! It really isn't all that hard.■

```
FRACMUL STAA ARG1        ;ARG1:=A
        CLRA             ;A:=0
MLOOP   LSR  ARG1        ;ARG1:=ARG1/2
        ASLB             ;CY:=MSB(B); B:=ASL(B,1)
        BCC  NONADD      ;IF CY=0 THEN SKIP THE ADDITION
        ADDA ARG1        ;ELSE A:=A+ARG1
NONADD  BNE  MLOOP       ;IF ARG2 NE 0 THEN REITERATE
        RTS              ;ELSE RETURN WITH RESULT IN A
ARG1    RMB  1           ;SINGLE BYTE TEMPORARY DATA AREA
```

Listing 1: The fractional multiplication routine. The result is returned to the main program in the accumulator.

```
        CLR  BVAL        ;INITIALIZATION
FILTER  LDA  A SAMPLE    ;A:= A/D SAMPLE (7-BITS)
        LDA  B #92       ;B:= 'AT'
        JSR  FRACMUL     ;MULTIPLY
        STA  A TEMP      ;TEMP:= ANSWER
        LDA  A BVAL      ;A:= BVAL
        LDA  B #179      ;B:= E (LN BASE) RAISED TO THE (-AT) POWER
        JSR  FRACMUL     ;MULTIPLY
        ADD  A TEMP      ;SUMMATION FOR LOW PASS FILTER
        STA  A BVAL      ;RETURN ANSWER
        RTI
BVAL    RMB  1
TEMP    RMB  1
```

Listing 2: Low pass filter routine. This routine can be followed using the flowchart of figure 3 or the block diagram of figure 4.

```
        SUB  A TEMP      ;CHANGE TO HIGH PASS
        NEG  A
```

Listing 3: To convert the low pass filter routine of listing 2 to a high pass filter routine, replace the ADD instruction with the above two lines. It can be seen that this is the only difference between the block diagrams of figures 2 and 7.

Technical Forum

Some Plotting Comments

T P Roberts
Kern Instruments Inc
111 Bowman Av
Port Chester NY 10573

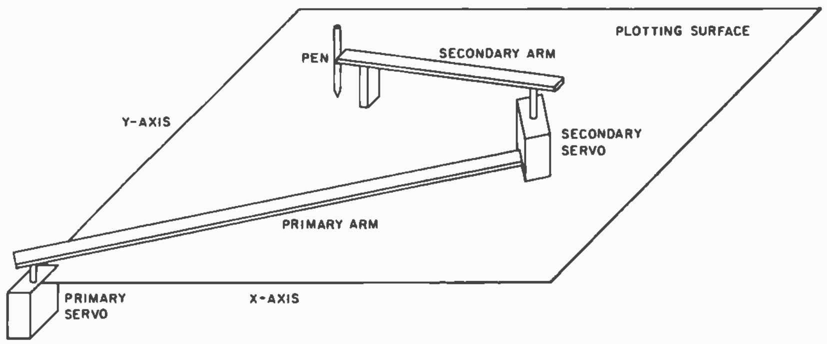


Figure 1: Layout for the articulated plotting method. The primary servomotor is stationary with a secondary servomotor on the far side of its lever arm. The secondary servomotor moves a second lever arm which contains a pen on its far end.

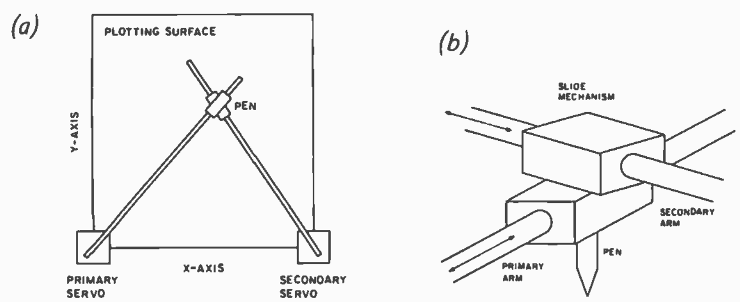


Figure 2: The intersection plotting approach uses two stationary servomotors. The pen is held precisely under the intersection of the two lever arms. Figure 2a is a graphical view of the plotter on the plotting surface. Figure 2b is a close-up view of the slide mechanism and the pen holder.

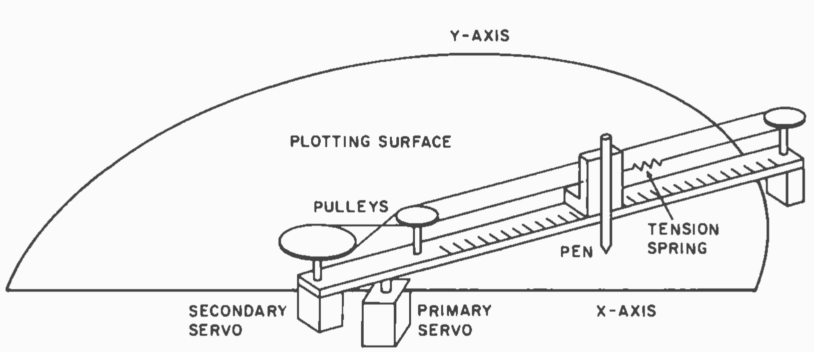


Figure 3: The polar plotting method uses a primary servo to swing the entire assembly across the plotting surface. A second servomechanism moves the pen along the lever arm.

The Servo Plotter

There are several ways to tackle the problem of building a plotter using the hobby servomechanism described briefly on pages 9 and 10 of March 1977 BYTE. As might be expected, the designs span the range between hardware and software intensive operation, the latter being less stable mechanically. The simplest to build, for example, requires by far the most software, is mechanically the least precise, and has the least theoretical accuracy.

This "articulated" design proposed in the earlier article is in the form of a chain as seen figure 1. A base mounted servo (primary) controls an arm with the secondary servo mounted on the far end. The second servo in turn controls an arm carrying the drawing pen.

The next simplest design is the intersection plotting method, where the x, y pen position is a function of two angles, determined by the servomotors, and the base line (distance between servos). Figure 2a shows the plane view of such a plotter. Figure 2b shows a close-up view of the two arms, slide mechanism and pen holder. Notice that the pen must fit within the slide mechanism to eliminate offset errors. This design does not utilize the full 180° sweep of the motors; thus accuracy is reduced over the plotting surface.

Figure 3 shows the polar method of plotting which uses the familiar polar coordinate system. Standard Cartesian-to-polar conversion can be used to produce the control data. The analog angular-to-linear conversion is accomplished by the simple pulley arrangement.

The servo portion of this device may, of course, be located anywhere along the radial arm of the plotter. A small offset correction is required if the pen travel is not aligned with the primary servo axis.

The final design shown in figure 4 uses two of the linear devices shown in figure 3. The primary device is rigidly attached to the plotting table, but the pen holder has been replaced with the secondary device mounted at right angles to the first. This, then, is an XY plotter. The only software required to operate this type of plotter with the servos being discussed is to scale the coordinates into timing data.

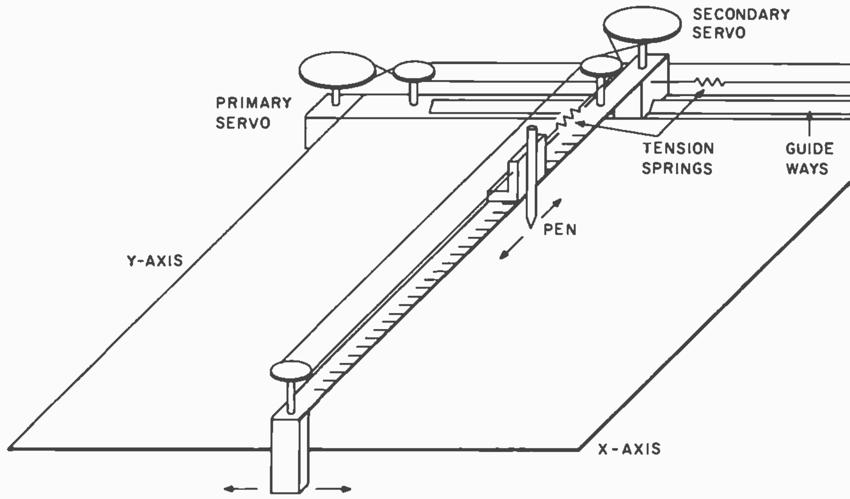


Figure 4: The XY plotter uses two of the polar plotting devices. The moving pen is replaced by another servomechanism with the moving pen contained on its lever arm.

Accuracy

Let's now take a look at two important aspects of each system: ease of construction and accuracy.

As these servomechanisms are to be operated by computer in discrete steps, positioning is possible only to the nearest grid intersection corresponding to these steps. The theorectical plotting accuracy is limited to one half the grid spacing.

The tangential distance between radial lines 25 cm long and 0.04° apart is 0.17 mm. An XY plotter would give a square grid with spacing of 0.054 mm over a 25 cm square area. Repeatability or precision of pen placement depend largely on mechanical design and construction.

Due to the compounding of errors and large moment arms in its design, the articulated plotter has poor precision relative to the other designs. The polar plotter suffers from the same problem to a lesser extent. The problem with the polar design, as with the XY plotter, is the mounting of one device upon another without causing undue instability. The intersection plotter suffers from theorectical, rather than mechanical, instability. As with the articulated plotter, its full range of 180° cannot practically be used. Also, the area near the base line does not have frequent grid intersections.

Plotting with Servomechanisms

The servomechanisms considered here respond only to positioning commands. The XY plotter, for example, is restricted to drawing straight lines parallel to each axis and at a 45° angle. Other plotter designs have similar limitations. All lines not so situated must be composed of small increments of these lines, giving the final product a sawtooth appearance. Computer time required to compute these small increments will be significant, except in the case of the XY plotter where servo motions for small line segments are proportional to motions for the entire line. This property, in addition to the conversion formulas, makes the XY plotter by far the most attractive from a calculation point of view.

CONVERSION FORMULAS

Articulated Plotter

In order to position the primary and secondary servos at their proper angles ($\theta 1$ and $\theta 2$ respectively) to provide the required x, y pen position, it is convenient to first convert these values to the polar coordinate system. Equation 1, below, shows the conversion by which the radial distance, R, from the origin (the primary servo axis), and the angle θ (the slope from the origin to x, y) (see figure 5) are found.

```
Equation 1:
R = SQRT(X**2 + Y**2)
θ = ARCTAN(Y/ABS(X))
or θ = 180 - ARCTAN(Y/ABS(X)), when X
is negative.
```

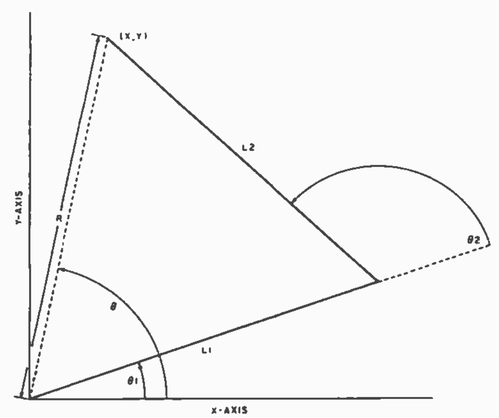


Figure 5: Coordinate plotting scheme for the articulated plotter. The lever arms are indicated by L1 and L2. $\theta 1$ is the angle generated by the position of the lever arm with respect to the primary lever arm.

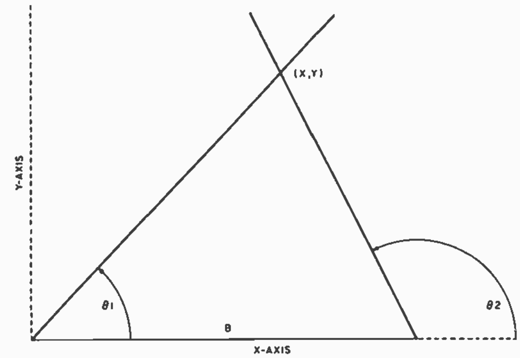


Figure 6: Arrangement for the intersection plotter. The distance B is the distance between the two servomotors. The two lever arms intersect at point x, y.

The values of $\theta 1$ and $\theta 2$ are then found from equation 2 where L1 and L2 are the lengths of the primary and secondary arms respectively:

Equation 2:
$\theta 2$ = ARCCOS((R**2 − L1**2 − L2**2)
/(2*L1*L2))
$\theta 1$ = θ − ARCTAN(L2*SIN($\theta 2$)/(L1+L2*
COS($\theta 2$)))
or $\theta 1 = \theta - \theta 2/2$, when L1 = L2.

Note that using this formula the angle produced at the origin by the pen position and primary arm is exactly one half the angle at the secondary servo, when the arms are of equal length. *[The formulas needed for plotting with this arrangement are fundamentally similar to positioning a singly jointed robot arm. . .RC]* This conversion would no doubt be difficult to handle in a low level language.

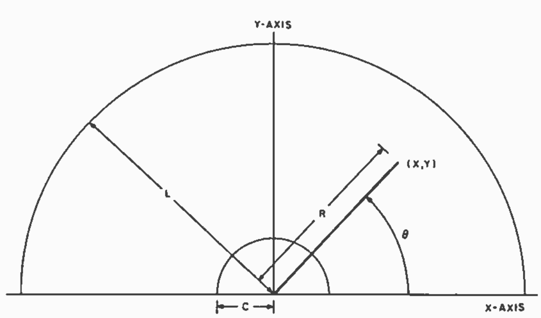
Intersection Plotter

The formulas required by the intersection method (see figure 6) are considerably less complex.

Equation 3:
$\theta 1$ = ARCTAN(Y/X)
$\theta 2$ = 180 − ARCTAN(Y/(B−X))

B is the distance along the X axis between servos.

Figure 7: The plotting arrangement for the polar coordinates. Length L is the total length of the plotter's arm. The value R is the distance of the pen from the plotter arm's origin. C is the closest the arm can approach the origin. θ is the angle of the primary servomechanism at the origin with respect to the horizontal axis.



Polar Plotter

The polar plotter (figure 7) also uses equation 1 to convert Cartesian coordinates to polar form. In addition, the value of R must be scaled into angular form as follows:

Equation 4:
θ = (R − C)*180/L

L is the radial distance produced by rotating the secondary servo 180°, and C is the minimum radial plotting distance.

XY Plotter

Equation 5 gives the angular values required by the XY plotter.

Equation 5:
$\theta 1$ = X*180/L1
$\theta 2$ = Y*180/L2

L1 and L2 are the linear motions produced by 180° rotation of the primary and secondary servos, respectively.

Converting Angle Requirements to Timing Data

Assuming required pulse widths from 1.3 to 3.6 ms, as noted in March 1977 BYTE, equal steps of about 0.04° throughout the 180° range of the mechanism would be provided with counts from 2600 to 7200, assuming a 2 MHz clock rate. The number of counts needed to produce a given angle is then given by equation 6.

Equation 6:
N = 2600 + θ*230/9.

The count value, N, is simply truncated, or rounded off, for better precision.

This high precision timing would require external hardware to receive the data, count down at the proper rate, and interrupt the processor at completion. The alternative is a software loop with steps of 0.0075 ms, based on an 8080 chip requiring 15 cycles for decrement and branch on zero. With the XY plotter this would give a grid spacing, over a 25 cm square area, of 0.81 mm.

Conclusions

From the information presented so far, it appears that the more easily constructed plotter designs have inherently less precision, use a larger grid spacing, and require more complex, time consuming software. The most easily programmed device, the XY plotter, requires more exacting construction. As a compromise, the intersection plotter is a good choice, being simple of construction without requiring too much computation.

Those hackers who plan to do any great amount of plotting, though, will do well to consider the XY plotter. For many uses too, software timing should prove sufficient.

Further Refinements

To fully automate plotting, a solenoid could be attached to the pen holder to lift and drop the pen under program control. A small servo should be able to be activated directly from the output latch, at least through a transistor.

Servo speed control would be another nicety, allowing fast, processor efficient straight-line drawing, and producing a higher quality line. For example, if the X axis servo were to move at a speed twice that of the Y axis servo, a straight line at an angle of 25° would be produced. Similarly, lines could be drawn at any angle.■

A Standard for Writing Standards

David A Wallace
146 Westford St
Chelmsford MA 01824

I'm sick to death of save the world articles proposing standards for software data structures and object code formats which start by assuming that the author's pet descriptor is the ultimate and final word to be said about the subject. I am therefore proposing the following as a standard for software standards:

- Be humble: Don't make grandiose claims about the universal applicability of your structure. Instead, define the limits of the range of applications to the best of your ability. Often it is as useful to know where something cannot apply as to know where it can.
- Plan for change: Everything in this universe either evolves or becomes defunct, including software. For example, set up the structure of your construct so that the first byte (word, field, whatever) represents the revision number of the specification which describes this structure. That way, if you have a data base which corresponds to revision 3 of the specification for random files and you've just recompiled the program which updates the data using a compiler whose random file operators correspond to re-

vision 7, the revision 7 processor can figure out that it has been passed an obsolete structure and call the revision 3 processor to sort out the mess.

- Don't begrudge the space used by an obsoleted field: Far too often a programmer will remove a data field which was made obsolete, thus moving all subsequent fields out of their previous positions. This misguided attempt to conserve space has the effect that the positions of fields whose meanings did not change are constantly shuffled from revision to revision, resulting in confusion to programmers and needless complexity in programming.

  If instead of removing the obsolete field, the field is merely ignored, the current revision program would find data in expected places and perhaps process nearly all of the structure before having to invoke some sort of routine to process the obsolete fields. This technique involves less execution space than having to roll in an entire program to process the obsolete data when the current revision finds all fields misplaced. Additionally, this technique implies that the newer format specification is always at least as long as the older format. This means that when an earlier revision must be invoked by the current one, all data which the older revision program needs has already been fetched by the newer revision program, which simplifies parameter passing.
- Wait at least one major revision of your system before redefining an obsoleted field for another purpose. This gives you time to change your mind if it turns out that the field in question really is necessary after all.

If all of the above rules are followed rigorously, you should never again have to translate or reformat files and recompile programs when you make a change to the operating system of your machine. If all of the above sounds suspiciously like another of those save the world software standards, I'm sorry; I guess the disease must be contagious! ■

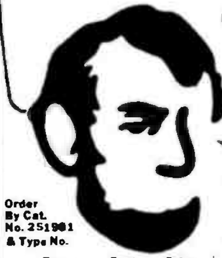
Technical Forum is a feature intended as an interactive dialog on the technology of personal computing. The subject matter is open-ended, and the intent is to foster discussion and communication among readers of BYTE. We ask that all correspondents supply their full names and addresses to be printed with their commentaries. We also ask that correspondents supply their telephone numbers, which will be printed unless we are explicitly asked to omit them.

THIS SPECIAL ONE CENT SALE IS FOR MAGAZINE ADS ONLY

POLY PAKS® BIGGEST 1¢ GIVEAWAY SALE!

IT MAKES "CENTS" TO GIVE OUR CUSTOMERS THE BEST BARGAINS!

TTL'S, BUY ONE AT SALE PRICE, GET 2ND FOR ONLY 1¢ MORE



Honest Abe Penny Sale!

Order By Cat. No. 251981 & Type No.

Type	Each	2 for
SN7400	$.19	$.20
SN7401	.19	.20
SN7402	.35	.38
SN7403	.19	.20
SN7404	.25	.26
SN7405	.19	.20
SN7406	.19	.20
SN7408	.21	.22
SN7410	.19	.20
SN7413	.39	.40
SN7414	.68	.68
SN7416	.29	.30
SN7417	.35	.36
SN7420	.19	.20
SN7423	.29	.30
SN7426	.25	.26
SN7427	.25	.26
SN7430	.29	.30
SN7432	.25	.26
SN7437	.25	.26
SN7438	.29	.30
SN7440	.19	.20
SN7442	.69	.70
SN7443	.69	.70

Type	Each	2 for
SN7444	.65	.66
SN7445	.99	1.00
SN7446	1.35	1.36
SN7447	1.25	1.26
SN7448	1.35	1.36
SN7450	.19	.20
SN7451	.19	.20
SN7453	.19	.20
SN7454	.19	.20
SN7455	.19	.20
SN7460	.19	.20
SN7462	.19	.20
SN7464	.19	.20
SN7465	.19	.20
SN7470	.25	.26
SN7471	.25	.26
SN7472	.25	.26
SN7473	.55	.56
SN7474	.29	.30
SN7475	.79	.80
SN7476	.59	.60
SN7478	.59	.60
SN7480	.29	.30
SN7482	.39	.40

Type	Each	2 for
SN7483	.99	1.00
SN7485	1.49	1.50
SN7486	.39	.40
SN7488	1.95	1.96
SN7490	.69	.70
SN7491	.79	.80
SN7492	.45	.46
SN7493	.49	.50
SN7494	.69	.70
SN7495	.69	.70
SN7496	.69	.70
SN7498	.69	.70
SN74100	1.49	1.50
SN74107	.39	.40
SN74112	.25	.26
SN74113	.19	.20
SN74114	.25	.26
SN74121	.49	.50
SN74123	.69	.70
SN74125	.59	.60
SN74126	.39	.40
SN74132	1.25	1.26
SN74140	.99	1.00
SN74141	1.49	1.50
SN74145	1.25	1.26
SN74148	1.75	1.76
SN74150	.99	1.00
SN74151	.99	1.00
SN74153	.99	1.00
SN74154	1.75	1.76
SN74155	.79	.80
SN74156	.69	.70
SN74157	.99	1.00
SN74158	.99	1.00
SN74160	1.25	1.26
SN74161	1.25	1.26
SN74163	1.19	1.20
SN74164	.99	1.00
SN74165	1.25	1.26
SN74166	1.25	1.26
SN74173	1.25	1.26
SN74174	1.49	1.50
SN74175	.99	1.00
SN74177	.79	.80
SN74179	1.49	1.50
SN74180	.49	.50
SN74182	.69	.70
SN74190	1.49	1.50
SN74191	1.75	1.76
SN74192	.85	.86
SN74193	.85	.86
SN74194	1.25	1.26
SN74195	.49	.50
SN74197	.75	.76
SN74199	1.75	1.76
SN74200	5.50	5.51
SN74251	1.75	1.76
SN74284	4.50	4.51
SN74285	4.25	4.26

POP-AMPS AT "CENT-CIBLE" PRICES

Buy ONE At Sale Price, Get 2ND For Only 1¢ More - Order By Type No.

Type	Each	2 for
LM300H	$.49	$.50
LM301H-V	.45	.46
LM307V-N	.45	.46
LM308V-H	.79	.80
LM309K	1.79	1.80
LM311H-V	.79	.80
LM318V	1.29	1.30
LM320H-5, 12, 15	1.19	1.20
LM320K-12, 15	1.19	1.20
LM320T-6, 15, 24	1.19	1.20
LM322N	1.19	1.20
LM324N	1.75	1.76
LM339N	1.09	1.10
LM340K-5, 6, 8, 12, 15, 18, 24V	1.19	1.20
LM340T-5, 6, 8, 12, 15, 18, 24	1.19	1.20
LM350N	.59	.60
LM370N-H	1.29	1.30

Type	Each	2 for
LM374H	1.79	1.80
LM376V	.29	.30
LM377N	2.25	2.26
LM379	7.95	7.96
LM381N	1.69	1.70
LM387V	1.19	1.20
LM531H	1.49	1.50
LM532H-N	.29	.30
LM555V	.75	.76
LM558V	.79	.80
LM561N	1.00	1.01
LM565N	.99	1.00
LM567V	1.95	1.96
LM703H	.49	.50
LM704H	.29	.30
LM709N-H	.25	.26
LM710N	.39	.40
LM733N	.79	.80
LM739N	.89	.90

Type	Each	2 for
LM741V-N-H	.29	.30
LM1304	.79	.80
LM1310	.99	1.00
LM1312	1.99	2.00
LM1414V	.79	.80
LM1458V	.69	.70
LM1800N	.99	1.00
LM3028H	.65	.66
LM3900N	.49	.50
LM3909V	1.75	1.76
LM4195	1.95	1.96
LM4250	1.19	1.20
LM75451	.39	.40
LM75453	.39	.40
LM75491	.79	.80
LM75492	.79	.80
LM75494	.59	.60
PA239	1.29	1.30
PA263	1.50	1.51

PENNY SALE PRICES LISTED ARE GOOD TILL MAR. 15, 1978
WE RESERVE THE RIGHT TO LIMIT QUANTITIES!!!!!!

WOW! MORE 'PENNY SALE' ITEMS!

TRANSFORMER PENNY SALE!

BUY ONE AT SALE PRICE, 2ND FOR ONLY 1¢ MORE
• 110VAC Primaries

Cat. No.	Output V.	Amps	Misc.	Sale Each	1¢ Sale 2 for
2S3399	6.3V	500ma	Metal encased	$1.98	$1.99
2S3814	6.3V	1A	Open frame	$2.49	$2.50
2S3412	12V	300ma	Open frame	$1.95	$1.96
2S4029	12V	1A	Open frame	$2.49	$2.50
2S4028	12V	1A	Metal encased	$2.95	$2.96
2S3323	24VCT	300ma	Open frame	$1.95	$1.96
2S3937	24V	1A	Open frame	$1.95	$1.96
2S2773	48VCT	2A	Metal encased	$2.95	$2.96
2S3875	110V	300ma	Isolation	$1.19	$1.20

1N4000 Epoxy Rectifiers

PENNY SALE!

• Miniature!
• 1.5 Amp!

Order by Cat. No. and Type No.

Cat. No.	Type No.	PIV	Sale	1¢ Sale
2S2377	1N4001	50	10 for $.65	20 for $.66
2S2378	1N4002	100	10 for .75	20 for .76
2S2379	1N4003	200	10 for .85	20 for .86
2S2380	1N4004	400	10 for .99	20 for 1.00
2S2381	1N4005	600	10 for 1.29	20 for 1.30
2S2382	1N4006	800	10 for 1.39	20 for 1.40

DIP SWITCHES

BUY ONE AT SALE PRICE, 2ND FOR ONLY 1¢ MORE!

Cat. No.	Switches	Sale Each	2 for
2S3668	2	$.77	$.78
2S3669	3	.88	.89
2S3021	4	.99	1.00
2S3670	5	1.19	1.20
2S3671	6	1.29	1.30
2S2677	7	1.79	1.80

2-Amp Epoxy Bridge Rectifiers

PENNY SALE! BUY ONE AT SALE PRICE, GET 2ND FOR ONLY 1¢ MORE

• Full Wave! • TO-5 case!!

Order by Cat. No. 251346 and PIV

PIV	Sale Each	1¢ Sale 2 for
50	$.59	$.60
100	.69	.70
200	.79	.80
400	.89	.90
600	.99	1.00
800	1.19	1.20
1000	1.29	1.30

MICROPROCESSORS! MEMORIES SUPPORT!

Order By Cat. No. 253459 & Type No.

Type	Sale
MC6800L	$24.95
8080A	14.95
8008	9.95
1101	.69
1103	1.29
1702A	5.95
2102-L1	1.69
2111	5.95
2112	2.49
2708	19.95
MK4200P11	3.95
MM5202	6.95
MM5203	8.95
MM5260	.99
MM5262	.99
8212	3.95
8216	3.95
8224	4.95
8228	9.95
8251	11.50
8255	11.95

3/8"Sq. POTENTIOMETERS

PENNY SALE!

- 253863 25 Turn upright, type 64
- 253864 25 Turn flat, type 64
- 253866 Single turn flat, type 63

SPECTROL 'SKINNY-TRIMS'

3/8" square. Screwdriver shaft, 20% tolerence, 1/2 watt. Cermet construction. PC leads. Order by Cat. No. and Value.

Choose any 2 for 70¢ $.69 each

OHMS: 10 ••, 20 ••, 50 ••, 100 •, 200 •, 500 •, 1K •, 2K •, 5K •, 10K •, 20K •, 50K ••, 100K ••, 200K ••, 500K ••, 1 Meg ••

•Available in all types
••Available in Cat. No. 253863 only.

1st Time Offered! It's Different! It's Inflation Fighting! New!

KIT KING®

KITS BY POLY PAKS®

SUPER ECONO KITS

ONLY $1.98 — BUY 10 KITS & CHOOSE 11TH FREE!

* MONEY BACK GUARANTEE * AVG WT. 6 OZS.

