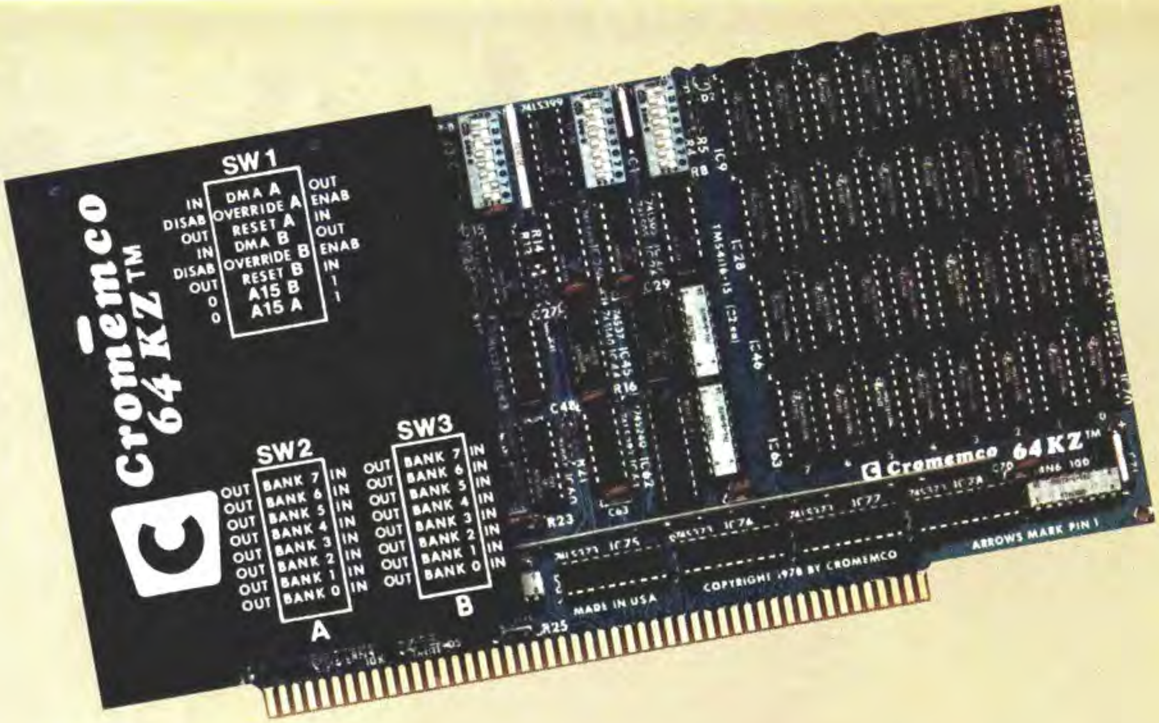
JULY 1979 Volume 4, Number 7 $2.00 in USA/$2.4[illegible] Canada

# BYTE®

the small systems journal

A McGRAW-HILL PUBLICAT[illegible]


AUTOMATING ECLIPSES




# You can do surprising things when you have 64 kilobytes of fast RAM on one card

**4 MHz FAST—AND EXPANDABLE**

Here's 64 kilobytes of memory on one RAM card. Yes, we mean 512K bits of read/write memory on this single card.

And, yes, we mean it's fast. With 150-nanosecond chip access times — so the card can operate in fast Z-80 systems with no wait states. Repeat, no wait states.

**EXPANDABLE ON TWO LEVELS**

Not only does the new Model 64KZ give you a large, fast RAM but it is expandable on two levels.

First, through our Cromemco Bank Select feature, you can expand to 512 kilobytes in eight 64K banks.

Or, with our Extended Bank Select feature, you can expand memory space to as much as 16 megabytes.

This expandability we call your obsolescence insurance.

The legend on the card's heat sink is an easy reference for address and bank selection.

**BENCHMARK IT**

Obviously, the speed and memory capacity of this new card give you a lot of power.

You can see that for yourself in our new 7-station Multi-User Computer System which uses these Model 64KZ cards. This S100-bus system outperforms the speed of many if not most timesharing systems of up to 10 times the Cromemco price.

And yet where some of these much more expensive and cumbersome systems clearly slow to a snail's pace when timesharing, the Cromemco system using Bank Select switching runs surprisingly fast.

**SEE IT NOW**

See the new Model 64KZ at your computer dealer now. Study the literature on it. See how for only $1785 you can get around that ever-present barrier of memory that's too little and too slow.