Quantity Description (Order by Cat. No., see parenthesis)	Sale
10—CALCULATOR KEYBOARDS, up to 20 keys (2S1524)	$1.98
8—LINE CORDS, 8-ft. lg, twin #18 wire, plug (2S3843)	1.98
30—NEON LAMPS, NE-2 style, red glow, leads (2S2613)	1.98
3—1702A ROMS, factory fallouts, hobby, useable (2S3729)	1.98
40-FT SHIELD CABLE, 1-cond.-n-shield, #22 wire, vinyl jack (2S3577)	1.98
50—TRANSISTOR ELECTRO'S, asstd values, upr'tes, axials (2S2747)	1.98
3—SOUND TRIGGERS, handclap triggers scr, w/amp (2S3625)	1.98
8—IC SOCKETS, incl. 4-14's and 4-18's, low prof (2S3621)	1.98
15—6V TEST INDICATORS, w/leads, grain-o-wheat (2S1526)	1.98
3—GLOW'N PANELS, 3x1", 110vac, glow-n-dark, green (2S3650)	1.98
400 PARTS ON A BOARD, greatest asst on p.c. boards (2S3401)	1.98
150—METAL FILM RESISTORS, 1/4 watts, color-coded, leads (2S3413)	1.98
15—JUMBO "LEDS" RED, prime, leads, like MV5054 (2S3369)	1.98
30—MINI TRIM POTS, to 1 meg, 1 turn, 1/4W (2S3345)	1.98
10—VOLTAGE REGULATORS, TO-3 case, 309K etc. hobby (2S3330)	1.98
30—PANEL SWITCHES, rotaries, slide, mod. 110vac, etc. (2S3268)	1.98
200-PC. RESISTOR SPECIAL, 1/4 to 2W, metal, carbon (2S3054)	1.98
40—AXIAL ELECTRO'S, asstd values to 100 mf, volts (2S3227)	1.98
40—UPRIGHT ELECTRO'S, asstd volts, values to 110mf (2S3226)	1.98
150—MOLEX SOCKETS, pins fit all ic's, most pop'lr (2S3144)	1.98
100—TERMINAL STRIPS, tie, to 8 lugs, for solder (2S3136)	1.98
15—SLIDE CONTROLS, asst values, vol, trble, bass, etc. (2S3057)	1.98
10—PHOTO ELECTRIC CELLS, CDS pancake styles, asst (2S3052)	1.98
200—HALF WATTERS, color coded, carbon, prec, metal (2S3046)	1.98
100—NATIONAL IC BONANZA, linears, 7400's, roms, clocks, untested (2S2860)	1.98
40—HOBBY LEDS, asst shapes, red, untested mostly useable (2S2859)	1.98
15—LM340T VOLTAGE REGS hobby, useable 5 to 24V, TO-220 (2S2635)	1.98
100—CAPACITOR SPECIAL, poly's, dips, molded, mica, discs (2S2735)	1.98
100—TWD WATTERS, carbon, metal, prec, etc. marked, leads (2S2735)	1.98
100—POLYSTYRENE CAPS, asst values, voltages, hi-Q (2S2729)	1.98
30—SLIDE SWITCHES, spst to dpdt, many types (2S2726)	1.98
15—NPN POWER TRANSISTORS, TO-3 2N3055 rejects, useable (2S2617)	1.98
200—PRE-FORMED RESISTORS, color-coded, p.c. work, asst values (2S2609)	1.98
100—PLASTIC TRANSISTORS, TO-92, asst values, mostly useable (2S2604)	1.98
100—DISC CAPS, npo's, N750, long leads (2S2598)	1.98
60—DIPPED MYLAR CAPS, asst values & voltages, finest (2S2597)	1.98
200—PRECISION RESISTORS, 1%, 1/4 to 2, marked, leads (2S2428)	1.98
30—VOLUME CONTROLS, single, doubles, asst shafts values (2S2421)	1.98
150—SWITCHING DIODES, 1N4148, most pop'lr, mostly good (2S2418)	1.98
75—SN7400 IC's, marked, mostly useable, wide asst. of SN's (2S2415)	1.98
100—1N4000 RECTIFIERS, 1 amp, sil, most pop, asst, mostly good (2S2417)	1.98

"KIT KING" $1.19 KITS - "The Econo's"

BUY 5 KITS — CHOOSE THE 6TH FREE! MONEY BACK GUARANTEE!

Quantity Description (Order by Cat. No., see parenthesis)	Sale
30—RADIO-N-TV KNOBS, asst sizes, colors, shapes (2S217)	$1.19
10—PRS PHONO PLUGS-N-JACKS, RCA style, 10 plugs, 10 jacks (2S35-402)	1.19
35—DIPPED SILVER MICA CAPS, includes rect types too, asst (2S455)	1.19
75—PREFORMED DISC CAPS, for p.c. use, asst values, up to 2KV (2S1219)	1.19
10—NE-2's WITH RESISTOR, for 110vac use, neon, leads (2S1222)	1.19
12—TO-5 HEAT SINKS, for TO-5 metal & plastic transistors, fin type (2S1832)	1.19
12—TRANSISTOR SOCKETS, asst styles for NPN n PNP types (2S1905)	1.19
20—THERMISTORS, a resistor that changes with temp, protects! (2S2048)	1.19
40—INSULATED RF CHOKES, resistor n sd. styles, coded (2S3203)	1.19
30—WIRE NUTS, ties, 2 bare ended wires together (2S3724)	1.19
3—BLANK PC BOARDS, 4x6's, double sided, G-10 (2S3833)	1.19
2—BLANK PC BOARDS, 6x6's, double sided, G-10 (2S3824)	1.19
10—OPEN FACE READOUTS, singles, doubles, some missing segs (2S3952)	1.19
3—"PIGGY-BACK" IC SOCKETS, will hold 2-14 pin IC's-on-a-slab (2S3486)	1.19
FIBER OPTIC LIGHT PIPE, 12" long highly polished ends, vinyl jack (2S2693)	1.19
4—FERRITE "STICK" ANTENNA, 7"x1/2". HH Scott, Fisher hi-fi eqpt (2S3400)	1.19
60—TUBULAR CAPS, asst molded, plastic, mylar, values, volts (2S219)	1.19
55—LOW NOISE RESISTORS, for hi-fi, metal films 5% 1/4 to 2W (2S220)	1.19
50—POWER RESISTORS, wirewound, vit, sq, up 25 w. asst (2S228)	1.19
60—COILS AND CHOKES, r-f, ant, parasitic, i-f, etc. (2S297)	1.19
75—HALF WATTERS, resistors, color-coded, carbo-film, most pop (2S454)	1.19
10—SOLAR CHIPS, silicon, "broken energy", each chip 0.5v. asst mils (2S5083)	1.19

Low Power ICs

Order By Cat. No. 253667 & Type No.

Type	Sale
74LS00	$.32
74LS02	.32
74LS04	.35
74LS08	.32
74LS10	.32
74LS11	.32
74LS13	.64
74LS20	.32
74LS21	.32
74LS22	.32
74LS27	.39
74LS30	.32
74LS32	.39
74LS37	.45
74LS38	.45
74LS42	1.19
74LS47	.99
74LS74	.49
74LS90	.89
74LS92	.89
74LS93	.89
74LS109	.58
74LS112	.58
74LS113	.58
74LS114	.49

Type	Sale
74LS132	1.19
74LS138	1.24
74LS139	1.24
74LS151	1.25
74LS153	1.25
74LS155	1.25
74LS160	1.47
74LS161	1.47
74LS162	1.47
74LS163	1.47
74LS168	1.68
74LS169	1.68
74LS173	1.68
74LS174	1.05
74LS190	1.77
74LS191	1.75
74LS192	1.75
74LS193	1.75
74LS195	1.25
74LS197	1.25
74LS257	1.35
74LS266	.54
74LS366	.66
74LS368	.69
74LS390	2.95

C-MOS

Order By Cat. No. 252320 & Type No.

Type	Sale
CD4000	$.29
CD4001	.29
CD4002	.29
CD4006	1.19
CD4007	.29
CD4008	.79
CD4009	.59
CD4010	.59
CD4011	.29
CD4012	.29
CD4013	.69
CD4015	1.19
CD4016	.49
CD4017	1.19
CD4018	1.19
CD4019	.49
CD4020	.99
CD4021	1.29

Type	Sale
CD4022	1.19
CD4023	.29
CD4024	.79
CD4025	.34
CD4027	.69
CD4028	.89
CD4029	1.19
CD4030	.49
CD4033	1.60
CD4035	.99
CD4040	1.19
CD4041	1.25
CD4042	.88
CD4046	1.79
CD4049	.49
CD4066	.79
CD4071	.29

FREE! DOUBLE BONUS

CHOOSE ANY $1.19 ITEM, PLUS OUR $25* SURPRIZE PAK FREE

GET BOTH ITEMS FREE WITH ANY $15 ORDER!

*We think, add 25c for handling!

Send for your FREE POLY PAKS CATALOG FEATURING THE WORLD'S BEST BARGAINS IN ELECTRONICS

Terms: Add postage Rated: net 30
Phone: Wakefield, Mass. (617) 245-3829
Retail: 16-18 Del Carmine St., Wakefield.

MINIMUM ORDER — $6.00

POLY PAKS®

C.O.D.'s MAY BE PHONED

P.O. BOX 942S
LYNNFIELD, MA.

Circle 96 on inquiry card.

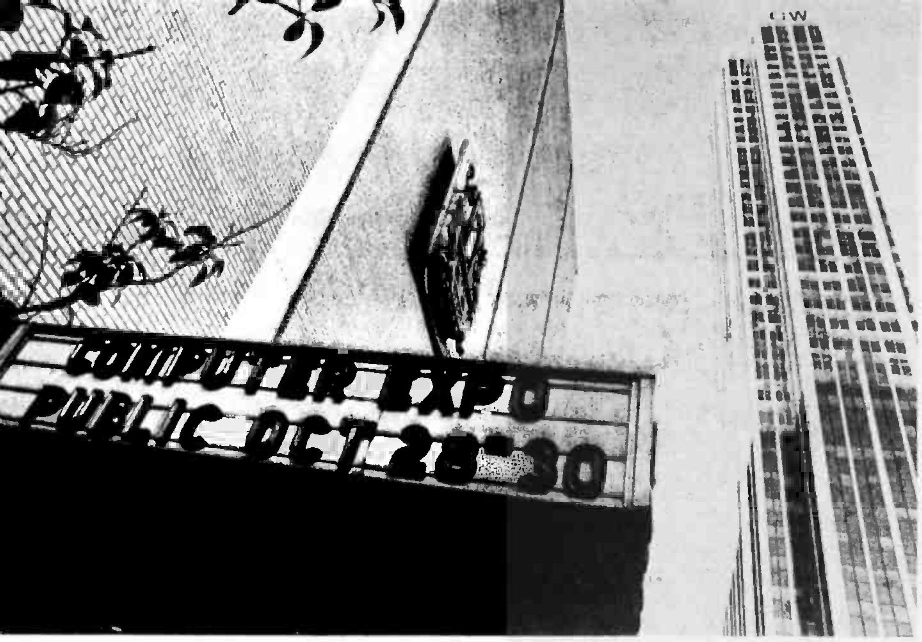


Photo 1: Entrance to the New York Coliseum Personal Computer Expo Show. Gulf and Western building is to the right.

Photo 2: A young computer fan attempts to hit the enemy ship in Apple Computer's Star Wars game at the Computer Mart of NY booth.

Photo 3: Some of the 14,000 people who saw the show.

New York's Coliseum, the location of many IEEE, AFIPS (NCC) and other technical shows over the years, played host to the first annual Personal Computer Expo last October 28, 29 and 30. Over 80 exhibitors were on hand at 150 booths, selling everything from resistors to complete computer systems with floppy disks and color graphics.

The exhibition floor filled rapidly on Friday morning. Many of the people I spoke to said the show was their first exposure to personal computing. Visitors flocked to the more spectacular exhibits like the Digital Group's talking computer and Heathkit's Star Wars game, or tried their skill at programming the new appliance computers, such as the Commodore PET and the Radio Shack TRS-80.

At the MITS booth, the emphasis was on business software. The business men and women who attended the show were able to choose from a number of sophisticated systems on view at MITS and other booths.

Some of the Highlights

Summagraphics featured an interesting device called the Bit Pad, apparently the first of its kind in the personal computing market. The Bit Pad is a digitizing tablet complete with stylus that allows you to quickly enter drawings or writing into a computer. For the floppy disk enthusiasts, Alpha Micro Systems displayed their AM-400 hard surface disk; more floppies were on hand at Per Sci and Realistic Controls. Ohio Scientific showed their Challenger III, a most unusual computer that contains three processors: the 6502A, 6800 and Z-80.

A nonprofit organization called Computers for the Handicapped was represented by Warren Dunning (5939 Woodbine Av, Philadelphia PA 19131) and Richard Moberg (404 South Quince St, Philadelphia PA 19147). The purpose of the group is to be a clearinghouse of information regarding the use of computers to help the handicapped. The goal is to get the people with the ideas and needs together with the people with the computer know-how so that development of these systems can begin.

A Record Crowd

By Sunday evening, over 14,000 people had attended the Expo, making it the biggest personal computing show ever, and giving added impetus to this young and growing field.■

New York Notes

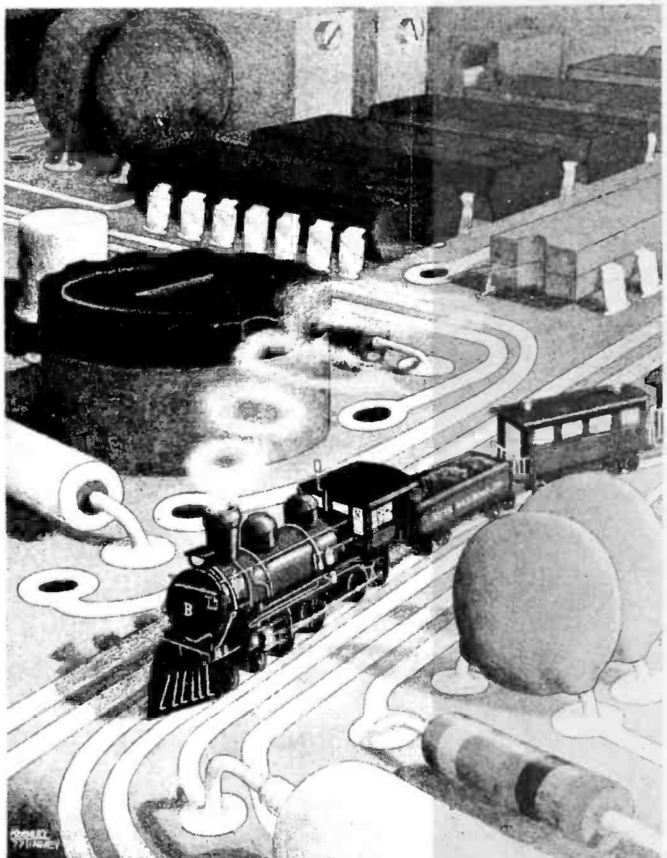
by Chris Morgan, Editor

Photos by Fritz Wetherbee

Photo 4: A happy group plays Space War at the Heathkit booth.

Photo 5: Alpha Micro System's hard surface disk, one of the most sophisticated devices at the show.

Photo 6: An array of new and used equipment offered by the Computer Warehouse Store (of Boston).



In unusual cases, processing may exceed 30 days.

COMPUTER ENGINEERING

THE MIDNIGHT SPECIAL . . .
. . . a full color limited edition poster for your Casey Jones fantasies.

The poster is 16½ by 21½ inches (41.9 by 54.6 cm) with a white border. The colors are the same as the original by Robert Tinney, which graces our July cover, minus the BYTE logo. The price is $3.00, plus 50¢ postage. It is shipped unfolded, in a mailing tube.

ORDER NOW. SUPPLIES ARE LIMITED.

BITS Inc
70 Main Street
Peterborough NH 03458

_______ Posters $3.00 each plus $ .50 mailing

☐ Bill BankAmericard
☐ Bill Master Charge
☐ Check Enclosed $

No.__________________ Exp. Date______

Name__________________

Address__________________

City______________ State_____ Zip_______

ESAT-100
Economical Stand Alone Terminal
Assembled, Tested, Burned-In........$239.00

For Teletype, SCMP, Joll, Kim SBC 80 Users. A reliable, inexpensive, self-contained Communicating Terminal, completely assembled, burned-in, and tested $239.00

SPECS 32 character x 16 lines x 2 pages of 5 x 7 dot matrix 64 character ASCII communicating with a serial, asynchronous 11 unit code. TTL compatible from 300 to 9600 baud. Keyboard Controls are back and forward space, line feed, clear page, or to end of line, select page 1 or 2, full/half duplex, local/remote, cursor on/off, odd/even/no parity. Output to TV Monitor is Composite Video, 75 Ohms. Keyboard required is parallel output 7 unit ASCII with negative true strobes. Keyboard may tap up to 200 Mo from the ESAT 100 on-board 5V power supply. Power required is 110VAC @ 7 watts

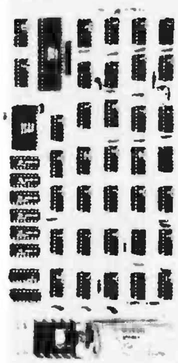
COMMENTARY At this writing (10-1-77), the ESAT 100 is the only Stand Alone Terminal board requiring only black and white TV set and ASCII Keyboard. You do not have to have a S-100 Bus Machine, or even a computer. May be used in conjunction with a Modem and your home TV set to provide a time-share type terminal at any Baud rate you desire.

Note, commercial terminals use an 80 character x 24 line format. However, we have chosen 32 character x 16 lines for television set applications because of the limited resolution available on most TV sets.

Nonetheless, for those of you who are the owners of either high quality video monitors, or the best of Japan's TV receivers, we offer the

Scrollboard Adapter Kit (designed to fit on ESAT-100) with 64 characters x 16 lines and Automatic Scrolling for $29.95

M & R Supermod R F Modulator for antenna connection with your TV set. Runs off of ESAT 100 power supply color and black & white for $24.95



CAP. SPECIAL:
The Highest Quality,
By-pass anywhere:
VK06 0.1/100V for $0.29

Full Color TV Game Kit:
Includes MM57100 Game Chip, MM53104 Clock Gen., LM1889 N Color Modulator, 3.58 MHz color burst oscillator crystal, variable cap for crystal adj. and PC board. Very impressive on color sets. All data sheets and schematics included...$25.95

Direct Reading Capacitance Meter Kit:
Here is a handy shop item incorporating a reliable and accurate measurement method which works from 01 pico + d to 10 micro + d. Includes large 6" scale analog meter movement, all electronic parts and full documentation and theory of operation. Requires 5V and +-12V power. Makes accurate readings directly and instantly! No tuning, fiddling or Interpolation required. Full documentation and theory...$39.95

5V Regulated Power Supply Kit:
Provides regulated 5V @ approx. 250 ma for benchwork, breadboards and small projects. Includes LM340T-5 regulator, rectifiers, filter caps, and wall-plug transformer. Full notes...$6.95

Nicad Batteries and Charger:
2 ea 2AH 'C' size Nicads (great for memory power-fail CKTS, cold weather flashlights, etc. and charger...$13.95

LINEAR

LM380N	1.39
LM340T-5,12,15	.99
NE560	2.95
NE561	2.95
NE565	1.95
NE566	1.49
NE567	1.49
LM1812	4.95
LM1889	4.95

CPU SPECIALS

8080A	10.95
Z80 (2mHz) incl.	
18mHz Xtal	33.95
Z80A (4mHz) incl.	
36mHz Xtal	39.95

OPTOCOUPLERS

4N26	1.00

PROMS

8223 Special	10/9.95
82S115 8 x 512	15.95

EPROMS

2708	650nS	12.95
2708	450nS	16.95
C1702A	1000nS	4.95
MM5203Q		4.95
MM5204Q		9.95

8K Static Memory Boards
Tested, Burned-In, Complete
$179.00 for 450nS
$239.00 for 250nS
$ 19.00 for Bare Board

VERBATIM Removable Magnetic Storage Media PRICE REDUCTION!

	1-9	10-25	26-100
Minidiskettes	4.79	4.65	4.45

MD525-01	(Soft Sector) for: Intelligent Systems, Magnovox, Microkit, and Vector Graphics
MD525-10	(10 Sector, Hard) for: Digi-log, North Star, Polymorphic, Tei Inc., and Wang
MD525-16	(16 Sector, Hard) for: Altair, Comtek, Micropolis, R2E, Realistic Controls, and Teleray- Research Inc.

	1-9	10-25	26-100
Standard Size Diskettes	5.99	5.33	4.79

FD34-1000	(Soft Sector, IBM Std.)
FD32-1000	(Hard Sector, Inner dia.)
FD65-1000	(Hard Sector, Outer dia.)

	1-9	10-25	26-100
Cassettes R-300 Digital Direct	5.25	4.99	4.35

RAMS

21L02-1	450nS	1.25
21L02	250nS	1.50
2111AL-4	450nS	2.30
C3107B	60nS	1.99
PD411D-4	135nS	3.99

and OTHERS

AY5-1013	4.95
MM5369	1.69
MM5320	6.00
2513 Up.	5.95
2513 Lo.	6.95
IN4005	11/1.00
IN4148	15/1.00

MERRY CHRISTMAS!

Electrolabs
POB 6721,
Stanford, CA 94305
415-321-5601

SATISFACTION 100% GUARANTEED!

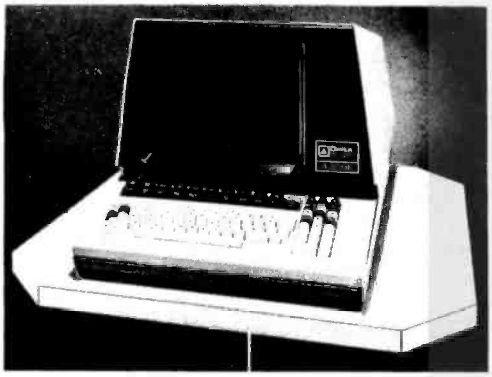
TO ORDER:
Send check or money order and include $1.00 for shipping, 1.00 (optional) for insurance, and please include 6½% sales tax if you are a California state resident. COD orders add .85 Thank you.

Circle 45 on inquiry card.

What's New?

PERIPHERALS

Text Handling Terminal Features Automatic Justification

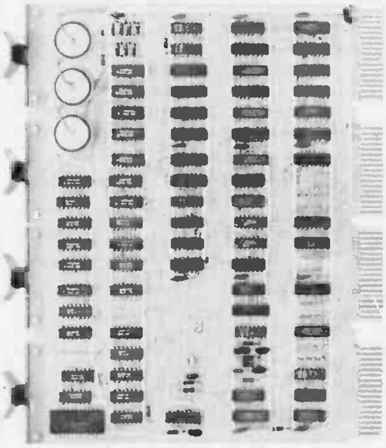


The Delta 4300E, a new video display terminal designed for text processing applications, features automatic word wrap, automatic ragged right sentence and paragraph justification, and automatic justification of new insertions.

Editing features include searching for a specific word or word string, and optionally deleting or replacing the string found with a new word or word string. Text is stored in a 4 K character memory, expandable to 8.5 K. The display presents a full 128 upper and lower case character set in 25 lines of 80 characters each. Automatic scrolling is provided, and a "paging" feature permits recall of information that has been scrolled off the screen. Communication speeds up to 9600 bps are available, and a serial printer port can be added as an option. The 4300E is offered by Delta Data Systems Corp, Woodhaven Industrial Park, Cornwells Heights PA 19020, (215) 639-9400.■

Circle 615 on inquiry card.

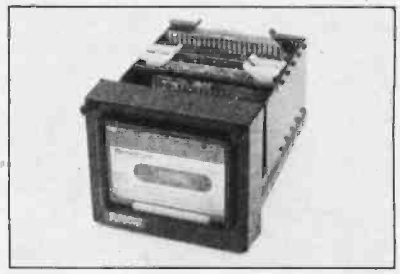
Time and Date Board for LSI-11



This battery operated accessory board provides calendar and real time clock functions for the LSI-11 or PDP-11 computer. The TCU-50 board for the LSI-11 provides the month and day, and the time in hours, minutes and seconds in response to a read instruction. The TCU-100, for the PDP-11, also includes an interrupt feature which can be set to interrupt the system at a specific time or at regular intervals. The rechargeable batteries are good for three months of use. The units are shipped running and preset to the correct date and local time at the customer's location. The TCU-50 is $325, and the TCU-100 is $495 in single quantites, from Digital Pathways Inc, 4151 Middlefield Rd, Palo Alto CA 94306, (415) 493-5544.■

Circle 617 on inquiry card.

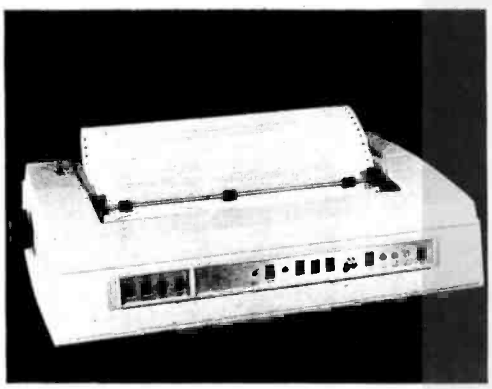
Cassette Recorder Meets ANSI, ECMA Standards



This new compact, lightweight digital cassette recorder can be used with any ECMA-34 compatible reader, minicomputer or terminal as well as ANSI compatible devices such as the Texas Instruments Silent 700. The Model 819-34 measures 4.5 by 4 by 7 inches (11.4 by 10.2 by 17.8 cm), weighs 3 pounds (1.4 kg) and requires 500 mW while running or 20 μW in standby mode. The unit features parallel input of up to 32 bits, a data rate of 50 bits per second and a formatted capacity of 1 million bits. Analog to digital and 16 channel multiplexer cards may be added to the unit's card cage. The Model 819-34 is $995 from Memodyne Corporation, 385 Elliot St, Newton Upper Falls MA 02164, (617) 527-6600.■

Circle 618 on inquiry card.

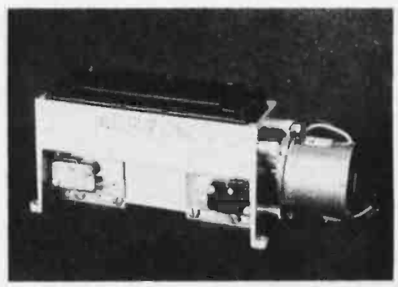
Daisywheel Printer Takes Byte Sized Commands



Interfacing a daisywheel printer to a microcomputer is simplified with this new OEM model, which takes its commands in serial or parallel byte format instead of the 13 bit format of prior mechanisms. The Sprint Micro 5 includes a built-in microprocessor with a set of 58 commands, which can be used to define format and character spacing, hammer intensity, ribbon color, vertical and horizontal tabs, and select normal, program or graphics modes. The optional RS232C interface, which includes a 224 character buffer, allows the printer to receive parallel and serial data from two sources simultaneously. For terminal builders, a send receive cover and plug-in facilities for a keyboard are offered. The control panel includes 11 switch selectable functions, such as full or half duplex, data rate, form length, and 10 or 12 pitch spacing. The Sprint Micro 5 is available in two models, with printing speeds of 45 or 55 characters per second. The 45 cps model is priced at $1675 in quantities of 50, and the optional RS232C interface is $100 from Qume Corporation, 2323 Industrial Pky W, Hayward CA 94545, (415) 783-6100.■

Circle 616 on inquiry card.

Credit Card Magnetic Stripe Reader



This device reads or writes information on the magnetic stripes of credit cards conforming to the International Standards Organization (ISO) and American National Standards Institute (ANSI) conventions. The magnetic head travels along a precision lead screw running in ball bearings to read or write on the stripe, and a spring loaded design minimizes card wear and provides optimum signal output. The ANSI standard provides for recording of up to 600 bits per track, but an improved design, for which patents are being sought, is capable of reading and writing up to 1024 8 bit bytes on the stripe. Interfacing options include TTL clock and data levels, buffered RS232 signals and a direct microprocessor bidirectional bus interface. Prices start at $296 for the reader mechanism with TTL interface in single quantities, from Vertel Industries, 167 Worcester St, Wellesley Hills MA 02181, (617) 235-2330.■

Circle 619 on inquiry card.

cybercom BOARDS

MB-1 MK-8 Computer RAM, (not S-100), 4KX8, uses 2102 type RAMs, PCBD only $22

MB-3 1702A EROM Board, 4KX8, S-100, switchable address and wait cycles, kit less PROMS $65

MB-4 Basic 4KX8 ram, uses 2102 type rams, may be expanded to 8KX8 with piggybacking, S-100 buss. PC board $30

MB-6 Basic 8KX8 ram uses 2102 type rams, memory protect in 256 to 8K switchable S-100 buss. PCBD $35

MB-7 16KX8, Static RAM uses μP410 Protection, fully buffered.
PCBD .... $30.00 KIT $525.00

MB-8 2708 EROM board, S-100, 8KX8 or 16KX8 kit without PROMS $85

MB-9 4KX8 RAM/PROM Board uses 2112 RAMS or 82S129 PROM kit without RAMs or PROMs $80

IO-2 S-100, 8 bit parallel I/Oport, 2/3 of board is for kludging.
Kit ...... $55 PCBD $30

IO-4 Two serial I/O ports with full handshaking 20/60 ma current loop. Two parallel I/O ports.
Kit $150

VB-1 64X 16 video board, upper lower case Greek, composite and parallel video with software, S-100.
Kit .... $150 PCBD $30

SP-1 Music synthesizer board, S-100, computer controller wave forms, 9 octaves, 1V rms 1/2% distortion, includes software kit $200

Altair Compatible Mother Board, 11 x 11 1/2 x 1/8".
Board only ... $45 With 15 connectors $105

Extender Board full size. Board only $9
With connector $13.50

Solid state music Cybercom boards are high quality glass board with gold finger contacts. All boards are check for shorts. Kits only have solder mask. 90 day guarantee on Cybercom kits.

Non-electrical cosmetic rejected PCBD from Cybercom.
IO-2 .... $21 MB-6 .... $21 VB-1000 .... $25

WMC inc. WAMECO INC.

MEM-1 8KX8 fully buffered, S-100, uses 2102 type rams. PCBC $30

Mother Board 12 slot, terminated, S-100, board only $35

CPU-1 8080A Processor board S-100 with 8 level vector interrupt PCBD $30

10% discount on 10 or more of WAMECO PCBD in any combination.

NEW! All IC's Sockets & hardware for WAMECO CPU-1 include all prime Eowa, 8214, 8224, 8212. PCBD not included $65

All ICs, sockets & hardware for WAMECO MEM-1 includes prime 2102AL-4's. PCBD not included. Order PCBD separately $135

Special 2102AL-4 1K x 1 ram 1/3 less power than 21L02 type rams, with power down, prime from NEC. Ea. 2.00; 32 ea. 1.80; 64 ea. 1.70; 128 ea. 1.60; 256 ea. 1.50.

9080A AMD 8080A (Prime)	20.00
8212/74S412 Prime	4.00
8214 Prime	8.30
8216 Prime	4.95
8224 Prime	5.00
8228 Prime	8.90
8251 Prime	14.50
8255 Prime	14.50
1702A-6 AMD 402A Prime	5.00
TMS-6011 UART Prime	6.95
2513 Char Gen Upper Prime	11.00
2513 Char Gen Lower Prime	11.00
1702A Intel Not Prime	4.00

74L00	.25	74LS00	.40	1101	1.25
74L01	.25	74LS01	.50	1103	1.25
74L02	.25	74LS02	.40	2101	4.50
74L03	.25	74LS03	.40	2111-1	3.75
74L04	.30	74LS04	.45	2112	4.50
74L05	.40	74LS05	.45	2602	1.60
74L06	.30	74LS08	.40	4002-1	7.50
74L08	.40	74LS10	.40	4002-2	7.50
74L09	.40	74LS12	.55	MM5262	1.00
74L10	.30	74LS20	.40	7489	2.00
74L20	.35	74LS22	.45	74200	4.95
74L26	.40	74LS27	45	74C89	3.00
74L30	.40	74LS30	.40	82S06	2.00
74L32	.45	74LS37	.60	82S07	2.00
74L42	1.50	74LS38	.60	82S17	2.00
74L51	.35	74LS42	1.50	8223	2.50
74L54	.45	74LS51	.40	82S23	3.00
74L55	.35	74LS54	.45	82S123	3.00
74L71	.30	74LS55	.40	82S126	3.50
74L73	.55	74LS73	.65	82S129	3.50
74L74	.55	74LS74	.65	82S130	3.95
74L75	1.20	74LS76	.65	82S131	3.95
74L78	.90	74LS151	1.55	IM5600	2.50
74L85	1.40	74LS174	2.20	IM5610	2.50
74L86	.75	74LS175	1.95	IM5603	3.00
74L89	3.50	74LS192	2.85	IM5604	3.50
74L90	1.50	2501B	1.25	IM5623	3.00
74L91	1.50	2502B	3.00	IM5624	3.50
74L93	1.70	2507V	1.25	MMI6330	2.50
74L95	1.70	2510A	2.00	DM8573	4.50
74L98	2.80	2517V	1.25	DM8574	5.50
74L123	1.50	2519B	2.80	DM8575	4.50
74L164	2.50	2532B	2.80	DM8576	4.50
74L165	2.50	2533V	2.80	DM8577	3.50
74L192	1.25	DM8131	2.50	DM8578	4.00
74L193	1.20	N8263	3.50	2.4576 MHZ	
MH0026	2.95	MC1489	1.50	XTAL	7.20
MC1488	1.50	DM8837	1.50		

MIKOS

419 Portofino Drive
San Carlos, California 94070
Please send for IC, Xistor and Computer parts list.

Check or money order only. If you are not a regular customer and your order is large please send either a cashier's check or a postal money order, otherwise there will be a delay of two weeks for the check to clear. All items post paid in the U.S. Calif. residents add 6% tax. Money back 30 day guarantee. We cannot accept returned IC's that have been soldered to. Prices subject to change without notice. **$10 minimum order. $1.00 service charge on orders less than $10.**

THE ALPHA-1 SYSTEM ✔RATED A BEST BUY IN MASS STORAGE SYSTEMS

✔APPLICATIONS

- BUSINESS applications include mailing lists, payroll, billing, and inventory.
- CASSETTE BACKUP for disk-based Systems not only provides large amounts of storage at low cost, but also provides for convenient storage of historical records.
- DEVELOPMENT SYSTEM features include a powerful operating System with an Editor, Assembler, and Debugger, plus a variety of System utilities which speed development.
- OEM applications include P.O.S. data capture, word processing systems, audio-visual presentation systems, telephone call transfer systems.

✔HARDWARE

- Stores greater than 500K bytes per side of a C-60 tape.
- Access a file in 17 seconds average on a C-60 tape.
- Load 8K of data in less than 11 seconds (6250 baud).
- 100% interchangeability of cassettes with no adjustments required or allowed.
- Compatible with all popular S-100 Bus Microcomputers.
- Audio track under computer control.
- Eliminates the need for ROM/PROM monitors.

✔SOFTWARE

- MCOS, a powerful stand-alone cassette operating system, is operationally much simpler than a D.O.S., handles variable length named files, will update a file in place, packs or copies tapes with a single command.
- EXTENDED BASIC with MCOS permits array handling and concatenation of files, plus all capabilities of MCOS.

✔PRICES START AT $240

✔FREE BUYERS GUIDE

If you are shopping for a tape or disk system for your S-100 Bus Computer System, you do not have all the facts until you have the MECA "BUYERS GUIDE TO MASS STORAGE." This 10 page guide book provides a framework for evaluating cassette, cartridge, and disk-based systems. Write for your copy today.

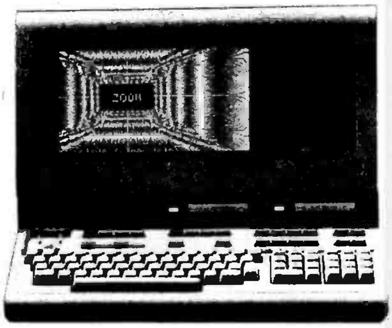
For complete information including the Dealer nearest you, write or phone:

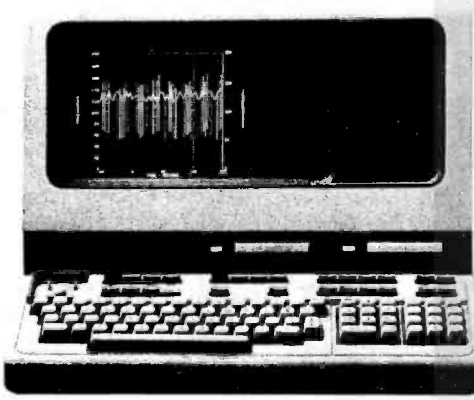
meca
7026 O.W.S. Road, Yucca Valley, CA 92284
(714) 365-7686

What's New?

PERIPHERALS

Zoom and Pan with This Graphics Terminal

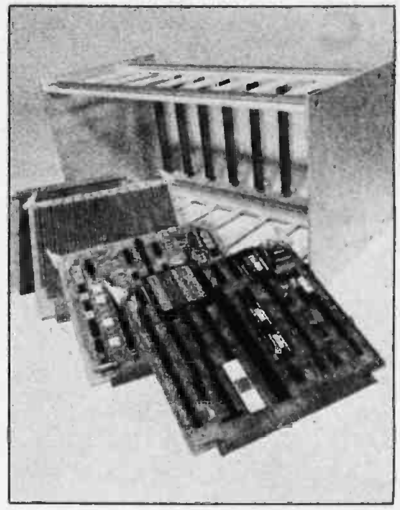




This raster scan graphics terminal with a built-in microprocessor performs a variety of graphics operations independently of the computer to which it may be connected. The terminal features separate alphanumeric and graphics memories, with 8 K bytes of memory (expandable to 12 K) allotted to a 24 line by 80 character alphanumeric display with 9 by 15 dot character cells, and 256 K bits of memory for graphics with a 360 by 720 dot resolution. Graphics capabilities include "rubber band" line drawing, zoom magnification of any portion of the graphics memory up to 16 times, and panning through any portion of the magnified display which is not in the viewing window. An automatic plotting feature for tabular data guides the operator through a simple menu of questions about plotting parameters, and then generates a fully labeled plot with as few as three keystrokes. Optional built-in cartridge tape drives provide up to 220 K bytes of local data storage. The Hewlett-Packard 2648A graphics display terminal is $5500 in single quantities, or $7100 with cartridge tape drives, from Hewlett-Packard Company, 1501 Page Mill Rd, Palo Alto CA 94304, (415) 493-1501.■

Circle 620 on inquiry card.

Peripheral Boards for Z-80



This new family of peripheral and accessory boards for the Z-80 based MCB series includes the MAD-ONE multiple channel analog interface card with software programmable gains ($595), the Model 606 programmable gain amplifier and filter card with dual channel inputs ($395), the Model 602 prototyping board with or without wire wrap pins ($75), the Model 605 extender card ($95), and the Model 604 card cage with eight card slots ($210). The boards are available from Signal Laboratories Inc, 202 N State College Blvd, Orange CA 92668, (714) 634-1533.■

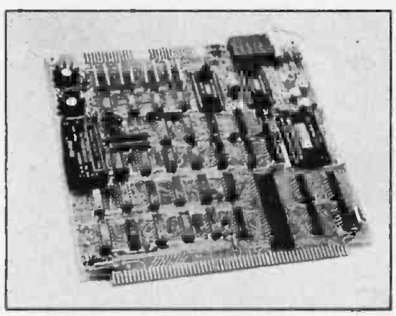
Circle 623 on inquiry card.

Low Cost Hobbyist Keyboard

This inexpensive keyboard features a versatile interface which allows user selection of data and strobe polarity, parity sense, upper case alpha lock, and access to three user definable keys for custom code or function assignment. The Model 753 keyboard provides ASCII encoding for 53 keys in the standard Teletype format, employs KBM keyswitches for reliability and is said to be guaranteed. When built from a kit, estimated construction time is two hours. The Model 753 is $59.95 in kit form or $71.25 assembled and tested. Also available is a custom plastic enclosure, Model 701 ($14.95), which is precut for the Model 753 keyboard. Delivery is from stock, from George Risk Industries Inc, GRI Plaza, Kimball NB 69145, (308) 235-4645.■

Circle 621 on inquiry card.

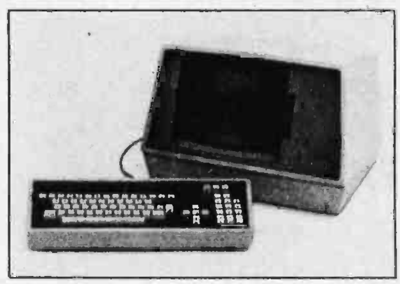
Analog Boards from Zilog



Two new analog boards have been added to Zilog's MCB family of Z-80 microcomputer boards. The Z80-AIB board features 32 analog input channels, analog to digital converter gain ranges up to 0 to 10 V, amplifier gain ranges of 1 to 1000, and 12 bits of conversion resolution. The Z80-AIB is $575, or $675 with an optional DC to DC converter. The Z80-AIO board features 32 input channels and two analog output channels. Output resolution is also 12 bits. The Z80-AIO is $775, or $875 with the DC to DC converter, from Zilog, 10460 Bubb Rd, Cupertino CA 95014, (408) 446-4666.■

Circle 622 on inquiry card.

Smart Terminal



This "smart" editing terminal features an option for use with the Burroughs TD-800 series polling protocol. The detachable keyboard generates the full ASCII character set and has 16 or 32 special function keys. Editing features include tab, back tab and columnar tab operations, protected fields, absolute cursor addressing and cursor position reading. The Burroughs polling features include specific, broadcast and fast selection and multipoint contention mode. The D300 Teletype compatible version is $1645, and the D400 with Burroughs polling features is $1895, both in quantities of 25 with 45 day delivery, from EECO, 1441 E Chestnut Av, Santa Ana CA 92701, (714) 835-6000.■

Circle 624 on inquiry card.

Circle 92 on inquiry card.

✶ YOUR BEST BUY IN WIRE WRAP SUPPLIES ✶

PRECUT WIRE

WHY BUY WIRE ON ROLLS?

PRECUT & STRIPPED WIRE IS:

- **Fast** - No more cutting & stripping by hand
- **Reliable** - Good, clean, uniform strip
- **Economical** - Cheaper than using bulk wire

Precut Wire	Bulk Wire
100 pcs of 3" at $.82 = 3¼¢/ft.	50 ft. roll at $1.99 = 4¢/ft.
100 pcs of 6" at 1.06 = 2¢/ft.	100 ft. roll at 2.95 = 3¢/ft.
Wire Kit 1 at $6.95 = 2 1/3¢/ft.	

30 Kynar stripped 1" on each end. Lengths are overall
Colors: Red,Blue,Green,Yellow,Black,Orange,White
Wire packaged in plastic bags. Add 25¢/length for tubes.

	100	500	1000	5000
2½ in.	.78	2.40	4.30/K	3.89/K
3 in.	.82	2.60	4.71/K	4.22/K
3½ in.	.88	2.80	5.12/K	4.55/K
4 in.	.90	3.00	5.52/K	4.88/K
4½ in.	.94	3.21	5.93/K	5.21/K
5 in.	.98	3.42	6.34/K	5.52/K
5½ in.	1.02	3.65	6.75/K	5.86/K
6 in.	1.06	3.85	7.16/K	6.19/K
6½ in.	1.15	4.05	7.57/K	6.52/K
7 in.	1.20	4.25	7.98/K	6.85/K
7½ in.	1.25	4.45	8.39/K	7.18/K
8 in.	1.29	4.65	8.80/K	7.53/K
8½ in.	1.32	4.85	9.21/K	7.84/K
9 in.	1.36	5.05	9.62/K	8.17/K
9½ in.	1.40	5.25	10.03/K	8.50/K
10 in.	1.45	5.51	10.44/K	8.83/K
Addl. inches	.10	.41	.82/K	.66/K

WIRE KITS

#1	$6.95			#2	$19.95				
250	3"	100	4½"	250	2½"	250	4½"	250	6"
250	3½"	100	5"	500	3"	250	5"	100	6½"
100	4"	100	6"	500	3½"	100	5½"	100	7"
				500	4"	1 250 ft. Roll Bulk			

Choose One Color or Assortment

WIRE WRAP SOCKETS

	1-9	10-24	25-99	100-249	250-999	1K-5K
8 pin*	.41	.38	.35	.31	.29	.27
14 pin*	.42	.39	.36	.32	.29	.27
16 pin*	.46	.43	.39	.35	.32	.30
18 pin*	.63	.58	.54	.47	.44	.41
20 pin	.84	.78	.71	.63	.59	.54
22 pin*	1.30	1.20	1.10	.95	.90	.84
24 pin	.91	.84	.78	.68	.64	.59
28 pin	1.25	1.15	1.08	.95	.89	.82
40 pin	1.65	1.55	1.42	1.25	1.15	1.09

Gold 3-level Closed Entry Sockets

* End & Side Stackable — All prices include gold

Tin sockets and 2-level sockets available

INTERCONNECT CABLES

Ribbon cable connectors for connecting boards to front panels, or board to board.

	SINGLE ENDED			DOUBLE ENDED		
	14 pin	16 pin	24 pin	14 pin	16 pin	24 pin
6"	1.24	1.34	2.05	2.24	2.45	3.37
12"	1.33	1.44	2.24	2.33	2.55	3.92
24"	1.52	1.65	2.63	2.52	2.76	4.31
48"	1.91	2.06	3.40	2.91	3.17	5.08

WIRE WRAP TOOLS

$34.95

HOBBY WRAP
Model BW 630
With Free Wire Kit 1
($6.95 Value)

Batteries & Charger	$11.00
WSU 30 Hand Wrap-Unwrap Strip Tool	5.95
WSU 30M, for Modified Wrap	6.95
BT 30 Extra Bit	2.95

WIRE WRAP BOARDS

				1-4	5-9	10-24	WW Connector
H-PCB-1	4x4½	44	Buses on both sides	4.75	4.50	4.00	3.00
3662	4x6	44	Blank	6.25	5.85	5.50	3.00
3682	4x10	44	Buses on one side	10.50	10.00	9.50	3.00
3682-2	4x6	44	Buses on one side	9.50	9.00	8.25	3.00
4066-4	4x6	72	Buses on both sides	13.50	13.00	12.50	4.00
3719-1	4x6	72	Blank	9.00	8.50	8.00	4.00
3719-4	4x10	72	Blank	11.00	10.50	10.00	4.00
4350	7x9½	80	Buses on both sides	17.50	16.50	16.00	7.00
8800V	10x5.3	100	Buses on both sides	19.50	17.96	16.95	5.00
169P84	8½x17	–	Blank	6.25	5.75	5.25	–

SOCKET SALE!

	1-9	10-49	50 up
14 pin Gold WW	.34	.32	.30
16 pin Gold WW	.36	.35	.33
24 pin Gold WW	.75	.70	.68

Prices good through 3/1/78 when purchased with Wire Kit #1 or #2

Ordering Information:

- Orders under $25 and COD's, add $2
- All others, shipped Ppd in U.S. via UPS
- For Blue Label (Air) or 1st Class, add $1
- We accept Visa & Mastercharge
- Most orders shipped same day

Dealer Inquiries Invited

PAGE DIGITAL ELECTRONICS
135 E. Chestnut Street 4A
Monrovia, California 91016
Phone (213) 357-5005

APPLE II I/O BOARD KIT

Plugs Into Slot of Apple II Mother Board

FEATURES:
- 1 8 Bit Parallel Output Port (Expandable to 3 Ports)
- 1 Input Port
- 15mA Output Current Sink or Source
- TTL or CMOS Compatible
- Addressable anywhere in memory output area
- Can be used for peripheral equipment such as printers, floppy discs, cassettes, paper tapes, etc.

KIT INCLUDES:
P.C. Board, I.C.'s Sockets and Assembly Manual.

PRICE:
1 Input and 1 Output Port for $49.00
1 Input and 3 Output Ports for $64.00

DEALER INQUIRIES INVITED

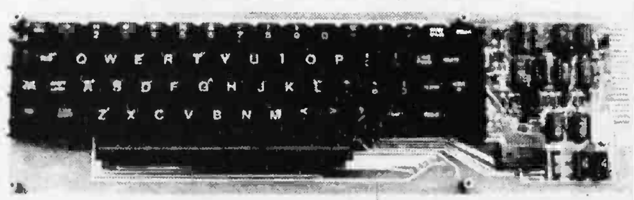
INSTRUCTION MANUAL with SOFTWARE for Interfacing to SWTPC PR-40 Printer, $10.00

LD-130 3 Digits A/D Conv. $11.95	6800 mpu 15.95 6820 pia 13.00 6850 acia 12.00	2708 E Prom 1Kx8 450ns $14.95

SHIPPING AND HANDLING KEYBOARD $3.50. OTHERS – $1.00.
California residents add 6% sales tax

ELECTRONICS WAREHOUSE Inc.
1603 AVIATION BLVD.
REDONDO BEACH, CA. 90278
TEL. (213) 376-8005
WRITE FOR FREE CATALOG
You are invited to visit our store at the above address

3RD GENERATION ONLY $63.00
ASCII KEYBOARD KIT



FURTHER IMPROVEMENTS, MORE FEATURES

- TTL Logic Circuits
- Power: +5V, 275mA
- Upper and Lower Case
- Full ASCII Set (Alpha Numeric, Symbols, Control)
- 7 ot 8 Bits Parallel Data
- Optional Serial Output
- Selectable Positive or Negative Strobe, and Strobe Pulse Width
- 'N' Key Roll-Over
- Fully Debounced
- Carriage Return Key
- Repeat Function Key
- Shift Lock, 2 Shift Keys
- 4 User Defineable Keys
- P.C. Board Size: 17-3/16" x 5"

OPTIONS:
- Metal Enclosure (Painted IBM Blue and White $25.00
- 18 Pin Edge Con. $2.00
- I.C. Sockets . . . $4.00
- Serial Output (Shift Register) $2.00
- Upper Case Lock Switch (for Capital Letters and Numbers) $2.00

KIT Includes: Keyboard, P.C. Board, all required components and assembly manual.

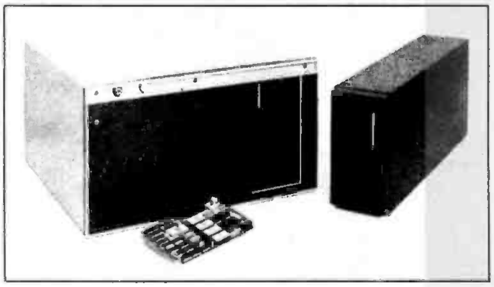
NOTE: If you have this 63 Key Teletype Keyboard you can buy the Kit without it for $44.95.

Circle 48 on inquiry card.

What's New?

MASS STORAGE

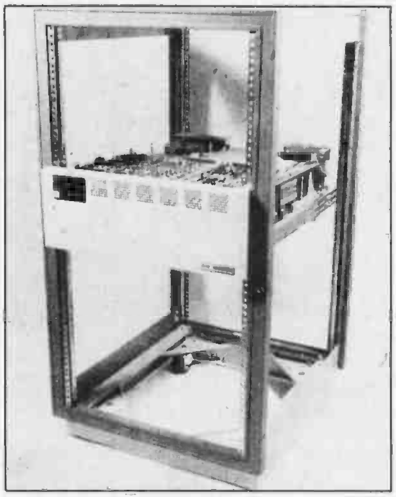
Intelligent Floppy Available in Several Styles



These floppy disk drives feature voice coil positioning for faster access and a microprocessor based intelligent controller. The Model 277 Dual Diskette Drives can be packaged in a variety of configurations: 1) a system with one or two dual drives, controller, power supply and cabling enclosed in a 19 inch rack mountable chassis; 2) a one or two drive system with power supply and cabinet but without the controller, and 3) a "slimline" system which incorporates one dual drive and power supply in a tabletop chassis. The intelligent controller has its own 8080 microprocessor and internal disk operating system in firmware. On command, the 1070 controller can perform all file management functions including disk formatting and initializing. Voice coil positioning is said to be seven to ten times faster than other methods, with an average seek time of 36 ms. Interfaces are available for most of the popular microcomputers. The systems range in price from $740 for the controller only to $3995 for the two drive (four spindle) system with controller, from PerSci Inc, 12210 Nebraska Av, W Los Angeles CA 90025, (213) 820-3764.■

Circle 625 on inquiry card.

New 74 Megabyte Hard Disk



Priced at $6000, the C-D74 disk drive provides a 35 millisecond average access time to any of 74 million bytes of information. With 12 tracks on a cylinder, the device can access any of 220,000 bytes in 5 ms. Single track seek time is 10 ms, and the disk's data transfer rate is 7.3 million bits per second. With its large storage capacity and fast access time, the device is said to be adequate to store all the records of a medium size company. The C-D74 uses "Winchester" technology in a nonremovable sealed chamber drive with a rotary arm positioner, and can run 24 hours a day without worry of disk wear. The C-D74 disk drive, cable, interface for an OSI Challenger and OS-74 operating system software is $6000 FOB the shipper's plant, from Ohio Scientific Instruments, Hiram OH 44234, (216) 569-7905.■

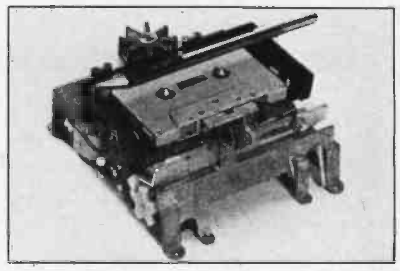
Circle 627 on inquiry card.

Million Byte Floppy Disk System

This floppy disk system comprises four drives in a "dual dual" configuration, a controller, power supply and chassis, enclosure, cabling, and a new BASIC software package. The Meta-Floppy 1054 Mod II will plug into any 8080 or Z-80 based computer using the Altair (S-100) bus and features an all steel head positioner system, electronics capable of reading disks whose signal strength is weak, file protect circuitry and a disk insertion interlock, and lighted numerals to show the logical address of each drive. Track to track access time is about 30 ms, and the data transfer rate is 250,000 bytes per second. The BASIC language system supports line printer spooling and chaining of program segments. The 1054 is $3220 in single quantities, from Micropolis Corp, 7959 Deering Av, Canoga Park CA 91304, (213) 703-1121.■

Circle 633 on inquiry card.

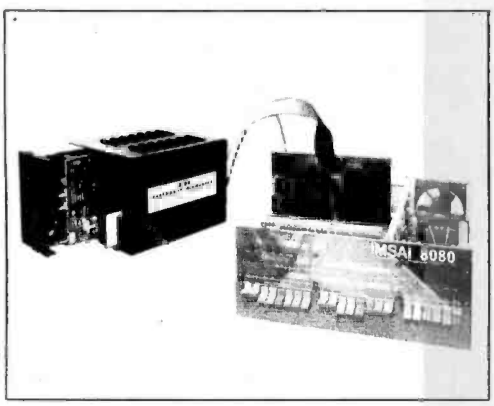
AC Capstan Motor for Cassette Transport



The Phi-Deck cassette tape transport is now available with a fixed speed AC capstan motor. Features of the new model include four motor control, remote control capabilities, fast start and stop, less than 30 seconds rewind time, and speeds from 1 to 10 inches per second. TTL compatible control boards are available for the transport, as are options such as beginning and end of tape sensing, cassette in place sensing, etc. The transport is $149 in single quantities and less than $100 in quantities of 500, from Triple I Inc, POB 18209, Oklahoma City OK 73118, (405) 521-9000.■

Circle 634 on inquiry card.

Minifloppy with FORTRAN IV



This Altair (S-100) bus compatible floppy disk kit allows you to run FORTRAN programs on an 8080 based system with at least 20 K bytes of programmable memory. The kit includes a Shugart SA400 minifloppy disk drive, cables and cabinet, a floppy disk interface board kit, a disk operating system with file management, a text editor, and FORT//80, a FORTRAN IV system for the 8080 from Unified Technologies of Canada. The interface board can control two minifloppy drives and includes a bootstrap and diagnostic program in ROM. Also included are 8 bit parallel input and output ports. The disk operating system (FDOS) manages named files and includes a "sysgen" program for custom tailoring of the operating system IO routines. The FORTRAN system includes double precision arithmetic, in line machine code, FORTRAN control over interrupts, and direct interface of custom IO drivers to FORTRAN READ and WRITE statements. A 90 day warranty and a 2 year software and documentation update service are provided. The package costs $1095 as a kit, or $1220 assembled and tested. A second minifloppy drive is available as a kit for $449, or $495 assembled and tested, from Realistic Controls Corporation, 3530 Warrensville Center Rd, Cleveland OH 44122, (216) 751-3158.■

Circle 626 on inquiry card.