For high reliability all Cromemco memory cards are burned in at the factory in these temperature-controlled ovens.

Cromemco Multi-User System shown with 7 stations

**Cromemco**
incorporated
280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400
Tomorrow's computers now

Circle 80 on inquiry card.


# Here's how you can be fully computerized for so much less than you thought

**BUSINESS — EDUCATION — ENGINEERING — MANUFACTURING**

We are pleased to announce the first professional time-sharing system in the microcomputer field.

Naturally, it's from Cromemco.

This new multi-user system will do all of the tasks you usually associate with much more expensive time-sharing computers. Yet it's priced at an almost unbelievably low figure.

Look at these features:

- **You can have up to 7 terminals plus a fast, 132-column line printer**
- **You can have a large system RAM memory that's expandable to ½ megabyte using the Bank Select feature**
- **Each user has an independent bank of RAM**
- **You can have floppy disk storage of up to 1 megabyte**
- **You have confidentiality between most stations**
- **And, make no mistake, the system is fast and powerful. You'll want to try its fast execution time yourself.**


**PROGRAMMERS LOVE OUR BASIC**

This new system is based on Cromemco's well-known System Three Computer and our new Multi-User BASIC software package.

Programmers tell us that Cromemco Multi-User BASIC is the best in the field. Here are some of its attractions:

- **You can use long variable names and labels up to 31 characters long — names like "material on order" or "calculate speed reduction."**
- **You get many unusual and helpful commands that simplify programs and execution — commands such as PROTECT, LIST VARIABLES, NOLIST, and many more.**
- **No round-off error in financial work (because our BASIC uses binary-coded decimal rather than binary operation). And we've still been able to make it FAST.**
- **Terminals and printer are interrupt-driven — no additional overhead until key is pressed.**
- **The conveniences in this Multi-User BASIC make it much easier to write your own application software.**
- **A line editor simplifies changes.**

**BENCHMARK IT — NOW**

In the final analysis, the thing to do is see this beautiful new system at your dealer. See its rugged professional quality. Evaluate it. Benchmark it for speed with your own routine (you'll be agreeably surprised, we guarantee you).

Find out, too, about Cromemco's reputation for quality and engineering.

Look into it now because you can have the capabilities of a fully computerized operation much quicker and for much less than you ever thought.

**Cromemco**
incorporated
Microcomputer Systems
280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400



Circle 80 on inquiry card.

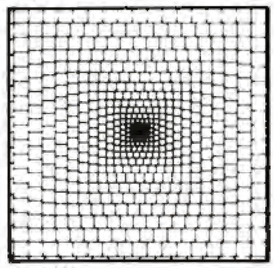
## Foreground



*page 8*

*page 34*

*page 54*



*page 105*

## Background



## Nucleus



Cover Art: AUTOMATING ECLIPSES, *by Robert Tinney*


BYTE is published monthly by BYTE Publications Inc, 70 Main St, Peterborough NH 03458. A wholly-owned subsidiary of McGraw-Hill, Inc. Address all mail except subscriptions to above address: phone (603) 924-7217. Address subscriptions, change of address, USPS Form 3579, and fulfillment questions 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 (ISSN 0360-5280). Subscriptions are $18 for one year, $32 for two years, and $46 for three years in the USA and its possessions. In Canada and Mexico, $20 for one year, $36 for two years, $52 for three years. $32 for one year air delivery to Europe. $32 surface delivery elsewhere. Air delivery to selected areas at additional rates upon request. Single copy price is $2 in the USA and its possessions, $2.40 in Canada and Mexico, $3.50 in Europe, and $4 elsewhere. Foreign subscriptions and sales should be remitted in United States funds drawn on a US bank. Printed in United States of America.





BYTE® is available in microform from University Microfilms International, 300 N Zeeb Rd, Dept PR, Ann Arbor MI 48106 USA or 18 Bedford Row, Dept PR, London WC1R 4EJ ENGLAND.

Subscription WATS Line: (800) 258-5485

Office hours: Mon-Thur 8:30 AM — 4:30 PM
Friday 8:30 AM — Noon

Robert Tinney's cover painting, "Automating Eclipses", symbolizes this month's theme of computers and weather. See this month's theme articles by Stephen P Smith and Neil Dvorak, and the editorial by Carl Helmers.