All Prime Quality — New Parts Only

Lowest Prices — Satisfaction Guaranteed

EDGE CARD CONNECTORS:

Bifurcated Contacts. Not tin. *Gold over nickel.* 50/100 Pin (.100, 1.25, + .156 Pin spacing). Double Read out.

50/100 Altair Type – Dip Solder Pins	$4.25 ea	5 pcs	$4.00 ea
50/100 Imsai Type – Dip Solder Pins	$4.25 ea	5 pcs	$4.00 ea
50/100 Imsai Type with Guides	$4.50 ea	5 pcs	$4.25 ea
Imsai Guides only	$ .30/pr	5 pr	$ .25/pr

All other contacts available: Solder Eyelet, Wire Wrap, etc. *All gold.*
Many other types available: 10/20, 15/30, 18/36, etc.
When ordering: *specify type of contact.*

25 PIN SUBMINIATURE CONNECTORS:

Gold Plated Contacts.

DB25P Plug	$2.60 ea	5 pcs	$2.25 ea
DB25S Socket	$3.75 ea	5 pcs	$3.50 ea
DB51212-1 Hood (grey)	$1.00 ea	5 pcs	$ .95 ea
DB51226-1A Hood (black)	$1.10 ea	5 pcs	$1.00 ea

SAVE:
Buy a complete set:
1 Plug, 1 Socket, 1 Hood (any) $6.50/set
5 Sets $6.25/set

2708 1KX8 PROM 450 NS
$16.75 ea
5 pcs or more $15.70 ea

8080A – PRIME
$12.00 ea
5 pcs $11.50 ea

Write for larger quantity discounts.

Dealers welcome.

Minimum order $10.00: Add $1.00 for shipping and handling. Orders over $25.00: We pay the shipping: Calif. residents add 6% tax. No COD's: For immediate shipment send money order or cashier's check.

When ordering: State method of shipment: Mail or UPS.

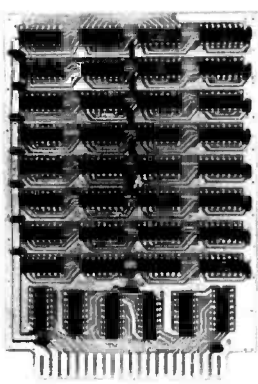
ORDER FROM:

Beckian Enterprises

P.O. Box 3089 Simi, Calif. 93063

4K RAM BOARD

Assembled and tested. **$89.95**



See kit below.

4K RAM BOARD KIT

450ns Access RAMs
Fully Buffered
Low Power **$79.95**
Static
5V only
4½ x 6 inch board

Buy 4 RAM Board kits at $79.95 each and an 8 slot Mother Board is yours **FREE.**

Includes 8 connectors and card guides.

MOTHER BOARD

8 SLOT 44 PIN BUS
50 Pin Edge Connector

Mother Board $20.00 ea
Connectors 2.50 ea
Card guides for above $10.00 per set.

KIM INTERFACE

Board plugs into Kim expansion connector
Connector for our Mother Board
Full memory decoding
Write protect for (4) 4K bytes
Address and data Bus buffers

2102 — $1.20 Multiples of 25 only.
Low power, 450 n.s. Access and cycle.

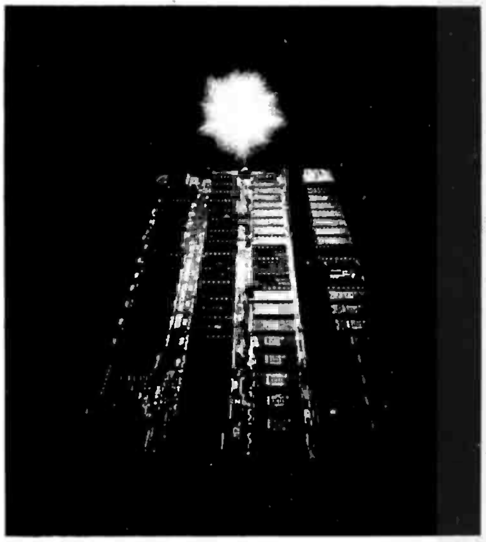
8K EPROM BOARD
5 Volt only For our bus

MAKE CHECK OR MONEY ORDER PAYABLE TO:

Kathryn Atwood Enterprises
P.O. Box 5203, Orange, CA 92667

Discounts available at OEM quantities. For orders less than $25.00 total, add $1.25 for shipping. California residents add 6% sales tax. Estimated shipping time 2 days ARO with money order. For checks allows 7 days for check to clear.

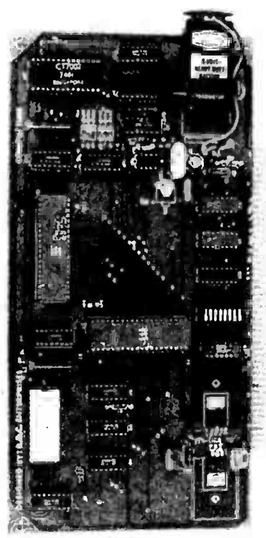
Semibuild This Semikit



A semikit is a fully assembled and wave soldered printed circuit board with pretested integrated circuits, which the user need only test and burn in on his or her own computer. It is designed to eliminate common kit building problems such as bad solder joints, heat damaged components and faulty integrated circuits. Documentation is included with the semikit for the test and burn in procedures. The first semikit is the 16KRA memory board, which includes 16 K bytes of programmable memory in 4 K independently addressable blocks, with an invisible refresh and a worst case access time of 400 ns. The 16KRA is $369 in semikit form and $399 tested and burned in, from Processor Technology Corp, 6200 Hollis St, Emeryville CA 98608, (415) 652-8080.■

Circle 628 on inquiry card.

CompuTime Offers Clock, Calendar and Calculator on One Board



CompuTime has announced an Altair (S-100) bus compatible PC board which combines a real time clock, calendar and 40 function scientific calculator in one package.

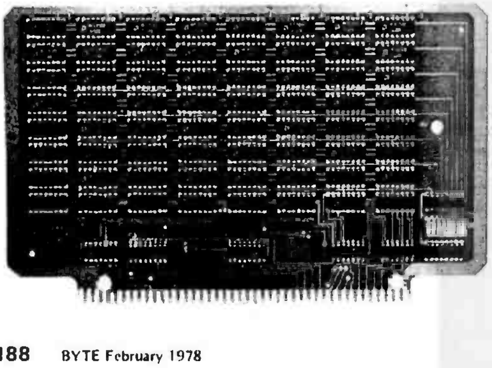
Applications for the clock and calendar include stamping output listings with time and date, plus alarm and timing operations which can be implemented by means of two coincidence counters provided on the board. If power is shut down, a battery backup system is provided.

The 40 function calculator enables the computer to handle floating point, trigonometric and algebraic problems as well as basic math functions.

The package is available in three configurations: time, date and calculator, kit price $199; time and date only, kit price $165; and calculator only, kit price $149. The boards are also available assembled and tested. Contact CompuTime, POB 417, Huntington Beach CA 92648, (714) 638-2094.■

Circle 631 on inquiry card.

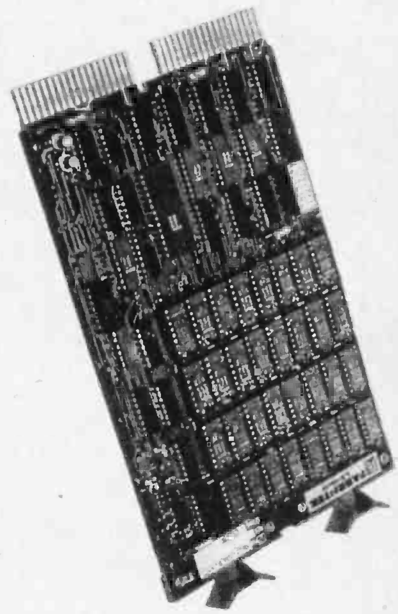
Memory Board for EXORciser and MEK6800



An 8 K static memory board now available is pin and signal compatible with the bus used in the Motorola EXORciser, Micromodules, and the MEK6800D1 and MEK6800D2 Evaluation Kits. The 9626 board features full 16 bit address decoding and buffered address, data and control lines. The 9626 is $350 in single quantities and $210 in lots of 100 from Creative Micro Systems, 6773 Westminster Av, Westminster CA 92683, (714) 892-2859.■

Circle 629 on inquiry card.

Add-on Memory for LSI-11 and PDP-11/03



A new high density memory card for the LSI-11 and PDP-11/03 is available from Fabri-tek Inc, 5901 S County Rd 18, Minneapolis MN 55436, (612) 935-8811. The LS-IN-11 provides 8, 16, 24 or 32 K bytes of memory on a single card with a 2 slot connector, using 8 K or 16 K dynamic MOS n-channel memory chips. A typical low quantity price for the 16 K version of the card is $1085 with a 12 month warranty.■

Circle 632 on inquiry card.

Attention Readers and Vendors...

Where Do New Product Items Come From?

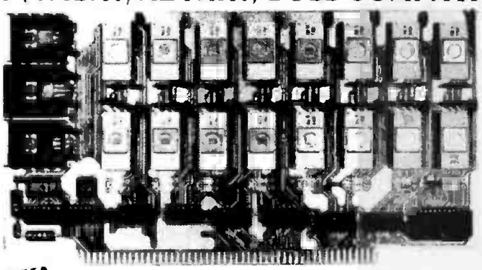
The information printed in the new products pages of BYTE is obtained from "new product" or "press release" copy sent by the promoters of new products. If in our judgment the neat new whiz-bang gizmo or save the world software package is of interest to the personal computing experimenters and homebrewers who read BYTE, we print the information in some form. We openly solicit such information from manufacturers and suppliers to this marketplace. The information is printed more or less as a first in first out queue, subject to occasional priority modifications.

LOW PROFILE IC SOCKETS	AUGAT WIRE WRAP SOCKETS	TRIMMER CAP	P. C. MOUNT VOLUME CONTROL
14 PIN - $.18 22 PIN - $.33 16 PIN - $.20 28 PIN - $.48 18 PIN - $.25	16 PIN Gold Plated $.49 ea. 10 FOR $3.95 LIMITED STOCK	2 - 20 PF CER. 4/$1 Mil. Quality	Nylon Shaft. 1K or 100K. YOUR CHOICE 4/$1

16K E-PROM CARD

$69.95 (KIT)

S-100 (IMSAI/ALTAIR) BUSS COMPATIBLE



IMAGINE HAVING 16K OF SOFTWARE ON LINE AT ALL TIME!

KIT FEATURES:

1. Double sided PC Board with solder mask and silk screen and Gold plated contact fingers.
2. Selectable wait states.
3. All address lines and data lines buffered!
4. All sockets included.
5. On card regulators.

USES 2708's!

KIT INCLUDES ALL PARTS AND SOCKETS! (EXCEPT 2708's)

ADD $25 FOR ASSEMBLED AND TESTED

WOW! DEALER INQUIRES INVITED

SPECIAL OFFER: Our 2708's (450 NS) are **$12.95** when purchased with above kit.

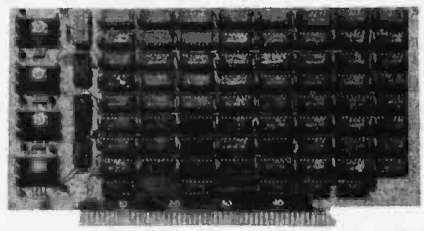
FULLY STATIC! $149.00 KIT

8K LOW POWER RAM KIT!

KIT FEATURES:

1. Double sided PC Board with solder mask and silk screen layout. Gold plated contact fingers.
2. All sockets included!
3. **Fully** buffered on all address and data lines.
4. **Phantom** is jumper selectable to pin 67.
5. **FOUR** 7805 regulators are provided on card.

S-100 (IMSAI/ALTAIR) BUSS COMPATIBLE



USES 21LO2-1 RAM'S.

PRICE WAR!
For a limited time only:
Buy two 8K Kits for **$129 ea.**

Fully Assembled and Burned In —	**$179.00**
Blank PC Board With Documentation —	**29.95**
Low Profile Socket Set –	13.50
Support IC's (TTL and Regulators) –	9.75
Bypass CAP's (Disc and Tantalums) –	4.50

3W AUDIO AMP MODULE

Fully assembled and tested. With schematic. 4, 8 or 16 OHMS.

$3.95 NEW!

JUMBO RED LED'S

New by G.E. Like MV5024 #SSL-22

6 FOR $1   25 FOR $3.75

RCA HOUSE #2N3772

NPN Power Transistor. 30 AMP. 150 W. VCEO-60. TO-3. Vastly out performs 2N3055. Reg. List $3.04

2 FOR $1

NEW! **4K STATIC RAM'S** NEW!

2114. The industry standard. 18 PIN DIP. Arranged as 1K X 4. Equivalent to FOUR 21LO2's in ONE package! TWO chips give 1K X 8. with data.

2 FOR $24   450 N.S.!   8 FOR $85

MOTOROLA 7805R VOLTAGE REGULATOR

Same as standard 7805 except 750 MA OUTPUT. TO-220. 5VDC OUTPUT.

$ .44 each   10 FOR $3.95

450 NS!   2708 EPROMS   450 NS!

Now Full Speed! Prime new units from a major U.S. Mfg. 450 N.S. Access time. 1K x 8. Equiv. to 4 1702 A's in one package!

$15.75 ea.

Special Offer: **$12.95 ea.** when purchased with our 16K EPROM Kit.

Z - 80 PROGRAMMING MANUAL

By MOSTEK, the major Z - 80 second source. The most detailed explanation ever on the workings of the Z - 80 CPU CHIPS. At least one full page on **each** of the 158 Z - 80 instructions. A **MUST** reference manual for any user of the Z - 80. 300 pages. Just off the press! A D.R.C. exclusive! **$12.95**

PRICE WAR!

21L02-1 LOW POWER RAM! Prime. 500 N.S. FAST! **8 FOR $9.95**

8 POSITION DIP SWITCH

By CTS. Fits 16 Pin Socket. **$1.95**

TERMS: ORDERS UNDER $15 ADD $ .75. NO C.O.D. WE ACCEPT VISA, MASTER CHARGE AND AMERICAN EXPRESS CARDS. MONEY BACK GUARANTEE ON ALL ITEMS. TEXAS RESIDENTS ADD 5% SALES TAX.

Digital Research Corporation

P. O. BOX 401247 • GARLAND, TEXAS 75040 • (214) 271-2461

CALL FOR OUR CATALOG!

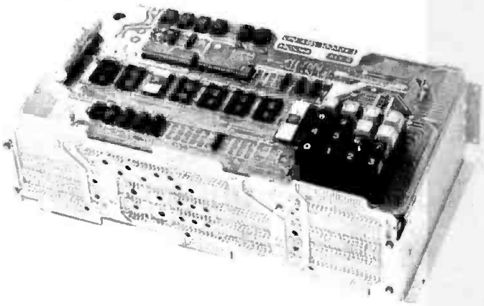
WE PAY POSTAGE!

Circle 41 on inquiry card.

What's New?

SYSTEMS

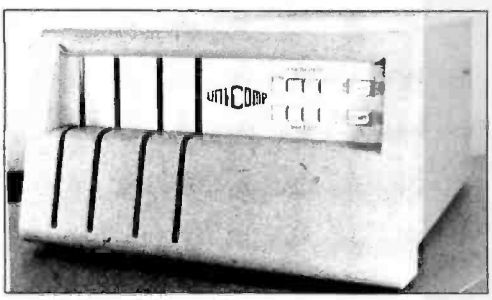
6800 Based System for OEMs



Designed especially for original equipment manufacturers, the MBC Microcomputer System comes complete with a keyboard and 6 digit hexadecimal display, a system monitor, general purpose board, four slot motherboard and flexible mounting system. The mounting frame, which will accept three peripheral boards, is provided with brackets to allow for front, back, side or 19 inch rack panel mounting. The main computer board has room for up to 768 bytes of programmable memory, 2.5 K bytes of read only memory, and a current loop, RS232 and cassette interface. The M68-MBC is $695 from Electronic Product Associates Inc, 1157 Vega St, San Diego CA 92110, (714) 276-8911.■

Circle 641 on inquiry card.

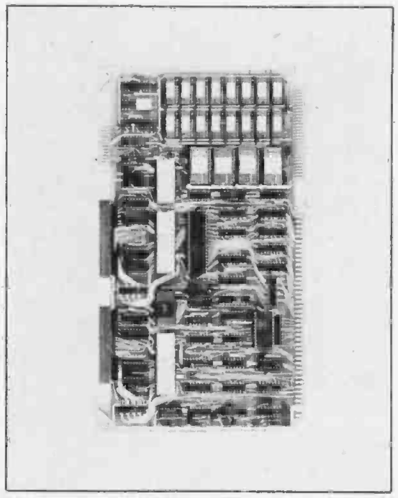
Integrated Package Based on LSI-11



The LSI based system includes a large backplane, dual drive floppy disk and and power supply all in one package. The SS-11/15 is available in a single 10.5 inch rack or a tabletop mounting enclosure, and includes a 15 quad slot backplane, console interface and switch register, diagnostic and bootstrap ROM bus terminator, and distributed refresh controller. The system is compatible with Digital Equipment Corp software such as the RT-11 and RSX-11/S operating systems and multiuser BASIC, FORTRAN, and MACRO-11. The SS-11/15 is backed by a one year warranty and is delivered with all unused card slots occupied by bus grant continuity boards to simplify testing of custom interfaces, from Unicomp Inc, 8950 Westpark, Suite 312, Houston TX 77063, (713) 782-1750.■

Circle 640 on inquiry card.

A New Single Board Z-80 Computer



Monolithic Systems Corporation has introduced the SBC-80 Multibus compatible computer featuring 8 K bytes of static programmable memory and 8 K of erasable programmable read only memory sockets with serial and parallel IO ports. Designated the MSC 8001, the single board computer uses the Z-80 processor and has up to 4 MHz clock speed. It is electrically and mechanically compatible with the SBC 80 systems, operating as a master module in the Multibus scheme.

The two parallel IO ports consist of parallel peripheral interface circuits with buffers and terminators to protect all internal MOS circuitry. A total of 48 lines are available. They can be configured for either positive or negative logic signals.

The serial IO port of the MSC 8001 supports RS232C, TTL or current loop compatible serial IO devices with programmable data transfer rate. Asynchronous and synchronous data formats can be programmed. The current loop interface is optically isolated to protect the MSC 8001 from transients or ground loops caused by peripheral equipment.

Real time processing is provided with the 8253 interval timer. The timer contains three 16 bit counters which operate independently. One is dedicated to the serial IO port and the other two are available for general use. The unit provides eight levels of fully vectored priority interrupts. The memory is available with either 4 K or 8 K of standard 18 pin, 4 K x 1 static programmable memory.

The MSC 8001 single board computer is $845, including all interface elements and 8 K bytes of programmable memory.

Contact Dick Lorimor, Monolithic Systems Corporation, 14 Inverness Dr E, Englewood CO 80110, (303) 770-7400.■

Circle 637 on inquiry card.

LSI-11 Based Floppy Disk System

This dual drive floppy disk system is available with or without an integral LSI-11 processor and is said to be identical in function to the PDP-11V03. The Micro-Flop 11 uses the Shugart SA800 disk drive with the SA850 double sided disk available as an option. The disk controller features a dedicated 8080 microprocessor which implements a disk self-test feature independently of the LSI-11. A front panel console, 10.5 inch (26.7 cm) enclosure, and the Digital Equipment Corporation H9270 backplane are included. The Micro-Flop 11 is $3,350 without the LSI-11 and $4,290 with the LSI-11 included, from Charles River Data Systems, 235 Bear Hill Rd, Waltham MA 02154, (617) 890-1700.■

Circle 638 on inquiry card.

Commodore Ships First PET Computers

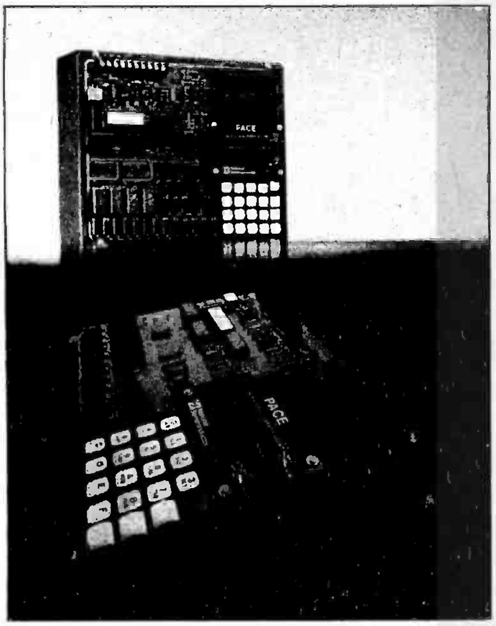
The PET computer made its debut recently as the first 100 units were shipped to waiting customers in mid October 1977. Here Commodore Systems Division Director Chuck Peddle is pictured with the PETs undergoing final checkout. Shipments were made about six weeks later than expected, according to Peddle. The delay was due in part to time consuming quality control measures and the material flow problem in starting up the production lines. "In this business," Peddle argued, "six weeks is actually pretty good." Many of the first units were delivered to customers who intend to develop software for the PET. Commodore plans to create a publishing house for programs developed by users as well as employees. The company plans to increase production of the PET computers to several thousand per month by early 1978. The basic PET with 4 K memory is priced at $595, while the 8 K memory version is $795, from Commodore Business Machines Inc, 901 California Av, Palo Alto CA 94304, (415) 326-4000.■

Circle 639 on inquiry card.

What's New?

SYSTEMS etc

Low Cost 16 Bit Microprocessor Development System



This low cost development system (LCDS) lets the user gain experience with and develop programs for the 16 bit PACE microprocessor for a basic cost of only $585. Fully assembled on a printed circuit card, the LCDS includes the microprocessor, 1 K 16 bit words of programmable memory, sockets for 1 K words of programmable read only memory, a 20 key dual function keyboard, a six digit light emitting diode display, a timer, input output buffers and bidirectional transceivers. On board ROM contains a system monitor for the keyboard, display, and control of input output subroutines. Both a 20 mA current loop interface and an RS232 port are provided. Three prewired, 72 pin sockets allow for additional memory or for expansion of the LCDS interface bus. Expansion boards may be plugged directly into these sockets, or a prewired cable assembly can be used to connect the unit to an expansion chassis. Maximum system memory is 60 K words. The LCDS microprocessor can be isolated from the system bus, allowing an external PACE to use LCDS memory and peripherals. This feature makes it easier to check out prototyping hardware as it is developed. Documentation includes an 80 page *Microprocessor System Design Manual*, a 96 page *LCDS User's Manual*, a 112 page *Assembly Language Programmer's Manual*, data sheets and schematic drawings. The unit requires a 5 V power supply delivering 2.8 A plus additional current for any memory expansion cards, and a 12 V supply for the RS232 interface. The LCDS is priced at $585; expansion options include the IPC-16C/011 card including 1 K words of programmable memory for $170, the IPC-16C/012B card providing sockets for 2 K words of read only memory for $139, and the IPC-16P/802 expansion cable assembly for $145, from National Semiconductor Corp, 2900 Semiconductor Dr, Santa Clara CA 95051, (408) 737-5000.■

Circle 642 on inquiry card.

Correction

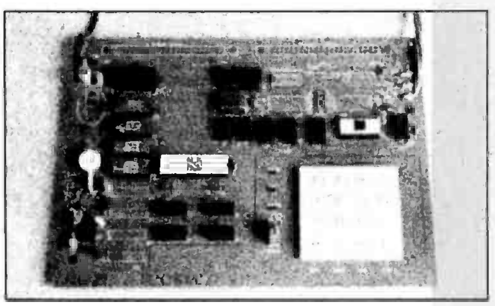
On page 208 of the December 1977 BYTE we gave the incorrect address for ordering Hewlett-Packard's HP-01 wristwatch calculator. The unit is presently being marketed only through jewelry stores. To obtain the name of the store nearest you, call toll free (800) 648-7111; in Nevada, call 329-2700. Our thanks to HP's Mike Rosenthal for this information.■

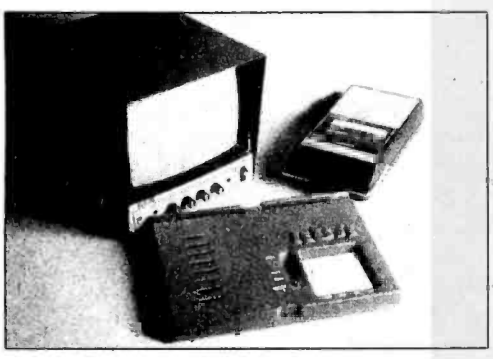
For Z-80 Users

The Z80-PDS program development system includes a floppy disk drive with up to 300 K bytes of online storage, 3K bytes of read only memory and 16 K bytes of programmable memory, and serial IO with RS232 or strappable current loop interface. Software for the system includes a disk resident operating system, editor, assembler, debugger and file handling utilities. The Z80-PDS can be used with any standard CRT or hard copy terminal at data rates from 110 to 19,200 bps. The system may also connect directly to a soon to be available optional keyboard and video monitor by means of the Z80-VDB video display board. Other optional modules are the Z80-PPB programmer board, Z80-IOB input output board, and the Z80-SIB serial IO board. The card enclosure measures 15 by 10 by 4 inches and weighs only 5 pounds, while the disk unit, 16 by 4.75 by 9 inches, weighs 10 pounds. The program development system is priced at $2850 in single quantities from Zilog, 10460 Bubb Rd, Cupertino CA 95014, (408) 446-4666.■

Circle 643 on inquiry card.

COSMAC Based Kit Aimed at Hobbyist





This low cost hobby computer kit features video display and audio cassette recorder IO as well as a low level interpretive programming language especially designed for the creation of compact games and graphics. The COSMAC VIP is based on the CDP1802 microprocessor and uses the CDP1861 video chip to control the video display. The VIP is built on a single 8.5 by 11 inch printed circuit card and provides 2 K bytes of programmable memory using 4 K bit static memory chips, and 512 bytes of read only memory containing a monitor program which permits the user to examine and alter memory, save and load programs on cassette tape, and examine the processor registers. The cassette interface operates at 100 bytes per second using any reasonably good audio cassette recorder. The CHIP-8 programming language simplifies the task of programming video games in hexadecimal code. CHIP-8 has 31 instructions in a 2 byte format for functions such as displaying a pattern on the video display, generating a random number, sounding a tone, etc, and provides 16 one byte variables and subroutine nesting capability. Memory expansion to 4 K bytes and parallel IO expansion to 19 lines can be achieved by inserting additional integrated circuits on the printed circuit board, and additional memory and peripherals can be added through the 44 pin memory and input output expansion connector sockets on the board. The VIP user's manual, said to be written by a hobbyist for hobbyists, contains detailed information on kit assembly, operating procedures, CHIP-8 programming techniques, test programs and trouble shooting hints, and system expansion instructions. The manual also includes program listings for 20 video games which can be immediately entered and played by the user without having to learn programming.

Priced at $275, the COSMAC VIP is available from RCA Solid State Division, POB 3200, Somerville NJ 08876, (201) 685-6423.■

Circle 644 on inquiry card.

PLETE SATIS[illegible]ON IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION

HOBBY WORLD

HUGE DISCOUNTS!

Savings up to 70% on major brand IC parts and computer kits.