# In This BYTE

When entering large amounts of graph associated data into a computer, a graphic tablet that allows you to digitize the data is a great help. Stephen P Smith uses such a device, the Summagraphics Bit Pad digitizer, to perform the **Graphic Input of Weather Data.** *page 18*

This month "Ciarcia's Circuit Cellar" explores the use and interfacing of Texas Instruments and General Instrument sound generators. Find out how you can let your computer **Sound Off.** *page 34*

In part 1 of **A Model of the Brain for Robot Control,** James Albus defined the notation we used for his brain model. This month he describes a neurological model that can store and recall a broad class of mathematical functions. *page 54*

Much computer art employs the calculating ability of the machine to make drawings expressing mathematical relationships. Kurt Schmucker defines two classes of such drawings and describes methods for producing them in **The Mathematics of Computer Art.** *page 105*

To forecast weather, you need to know wind speed and direction. By using modern technology, we can do without whirling mechanical assemblies. Neil Dvorak shows us how to use electronic components and computer programs to measure the wind in **Sonic Anemometry for the Hobbyist.** *page 120*

In part 2 of **The Nature of Robots,** William T Powers presents a BASIC simulation of a control system. By experimenting with this simulator, the reader is able to work with the concepts of a closed loop control system. *page 134*

**Creativity in Computer Music** by Hubert S Howe Jr is a survey of some recent work in music theory, analysis, sound generation, and composition done with computers. Microcomputers can now be used for much of the work formerly done by large scale computers a decade ago. *page 158*

After you have successfully hunted the Wumpus, and destroyed all the Klingons, what is your next step? Roger Chaffee suggests you try your hand in some caves, searching for hidden treasure. Enter the world of suspense and danger on your **Quest** for riches. *page 176*

Building a computer from scratch as an amateur is the historical root of the personal computer field. In this issue, Carl Helmers begins an informal series of articles on a new homebrew project: a general purpose 6809 system. The computer itself has an intended application to music, but the design and construction of this homebrew project are quite general. See **Photo Essay: Physical Hardware of a New Computer Backplane.** *page 194*

Mouse is a programming language that contains many features usually associated with high level programming languages and can be implemented with minimal resources. It is of interest to people who enjoy obtaining dramatic results with little effort and to those who have a system which is too small to support a conventional high level language. Peter Grogono describes the implementation of Mouse by means of a Pascal program which can be used to write an assembly language version. Indications of how this might be done are provided in **Mouse: A Language for Microcomputers.** *page 198*

When working with subroutines, the concept of passing parameters can be confusing. W D Maurer describes three methods of passing parameters (call by value and result, call by reference, and call by name) in his article **Subroutine Parameters.** *page 226*

# Letters

## SERIOUS COMMENTS ON AMENDED BASIC

I enjoyed the humor that Robert Bass used in his Languages Forum on "Amended BASIC" (April 1979 BYTE, page 238). Most of his suggested additions to the BASIC language were of the April Fool's Day variety; however, his FORGET statement, though included in jest, does have some merit.

Frequently, the programs I write have hard-to-find bugs in them. These programs may have long printouts of instructions on how to use them. However, when debugging programs, it is irritating to have to wait for all of these printouts before the program really starts. I usually change all of those PRINT statements into comments by inserting a REM before them. A FORGET statement, however, would be far more convenient. The programmer could include a statement like FORGET 100-210, 320, 400-460 at the beginning, and then run the program. The BASIC will treat the statements listed in the FORGET statement as remarks. When you are finished debugging the program, remove the FORGET statement, or change it into a remark. In this way, you need to change only one statement in your program, instead of numerous statements as in our current BASIC.

Another suggestion I would like to see implemented in BASIC is a variation of the RESTORE statement. Presently the RESTORE statement sets the DATA pointer to the start of the list of DATA. However, sometimes it is convenient to have the pointer set at a different point. I suggest that a statement of the form "RESTORE 300" be implemented. This would set the DATA pointer to the first set of DATA at or following line 300. A variation of this might be "ON K RESTORE 300, 310,320." This statement would be analogous to the "ON K GOTO 300, 400, 500" statement. Both versions would allow immediate access to DATA. At present you must RESTORE the pointer to the beginning of the DATA, and then use dummy variables to READ to the DATA you actually want.