74L$00 TTL

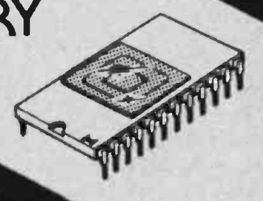
74L$00	$ .21	74L$93	.52	74L$192	.90
74L$02	.21	74L$109	.36	74L$196	.80
74L$04	.24	74L$112	.36	74L$197	.80
74L$08	$ .21	74L$113	.36	74L$221	1.06
74L$10	.21	74L$114	.36	74L$257	.71
74L$14	.85	74L$125	.46	74L$258	.70
74L$20	.23	74L$126	.46	74L$266	.26
74L$21	.23	74L$132	.75	74L$283	.72
74L$22	.23	74L$138	.70	74L$365	.55
74L$30	.23	74L$139	.70	74L$366	.55
74L$32	.30	74L$151	.65	74L$367	.55
74L$37	.31	74L$153	.66	74L$368	.55
74L$38	.31	74L$154	1.00	74L$386	.39
74L$42	.60	74L$157	$.62	81L$95	.77
74L$47	.75	74L$160	.82	81L$96	.77
74L$48	.72	74L$161	.82	81L$97	.77
74L$73	.35	74L$162	.82	81L$98	.77
74L$74	.35	74L$163	.82		
74L$75	.53	74L$164	.98		
74L$76	.37	74L$174	.75		
74L$86	.36	74L$175	.79		
74L$90	.52	74L$190	.90		
74L$92	.52	74L$191	.90		

SPECIAL

1702A MEMORY

JUST

$4.75 ea.



74XX

7400	.11			74100	.28
7401	.13			74107	.28
7402	.13			74109	.22
7403	.13	7441	.70	74121	.29
7404	.15	7442	.37	74122	.38
7405	.13	7443	.59	74123	.48
7406	.16	7444	.59	74132	.65
7407	.23	7445	.65	74141	.70
7408	.17	7446	.62	74145	.65
7409	.17	7447	.59	74150	88
7410	.13	7448	.60	74151	.61
7411	.18	7450	.13	74153	61
7412	.13	7451	.13	74154	.95
7413	.25	7453	.13	74157	.55
7414	.61	7454	.13	74161	.55
7416	.24	7460	.14	74163	.55
7417	22	7470	.26	74164	85
7420	.13	7472	.21	74165	.90
7423	.25	7473	.21	74170	1.69
7425	.25	7474	.27	74173	1.10
7426	.22	7475	.45	74175	.85
7427	.17	7476	.28	74175	.75
7430	.13	7479	5.00	74176	.69
7432	.23	7480	.31	74177	.70
7437	.21	7481	.95	74180	.65
7438	.21	7482	.50	74181	1.75
7439	25	7483	.62	74190	.95
7440	.13	7485	.75	74191	.95
		7486	.26	74192	.79
		7489	1.75	74193	.80
		7490	.40	74195	.49
		7491	.51	74221	.86
		7492	.40	74251	1.00
		7493	.40	74365	.62
		7494	.60	74366	.62
		7495	.60	74367	.62
		7496	.60	74368	.62

SUPPORT DEVICES

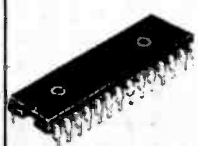
6820	5.00
6850	5.00
8212	3.45
8214	9.50
8216	3.75
8224	3.75
8228	6.50
8226	3.85
8238	7.95

MEMORY

1702A	4.75
2708	13.50
2716	32.95
21L02 450ns	1.25
21L02 250ns	1.60
1101	.75

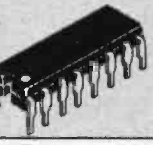
MICROPROCESSOR

8080A	$11.95
Z-80	24.95
Z-80A	34.95
6800	16.50



CMOS

		4023	$ .16
		4024	.67
4001	.16	4025	.16
4002	.16	4028	.74
4006	.99	4030	.22
4007	.16	4035	.98
4008	.79	4040	.99
4009	.37	4049	.34
4010	.37	4050	.34
4011	.16	4051	.89
4012	.16	4052	1.15
4013	.29	4053	.89
4014	.74	4066	.59
4015	.74	4071	.19
4016	.29	4081	.23
4017	.79	4082	.23
4018	.79	4507	.39
4019	.23	4510	1.05
4020	.84		
4021	.89		
4022	.89		



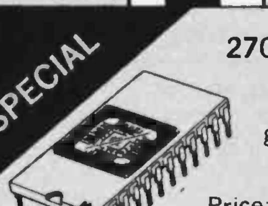
KITS

WAMECO
S-100 P.C. Boards
- 8K RAM 28.00
- Z-80 CPU 28.00
- 12-Slot Mother Board 33.00

ITHACA AUDIO
S-100 P.C. Boards
- 8K RAM 28.00
- Z-80 CPU 28.00

SOLID STATE MUSIC
S-100 Kits & Bare Boards
- MB-3 2K/4K EPROM
  Uses 1702A EPROMS
  Kit w/o EPROMS 59.95
- MB-4 4K STATIC RAM
  Kit 95.00
  Bare Board 25.95
- MB6A 8K STATIC RAM
  Kit $179.95
  Bare Board 25.95
- MB7 16K STATIC RAM
  Kit 435.00
  Bare Board 25.95
- MB8 8K/16K EPROM
  Uses 2708's
  Kit Less EPROMs 75.95

SPECIAL



2708 Special

Buy 10
get 1 free

Price: $13.50 each

450 μs Full Specification Part

KITS

SOLID STATE MUSIC
- MB9 STATIC PROM/RAM
  Kit Less Memory 72.00
- VB1A VIDEO INTERFACE
  Kit 129.95
  Bare Board 25.95
- IO2 PARRELL I/O And Kludge
  Kit 49.95
  Bare Board 25.95
- IO4 2+2 I/O
  Kit 139.95
- SB-1 MUSIC SYNTHESIZER
  Kit with Software 174.95
- MT-1 15-Slot Mother Board 39.95
- XB-1 EXTENDER BOARD
  Bare Board 8.99
- SSM 8080 MONITOR V1
  ON 2-2708 47.00
  ON 8-1702A 47.00

LINEAR

LM 301H	.35		
LM 301AH	.38		
LM 301AM	.30		
LM 304H	1.00		
LM 305H	$ .60	LM 358M	.99
LM 307M	.30	LM 379M	3.99
LM 308H	1.00	LM 380M	.99
LM 308AH	.95	LM 380N	.75
LM 308N	1.00	LM 381N	1.79
LM 311M	.90	LM 386N	.99
LM 311N	.85	LM 387N	.69
LM 311H	.90	LM 3900N	.40
LM 312H	1.25	LM 555M	.39
LM 317H	3.45	LM 556N	.88
LM 317K	3.70	LM 567M	1.25
LM 318M	1.15	LM 723N	.49
LM 319H	1.25	LM 741H	$ .35
LM 321H	2.02	LM 741M	.19
LM 322H	2.02	LM 747H	.79
LM 324N	.72	LM 747N	.79
LM 325H	2.25	LM 1458M	.59
LM 339N	.99	LM 14558H	.55
LM 343H	2.99	LM 4250CH	.89

REGULATORS

		LM 320K-18	1.75
		LM 320K-24	1.75
		LM 320T-5	1.10
LM 309H	1.10	LM 320T-12	1.10
LM 309K	.99	LM 320T-15	1.10
LM 320H-5	1.19	LM 340K-5	.85
LM 320H-6	1.19	LM 340K-8	1.35
LM 320H-8	1.19	LM 340K-12	.85
LM 320H-12	1.19	LM 340K-15	.85
LM 320H-15	1.19	LM 340K-18	1.35
LM 320H-18	1.19	LM 340K-24	.85
LM 320H-24	1.19	LM 340T-5	.95
LM 320K-5	1.35	LM 340T-8	.95
LM 320K-6	$1.75	LM 340T-12	.95
LM 320K-8	1.75	LM 340T-15	.95
LM 320K-12	1.35	LM 340T-18	.95
LM 320K-15	1.35	LM 340T-24	.95

Circle 56 on inquiry card.

HOBBY WORLD

5716 W. Manchester Ave.
Suite #5
Los Angeles, CA 90045

TELEPHONE ORDERS:
Call (213) 641-4200

master charge

BANKAMERICARD VISA

☐ Cash
☐ COD

Charge My
☐ M.C.
☐ BAC (VISA)

Sig.

☐ Send your complete catalog, quickly.
☐ Please send me the following items I have listed below:

Stock No.	Qty.	Price
	Postage/Handling	$1.50

Name

Address

City State Zip

Satisfaction 100% Guaranteed

California Residents Add 6% Sales Tax

Note: Minimum Order $10.00, 5% Discount over $100.00

NEW 1978

IC MASTER

- Over 40,000 IC's listed.
- Over 2,000 pages.
- Updated every 90 days.

Retail Value
$55.00

Your Price
$48.50

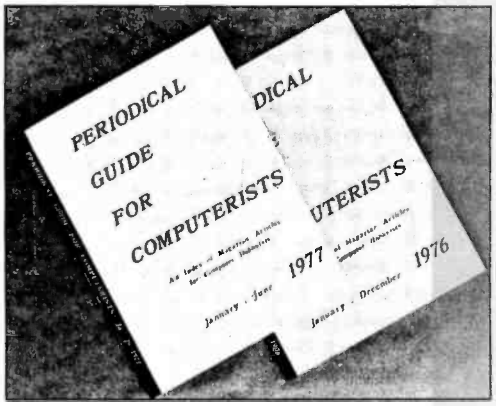
IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE

IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR

What's New?

PUBLICATIONS

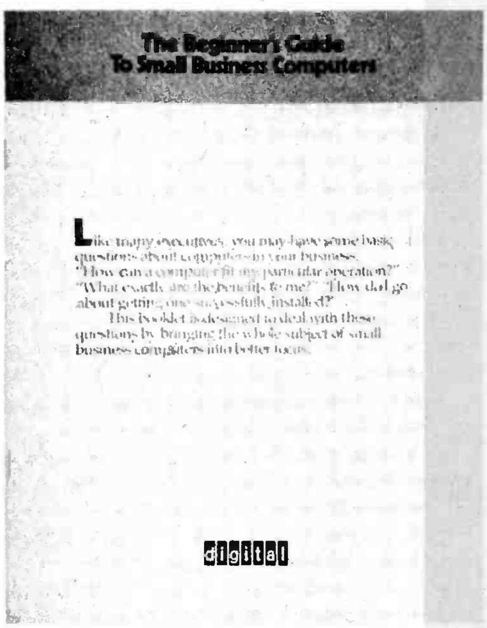
Guide Cross Indexes Personal Computing Magazines



The January to June *Periodical Guide for Computerists* indexes 1080 articles from 23 hobby and professional computer publications. Articles, editorials, book reviews, and letters from readers which have relevance to the personal computing field are indexed by subject under 90 categories. The 32 page book is available postpaid for $3 from E Berg Publications, 1360 SW 199th Ct, Aloha OR 97005, (503) 649-7495, or from local computer stores.■

Circle 645 on inquiry card.

Guide to Small Business Computers



This free brochure, said to unravel the mysteries of small business computing systems, details a step-by-step approach to matching computer capabilities with business needs. Copies are available from Digital Equipment Corporation, Communications Services, Brochure EA 07430, 444 Whitney St, Northboro MA 01532, (617) 897-5111.■

Circle 646 on inquiry card.

Computerlogue Stardated Fall 1977

This 22 page microcomputer catalog includes products from all the major manufacturers. Separate prices are given for credit card purchases and for cash purchases, which receive a discount. *Computerlogue* is available from Computer Enterprises, POB 71, Fayetteville NY 13066, (315) 637-6208.■

Circle 647 on inquiry card.

Signature Analysis: A New Applications Note from Hewlett-Packard

Signature analysis is a new technique for debugging microcomputer circuitry and other circuitry designed around bus architecture. Data bit streams, which are common in this type of architecture, present special problems when fault analysis is required. Hewlett-Packard details some of the new techniques used in signature analysis in its free 50 page *Applications Note 222*, available from the Inquiries Manager, Hewlett-Packard Company, 1501 Page Mill Rd, Palo Alto CA 94304, (415) 493-1501.■

Circle 648 on inquiry card.

Bubble Memory Report

This report analyzes the impact of bubble memory technology on end user products such as point of sale terminals and word processors, programmable calculators and home computers and the implications for competitive memory systems such as cassettes and disks, charge coupled devices and MOS memory. A complete facilities plan for production of bubble memories is included. Other reports such as "Small Business Systems Industry Report" and "Data and Word Processing Opportunities in the Automation of Legal Work" are also available. The bubble memory report is $995 from Small Business Systems, 4320 Stevens Creek Blvd, Suite 230, San Jose CA 95129 (408) 243-8121.■

Circle 649 on inquiry card.

Brochure Describes Analog Peripherals

This 16 page brochure provides the specifications for the SineTrac 800 series of data acquisition cards for the Intel MDS-800 and SBC-80/20/10 microcomputers. Accepting 32 or more analog channels, the high speed SineTrac 800 communicates over the processor bus as an addressable IO device. The brochure is available from Datel Systems Inc, 1020 Turnpike St, Canton MA 02021, (617) 828-8000.■

Circle 650 on Inquiry card.

At Last, a Microcomputer Troubleshooting Manual

What do you do when your newly assembled microcomputer kit doesn't work? Thousands of hobbyists undoubtedly have been in this predicament, and most learn the art of troubleshooting digital circuits the hard way. This manual, written by a test engineer and technical writer, may provide a short cut. It offers general hints and specific procedures for finding and curing common problems arising with the components of microcomputer systems. Separate sections treat general problem solving approaches, troubleshooting newly assembled equipment, and fixing a system which has worked properly prior to the latest failure. Typical problems with processor boards, memory boards and television interface boards are treated in some detail. A glossary of terms and a list of recommended component suppliers is included. $5 from Micro-Info Associates, POB 849, Castroville CA 95012.■

Circle 651 on inquiry card.

Notes for Altair Computer Users

Computer Notes is a monthly publication for owners of Altair computers, providing tutorial articles, hints and project ideas. The September 1977 issue includes articles on building your own video display and on robot mechanics. *Computer Notes* is free to Altair computer owners; 50 cents per issue or $5 per year from MITS Inc, 2450 Alamo SE, Albuquerque NM 87106.■

Circle 652 on inquiry card.

DIODES/ZENERS

1N914	100v	10mA	.05
1N4005	600v	1A	.08
1N4007	1000v	1A	.15
1N4148	75v	10mA	.05
1N753A	6.2v	z	.25
1N758A	10v	z	.25
1N759A	12v	z	.25
1N4733	5.1v	z	.25
1N5243	13v	z	.25
1N5244B	14v	z	.25
1N5245B	15v	z	.25

SOCKETS/BRIDGES

8-pin	pcb	.25	ww	.45
14-pin	pcb	.25	ww	.40
16-pin	pcb	.25	ww	.40
18-pin	pcb	.25	ww	.75
22-pin	pcb	.45	ww	1.25
24-pin	pcb	.35	ww	1.10
28-pin	pcb	.35	ww	1.45
40-pin	pcb	.50	ww	1.25

Molex pins .01 To-3 Sockets .45

2 Amp Bridge 100-prv 1.20

25 Amp Bridge 200-prv 1.95

TRANSISTORS, LEDS, etc.

2N2222	NPN	(Plastic .10)	.15
2N2907	PNP		.15
2N3906	PNP		.10
2N3054	NPN		.35
2N3055	NPN	15A 60v	.50
T1P125	PNP	Darlington	.35
LED Green, Red, Clear			.15
D.L.747	7 seg 5/8" high com-anode		1.95
XAN72	7 seg com-anode		1.50
FND 359	Red 7 seg com-cathode		1.25

C MOS

4000	.15
4001	.20
4002	.20
4004	3.95
4006	1.20
4007	.35
4008	.95
4009	.30
4010	.45
4011	.20
4012	.20
4013	.40
4014	1.10
4015	.95
4016	.35
4017	1.10
4018	1.10
4019	.60
4020	.85
4021	1.35
4022	.95
4023	.25
4024	.75
4025	.35
4026	1.95
4027	.50
4028	.95
4030	.35
4033	1.50
4034	2.45
4035	1.25
4040	1.35
4041	.69
4042	.95
4043	.95
4044	.95
4046	1.75
4049	.70
4050	.50
4066	.95
4069	.40
4071	.35
4081	.70
4082	.45

– T T L –

7400	.15	7473	.25	74176	1.25	74H72	.55	74S133	.45		
7401	.15	7474	.35	74180	.85	74H101	.75	74S140	.75		
7402	.20	7475	.35	74181	2.25	74H103	.75	74S151	.35		
7403	.20	7476	.30	74182	.95	74H106	.95	74S153	.35		
7404	.15	7480	.55	74190	1.75			74S157	.80		
7405	.25	7481	.75	74191	1.35	74L00	.35	74S158	.35		
7406	.35	7483	.95	74192	1.65	74L02	.35	74S194	1.05		
7407	.55	7485	.95	74193	.85	74L03	.30	74S257 (8123)	.25		
7408	.25	7486	.30	74194	1.25	74L04	.35				
7409	.15	7489	1.35	74195	.95	74L10	.35	74LS00	.35		
7410	.10	7490	.55	74196	1.25	74L20	.35	74LS01	.35		
7411	.25	7491	.95	74197	1.25	74L30	.45	74LS02	.35		
7412	.30	7492	.95	74198	2.35	74L47	1.95	74LS04	.35		
7413	.45	7493	.40	74221	1.00	74L51	.45	74LS05	.45		
7414	1.10	7494	1.25	74367	.85	74L55	.65	74LS08	.35		
7416	.25	7495	.60			74L72	.45	74LS09	.35		
7417	.40	7496	.80	75108A	.35	74L73	.40	74LS10	.35		
7420	.15	74100	1.85	75110	.35	74L74	.45	74LS11	.35		
7426	.30	74107	.35	75491	.50	74L75	.55	74LS20	.35		
7427	.45	74121	.35	75492	.50	74L93	.55	74LS21	.25		
7430	.15	74122	.55			74L123	.55	74LS22	.25		
7432	.30	74123	.55	74H00	.25			74LS32	.40		
7437	.35	74125	.45	74H01	.25	74S00	.55	74LS37	.35		
7438	.35	74126	.35	74H04	.25	74S02	.55	74LS40	.45		
7440	.25	74132	1.35	74H05	.25	74S03	.30	74LS42	1.10		
7441	1.15	74141	1.00	74H08	.35	74S04	.35	74LS51	.50		
7442	.45	74150	.85	74H10	.35	74S05	.35	74LS74	.65		
7443	.85	74151	.75	74H11	.25	74S08	.35	74LS86	.65		
7444	.45	74153	.95	74H15	.30	74S10	.35	74LS90	.95		
7445	.65	74154	1.05	74H20	.30	74S11	.35	74LS93	.95		
7446	.95	74156	.95	74H21	.25	74S20	.35	74LS107	.85		
7447	.95	74157	.65	74H22	.40	74S40	.25	74LS123	1.00		
7448	.70	74161	.85	74H30	.25	74S50	.25	74LS151	.95		
7450	.25	74163	.95	74H40	.25	74S51	.45	74LS153	1.20		
7451	.25	74164	.60	74H50	.25	74S64	.25	74LS157	.85		
7453	.20	74165	1.50	74H51	.25	74S74	.40	74LS164	1.90		
7454	.25	74166	1.35	74H52	.15	74S112	.90	74LS367	.85		
7460	.40	74175	.80	74H53J	.25	74S114	1.30	74LS368	.85		
7470	.45			74H55	.25						
7472	.40										

LINEARS, REGULATORS, etc.

8266	.35	LM320K5 (7905)	1.65	LM340T24	.95	LM723	.50
MCT2	.95	LM320K12	1.65	LM340K12	2.15	LM725	1.75
8038	3.95	LM320T5	1.65	LM340K15	1.25	LM739	1.50
LM201	.75	LM320T12	1.65	LM340K18	1.25	LM741 (8-14)	.25
LM301	.25	LM320T15	1.65	LM340K24	.95	LM747	1.10
LM308 (Mini)	.75	LM339	.95	LM373	2.95	LM1307	1.25
LM309H	.65	7805 (340T5)	.95	LM380	.95	LM1458	.95
LM309K (340K-5)	.85	LM340T12	1.00	LM709 (8,14 PIN)	.25	LM3900	.50
LM310	1.15	LM340T15	1.00	LM711	.45	LM75451	.65
LM311D (Mini)	.75	LM340T18	1.00			NE555	.50
LM318 (Mini)	.65					NE556	.95
						NE565	.95
						NE566	1.75
						NE567	1.35

9000 SERIES

9301	.85
9309	.35
9322	.85
95H03	.55
9601	.75
9602	.50

MEMORY CLOCKS

74S188 (8223)	3.00
1702A	6.95
MM5314	3.00
MM5316	3.50
2102-1	1.75
2102L-1	1.95
TR 1602B/ TMS 6011	6.95
8080AD	15.00
8T13	1.50
8T23	1.50
8T24	2.00
2107B-4	4.95

INTEGRATED CIRCUITS UNLIMITED

7889 Clairemont Mesa Boulevard, San Diego, California 92111

(714) 278-4394 *(Calif. Res.)*

All orders shipped prepaid — No minimum

Open accounts invited — COD orders accepted

Discounts available at OEM Quantities — California Residents add 6% Sales Tax

All IC's Prime/Guaranteed. All orders shipped same day received.

24 Hour Toll Free Phone 1-800-854-2211 — MasterCharge / BankAmericard / AE

SPECIAL DISCOUNTS

Total Order	Deduct
$35 - $99	5%
$100 - $300	10%
$301 - $1000	15%
$1000 - Up	20%

Circle 59 on inquiry card.

COMPU/TIME CT 100

COMPU/TIME offers A Real Darn Clever Enhancement to users of IMSAI/ALTAIR Microprocessors

S100 BUS COMPATIBLE

TIME & CALENDAR

COMPU/TIME	CT100	$199 Kit	$245 Assembled
COMPU only	C101	$149 Kit	$189 Assembled
TIME only	T102	$165 Kit	$205 Assembled
COMPU/TIME PC Board only			$ 80

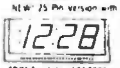
MM16 EPROM

- Utilizing up to 16 2708 EPROMS
- S-100 Bus Computer Systems
- Memory capacity of 8K or 16K bytes by DIP
- 8K boundary addressing by DIP Switch
- 0 to 4 wait cycles by DIP Switch
- Data output address input lines fully buffered
- Hi-grade glass-epoxy with plated-thru holes
- Epoxy solder masked

$99.00

FCS 8000A — 3½ Digit · .8" Display

NEW 25 Pin version with colon & am pm indicator



FRONT VIEW FSC8000

SPECIAL

$4.95 EA.

- MAXIMUM FORWARD CURRENT 25 mA
- Connects almost one for one with 3817 3817A or D 3817 available at $5.00 each
- Typical segment current 8mA except colon 10 mrs b & c and 10 mm a & d which are 16 mA
- Forward voltage drop 1.5 volts

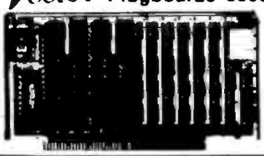
Vector

SLIT-N-WRAP WIRE WRAP TOOL

- Slits and opens insulation exposing bare wire
- No pre-cutting or pre-stripping
- Comes complete with two 100 ft spools #28 AWG wire

Model P180 $24.50

Vector Plugboards 8800V



Universal Microcomputer/Processor plugboard, use with S 100 bus complete with heat sink & hardware 5.3 x 10 x 1.16

$19.95

CHANNEL F — FAIRCHILD VIDEO ENTERTAINMENT SYSTEM

$129.95

- Freeze Action • Speed Option
- Automatic time and scorekeeping
- Battery free AC operation
- Dual controls with 8-way action
- Built-in Pro Hockey and Tennis games
- Easy hook-up on any B/W or Color TV
- Factory warranty

Channel F — additional cartridges — $17.95 ea.

0811 · Tic-Tac-Toe/Shooting Gallery Quadra-doodle Doodle
0812 · Desert Fox Shooting Gallery
0813 · Blackjack (1 or 2 players)
0814 · Spitfire (1 or 2 players)
0815 · Spacewar (2 players)
0818 · Magic Numbers (computer logic)
0819 · Drag strip (1 or 2 players)
0820 · Maze (2 players)
0822 · Baseball (2 players)

D—SUB CONNECTORS

NO. PINS	PART NO.	PRICE	COVER PRICE
9	DE-9P	1.49	1.25
9	DE-9S	2.15	
15	DA-15P	2.11	1.50
15	DA-15S	3.10	
25	DB-25P	3.00	1.50
25	DB-25S	4.00	
37	DE-37P	4.14	2.00
37	DE-37S	6.00	
50	DD-50P	5.40	2.25
50	DD-50S	8.00	

EDGE CONNECTORS

NO	PINS		TYPE	
20	DUAL	10 PIN	GOLD	$ .50
30	DUAL	15 PIN	GOLD	.75
44	DUAL	22 PIN	GOLD	1.95
44	DUAL	22 PIN	GOLD	2.50
80	DUAL	40 PIN	GOLD	4.95
86	DUAL	43 PIN	GOLD(6800)	5.00
100	DUAL	50 PIN	GOLD(IMSAI/ALTAIR)	4.25
100	DUAL	50 PIN	GOLD(IMSAI/ALTAIR)	4.95
100	DUAL	50 PIN	GOLD(NO EARS-IMSAI)	3.50
100	DUAL	50 PIN	TIN (.1" SPACING)	3.25

ALTAIR/IMSAI CARD GUIDES .25¢ EACH

CLOCK CHIPS

MM5309	6 Digit, BCD Outputs, Reset PIN	$9.95
MM5311	6 Digit, BCD Outputs, 12 or 24 Hour	4.95
MM5312	4 Digit, BCD Outputs, 1 PPS Output	4.95
MM5314	6 Digit, 12 or 24 Hour, 50 or 60 Hz	4.95
MM5316	4 Digit, Alarm, 1 PPS Output	6.95
MM5318	Video Clock Chip For Use With (MM5841 $9.95)	9.95
CT7001	6 Digit, Calendar, Alarm, 12 or 24 Hour	5.95

Wire Wrap

	1-24	25-99	100-999	1K
10	39	36	32	26
14	34	33	31	29
16	36	34	32	30
18	70	60	54	40
20	88	75	67	55
22	95	80	72	59
24	95	80	72	59
28	95	84	80	71
36	1 40	1 25	1 08	83
40	1 56	1 40	1 20	89

I C SOCKETS

Lo Profile-Solder Tin

	1-24	25-99	100-999	1K & Up
(8)	15	14	13	12
(14)	25	20	16	14
(16)	25	20	18	16
(18)	28	27	26	20
(20)	34	31	30	239
(22)	36	35	34	28
(24)	36	35	34	289
(28)	44	43	42	367
(40)	60	58	57	493

7400 TTL Series

7400	.18	7441	.75	7496	.80	74160	1.30
7401	.20	7442	.50	7497	4.00	74161	1.30
7402	.20	7443	1.20	74100	1.25	74162	1.90
7403	.20	7445	1.05	74107	.40	74163	1.40
2404	.20	7446	1.05	74109	.45	74164	1.50
7405	.25	7447	.85	74110	.80	74165	1.40
7406	.35	7448	.95	74116	2.00	74166	1.50
7407	.35	7450	.20	74120	1.25	74167	3.00
7408	.25	7451	.20	74121	.55	74170	2.00
7409	.25	7453	.20	74122	.45	74172	9.75
7410	.20	7454	.20	74123	.95	74173	1.50
7411	.25	7460	.20	74125	.55	74174	1.10
7412	.40	7470	.40	74126	.60	74175	1.20
7413	.75	7472	.35	74128	.65	74176	1.50
7416	.35	7473	.40	74132	1.50	74177	.90
7417	.40	7474	.40	74136	1.80	74180	1.00
7420	.20	7475	.70	74141	1.15	74181	2.00
7422	.75	7476	.40	74142	4.00	74182	.90
7425	.35	7479	2.00	74144	4.00	74184	2.00
7426	.30	7480	.69	74145	1.10	74185	2.00
7427	.35	7482	1.50	74147	2.50	74186	12.00
7428	.40	7483	.85	74148	1.75	74190	1.40
7429	.40	7485	1.10	74150	1.00	74191	1.25
7430	.25	7486	.40	74151	1.10	74192	1.10
7432	.30	7489	2.25	74153	1.10	74193	1.10
7433	.40	7490	.55	74154	1.10	74194	1.20
7437	.30	7491	1.10	74155	1.10	74195	1.00
7438	.35	7492	.60	74156	1.10	74196	1.10
7439	.36	7493	.60	74157	1.20	74197	1.30
7440	.20	7494	.85	74158	1.75	74198	1.50
		7495	.90	74159	3.60	74199	1.75

74LS00

74LS00	.29					74LS192	2.25
74LS01	.29					74LS193	2.25
74LS02	.29					74LS194	1.89
74LS03	.29					74LS195	1.87
74LS04	.29					74LS196	1.87
74LS05	.35	74LS60	.29	74LS138	1.25	74LS197	1.87
74LS08	.29	74LS73	.49	74LS139	1.25	74LS221	2.00
74LS10	.29	74LS74	.49	74LS151	1.25	74LS240	3.00
74LS11	.40	74LS75	.69	74LS153	1.25	74LS241	3.00
74LS12	.50	74LS76	.49	74LS154	3.65	74LS242	3.00
74LS13	.69	74LS78	.65	74LS155	1.25	74LS243	2.60
74LS14	1.70	74LS83AN	1.75	74LS156	1.85	74LS244	2.60
74LS15	.40	74LS85	2.25	74LS157	1.50	74LS247	1.68
74LS20	.29	74LS86	.69	74LS158	1.55	74LS248	1.35
74LS21	.40	74LS90	.89	74LS160	1.95	74LS251	1.91
74LS22	.40	74LS92	.89	74LS161	1.95	74LS253	1.75
74LS26	.39	74LS93	.89	74LS162	1.95	74LS257	1.75
74LS27	.39	74LS95	1.50	74LS163	1.95	74LS258	1.95
74LS28	.39	74LS96	1.89	74LS164	1.95	74LS259	4.25
74LS30	.29	74LS107	.55	74LS165	2.00	74LS260	.50
74LS32	.39	74LS109	.59	74LS166	2.00	74LS266	.68
74LS37	.48	74LS112	.59	74LS168	2.00	74LS273	2.50
74LS38	.48	74LS113	.60	74LS169	2.00	74LS279	.79
74LS40	.39	74LS114	.60	74LS170	4.00	74LS283	1.85
74LS42	1.20	74LS122	.60	74LS173	2.00	74LS289	6.16
74LS47	1.25	74LS123	1.25	74LS174	1.87	74LS290	1.56
74LS48	1.68	74LS124	1.80	74LS175	1.95	74LS293	1.56
74LS51	.29	74LS125	.87	74LS181	3.69	74LS295	2.00
74LS52	.29	74LS126	1.07	74LS189	6.16	74LS298	2.00
74LS54	.29	74LS132	1.25	74LS190	2.49	74LS352	1.65
74LS55	.29	74LS136	.59	74LS191	2.49	74LS367	.87
						74LS390	3.00
						74LS670	3.95

CMOS

34001	.40	4050	.61	4517	8.50		
4000	.25	4051	1.10	4518	1.65		
4001	.25	4052	1.10	4519	.90	MM74C173N	1.39
4002	.25	4053	1.10	4520	1.65	MM74C174N	1.39
4004	3.50	4060	3.25	4521	3.25	MM74C175N	1.39
4006	1.40	4061	7.00	4522	1.75	MM74C192N	1.71
4007	.25	4063	2.50	4527	3.00	MM74C193N	1.71
4008	1.25	4066	.85	4528	1.75	MM74C195N	1.61
4009	.48	4067	6.00	4583	1.45	MM74C200N	10.45
4010	.48	4068	.35	4584	.75	MM74C221N	2.13
4011	.25	4069	.35			MM74C901N	.84
4012	.25	4070	.85			MM74C902N	.84
4013	.60	4071	.35			MM74C903N	.84
4014	1.25	4072	.35	MM74C00N	.38	MM74C905N	11.20
4015	1.25	4073	.35	MM74C02N	.38		
4016	.59	4075	.35	MM74C04N	.38	MM74C906N	.84
4017	1.25	4076	1.85	MM74C08N	.38	MM74C907N	.84
4018	1.25	4077	.42	MM74C10N	.38	MM74C908N	3.80
4019	.70	4078	.35	MM74C14N	2.18	MM74C909N	2.52
4020	1.25	4081	.35	MM74C20N	.38	MM74C910N	10.45
4021	1.25	4082	.35	MM74C30N	.38	MM74C914N	2.18
4022	1.25	4085	1.35	MM74C32N	.38		
4023	.35	4086	1.45	MM74C42N	1.42	MM74C915N	1.71
4024	1.00	4089	3.00			MM74C918N	4.19
4025	.35	4093	1.75	MM74C48N	2.13	MM74C922N	5.65
4026	2.25	4098	2.50	MM74C73N	.84	MM74C923N	5.75
4027	.60	4160	1.75	MM74C74N	.87	MM74C925N	12.00
4028	1.25	4161	1.75	MM74C76N	.84		
4029	1.50	4162	1.75	MM74C83N	2.00	MM74C926N	12.00
4030	.60	4163	1.75	MM74C85N	2.00		
4032	1.60	4174	1.75	MM74C86N	.99	MM74C927N	12.00
4033	2.00	4175	1.60	MM74C89N	6.75		
4034	3.50	4194	1.80	MM74C90N	1.32	MM74C928N	12.00
4035	1.60	4501	.38	MM74C93N	1.32		
4038	1.60	4502	1.75	MM74C95N	1.61		
4040	1.50	4503	1.15	MM74C107N	1.89	MM80C97N	.84
4041	1.45	4506	.70	MM74C150N	5.67	MM80C98N	.84
4042	1.25	4507	1.00	MM74C151N	3.80		
4043	1.20	4508	4.00	MM74C154N	5.67		
4044	1.25	4510	1.75	MM74C157N	3.40		
4045	2.50	4511	2.00	MM74C160N	1.71		
4046	1.95	4512	1.50	MM74C161N	1.71		
4047	2.50	4514	4.75	MM74C162N	2.57		
4048	1.00	4515	4.75	MM74C163N	1.71		
4049	.61	4516	1.75	MM74C164N	1.61		
				MM74C165N	1.61		

LINEAR

LM300H	1.70	LM702H	.79	LM2111N	1.35
LM301AH	.35	LM703LN	.45	LM2900N	.90
LM301AN	.35	LM703CH	.45	LM2901N	2.95
LM302H	.75	LM709CH	.40	LM2902N	3.00
LM306H	2.50	LM710CH	.60	LM2903N	1.60
LM307H	.55	LM710N	.75	LM2904N	1.60
LM307N	.35	LM711CH	.39	LM2917N	2.00
LM308AH	3.75	LM711CN	.39	LM3046N	1.25
LM308AN	3.00	LM723CH	.55	LM3053N	1.50
LM308H	1.00	LM723CN	.55	LM3064N	1.25
LM308N	1.00	LM725CH	2.50	LM3065N	1.00
LM310H	1.15	LM725CN	3.00	LM3067N	2.60
LM310N	2.00	LM733CH	1.00	LM3075N	1.75
LM311D	2.00	LM733CN	1.00	LM3089N	3.00
LM311H	.90	LM739N	1.19	LM3146N	2.00
LM311N	.80	LM741CH	.35	LM3302N	2.00
LM311N	.80	LM741CN	.35	LM3401N	.80
LM312H	2.70	LM741CJ	.39	LM3900N	.49
LM318H	1.50	LM741CN	.39	LM3905N	.89
LM318N	1.50	LM747CH	.79	LM3909N	.90
LM319N	1.25	LM747CN	.79	LM3911N	1.25
LM321H	3.00	LM748CH	.39	LM4024N	2.75
LM322N	3.00	LM748CN	.39	LM4044N	3.00
LM324N	1.65	LM760CN	3.00	LM4250CH	2.00
LM339AN	3.00	LM1303N	.90	LM4250CN	2.00
LM339N	.99	LM1304N	1.19	LM4558N	.75
LM348N	1.85	LM1305N	1.40	LM5556N	1.75
LM350N	1.00	LM1307N	.85	LM6558N	1.00
LM358N	1.00	LM1310N	2.75		
LM370H	1.15	LM1358N	1.00		
LM370N	2.75	LM1414N	1.75		
LM373N	2.95	LM1458H	1.30		
LM375N	3.00	LM1458N	.69		
LM377N	2.00	LM1468N	2.90		
LM380N	1.05	LM1488N	1.95		
LM380N	1.05	LM1489N	1.95		
LM380N	1.05	LM1496H	.95		
LM381N	1.75	LM1496N	.95		
LM382N	1.75	LM1556N	1.75		
LM386H	.85	LM1596J	2.75		
LM387AN	.75	LM1800N	1.90		
LM387N	.95	LM1808N	2.00		
LM388N	1.25	LM1810N	2.00		
LM389N	1.25	LM1812N	4.00		
NE531T/V	2.00	LM1820N	1.75		
NE550A	1.30	LM1830N	2.00		
NE555V	.39	LM1845N	1.75		
NE556B	1.25				
NE560B	5.00				
NE562B	5.00				
NE565B	1.50				
NE566B	2.00				
NE566T/V	1.75				
NE567V	1.59				

RCA LINEAR SERIES

CA3013	2.15
CA3023	2.56
CA3035	2.48
CA3039	1.35
CA3046	1.30
CA3059	3.25
CA3060	3.25
CA3080	.85
CA3081	2.00
CA3082	2.00
CA3083	1.60
CA3086	.85
CA3089	3.75
CA3130	1.39
CA3140	1.25
CA3401	.49
CA3600	3.50

MICROPROCESSOR CRYSTALS

FREQUENCY	price
1.0 MHz	5.95
1.8432 MHz	9.80
2.0 MHz	5.95
2.097152 MHz	8.50
2.4576 MHz	8.50
2.667 MHz	8.50
3.00 MHz	8.50
3.20 MHz	8.50
3.2768 MHz	8.50
3.579545 MHz	1.25
4.0 MHz	4.95
4.0 MHz	4.95
4.194304 MHz	4.95
4.91520 MHz	4.95

FREQUENCY	price
5.0 MHz	4.95
5.0688 MHz	4.95
5.185 MHz	4.95
5.7143 MHz	4.95
6.00 MHz	4.95
6.144 MHz	4.95
6.40 MHz	4.95
6.5536 MHz	4.95
8.0 MHz	4.95
10.0 MHz	4.95
18.00 MHz	4.95
18.432 MHz	4.95
20.0 MHz	4.95
22.1184 MHz	4.95
27.0 MHz	4.95
36.0 MHz	4.95
48.0 MHz	4.95

FOR PROCESSOR TECH

14.318 $4.95

DIP SWITCHES

		7 POSITION	1.80
4 POSITION	$1.50	8 POSITION	2.00
5 POSITION	1.60	9 POSITION	2.25
6 POSITION	1.70	10 POSITION	2.50

WIRE WRAP CENTER

HOBBY-WRAP TOOL-BW-630

$34.95

batteries not included

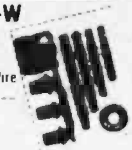
- Battery Operated (Size C)
- Weighs ONLY 11 Ounces
- Wraps 30 AWG Wire onto Standard DIP Sockets (.025 inch)
- Complete with built-in bit and sleeve

WIRE-WRAP KIT — WK-2-W

WRAP • STRIP • UNWRAP

- Tool for 30 AWG Wire
- Roll of 50 Ft White or Blue 30 AWG Wire
- 50 pcs each 1", 2", 3" & 4" lengths — pre-stripped wire.

$11.95



WIRE WRAP TOOL WSU-30

WRAP • STRIP • UNWRAP - $5.95

WIRE WRAP WIRE — 30 AWG

50 ft $1.95 1000 ft $15.00

SPECIFY COLOR — White - Yellow - Red - Green - Blue - Black

WIRE DISPENSER — WD-30

- 50 ft roll 30 AWG KYNAR wire wrap wire $3.45 ea.
- Cuts wire to desired length
- Strips 1" of insulation **Specify** — Blue-Yellow-White-Red

LIQUID CRYSTAL DIGITAL CLOCK-CALENDAR

For Auto, Home, Office
Small in size (2x2½x½)
Push button for seconds release for date.
Clocks mount anywhere with either 3M double-sided tape or VELCRO included
2 MODELS AVAILABLE
LCD-101 portable model runs on self-contained batteries for better than a year
LCD-102 runs on 12 Volt system and is back-lighted

LCD-101 or LCD-102 your choice. .... $33.95 ea.

Clear desk stand for $2.00

MA1003, 12V DC CLOCK MODULE

$17.95

Built in X'TAL controlled time base. Protected against automotive volt transients. Automatic brightness control with 0.3" green color display. Display turnoff with ignition "OFF"

JADE

Computer Products

5351 WEST 144th STREET
LAWNDALE, CALIFORNIA 90260
(213) 679-3313

RETAIL STORE HOURS M-F 9-7 SAT 9-5

Discounts available at OEM quantities. Add $1.25 for shipping. California residents add 6% sales tax.

1978 CATALOG NOW AVAILABLE

FOR ALL CUSTOMERS EXCEPT CALIF. CALL TOLL FREE 800-421-5809

MICROCOMPUTER

MICROPROCESSOR's

F-8	19.95
Z-80	25.00
Z-80A	35.00
CDP1802CD	24.95
AM2901	22.95
6502	12.95
6800	19.95
8008-1	8.75
8080A	15.95
TMS-9900TL	89.95

6800 SUPPORT

6810P	4.95
6820P	8.00
6828P	11.25
6834P	16.95
6850P	9.95
6852P	11.95
6860P	14.95
6862P	17.95
6880P	2.70

Z80 SUPPORT DEVICES

3881	12.95
3882	12.95

F-8 SUPPORT DEVICES

3851	14.95
3853	14.95

8080A SUPPORT DEVICES

8212	3.95
8214	9.95
8216	4.50
8224	4.95
8228	8.75
8238	8.00
8251	12.00
8253	28.00
8255	12.00
8257	25.00
8259	25.00

CHARACTER GENERATORS

2513 UP	6.75
2513 DOWN	6.75
2513 UP (5v)	9.95
2513 DOWN(5v)	10.95
MCM6571	10.80
MCM6571A	10.80
MCM6572	10.80
MCM6574	14.75
MCM6575	14.75

STATIC RAMS		1-24	25-99	100
21L02	(450)	1.50	1.40	1.25
21L02	(250)	1.95	1.80	1.50
21L11		4.25	4.10	3.95
1101A		1.49	1.29	1.10
2101-1		2.95	2.75	2.60
2102		1.25	1.15	1.00
2102-1		1.50	1.30	1.15
2111-1		4.00	3.50	3.25
2112-1		3.00	2.80	2.69
2114		17.95	16.95	16.50
4200A		12.95	12.50	11.95
5101C-E		11.95	11.25	10.25

WAVEFORM GENERATOR

8038	4.00
MC4024	2.50
566	1.75

KEYBOARD CHIPS

AY5-2376	13.95
AY5-3600	13.95

PROM'S

1702A	5.00
2704	15.00
2708	13.00
2716	38.00
3601	4.50
5203AQ	4.00
5204AQ	6.00
6834	16.95
6834-1	14.95
82S23B	4.00
8223B	2.70

DYNAMIC RAMS

1103	1.50
2104	4.50
2107A	3.75
2107B	4.50
2107B-4	4.00
TMS4050	4.50
TMS4060	4.50
4096	4.50
4116	42.00
MM5270	5.00
MM5280	6.00
MCM6605	6.00

UART'S

AY5-1013A	5.50
AY5-1014A	8.95
TR-1602A	5.50
TMS-6011	6.95
IM-6402	10.80
IM-6403	10.80

MISC. OTHER COMPONENTS

NH0025CN	1.70
NH0026CN	2.50
N8T20	3.50
N8T26	2.45
74367	.90
DM8098	.90
1488	1.95
1489	1.95
D-3207A	2.00
C-3404	3.95
P-3408A	5.00
P-4201	4.95
MM-5320	7.50
MM-5369	1.90
DM-8130	2.90
DM8131	2.75
DM-8831	2.50
DM-8833	2.50
DM-8835	2.50
SN74LS367	.90
SN74LS368	.90

KIM

KIM-1	245.00
6502	12.95
6520	9.00
6522	9.25
6530-002	15.95
6530-003	15.95
6530-004	15.95
6530-005	15.95

USRT

S-2350	10.95

LATE ADDITIONS

TMS 4044	$14.00
TMS 5501	$24.95

FLOPPY DISC CONTROLLER

1771B	55.95
1771B-01	59.95

8K STATIC RAM BOARD

ASSEMBLED & TESTED

250ns. $199.95
450ns. $150.00

* WILL WORK WITH NO FRONT PANEL
* FULL DOCUMENTATION
* FULLY BUFFERED
* S100 DESIGN
* ADEQUATELY BYPASSED
* LOW POWER SCHOTTKY SUPPORT IC'S

KIT

250ns. $169.95
450ns. $129.95

BARE BOARD W/SCHEMATIC $25.00

ADAPT YOUR MOTOROLA 6800 SYSTEM TO OUR S-100 8K RAM BOARD. KIT PRICE $12.95

JADE VIDEO INTERFACE KIT

FEATURES $89.95

S-100 Bus Compatible
32 or 64 Characters per line
16 lines
Graphics (128 x 48 matrix)
Parallel & Compositive video
On board low-power memory

Powerful software included for cursor, home, EOL, Scroll Graphics/Character, etc.
Upper case, lower case & Greek
Black-on-white & white-on-black

JADE PARALLEL/SERIAL INTERFACE KIT

S-100 $124.95 KIT

2 Serial Interfaces with RS232 interfaces or 1 Kansas City cassette interface.
Serial interfaces are crystal controlled.
Selectable baud rates.
Cassette works up to 1200 baud.
1 parallel port.

EDGE CONNECTORS

S-100 Altair Spacing $4.45
S-100 Imsai Spacing $4.45

IMSAI/ALTAIR S-100 COMPATIBLE

JADE Z80 KIT

—with PROVISIONS for ONBOARD 2708 and POWER ON JUMP

$135.00 EA. (2MHZ)
$149.95 EA. (4MHZ)

BARE BOARD $35.00

MOTHER BOARD

* 13 SLOT MOTHER BOARD w/front panel slot
* S-100 DESIGN
* FULL GROUND PLANE ON ONE SIDE
* RC NETWORK TERMINATION ON EVERY LINE EXCEPT PWR & GRD
* KLUGE AREA
* STRONG 1/8" THICK DOUBLE SIDED BOARD

BARE BOARD $35.00 KIT $85.00

THE PROM SETTER

WRITE and READ EPROM

1702A-2708-2716
5204-6834

* Plugs Directly into your ALTAIR/IMSAI Computer
* Includes Main Module Board and External EPROM Socket Unit
* The EPROM Socket Unit is connected to the Computer through a 25 Pin Connector
* Programming is accomplished by the Computer
* Just Read in the Program to be Written on the EPROM into your Processor and let the Computer do the rest
* Use Socket Unit to Read EPROM's Contents into your Computer
* Software included
* No External Power Supplies. Your Computer does it all
* Doubles as an Eight Bit Parallel I/O
* Manual included Delivery Less than 60 days

THE PROM SETTER
KIT $210.00 ASSEMBLED $375.00

JADE 8080A KIT

$100.00 KIT

BARE BOARD $35.00

PERSCI DISK DRIVE FOR S-100

Info 2000 S-100 DISK SYSTEM IMP COMPLETE
Info 2000 S-100 DISK SYSTEM
(includes dual drive, power supply, case, intelligent controller, adapter, cables, and disk monitor on EPROM) $2,650.00

COMPLETE TDL SOFTWARE PKG. FOR DISK $195.00

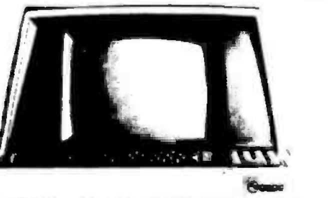
REAL TIME CLOCK FOR S-100 BUS

BARE BOARD $30.00 KIT $124.95

TV-1 VIDEO INTERFACE

You will want to know about the TV-1 Video to Televisior Interface Kit.
No need to buy a separate Video Monitor if you already own a TV set. Just connect the TV-1 between your system video output and the TV set antenna terminals - that's all there is to it - to convert your TV set to a Video Monitor, and at a much lower cost! PRICE $8.95

SOROC IQ 120 TERMINAL



A CAPABLE LOW COST. APPROACH TO REMOTE VIDEO DISPLAY TERMINALS.

* Cursor Control Keys Standard
* Numeric Key Pad Standard
* Line and Page Erase Standard
* Addressable Cursor Standard
* Switch Selectable Transmission from 75 to 19,200 bps Standard
* Communication Mode. HDX/FDX Block
* Interfaces: Printer Interface. RS232 Extension
* RS232C Interface
* Non-Glare Read Out Screen
* Protect Mode Standard
* Tab Standard

$975 00
Assembled

Price Includes
* Block Mode
* Lower Case
* 24 Line Option
* and shipping charge is on us.

JADE Computer Products

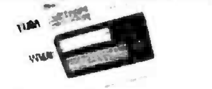
5351 WEST 144th STREET
LAWNDALE, CALIFORNIA 90260
(213) 679-3313

RETAIL STORE HOURS M-F 9-7 SAT. 9-5

Discounts available at OEM quantities. Add $1.25 for shipping. California residents add 6% sales tax

Circle 62 on inquiry card.

BREAD BOARD JUMPER WIRE KIT



Each kit contains 350 wires cut to 14 different lengths from 0.1" to 5.0".
Each wire is stripped and the leads are bent 90° for easy insertion.
Wire length is classified by color coding.
All wire is solid tinned 22 gauge with PVC insulation.
The wires come packed in a convenient plastic box.

JK1 . . . . $10.00 / kit

SOCKET JUMPERS

Mates with two rows of .025" sq. or dia. posts on patterns of .100" centers and shielded receptacles. Probe access holes in back. Choice of 6" or 18" length.

Part No.	No. of Contacts	Length	Price
924003-18R	26	18"	$ 5.38 ea.
924003-06R	26	6"	4.78 ea.
924005-18R	40	18"	8.27 ea.
924005-06R	40	6"	7.33 ea.
924006-18R	50	18"	10.31 ea.
924006-06R	50	6"	9.15 ea.

JUMPER HEADERS

Solder to PC boards for instant plug-in access via socket-connector jumpers. .025" sq. posts. Choice of straight or right angle.

Part No.	No. of Posts	Angle	Price
923863-R	26	straight	$1.28 ea.
923873-R	26	right angle	1.52 ea.
923865-R	40	straight	1.94 ea.
923875-R	40	right angle	2.30 ea.
923866-R	50	straight	2.36 ea.
923876-R	50	right angle	2.82 ea.

INTRA-CONNECTOR

Provides both straight and right angle functions. Mates with standard .10" x .10" dual row connectors (i.e. 3m, Ainsley, etc.) Permits quick testing of inaccessible lines.

Part No.: 922576-26 No. of contacts: 26 Price $6.90 ea.

INTRA-SWITCH

Permits instant line-by-line switching for diagnostic or QA testing. Switches actuated with pencil or probe tip. Mates with standard .10" x .10" dual-row connectors. Low profile design. Switch buttons recessed to eliminate accidental switching.

Part No.: IS-26 No. of contacts: 26 Price $13.80 ea.

CRYSTALS

THESE FREQUENCIES ONLY

Part #	Frequency	Case/Style	Price
CY1A	1.000 MHz	HC33/U	$5.95
CY2A	2.000 MHz	HC33/U	$5.95
CY2.01	2.010 MHz	HC33/U	$1.95
CY3A	4.000 MHz	HC18/U	$4.95
CY7A	5.000 MHz	HC18/U	$4.95
CY12A	10.000 MHz	HC18/U	$4.95
CY14A	14.31818 MHz	HC18U	$4.95
CY19A	18.000 MHz	HC18U	$4.95
CY22A	20.000 MHz	HC18/U	$4.95
CY30B	32.000 MHz	HC18/U	$4.95

CONNECTORS

PRINTED CIRCUIT EDGE-CARD

.156 Spacing-Tin-Double Read-Out

Bifurcated Contacts — Fits .054 to .070 P.C. Cards

15/30	PINS (Solder Eyelet)	$1.95
18/36	PINS (Solder Eyelet)	$2.49
22/44	PINS (Solder Eyelet)	$2.95
50/100	PINS (Wire Wrap)	$6.95
50/100A (.100 Spacing)	PINS (Wire Wrap)	$6.95

25 PIN-D SUBMINIATURE (RS232)

DB25P	PLUG	$3.25
DB25S	SOCKET	$4.95

HEAT SINKS

680-.75A 205-CB

205-CB	Beryllium Copper w-black finish for TO-5	$ .25
291-.36H	Aluminum for TO-220 Transistors & Regulators	$ .25
680-.75A	Black Anodized Aluminum for TO-3	$1.60
Dude 4	Black Anodized Aluminum — predrilled mounting holes for TO-3 — 4¾ x 1¼ x 2"	$1.75

Etching Kits (cannot be shipped via air)

32 X A-1	P.C. Etch Materials Kit enough for 5 circuit boards	$29.95 ea.
27 X A-1	Etched Circuit Kit Complete kit — only add water	$ 9.95 ea.

Plugboards

3662	6.5 X 4.5 X 1/16 Epoxy glass P-Pattern-44 P.C. Tabs-spaced .156"	$ 6.95 ea.
22/44	Mating connector for plugboard — 22 pin double readout	$ 2.95 ea.
8800V	Universal Microcomputer/Processor plugboard — Epoxy Glass — complete with heatsink and mounting hardware 5.313 X 10 X 1/16 copper clad	$19.95 ea.

Vector

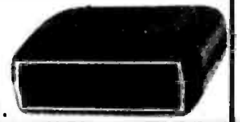
1/16 VECTOR BOARD

0.1 Hole Spacing P-Pattern

	Part No.	L	W	Price 1-9	10 up
PHENOLIC	64P44 062XXXP	4.50	6.50	1.72	1.54
	169P44 062XXXP	4.50	17.00	3.69	3.32
EPOXY GLASS	64P44 062WE	4.50	6.50	2.07	1.86
	84P44 062WE	4.50	8.50	2.56	2.31
	169P44 062WE	4.50	17.00	5.04	4.53
	169P84 062WE	8.50	17.00	9.23	8.26
EPOXY GLASS COPPER CLAD	169P44 062WEC1	4.50	17.00	6.80	6.12

INSTRUMENT/ CLOCK CASE

Injection molded unit. Complete with red bezel. 4½" x 4" x 1-9/16".



$3.95 ea.