James L Boettler
Director of the Computer Laboratory
Claflin College
Orangeburg SC 29115

## SQUISH BUGS

Regarding the April 1979 editorial about operating systems with bombed file systems:

First, we note that the UCSD file system, RT-11 for the PDP-11, and many other disk operating systems require periodic squishes to manage a disk. This is a foolhardy stunt as, in your case, one bad disk sector can prevent the entire squish from working. I feel that people who build and propagate file systems like this (without even any attempt to skip bad sectors!) are irresponsible.

It shows the need for better error recovery or a scheme which prevents the need for *squish* altogether, such as dynamic file space allocations (eg: CP/M).

Secondly, your need to write your own recovery program indicated a need for such a recovery program to come as a standard operating system utility. Disasters happen; the need is real.

To my knowledge, Motorola MDOS and Software Dynamic's SDOS (for the 6800) are the only microcomputer operating systems that provide both dynamic file allocation and disaster recovery programs.

The industry needs more systems like these.

Ira Baxter
Software Dynamics
2111 W Crescent Suite G
Anaheim CA 92801

## A FASTER MAILING LIST

In reference to Thomas E Doyle's article, "A Computerized Mailing List," (January 1979 BYTE, page 84) a few modifications might be helpful, particularly, as he expresses some concern on saving time in the discussion of Program 6.

Program 7 is the main concern of this letter. After the program locates the desired record to eliminate, it performs, in closing up the gap, what is commonly called *garbage collection*. It would appear to be more desirable to flag that record as an unused record and not perform garbage collection every time. Two possible ways to do this are: a special (in Mr. Doyle's application) call sign which is recognized as a *null record*; or add an additional variable to the list for each record. This additional variable could be easily used to indicate a variety of meanings for the remaining data on the record besides a null record. Then Programs 2 and 3 could be selective.

With the addition of this variable, all the other programs would have to be modified to take into account the change in the structure of the file. The disk file could be viewed as a collection of one or more sets of records, each set being zero or more consecutive records of *good* data, and ending with one null record. Program 1, having initialized the file, would then write one null record before ending. Program 6 would only search up to the first null record after the point of insertion (a null record must still exist at the end of the file, for Program 4 as well). Program 8 (to be

*Text continued on page 98*

# Editorial

## Computers and Eclipses

by Carl Helmers

The idea occurred last fall. An innocuous advertisement appeared in *Smithsonian Magazine,* trumpeting an exciting adventure called "Eclipse Over Big Sky" which would take place in February 1979 at the Big Sky ski resort in the southwestern part of Montana, near Yellowstone Park in Wyoming. Naturally, I sent away for the information advertised.

After receiving the literature, I made up my mind that a total solar eclipse was worth seeing, especially if it was to be the last one on continental North America for some forty years. So, I sent in my deposit and made plans to attend. It turned out (as I found when I arrived) that this expedition was one of a series of such expeditions organized by sociologist and eclipse buff Dr Phil Sigler of New York City. These expeditions had attended every total solar eclipse for the past eight or ten years. Using the latest in modern techniques, including reference to weather satellite data, they had found a necessary hole in the clouds at the right time in eight out of nine cases prior to this eclipse.

Previous expeditions had used cruise ships on the open ocean in order to implement the concept of "mobility" pioneered by Dr Ed Brooks of Boston University, the weather adviser for the operation. In order to utilize the same concept for the 1979 eclipse, some form of land mobility was required. An initial attempt to take advantage of an Amtrak route which paralleled the eclipse path was apparently squelched by the usual bureaucratic catch-22: "Sure you can rent the track, but we can't supply you with a train." Thus, mobility was achieved through the services of the Yellowstone Bus Company and a procession of 15 large buses. At 2 AM on eclipse day, this procession left the hotel for a six hour trip to central Montana, just west of a town called Roundup.

Taking pictures of a transient, two minute phenomenon is one of the goals of an eclipse expedition; the other goal being to simply watch this phenomenon with the naked eye or through a suitable telescope. When I say "naked eye," I mean it, despite all normal reactions which say "you can't look at an eclipse without protection." In actuality, there is absolutely no way to look at an un-eclipsed or partially eclipsed sun without using filters to avoid damaging your eyeballs or camera equipment.

However, this is the key difference with regard to a totally eclipsed sun: you can look at it directly. That last .1 percent that separates 99.9 from 100 percent makes all the difference in the world between the dull, filtered crescent sun of a partial eclipse and the incredibly beautiful natural phenomenon of a totally eclipsed sun. You can take excellent pictures, without filters, using 400 speed film and exposures of 1/30 to 1/2000 of a second (see photos 1 and 2). But, photographing the phenomenon is definitely a bit of a problem.