MICROPROCESSOR COMPONENTS

8080A	CPU	$16.00	CDP1802	CPU	$19.95
8212	8 Bit Input/Output	4.95	MC6800	8 Bit MPU	24.95
8214	Priority Interrupt Control	15.95	MC6820	Periph. Interface Adapter	15.00
8216	Bi-Directional Bus Driver	6.95	MC6810AP1	128 x 8 Static RAM	6.00
8224	Clock Generator/Driver	9.95	MC6830L8	1024 x 8 Bit ROM	15.00
8228	System Controller Bus Driver	10.95	Z80	CPU	29.95

CPU'S

8080A	Super 8008	16.00
2650	8 BIT MPU	26.50
P8085	CPU	29.95

SR'S

2504	1024 Dynamic	$ 3.95
2518	Hex 32 BIT	7.00
2519	Hex 40 BIT	4.00
2522	Dual 132 Bit SSR	2.95
2524	512 Dynamic	1.95
2525	1024 Dynamic	3.00
2527	Dual 256 BIT	3.95
2529	Dual 512 BIT	4.00
2532	Quad 80 BIT	3.95
2533	1024 Static	5.95
3341	Fifo	6.95
74LS670	16 x 4 Reg	3.95

UART'S

AY-5-1013	30K Baud	$5.95

ROM'S

2513(2140)	Char. Gen.-upper case	$ 9.95
2513(3021)	Char. Gen.-lower case	9.95
2516	Char. Gen.	10.95
MM5230	2048 BIT (512 x 4 on 256 x 8)	1.95

RAM'S

1101	256 x 1	Static	$ 1.49
2101	256 x 4	Static	5.95
2102	1024 x 1	Static	1.75
2107/5280	4096 x 1	Dynamic	4.95
2111	256 x 4	Static	6.95
7489	16 x 4	Static	2.49
8101	256 x 4	Static	5.95
8111	256 x 4	Static	6.95
8599	16 x 4	Static	3.49
21L02/91L02	1024 x 1	Static	2.25
74200	256 x 1	Static	6.95
93421	256 x 1	Static	2.95
MM5262	2K x 1	Dynamic	2 for 1.00
UPD414(2104)	4K	Dynamic 16 Pin	5.95

PROMS

1702A	2048	Famos	$ 9.95
5203	2048	Famos	14.95
82S23	32 x 8	Open C	5.00
82S123	32 x 8	Tristate	5.00
74S287	1024	Static	7.95
3601	256 x 4	Fast	3.95
2708	8K	Eprom	19.95
2716	16K	Eprom	34.95
6301-1	1024	Tri-State Bipolar	3.49
6330-1	256	Open Collector Bipolar	2.95

SPECIAL REQUESTED ITEMS

FCM3817	$5.00	11C90	19.95	7205	19.95	9368	3.95
AY-3-8500-1	8.95	4N33	3.95	ICM7045	24.95	LD110/111	25.00/set
AY-5-9100	17.50	8T20	7.50	ICM7207	7.50	95H90	11.95
AY-5-9200	14.95	8T97	2.00	ICM7208	22.00	MC3061P	3.50
AY-5-9500	4.95	HD0165	7.95	ICM7209	7.50	MC4016 (74416)	7.50
AY-5-2376	14.95	MCM6571	13.50	MK50240	17.50	MC1408L7	8.95
9374	1.95	MCM6574	13.50	DS0026CH	3.75	MC1408L8	9.95
82S115	25.00	MCM6575	13.50	TIL308	10.50		

CLOCK CHIPS

MM5309	$9.95
MM5311	4.95
MM5312	4.95
MM5314	4.95
MM5316	6.95
MM5318	9.95
MM5369	2.95
MM5841	9.95
CT7001	5.95

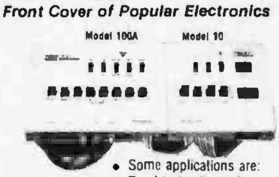
PARATRONICS

Featured on February's Front Cover of Popular Electronics

Logic Analyzer Kit Model 100A

$229.00/kit

Model 100A assembled $295.00



- Analyzes any type of digital system
- Checks data rates in excess of 8 million words per second
- Trouble shoot TTL, CMOS, DTL, RTL, Schottky and MOS families
- Displays 16 logic states up to 8 digits wide
- See ones and zeros displayed on your CRT, octal or hexadecimal format
- Tests circuits under actual operating conditions
- Easy to assemble — comes with step-by-step construction manual which includes 80 pages on logic analyzer operation.

(Model 100A Manual - $4.95)

- Some applications are:
  - — Troubleshooting microprocessor address, instruction, and data flow
  - — Examine contents of ROMS
  - — Tracing operation of control logic
  - — Checking counter and shift register operation
  - — Monitoring I/O sequences
  - — Verifying proper system operations during testing

PARATRONICS TRIGGER EXPANDER - Model 10

Adds 16 additional bits. Provides digital delay and qualification of input clock and 24-bit trigger word. — Connects direct to Model 100A for integrated unit).

Model 10 Kit — $229.00
Model 10 Assembled — $295.00
Baseplate — $9.95
Model 10 Manual — $4.95

B&K PRECISION 3½-Digit Portable DMM

- Overload Protected
- .3" high LED Display
- Battery or AC operation
- Auto Zeroing
- 1mv, 1Va, 0.1 ohm resolution
- Overange reading
- 10 meg input impendence
- DC Accuracy 1% typical

Ranges: DC Voltage ±0-1000V/
AC Voltage: 0-1000V
Freq. Response: 50-400 HZ
DC/AC Current: 0-100mA
Resistance, 0-10 meg ohm
Size: 6.4" x 4.4" x 2"

Model 2800 $99.95

Comes with test leads, operating manual and spare fuse.

Accessories:

AC Adapter BC-28	$9.00
Rechargeable Batteries BP-26	20.00
Carrying Case LC-28	7.50

CSC 100 MHz 8-Digit Counter

- 20 Hz-100 MHz Range
- .6" LED Display
- Crystal-controlled timebase
- fully Automatic
- Portable — completely self-contained
- Size — 1.75" x 7.38" x 5.63"
- Four power souces, i.e. batteries, 110 or 220V with charger 12V with auto lighter adapter and external 7.2-10V power supply

MAX-100 $134.95

ACCESSORIES FOR MAX 100:

Mobile Charger Eliminator use power from car battery — Model 100 — CLA $3.95
Charger/Eliminator use 110 V AC — Model 100 — CAI $9.95

CONTINENTAL SPECIALTIES

PROTO BOARD 6

$15.95
(6" long X 4" wide)

Other CS Proto Boards

PB100 - 4.5" x 6"	$ 19.95
PB101 - 5.8" x 4.5"	29.95
PB102 - 7" x 4.5"	39.95
PB103 - 9" x 6"	59.95
PB104 - 9.5" x 8"	79.95
PB203 - 9.75 x 6½ x 2¾	80.00
PB203A - 9.75 x 6½ x 2¾ (includes power supply)	129.95

PROTO CLIPS

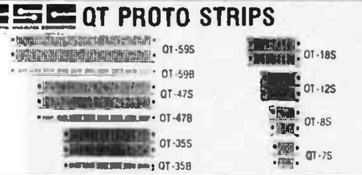
14 PIN	$4.50
16 PIN	4.75
24 PIN	8.50
40 PIN	13.75

DESIGN MATES

DM1 - Circuit Designer $69.95
DM2 - Function Generator $74.95
DM3 - RC Bridge $74.95

LOGIC MONITOR for DTL, HTL, TTL or CMOS Devices **$84.95**

CSC QT PROTO STRIPS



QT type	# holes	price
QT-59S	590	12.50
QT-59B	bus strip	2.50
QT-47S	470	10.00
QT-47B	bus strip	2.25
QT-35S	350	8.50
QT-35B	bus strip	2.00
QT-18S	180	4.75
QT-12S	120	3.75
QT-8S	80	3.25
QT-7S	70	3.00

Experimentor 300	$ 9.95
Experimentor 600	$10.95

$5.00 Minimum Order — U.S. Funds Only
California Residents — Add 6% Sales Tax

Spec Sheets - 25¢ — Send 35¢ Stamp for 1978 Catalog
Dealer Information Available

NOW OPEN SATURDAYS

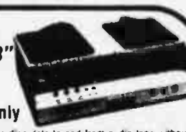
James ELECTRONICS

1978 CATALOG NOW AVAILABLE

1021-A HOWARD AVE., SAN CARLOS, CA. 94070
PHONE ORDERS WELCOME — (415) 592-8097

Advertised Prices Good Thru February

The Incredible "Pennywhistle 103"



$129.95 Kit Only

The Pennywhistle 103 is capable of recording data to and from audio tape without critical speed requirements for the recorder and it is able to communicate directly with another modem and terminal for telephone "hamming" and communications for the deaf. In addition, it is free of critical adjustments and is built with non-precision, readily available parts.

Data Transmission Method	Frequency-Shift Keying, full-duplex (half-duplex selectable).
Maximum Data Rate	300 Baud.
Data Format	Asynchronous Serial (return to mark level required between each character).
Receive Channel Frequencies	2025 Hz for space; 2225 Hz for mark.
Transmit Channel Frequencies	Switch selectable: Low (normal) = 1070 space, 1270 mark; High = 025 space, 2225 mark.
Receive Sensitivity	-46 dbm acoustically coupled.
Transmit Level	-15 dbm nominal. Adjustable from -6 dbm to -20 dbm.
Receive Frequency Tolerance	Frequency reference automatically adjusts to allow for operation between 1800 Hz and 2400 Hz.
Digital Data Interface	EIA RS-232C or 20 mA current loop (receiver is optoisolated and non-polar).
Power Requirements	120 VAC, single phase, 10 Watts.
Physical	All components mount on a single 5" by 9" printed circuit board. All components included.

Requires a VOM, Audio Oscillator, Frequency Counter and/or Oscilloscope to align.

NEW! BULB-ENERGY SAVER

Bulb lasts 3 or more times longer. Fits Standard Socket 6 watts to 200 watts

BES-1 1-9 $1.50 ea. 10+ 1.25 ea.

BULB-ENERGY SAVERS used for years by major industrial users — now available for home or office use. Bulb Savers can cut electrical bills by as much as 3%.

BULB-SAVERS lengthens light life by:

1. Acting as an electrical "shock absorber", turns the bulb on slowly, eliminating the "thermal shock". Bulb life increases 300 percent.
2. Banishes Current "Surges". Cushions line voltage surges when other loads cut power line.
3. Reduces Energy Consumption

DIGITAL STOPWATCH

- Bright 6 Digit LED Display
- Times to 59 minutes 59.59 seconds
- Crystal Controlled Time Base
- Three Stopwatches in One Times Single Event — Split & Taylor
- Size 4.5" x 2.15" x .90" (4½ ounces)
- Uses 3 Penlite Cells.

Kit — $39.95
Assembled — $49.95
Heavy Duty Carry Case $5.95

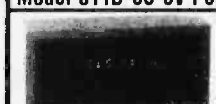
Stop Watch Chip Only (7205) $19.95

IMC 3½ DIGIT DPM KIT

- New Bipolar Unit
- Auto Zeroing
- .5" LED
- Auto Polarity
- Low Power
- Single IC Unit

Model KB500 DPM Kit $49.00
Model 311D-5C-5V Power Kit $17.50

JE700 CLOCK



The JE700 is a low cost digital clock, but is a very high quality unit. The unit features a simulated walnut case with dimensions of 6" x 2½" x 1". It utilizes a MAN72 high brightness readout, and the MM5314 clock chip.

12 or 24 Hour

115 VAC *KIT ONLY* **$16.95**

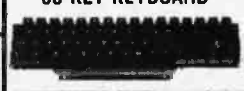
HEXADECIMAL ENCODER 19-KEY PAD

- 1 - 0
- ABCDEF
- Shift Key
- 2 Optional Keys

$10.95 each

63 KEY KEYBOARD

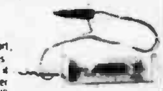
$29.95



This keyboard features 63 unencoded SPST keys, unattached to any kind of P.C.B. A very solid molded plastic 13" x 4" base suits most applications.

HD0165	Encoder Chip (encodes 16 Keys)	$7.95 ea.
AY-5-2376	Encoder Chip (encodes 88 Keys)	$14.95 ea.

JE803 PROBE



The Logic Probe is a unit which is for the most part, indespensible in trouble shooting logic families TTL, DTL, RTL, CMOS. It derives the power it needs to operate directly off of the circuit under test, drawing a scant 10 mA max. It uses a MAN3 readout to indicate any of the following states by these symbols: (H) - 1 (LOW) - 0 (PULSE) - P. The Probe can detect high frequency pulses to 45 MHz. It can't be used at MOS levels or circuit damage will result.

$9.95 Per Kit

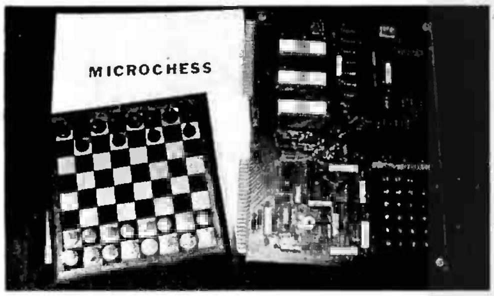
printed circuit board

T²L 5V 1A Supply

This is a standard TTL power supply using the well known LM309K regulator IC to provide a solid 1 AMP of current at 5 volts. We try to make things easy for you by providing everything you need in one package, including the hardware for only.

$9.95 Per Kit

Chess Program for SOL and KIM-1



This remarkable chess program runs in about 1 K bytes of memory on the KIM-1 microcomputer, yet plays an acceptable level of chess. No extra memory or peripherals are needed for the KIM-1 version; the built in hexadecimal keyboard and display are used to enter the player's moves and display the computer's replies. Another version of the program is tailored to the SOL computer, and enough documentation is provided to enable the user to adapt the appropriate version to other 6502 and 8080 based computers. The computer can be set up to play white or black, and can also play against itself. The program will follow a book opening which can be changed by the user, and data is provided for the French Defense, Giuoco Piano, Ruy Lopez, Queen's Indian and Four Knights openings. The level of the computer's play can be adjusted so that moves on the KIM-1 take 3 seconds ("super blitz"), 10 seconds ("blitz") or 100 seconds ("normal"). Because of size constraints the program doesn't handle castling, en passant captures and queening of pawns, but provision has been made for the user to manually execute these moves for either side. The program documentation includes a complete player's manual, a programmer's manual with a discussion of the playing strategy and method of analysis, basic flowcharts and state variable definitions, instructions for modifying the input and output routines, and suggestions for implementing strategy improvements. A commented assembly source code listing with symbol table and cross references is included, as well as a hexadecimal object code dump. The MICROCHESS program is available on a KIM-1 cassette for $13, on a SOLOS CUTS cassette for $18, and on paper tape for other 8080 based computers for $15, from Micro-Ware Ltd, 27 Firstbrooke Rd, Toronto Ontario CANADA M4E 2L2.■

Circle 653 on inquiry card.

Data File Program

Practical Programming Company, POB 3069, North Brunswick NJ 08902, has introduced a program product called the "Data File Program," designed for the 8080 or Z-80 processors and assembled to start at either address 0 or address 2000. This program is a form of editor which uses memory to create named files of data, with each file consisting of a number of records. The program includes a search feature, as well as facilities for editing data. This 1024 byte program is available for $10 (specify which origin, hexadecimal 0 or 2000, with your order).■

Circle 656 on inquiry card.

Assembly Language Aids for North Star Disk Users

The XEK package includes an assembler, autoline editor and disassembler, all using the North Star Disk Operating System for disk and terminal IO. Source and object programs can also be loaded from Tarbell format cassette tapes or from Intel hexadecimal format paper tapes. As many as six source or object files may be simultaneously resident in programmable memory. XEK comes with a user manual for $48 from the Byte Shop of Westminster, 14300 Beach Blvd, Westminster CA 92683, (714) 894-9131.■

Circle 658 on inquiry card.

Word Processing System for iCOM Floppy Users

This word processing system features text filling and justification, line centering and underlining, page numbering and top and bottom page titles, and variable line spacing. It comes in hexadecimal ASCII format on a data diskette ready to run under iCOM's FDOS II or III. Input to the word processor is created using the FDOS text editor, and formatted output is written back to a diskette. A driver for the Anderson Jacobson AJ 841 Selectronic terminal is also available. The package is $235 from Ortronics, 4753 Irvine Av, N Hollywood CA 91602, (213) 763-0404.■

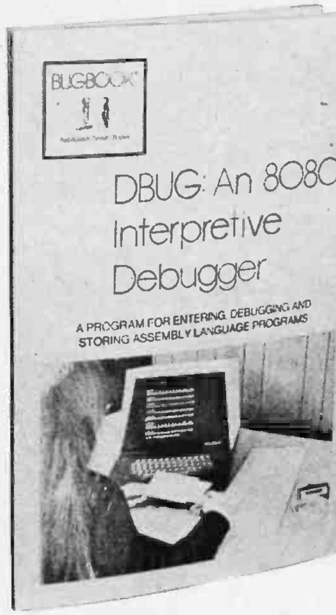
Circle 657 on inquiry card.

Get Your Editor and Operating System in a Poly Bag

Here's 8080 software offered on cassette and paper tape media, packaged in polyethylene bags. Offerings include the EDIT 3.0 text editor ($22.50), the COS 1.0 cassette operating system ($15), and the SOS 1.0 small operating system ($15) which includes utilities for the Tarbell cassette interface and Oliver paper tape reader. The SOS 1.0 and EDIT 3.0 user manuals contain 4.8 and 56 pages respectively. Dealers may place single quantity orders at the dealer discount to try the software and look over the documentation. Contact LSM Engineering, POB 3243, Orange CA 92665.■

Circle 571 on inquiry card.

BUGBOOK Writers Write Debug Book



This 100 page paperback, first in the BUGBOOK Application Series on assembly language programming, describes an interpretive debugger program for the 8080 which enables the user to enter and modify a program in memory and single step through program execution. The DBUG program was written for reading and punching paper tape on a Teletypewriter, but the IO routines can be easily changed to accommodate other peripherals. DBUG resides in 1 K bytes of memory, and a bootstrap loader for the *DBUG: An 8080 Interpretive Debugger* and hexadecimal listings of the DBUG program are given in the appendices. DBUG: An 8080 Interpretive Debugger sells for $5 from E&L Instruments Inc, 61 First St, Derby CT 06418, (203) 735-8774.■

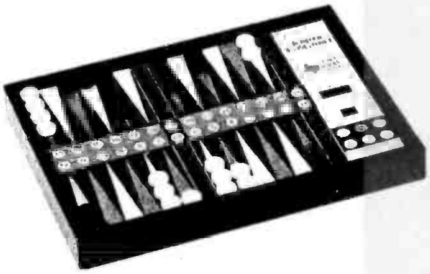
Circle 654 on inquiry card.

A Personal Data Base Management System

This data base management system should be useful in many applications where information must be stored, retrieved and modified. Commands are provided to create files, add, delete or list records in sequence or selectively, change fields within existing records, or search fields for a string or for integer values. The current system is designed for an Altair 8800b computer with one or two floppy disks and a minimum of 32 K bytes of memory. The PDMS system is supplied on an 8 inch floppy disk for $795, including a 40 page manual which contains a source listing of the program. The manual, which illustrates typical applications of the system, is available separately (without the source listing) for $20, from the Microware Division of Physical Biological Sciences Ltd, POB 47, Blacksburg VA 24060, (703) 951-9469.■

Circle 655 on Inquiry card.

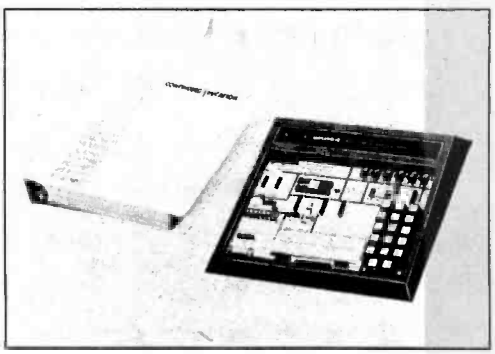
Backgammon, Anyone?



This backgammon board comes with something extra: a built-in opponent in the form of an Intel microprocessor controlled by a program in read only memory. The program generates random dice rolls, interacts with the playing keys and is said to base its moves on an analysis of the current board position using game theory and probabilistic methods. The backgammon game should be available in many retail outlets, and has been selected for inclusion in the Horschow's Collections and American Express gift catalogs. Priced at about $200 retail, the game is produced by Texas Micro Games Inc, 6230 Evergreen E, Houston TX 77081, (713) 778-9547.■

Circle 572 on inquiry card.

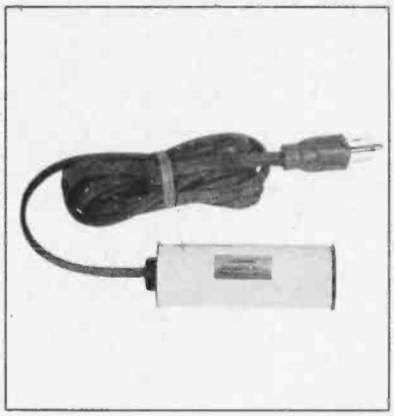
6800 Based Microcomputer Trainer from Heath



The EE-3401 self-instructional course provides tutorial material and hardware and software experiments in microprocessor operation, interfacing and programming. The course is designed to be used with the ET-3400 microprocessor trainer, which features the 6800 microprocessor, 256 bytes of programmable memory, 1 K byte read only memory monitor, and a 6 digit hexadecimal display and keyboard. Breadboarding sockets permit fast construction of experiments and special prototype circuits. The EE-3401 course and ET-3400 microprocessor trainer, priced at $89.95 and $189.95 respectively, are described in a free catalog available from Heath Company, Dept 350-460, Benton Harbor MI 49022, (616) 982-3236.■

Circle 578 on inquiry card.

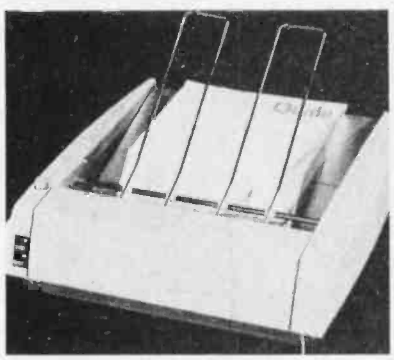
Low Cost Power Surge Protection



Has your computer ever lost its mind and memory during an electrical storm? This line cord transient suppressor will absorb repeated power surges from lightning or heavy duty electrical equipment, protecting delicate electronic circuits. Available in two prong ($11.50) or three prong ($14.50) plug form, these units can also be obtained with integral power line hash filtering, from Electronic Specialists Inc, POB 122, Natick MA 10760, (617) 655-1532.■

Circle 573 on inquiry card.

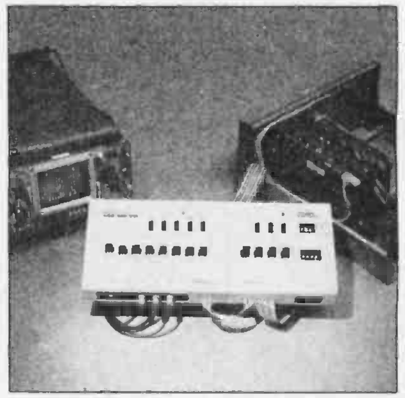
Paper Feeder for Word Processing Printer



This device eliminates the manual task of inserting fresh paper and removing typed documents from a printer in high volume word processing applications. The SpeedFeed interfaces to a Qume daisywheel printer, inserts sheets into the printer's platen, removes completed sheets and stacks them in an internal hopper. The hopper can hold up to 180 sheets of paper in sizes up to 12 by 14 inches (30.5 by 35.6 cm). Sensors automatically detect an end of paper condition. The SpeedFeed is $1390 in single quantities with 90 day delivery; substantial quantity discounts are available, from Qume Corporation, 2323 Industrial Pky W, Hayward CA 94545, (415) 783-6100.■

Circle 579 on inquiry card.

Low Cost Logic State Analyzer



The Model 100A logic state analyzer is said to be the lowest priced data domain instrument available. The basic unit operates as an 8 channel stand alone analyzer, offering a 16 word truth table display of ones and zeros on an ordinary oscilloscope, post-trigger and pre-trigger data collection, hexadecimal and octal formats, and both snapshot and repetitive display presentations. The Model 100A can be mated with the Model 10 expansion unit on an optional baseplate to provide a 24 bit logic analyzer capable of monitoring a microprocessor's full address and data bus. The expanded package also provides a user programmable digital delay for paging through programs up to 1000 steps long and a pass counter for monitoring loops. Selected bus operations can be captured and displayed with the clock and trigger qualifiers on the combined package. The units can be used with a variety of logic families and are capable of handling data rates in excess of 8 megabytes per second. The Model 100A basic unit and Model 10 expansion unit are each priced at $295 assembled or $229 as a kit. The optional baseplate is $9.95, and a separate owner's manual is available for $4.95, from Paratronics Inc, 800 Charcot Av, San Jose CA 95131, (408) 263-2252.■

Circle 576 on inquiry card.

Upgrade Kit for PolyMorphic Systems Users

The Poly 88 Disk Kit contains all mechanical parts and electronic assemblies needed to convert a Poly 88 microcomputer into a new System 8813 disk based system. The kit includes a chassis, walnut cabinet with brushed aluminum front panel, a 10 slot backplane, power supply and fan, floppy disk controller, 2 K bytes of read only memory, one floppy disk drive and two system diskettes. The conversion kit costs $1450 and is said to take only a few hours to install. Up to two more disk drives may be added at a cost of $590 each. The kit is available from PolyMorphic Systems Inc, 460 Ward Dr, Santa Barbara CA 93111, (805) 967-0468 or from PolyMorphic Systems dealers.■

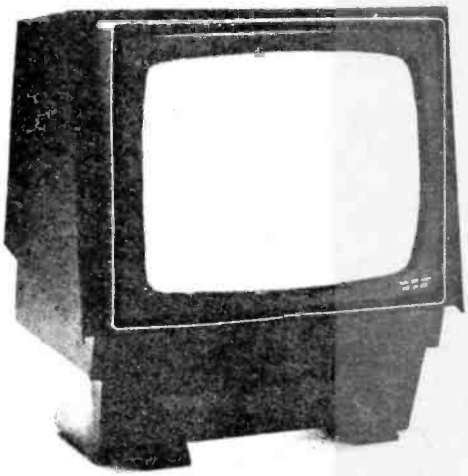
Circle 577 on inquiry card.

Fantastic $avings on Terminal Components

We have obtained a fairly large supply of professional CRT video monitors, encased in attractive metal cabinets with a simulated mahogany finish. We do not know the bandwidth capabilities of these 12" (diagonal) units; we have used them, however, to test our 24 x 80 video display board and have found them perfectly satisfactory. These units were manufactured for one of the largest data communications firms in the country. We are not allowed to use the name, and the nameplates had to be removed. Many of you have probably seen these units functioning. They are equipped with a standard video connector and have all the normal controls. They operate at 110 V, 60 cycles.

The units are in reasonably good condition cosmetically, although nearly all of them have a defect in the plastic anti-glare screen in front of the CRT tube. This screen could be readily removed or replaced.

We estimate that these units would sell new for between $150 and $200. We are offering them for sale in both functional and non-functional condition.



12CRT Used, operable* . . . . $ 59.95

12CRTNF Used, complete, known non-functioning . . . . . $ 39.95

*Units have been tested but are sold as-is. They are not represented as reconditioned units and may require minor repairs or adjustments.

Add the following charges for handling-shipping-insurance:
$2.50 Eastern Time Zone $3 Central $4 Mountain $5 Pacific

MiniMicroMart, Inc. also stocks a complete line of kits for building video display units.
Write for Information

COMMERCIAL MODEMS — *Limited Supply* — Acoustical coupler type and direct hard wire variety. Both operate at standard Bell Telephone frequencies, at up to 300 baud; they are Bell 103 compatible. They appear to be in new or equal-to-new condition. They are intended for communicating from a terminal to a time-share computer. Standard RS232C type connectors are supplied.

MDACP (acoustical)	$ 59.95
MDHW (hard wire)	44.95
MDACP-NF (used, suspect non-functional)	39.95
MDHW-NF (used, suspect non-functional	24.95

Add $2 for handling-shipping-insurance.

KEYBOARDS— *Limited Supply*
Attractive communications style keyboards; some in cases which match the monitor shown above. They are not ASCII encoded but the coding could be changed in software with PROMs or by replacing the circuitry with an encoding I.C. They key switch modules are of Cherry manufacture with an excellent feel. A schematic and limited modification information is supplied.

KBN brand new, in case	$ 37.95
KBU used, in case	27.95
KBUD used, in case, minor cosmetic defects	22.95
KBUNC used, no case	19.95

Add $2.50 for handling-shipping-insurance.

Write for free 64-page catalog featuring hundreds of items for minicomputer systems.

MiniMicroMart, Inc. 1618 James Street, Syracuse, N.Y. 13203 (315) 422-4467

Circle 77 on inquiry card.

Circle 19 on inquiry card.

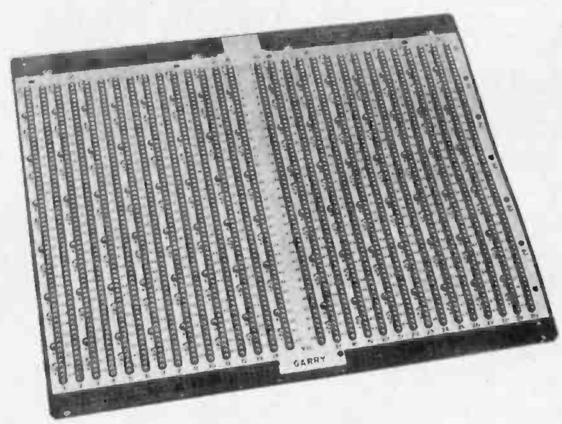
What's New? MISCELLANEOUS

Arabic and Farsi Character Sets

The Intecolor 8001 data terminal is now available with an Arabic and Farsi character set option which allows text in these languages to be entered from the keyboard and manipulated in BASIC. The shapes of certain characters are automatically changed depending on their position in a word as is the custom in these languages. Prices for the Intecolor systems with the Arabic and Farsi option start below $5000 from Intelligent Systems Corporation, 5965 Peachtree Cors E, Norcross GA 30071, (404) 449-5961.■

Circle 574 on inquiry card.

Universal Wire Wrap Panels



These wire wrap packaging panels consist of 30 columns, with 64 terminals per column, on 0.001 inch centers. The low profile IC socket contacts have 0.025 inch square terminations. Available with or without 108 pin connectors, the panels are priced at $1 to $2.50 per IC position from Garry Manufacturing Company, 1010 Jersey Av, New Brunswick NJ 08902, (201) 545-2424.■

Circle 575 on inquiry card.

Vector
8800V
Universal Microcomputer/processor plugboard, use with S-100 bus. Complete with heat sink & hardware. 5.3" x 10" x 1/16"
1-4 $19.95
5-9 $17.95
10-24 $15.96
8801-1
Same as 8800V except plain; less power buses & heat sink.
1-4 $14.95
5-9 $13.46
10-24 $11.96
3662 6.5" x 4.5" $7.65
3662-2 9.6" x 4.5" $11.45
P pattern plugboards for IC's Epoxy Glass 1/16" 44 pin con. spaced .156
NLS
SALE
List Sale
LM3A 3 dig 1% DC ...... $125. $110.
LM3.5A 3½ dig .5% DC ...... $147. $128.
LM40A 4 dig .1% DC ...... $190. $165.
LM4A 4 dig .03% DC ...... $227. $195.
SPECIFICATIONS
FEATURES
3690-12
Card Extender has 100 contacts — 50 per side on .125 centers — Attached connector — Is compatible with S-100 Bus Systems $25.00
3690 6.5" 22/44 pin .156 ctrs. Extenders $12.00
MS-15 MINISCOPE $289
With Rechargeable Batteries & Charger Unit
• 15 megahertz bandwidth.
• External and internal trigger.
• Time base — .1 microsec. to 0.5 Sec/div - 21 settings ± 3%.
• Battery or line operation.
• Automatic & line sync modes.
• Power consumption < 15 watts.
R644 P.C. receptacle 22/44 cont. .156 ctrs pierced solder eyelets tails gold $3.45
R644-3 P.C. receptacle 22/44 cont. .156 ctrs. Wire Wrap tails gold $4.49
R681 DIP Bus P.C. Edge Connectors solder tails on 140" spaced rows for ALTAIR mother boards. Fits .042 dia. holes. gold S-100 $5.00
R681-1 .025" sq. 3 wrap length posts (.62 long) on 250 spaced rows for wrapping gold $7.35
R681-2 .025" sq. 1 wrap length posts (.22 long) on 250 spaced rows for IMSAI mother boards. gold $5.85
R681-3 pierced solder eyelet tails. gold $7.35
8803
MOTHER BOARD FOR S100 BUS MICRO-COMPUTERS
Price: $29.50
LIQUID CRYSTAL DIGITAL CLOCK-CALENDAR
APPLIED MARKETING
12:28
• For Auto, Home, Office
• Small in size (2x2½x¾)
• Push button for seconds release for date.
• Clocks mount anywhere with either 3M double-sided tape or VELCRO included.
• 2 MODELS AVAILABLE:
LCD-101, portable model runs on self-contained batteries for better than a year.
LCD-102, runs on 12 volt system and is back-lighted.
• LCD-101 or LCD-102 your choice $34.95 ea.
• Clear desk stand for $2.00
WRAP POST
for .042 dia. holes (all boards on this page)
T-44 pkg. 100 $2.28
T-44 pkg. 1000 $14.00
A-13 hand installing tool $2.80
SLIT-N-WRAP
Wraps insulated wire on .025" square posts FOUR TIMES FASTER than regular manual wrap-post tools
Vector
NO PRE-STRIPPING • NO PRE-CUTTING • SPOOL-FED WIRE
P180 with two 100' spools of 28 ga. wire $24.95
P180-4T charger, wire includes $75.00
SLIT-N-WRAP WIRE NO. 28 GAGE INSULATED WIRE, 100' SPOOLS
W28-2-A Pkg. 3 Green
W28-2-B Pkg. 3 Red
W28-2-C Pkg. 3 Clear
W28-2-D Pkg. 3 Blue
$2.75 Package
Buy P180 and get 1 package of wire for 1c
Buy P160-4T and get 3 packages of wire for 1c
ACE • All-Circuit Evaluator for fast, solderless, plug-in circuit building and testing
Just plug in any components with leads to .032" dia. Interconnect with solid wire up to 20 ga. Assembled models too!
Model 236
14 & 16 PIN 3 LEVEL WIRE WRAP SOCKETS
14-T3 100 for $30.00
16-T3 100 for $30.00
50 of ea. for $32.00
MICRO-KLIP
for .042 dia. holes (all boards on this page)
T42-1 pkg. 100 $1.50
T42-1 pkg. 1000 $11.00
P-149 hand installing tool $2.03
NLS
SALE
SC-5 With Rechargeable Batteries & Charger Unit $79
FM-7 With Rechargeable Batteries & Charger Unit $168
2708 8K 450 ns EPROM FACTORY PRIME
1-9
10-24 $13.00
25+ Call for Price $12.50
24 PIN DIP PLUGS WITH COVERS
3 / $1.00
40 / $10.00
1/16 Vector BOARD
.042 dia holes on 0.1 spacing for IC's
2102L1 450ns RAM
Low Power—High Speed
8-24 $1.50
25-99 $1.40
100-249 $1.25
250 + $1.19
3 LEVEL GOLD WIRE WRAP SOCKETS
Price Breakthrough!
MA1003 CAR CLOCK
12:48
$17.95
Bright Green Fluorescent Display Crystal Time Base Assembled ... add switches and 12 VDC
SPECIAL
14CS2 100 for "14"
16CS2 100 for "16"
Solitron DEVICES INC CMOS
TAKE ADVANTAGE OF THIS SPECIAL PURCHASE OF 100% FACTORY PRIME PARTS
CM4006AE 2.00
CM4007AE ...... 8 / 1.00
CM4008AE ...... 3 / 1.00
CM4011AE ...... 8 / 1.00
CM4012AE ...... 5 / 1.00
CM4013AE ...... 4 / 1.00
CM4014AE ...... 3 / 2.00
CM4015AE ...... 3 / 2.00
CM4017AE ...... 2 / 1.00
CM4018AE 1.00
CM4019AE ...... 4 / 1.00
CM4020AE ...... 3 / 2.00
CM4022AE 1.00
CM4023AE ...... 8 / 1.00
CM4024AE ...... 3 / 1.00
CM4025AE ...... 5 / 1.00
CM4030AE ...... 7 / 1.00
CM4035AE ...... 3 / 2.00
CM4037AE 2.00
CM4041AE ...... 2 / 1.00
CM4047AE 1.00
CM4049AE ...... 4 / 1.00
CM4050AE ...... 4 / 1.00
CM4116AE ...... 4 / 1.00
(10% Discount on 100 pcs. or More, if a Minimum 25 per Part No. are Purchased.)
3677 9.6" x 4.5" $10.90
3677-2 6.5" x 4.5" $9.74
Gen. Purpose D.I.P. Boards with Bus Pattern for Solder or Wire Wrap. Epoxy Glass 1/16" 44 pin con. spaced .156
3682 9.6" x 4.5" $10.97
3682-2 6.5" x 4.5" $9.81
Hi-Density Dual-In-Line Plugboard for Wire Wrap with Power & Grd. Bus Epoxy Glass 1/16" 44 pin con. spaced .156
$5 Value BY Free
Just mention or send this COUPON and receive with your order — DIP PROJECT BOARDS — assorted sizes and types, epoxy glass, propunched and copperclad Boards.
PRIORITY ONE ELECTRONICS
10031 Woodley Ave. Sepulveda CA. 91343
Terms: VISA, MC, BAC, check, Money Order, C.O.D., U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Orders less than $75.00 include 10% shipping and handling; excess refunded. Just in case . . . please include your phone no.
Good thru March 15,1978
VISA
Send for our latest brochure.
phone orders welcome (213) 893-8202
OEM and Institutional Inquiries Invited.

Classified Ads

FOR SALE: 2708 programmer with preprogrammed PROM. Plugs into KIM-1 expansion connector and uses K-1 and K-2. Includes 6530 PIA, $95. −5, 12, 26 V power supply, $45. James Grina, 1284 Fitield Pl, St Paul MN 55108.

WANTED: Manual or timing information for a Burroughs A560 punch. R Six, 30725 Tennessee, Roseville MI 48066.

FOR SALE: Minicomputer, Digital Equipment PDP8-L, 4 K x 12 memory, Teletype IO, with complete original documentation, spare parts and much software $500. Hewlett-Packard 120A oscilloscope excellent condition with manual $100. Viatron cassette drive $15. Gary Hansen, POB 1337, Ft Davis TX 79734. (915) 426-3331.

WANTED: Datapoint 2200 processor version 2, used, for reasonable price. Send details by air mail to Carl DuBois, 69 Nasuli, Malaybalay, Bukidnon 8201 PHILIPPINES.

FOR SALE: BYTE number 1 for $10, BYTE numbers 2, 3, 4 and 5 for $5 each. These BYTES are now classics. All magazines in mint condition. Will sell any combination. Please send money order to D Mathews, POB 469, Lynden WA 98264.

WANTED: BYTE issues, June 1976 to March 1977. Evan H Foreman, POB F, Mobile AL 36601.

FOR SALE: Two running bit 483 — 16 K core memory minicomputers and an extra 4 K x 10 core memory (with all drivers and ps). Over 150 computer programs, including FORTRAN, timesharing on paper tape (source and object tapes) for the bit 483, all LBDs, wire list, some PCB negatives, almost complete set of tested spare PCB for bit, all above only $2500 FOB Phoenix. Ralph Greenthal, 5009 E Windsor Av, Phoenix AZ 85008.

SWAP: For computer of equal value: Heathkit HW-100 5 band radio, 10 thru 80 meter 180 W SSB/CW transceiver, converted to cover 11 meter CB in addition. AC power supply, speaker console and 4 BTV HUSTLER antenna. Perfect condition. Ron Dudeck, 1504 W I St, Ontario CA 91762.

FOR SALE: Honeywell 200 computer, tape drives, line printer, card reader; Dura Mach 10; Bendix G-15; Royal LGP-30. For details send stamped self-addressed envelope to S Lang, 730 Bridge, Apt 7, Davenport IA 52803.

ASSEMBLED SHERE BOARDS FOR SALE: CRT/1, CPU/2, 16 K memory. Contact Richard Likwartz (307) 362-5316.

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed notice to that effect. To be considered for publication, an advertisement should be clearly noncommercial, typed double spaced on plain white paper, and include complete name and address information. These notices are free of charge and will be printed one time only on a space available basis. Insertions should be limited to 100 words or less. Notices can be accepted from individuals or bona fide computer users clubs only. We can engage in no correspondence on these and your confirmation of placement is appearance in an issue of BYTE.

Please note that it may take three or four months for an ad to appear in the magazine.■

WANTED: Issue number 3, BYTE, November 1975, in good condition. Vincent M J Sumoski, A6-10 Lehigh Av, Gloucester Hts NJ 08030 (609) 456-4298.

WANTED: Back issues of BYTE. Issues 1 thru 10, 12 thru 13, Volume 2, Number 2. Good condition. John Berry, 1520 Aberdeen Av, Baton Rouge LA 70808.

FOR SALE: IMSAI 4 K EROM board with EROMS. Never used, $350. Associated Electronics PROM burner (burns 1702 and 8702 EROMS) with power supply, $260. J Williams, 2415 Ansdel Ct, Reston VA 22091.

FOR SALE: TVT-3 Terminal, never used. $95 or best offer takes. Also, I'm selling an ASCII keyboard with MOS encoder, adjustable parity, etc, for $20 "like new." David Tucker, 23681 Marlow, Oak Park MI 48237. Or call after 4:00 (313) 967-3130.

TELEPRINTERS FOR SALE: Model 15 (Baudot) with table, $65; Model 28ASRs, KSRs or ROs (Write); Model 33ASR with modem, $675. Some Model 35 (ASCII) equipment. Parts and supplies (paper, tape, ribbons). Model 33ASR wiring diagram packet, $5.75; Model 33 copyholders, $14; Model 33 readers and parts (write). Send SASE for complete list. Lawrence R Pfleger, 2141 N 52nd St, Milwaukee WI 53208.

USED COMPUTER TAPE FOR SALE: Standard ½ inch tape, 9 level 1600 bpi on 2400, 1200 and 600 foot reels. Came from a large data center which just switched computers. Also have about 100 old 2 foot by 2 foot programming boards plus several boxes of jumper wires. Also four boxes of hardware for raised computer room flooring. Call (904) 575-2183 or write Loren Moltedo, 2636-273 Mission Rd, Tallahassee FL 32304.

FOR SALE: ASR-33 Teletype with stand, paper punch and reader, for data 1610 modem, and documentation for all, $775. Russ Tremain, 1324 Mission St, Santa Cruz CA 95060. (408) 427-3656.

FOR SALE: Brand new 4 K EROMS, type MM-5204AQ 512 x 8 bit uV erasable, 750 ns, with rubber carriers and data sheets. $6.50 each, or all 12 (6144 bytes) for $75. Steven Hain, 40 Wilshire Dr, Sharon MA 02067. (617) 784-3374 weeknights.

FOR SALE: Tektronix RM15 oscilloscope with instruction and maintenance manual and new probe. Asking 160; I am willing to trade for a terminal (any type) or memory. Philip Kaaret, 1113 E State St, Ithaca NY 14850. (607) 272-9119.

WANTED: Any or all of the following issues of *HP-65 KEY NOTE*: volume 1, numbers 1 thru 5 and volume 2, number 1. Also interested in EE Pacs 1 and 2 and MATH Pacs 1 and 2 (software for discontinued HP-65). Will pay any reasonable price. James Damon, 1619 King St, Alexandria VA 22314.

FOR SALE: First four issues of BYTE (September thru December 1975) in new condition. Best offer over $20. Only highest bidder notified. Robert Kindt, 26 Russett Dr, RD #3, Allentown PA 18104.

FOR SALE: IMSAI MPU-A board with documentation and *Intel 8080 User's Guide*, all ICs are socketed, brand new, carefully assembled and tested, $130. Also four memory boards: two Vector 8 K static, $225 each, two MITS 4 K dynamic, $100 each and MITS 2 SIO board, $135. For more information call (518) 456-8717 or send SASE to Michael Favitta, 4 Sherwood Forest Rd, Albany NY 12203.

FOR SALE OR TRADE: Digital Group PHI-F assembled interface for cassette storage system. Purchased at $195, never opened or used, guaranteed brand new. 6400 bps rate, search speed 100 inches per second, recording density: 1600 FCPI. Test program tape, instruction manual, and board connectors included. First cashier's check or money order for $135 takes it. Also, will take best cash offer or trade for scope, DVM, or camera of equal value if not purchased outright. Let's make a deal! Mark Anglin, 929 Mills Rd, Wadsworth OH 44281. Phone: (216) 336-5769.

FOR SALE: Complete IMSAI 8080 System, 22 slot motherboard with ten connectors and fan; TDL 16 K memory board (250 ns); MIO board with cables; Processor Technology VDM board; Honeywell keyboard, XAM 100% solid state TV with video hookup; 2708 PROM Board with software for loading operating system from cassette. All manuals provided, plus instruction manual on operation of the complete system. Ready to be plugged in and run. $2500. Programming service also available. T Tai, 115 Bonny Ln, Collegeville PA 19426.

FOR SALE: New factory assembled Altair 8800B with 20 K static programmable memory (1 4 K board, 1 16 K board), 88 ACR cassette interface, serial interface, ADM-3A complete cursor control CRT. Extended BASIC software and all manuals; everything for only $2500 or best offer. Will consider selling units separately for best offer. Going away to college and must sell system. Please contact B Roberts, 505 E 82 St, New York NY 10028. (212) 734-5703.

FOR SALE: Okidata CP 110 printer (prints up to 110 characters per second) with RS-232 tractor feed, bidirectional printing and onboard self-test electronics. Less than one year old. New unit costs over $1700. Will sacrifice for $850. You pay shipping. Send SASE for sample of printout. Write to P Grivas, POB 3153, Walnut Creek CA 94598.

FOR SALE: Oliver papertape reader, $75. Polymorphics VTI 64 video board, $190. IMSAI audio cassette interface board, $50. Practically unused. J Williams, 2415 Ansdel Ct, Reston VA 22091.

FOR SALE: MITS 8080a mainframe. Never used, works, and is in good condition. Will sell for $400. Contact Tyrone Throop at 713 Erskine NW, Huntsville AL 35805, or call (205) 837-9246.

FOR SALE: DEC PDP 11/05 runs perfectly, has 8 K x 16 core memory. Extra boards, cables, service information, test tapes and documentation Included. First certified check for $1,500 plus shipping. Gerard C Plasse, 53 Main St, Oxford MA 01540. (617) 987-5588.

FOR SALE: New TEC-9900-SS-U super starter system kit 16 bit TI microprocessor, 32 bit IO, hardware multiply and divide, buffered bus, 20 mA or RS232, eight interrupt and sockets. Won at Hartford Ham Convention. List $299, first $175 takes it. Bunker Ramo keyboard #2200 and video display Model 2217 with self contained PS, schematics and manuals $95. John Keslo, 5 Belvina Cir, Pelham NH 03076. (603) 635-2508.

FOR SALE: SwTPC 6800 Computer with 4 K programmable memory. New and well constructed, 100% guaranteed, $250. A Cunningham, 1151 Seneca Pl, Charlotte NC 28210.

TRADE: Amateur radio equipment and testing equipment for an Altair 8800 or Imsai. Send for information to Dale Hutchinson, 10818 Brentway Dr, Houston TX 77070.

FOR SALE: Altair 8800a with 20 K programmable memory and ASR 33 Teletype (almost new), reader, punch, interfaces, fan, manuals, etc, 8 K MITS BASIC on paper tape. Cost almost $4000 new. Works perfectly but want to build my own now. Sacrifice this whole quality system for $2495, or just $1895 without the Teletype. First cashier's check or money order reserves this powerful computer and Teletype. Brian J Dowd, DDS, 1770 Century Cir NE, Atlanta GA 30345.

Reader Service

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from this list. Add a 13 cent stamp to the card, then drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by BYTE, bringing you a bigger BYTE.

**Reader Service inquiries not solicited. Correspond directly with company.*

BOMB
BYTE's Ongoing Monitor Box

Tightly Clustered

November's BOMB analysis provided two tight "clusters" of ratings in readers' reactions. The first cluster was a tie for first place between Steve Ciarcia's "Memory Mapped IO," page 10, and Burt Hashjzume's tutorial on "Floating Point Arithmetic," page 76. Each will receive a $100 bonus check for placing 1.6 standard deviations above the mean. The second place cluster consisted of two complete computer plans: "Kompuutar" by David Brader, page 94, and "Building a Computer From Scratch" by Hilary Jones, page 80. Each will receive a $50 bonus check for placing 0.7 standard deviations above the mean in readers' preferences. For November's voting, the standard deviation (σ) was 17% of the mean of 14 articles. Fill out your BOMB card with ratings from 0 to 10 for each article, plus any uncensored comments you have for this direct line to the editor's desk.■

Jim-pak® electronic components

One-Stop Component Center

QUALITY PRODUCTS

TTL

7400	7490	
7402	7492	
7404	7493	
7408	74107	
7410	74121	
7420	74122	
7430	74123	
7432	74125	(DM8093)
7442	74154	
7447	74161	(DM9316)
7473	74176	(DM8280)
7474	74177	(DM8281)
7475	74192	
7476	74193	
7485	74367	(DM8097)

Low Power TTL Schottky

74LS00	74LS83
74LS02	74LS85
74LS04	74LS86
74LS08	74LS90
74LS10	74LS109
74LS20	74LS123
74LS30	74LS136
74LS32	74LS138
74LS73	74LS151
74LS74	74LS175
74LS75	74LS367

C/MOS

4000	4029
4001	4044
4010	4046
4011	4049
4013	4050
4016	4051
4017	4069
4020	4071
4023	4081
4024	4511

LINEAR

LM301V	LM340T 15(7815)
LM307V	LM555V
LM309K	LM556N
LM311V	LM567V
LM320K 5(7905)	LM723N
LM320T 5(7905)	LM739N
LM320T 12(7912)	LM741V
LM320T 15(7915)	LM747N
LM339N	LM1458V(5558V)
LM340T 5(7805)	LM1556V
LM340T 12(7812)	LM3900N(CA3401)

MICROPROCESSOR

Z 80	2101
8080A	2102
8212	21L02
8224	7489
8228	MM5262
6800	1702A
6810	82S23
6830	2708
AY-5-1013	DM8835N
2513/2140	N8T97

DIODES

IN751	IN4148(IN914)
IN4733	IN4001
IN4734	IN4004
IN4742	IN4007
IN4744	MDA-980-3

TRANSISTORS

C106B1	2N3055
2N2222A	2N3904
2N2907A	2N3906
MJE2955	2N5129
MJE3055	2N5139

RESISTORS

ASST 1	ASST 5
ASST 2	ASST 6
ASST 3	ASST 7
ASST 4	

CAPACITORS

Aluminum Electrolytic

1mfd 50V	100mfd 50V
4.7mfd 50V	220mfd 50V
10mfd 50V	470mfd 50V
22mfd 50V	1000mfd 25V
47mfd 50V	2200mfd 16V

Ceramic Disc

10pf 50V	.001mfd 50V
47pf 50V	.0047mfd 50V
100pf 50V	.01mfd 50V
220pf 50V	.022mfd 50V
330pf 50V	.047mfd 50V
470pf 50V	.1mfd 50V

Dipped Tantalum

.1mfd 35V	2.2mfd 25V
.22mfd 35V	3.3mfd 25V
.33mfd 35V	4.7mfd 25V
.47mfd 35V	6.8mfd 25V
.68mfd 35V	10mfd 25V
1mfd 35V	15mfd 25V
1.5mfd 35V	33mfd 25V

SWITCHES

SPST 4 sta dip	SPDT push button
SPST 8 sta dip	SPDT push button
SPDT on off on toggle	SPST push button
	SPST push button
SPDT on none on toggle	SPST push button
	DPDT slide
DPDT on off on toggle	DPDT on none on toggle

SOCKETS

8 pin low profile	14 pin wire wrap
14 pin low profile	16 pin wire wrap
16 pin low profile	24 pin wire wrap
24 pin low profile	40 pin wire wrap
40 pin low profile	TO-3 Socket
Molex Pins	TO-5 Socket

Display LEDS Discrete

5x7 dot matrix	600 com. anode
300 com. anode	600 com. cathode
300 com. cathode	200LED Lamp-red

CONNECTORS

DB 25P Plug	DB25S Socket

DATA BOOKS*

7400/74LS Data Book
CMOS/Linear Data Book
Microprocessor/LED Data Book
*JIM-PAK Products only

A component product line developed for the independent dealer. Guaranteed, nationally advertised products. Complete **JIM-PAK** program includes national advertising, direct mail programs, store display racks, stock rotation plan and return policy. For dealer information, write or call **JIM-PAK**, 1021 Howard Avenue, San Carlos, CA 94070 (415) 592-8097.

★★

SEE YOUR LOCAL Jim-pak electronic components DEALER TODAY...

ALABAMA	Mobile	Lafayette Radio Electronics
ARIZONA	Fountain Hills	P & C Communications
CALIFORNIA	Bellflower	Earl's Hobby Shop
	Berkeley	Al Lasher Electronics
	Fontana	Fontana Electronics
	Long Beach	Scott Radio Supply
	Monterey	ZackIt
	Palo Alto	Zack Electronics
	Pasadena	Dow Radio Inc.
	Sacramento	The Radio Place
	Sacramento	Zackit
	San Carlos	J&H Outlet Store
	San Francisco	Zack Electronics
	San Jose	Quement Electronics
	Sunnyvale	Sunnyvale Electronics
	Vallejo	Zackit
	Walnut Creek	Byte Shop Computer Store
CANADA	Alberta (Calgary)	The Computer Shop
COLORADO	Aurora	Com Co Electronics
	Steamboat Springs	Norm's TV & Electronics
CONNECTICUT	Bridgeport	Computer World
FLORIDA	Lakeland	Lakeland Specialty Electronics
	Orlando	Altair Computer Center of Orlando
	Tampa	AMF Electronics
	Tampa	Microcomputer Systems
FRANCE	Paris	Computer Boutique
GEORGIA	Atlanta	Atlanta Computer Mart
HAWAII	Aiea	Delcoms Hawaii
	Honolulu	Integrated Circuit Supply
IDAHO	Idaho Falls	Audiotronics
ILLINOIS	Evanston	Itty Bitty Machine Co.
	Groveland	Moyer Electronics
ILLINOIS (Cont'd)	Mount Prospect	Tri-State Electronics
	Oak Park	Spectronics
INDIANA	East Chicago	Acro Electronics
	Hammond	Quantum Computer Works
KENTUCKY	Lexington	Radio-Electronic Equipment Co.
LOUISIANA	Baton Rouge	Davis Electronics Supply
MARYLAND	Baltimore	Computer Workshop of Baltimore
	Baltimore	Everything Electronic
	Rockville	Computer Workshop
	Towson	Computers, Etc.
MASSACHUSETTS	Medford	Tufts Electronics
	North Adams	Electronics Supply Center
	Waltham	Computer Mart Inc.
MICHIGAN	Grand Rapids	Micro Computer World
	Lansing	Fulton Radio Supply
MINNESOTA	Duluth	Northwest Radio of Duluth
	Eagan	Dacom Amateur Radio Center
MISSOURI	Parkville	Computer Workshop of Kansas City
MONTANA	Billings	Conley Radio Supply
NEBRASKA	Lincoln	Altair Computer Center
	Omaha	Omaha Computer Store
NEVADA	Las Vegas	Century 23
NEW JERSEY	Cherry Hill	Computer Systems Unlimited
	Hoboken	Hoboken Computer Works
	Pompton Lakes	Computer Corner of New Jersey
	Ramsey	Typetronic Computer Store
NEW YORK	Albany	Fort Orange Electronics
	New York	The Computer Stores Inc.
	New York	Computer Mart of New York
	Troy	Trojan Electronics
	White Plains	The Computer Corner
NORTH CAROLINA	Raleigh	Byte Shop of Raleigh
OHIO	Bucyrus	Mead Electronics
	Cincinnati	Digital Design
	Dayton	Altair Computer Center
	Reynoldsburg	Universal Amateur Radio
	Steubenville	Hosfelt Electronics
OKLAHOMA	Guymon	Sound Service
	Oklahoma City	Bits, Bytes & Micros
OREGON	Beaverton	Altair Computer Center
	Coos Bay	Herrick Electronix
	Ontario	Miller Electronics
PANAMA	Panama City	Sonitel, S.A.
PENNSYLVANIA	Hershey	Microcomputer Systems Inc.
	Murraysville	Computer Workshop of Pittsburgh
RHODE ISLAND	Cranston	Jabbour Electronics City
	Pawtucket	Jabbour Electronics City
SOUTH CAROLINA	No. Charleston	Technical Services Inc.
TENNESSEE	Knoxville	Byte Shop
	Memphis	Sere-Rose & Spencer Electronics
	Oak Ridge	Computer Denn
TEXAS	Dallas	Computer Shops Inc.
	Houston	Altair Computer Center
	Houston	Interactive Computers
	San Antonio	Sherman Electronics Supply
VIRGINIA	Alexandria	Computer Hardware Store
	Alexandria	Computers Plus
	Richmond	Computers-To-Go
	Springfield	Computer Workshop of North Virginia
	Virginia Beach	Heathkit Electronics Center
WASHINGTON	Bellevue	Altair Computer Center
	Longview	Progress Electronics
	Pasco	Riverview Electronics
	Spokane	Personal Computers
WEST VIRGINIA	Morgantown	The Computer Corner
	Morgantown	Electro Distributing Co.

Circle 63 on inquiry card.