During this past eclipse, I had only enough time to take about 15 exposures, with one lens change. This was done in −3 to 0 degree Celsius prevailing temperatures on an isolated road west of Roundup, Montana. My hands froze, and I probably did not get the optimal personal viewing, although the 1000 mm reflex telephoto lens of my camera acted as an excellent spotting scope through which to watch the sun for most of the eclipse.

# BEFORE YOU BUY COMPUTER #1, VISIT #1 COMPUTERLAND

If the truth is that you want a computer . . . then we want to be your computer store.

We're ComputerLand, the #1 computer store chain in the U.S. What's meaningful about that fact is, that ComputerLand has been chosen by more people as having what they've been looking for. And, since you're looking, let us tell you what you'll find, when you visit a ComputerLand store.

You'll find a product line that's continually evaluated to provide you with the widest and best selection in quality, brand name microcomputers anywhere. You'll find an enthusiastic and knowledgeable staff able to interpret all the equipment specifications, in terms of how they apply to you, and in a way you'll understand. You'll find demonstration areas where you can get a firsthand experience of running a computer yourself.

## COMPUTERS FOR BUSINESS

You'll find educational materials to give you a total insight into the world of microcomputers.

You'll find a fully equipped service department to provide whatever assistance is required to keep your computer running in top-notch condition. You'll find computer user's clubs to join, where you can share ideas with people as enthusiastic as yourself. And, with each new visit, you'll find excitement—from the people you deal with, the equipment they offer, and from your own ever-growing personal involvement.

Enough about us. How about what computers do. To attempt to describe all the things your computer might do, would be to describe your imagination. So instead, we'll briefly list some of the many things for which small computers are already being used.

**In business,** the advent of the versatile and compact microcomputer has put the benefits of computing within reach of small companies. With systems starting at less than $6000, the businessman can computerize things like accounting, inventory control, record keeping, word processing and more. The net result is the reduction of administrative overhead and the improvement of efficiency which allows the business to be managed more effectively.

## COMPUTERS FOR THE HOME

**In the home,** a computer can be used for personal budgeting, tracking the stock market, evaluating investment opportunities, controlling heating to conserve energy, running security alarm systems, automating the garden's watering, storing recipes, designing challenging games, tutoring the children . . . and the list goes on.

**In industry,** the basic applications are in engineering development, process control, and scientific and analytical work. Users of microcomputers in industry have found them to be reliable, cost-effective tools which provide computing capability to many who would otherwise have to wait for time on a big computer, or work with no computer at all.

## COMPUTERS FOR INDUSTRY

And now we come to you, which leads us right back to where we started: **If you want a computer, then we want to be your computer store.**

Whether you want a computer for the home, business or industry, come to ComputerLand first. We'll make it easy for you to own your first computer. Because, simply put, we really want your business. When you come right down to it, **that's what makes us #1.**

**ComputerLand Corp.**
**14400 Catalina St.**
**San Leandro, CA 94577**
**(415) 895-9363**
**Franchise Opportunities Worldwide.**

**ComputerLand®**
WE KNOW SMALL COMPUTERS

**ComputerLand Europe**
**Europa Terrassen**
**8 Rue Jean Engling**
**Dommeldange, Luxembourg**
**Phone 43 29 05 Telex 2423**

*Photo 1: A shot of the 1979 North American total solar eclipse taken with a relatively quick shutter speed. Note the prominences showing around the extremely dark disk of the sun as obscured by the moon. This picture was exposed shortly after totality began.*

*Photo 2: With a much slower shutter speed, elements of the solar corona begin to show. This picture was taken with about a 1/30th second exposure at f/11 using 400 ASA Ektachrome slide film. (I was too busy during the eclipse to take detailed notes for each exposure, so I can only observe that this was one of the longer exposures, probably at 1/30th second.)*

The problem is that if you spend your allotted time budget fooling around with the camera, you can miss a good portion of the event and its natural beauty. This is where the computer experimenter's inventiveness can come into play. Why not automate the exposure and picture taking sequences of the camera and telescope combination, so that once the first diamond rings of totality occur, a microcomputer can run through an open loop exposure sequence adapted to the camera equipment and the particular eclipse being viewed?

What are the functional requirements of such a device? Based on the recommendations of the expedition's photography advisers, George Keene of Eastman Kodak and Robert Little of Criterion Manufacturing (and confirmed by my own successful experience), the main requirement for achieving excellent photos with a 35 mm camera during an eclipse is to use a nonautomatic exposure technique which simply covers a range of shutter speed settings within a fixed aperture setting.

At each exposure of the film, different phenomena dominate the image. During short exposures, the extremely bright solar prominences are highlighted, with almost no corona visible (see photo 1). In longer exposures, one begins to see details of the fainter solar corona, while the inner prominence detail washes out due to overexposure.

During the transient events at the beginning ("second contact") and end ("third contact") of totality, a fixed aperture and shutter speed setting are appropriate, with a rather fast frame-to-frame timing. During totality, the film load of the camera should be spaced out over the balance of the 36 exposure magazine.

So, what we want the camera to do with its "n" exposures during the eclipse is to use a programmed sequence. The diagram of figure 1 shows a sequence that might have been ideal for me during the 1979 eclipse's 138 seconds of totality. In this figure, the events start at the last sliver of crescent sun when the filter is removed and a manual input starts the hypothetical computer sequence. Six shots are budgeted at 1/2 second intervals for the initial transient phenomenon called "Baily's beads" or "the diamond ring," depending upon the details of the sun shining through the lunar mountains.

The ideal case would then expose 24 frames at a uniform rate, covering an up and down sequence of exposure speeds. Finally, as the first bit of the departing transient starts to happen, the remaining six exposures would be used to capture the third contact "Baily's beads" or "diamond ring" effects as they occur. This would completely fill a single 36 exposure magazine of Kodak's excellent ASA 400 Ektachrome slide film. It sounds like a job for a microcomputer system as timing and control element, with suitable photographic peripherals.

How would this programmed sequence be possible? We want to use as much standard equipment as possible, for the purpose of reliability and to avoid total reinvention of the wheel. Fortunately, in contemporary photography, the motor drive is becoming an inexpensive and common accessory for the 35 mm SLR (single lens reflex) camera. This solves the problem of moving the film between frames. We need only set the motor drive on automatic and then the camera will take a picture and move the film to the next frame every time the exposure button is pushed. We must merely get the computer to push the shutter release button according to the timing diagram.

A relatively simple adaptation of a cable release to a solenoid actuator will serve to link the exposure button to the microcomputer sequencer. A suitable solid state relay power driver output from the computer will then press the button to take each picture. This, however, does not solve the problem of adjusting the shutter speed. Based on the current marketing literature of Nikon, I can get automated aperture control from an external source, but not control of the exposure time. Thus, the adaptation of my F2A camera will require careful thought and craftsmanship, of the same sort required for any other 35 mm camera body.

We will need a more elaborate combination of mechanical and electronic skills for this part of the operation. The adaptation of the camera shutter

# I've finally found a personal computer I respect. Compucolor II.

It's not surprising that professionals get excited about the Compucolor II. It's a totally-integrated 8080A system with full color graphics display, built-in 51K mini-disk drive, and the best cost performance ratio available in a personal computer.

The complete system is only $1495.* And that price includes 8K user RAM, RS-232C compatibility and random access file capabilities.

Our 8 foreground and background colors will boost your comprehension, while introducing you to an exciting new dimension in BASIC programming. The vector graphics have 16,484 individually-accessible plot blocks. And the 13" diagonal measure screen gives you 32 lines of 64 ASCII characters. You also have the flexibility that comes with 16K Extended Disk BASIC ROM.

Compucolor II offers a number of other options and accessories, like a second disk drive and expanded keyboard, as well as expandability to 32K of user RAM. Of course we also have a whole library of low-cost Sof-Disk™ programs, including an assembler and text editor.

Visit your nearest computer store for details. And while you're there, do some comparison testing. With all due respect to the others, once you see it, you'll be sold on the Compucolor II.

**Compucolor® Corporation**

Unretouched photo of screen

*U.S. Domestic price

Compucolor Corporation Post Office Box 569 Norcross, Georgia 30071 Telephone 404/449-5996

*Photo 3: A wide angle photograph showing the partially eclipsed sun in the southeastern sky, with one of several hot air balloons nearby. In order to get both the sun and the balloon, the sun is necessarily overexposed, and the balloon is underexposed.*

speed control to computer control requires machining skills with an amateur's lathe and milling setup.

The adapter is based upon a metal bracket which screws onto the camera at the tripod mounting socket which is standard on all 35 mm cameras. This bracket is set up with suitable spacers so that it will mount to the camera in a reliably repeatable fashion. The bracket can be mounted in the tripod socket since the telephoto lens or telescope used during an eclipse has its own mount on either a tripod or a true equatorial telescope base.

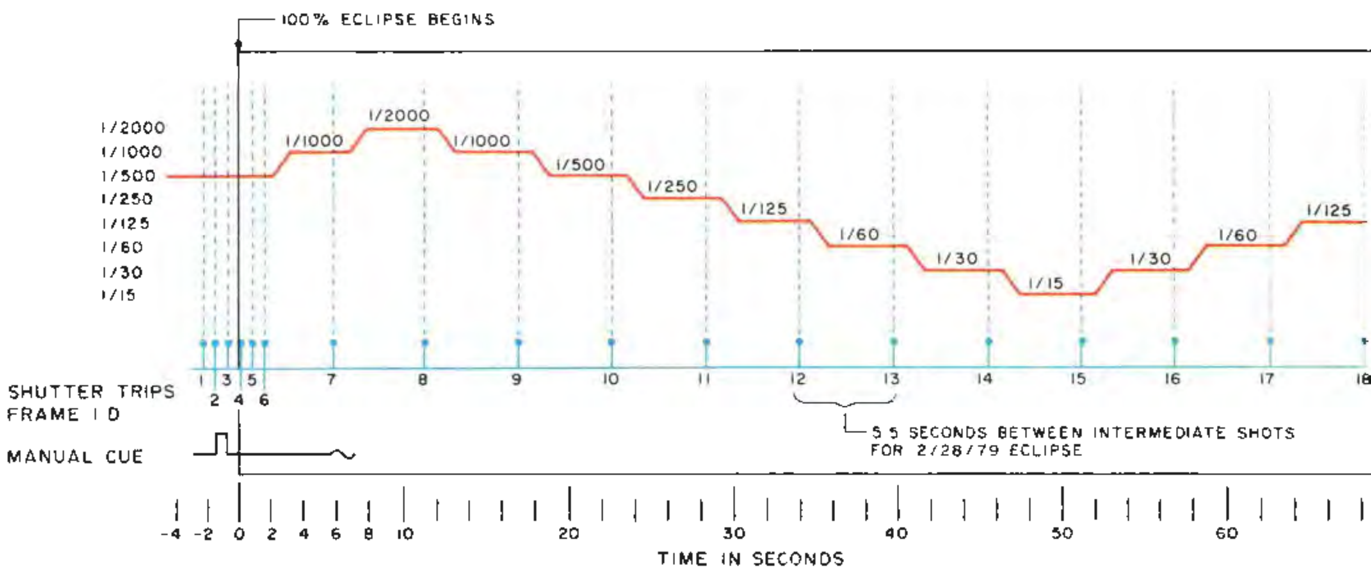
The adapter plate is then used to mount the small DC instrumentation motor and gearbox, which creates a reasonably high torque from the light duty motor. With my Nikon F2A, I need to turn the shutter speed control through an angle of approximately 10 degrees in about 1/4 to 1/2 of a second in order to accommodate the timing diagram of figure 1.

The output of the gearbox is a shaft which lines up with the axis of rotation of the shutter speed control. To this shaft is attached an optical position sensor created by passing the edge of a thin brass disk through an optoelectronic interrupter arrangement of the sort one can purchase from any optoelectronic company catalog. In the final stage of the custom fitting of this mechanism to the camera, each shutter position is marked on the disk and a small hole is punched in the disk. Thus, while the motor is turning the computer can tell when a given position has been reached, and the motor can be turned off.

The DC motor itself is controlled by a bidirectional electronic interface similar to the one shown in the article on the Terrapin Turtle by James A Gupton Jr ("Talk to a Turtle," June 1979 BYTE, page 74). This bidirectional interface allows us to turn the shutter speed knob to any setting, with the sensing of the shutter position returned by the optical interrupter. Use of a second interrupter for encoding of the first and last shutter speed settings will guarantee proper initialization and referencing of the speeds. This provides direct feedback of the limit stops in addition to the intermediate position information.

The one critical, unsolved problem in projecting this setup for my camera is the detail of driving the shutter speed control from the output of the gearbox. I will probably have to consider some potentially disastrous modifications to the camera. One possible method could be a tight fitting, carefully milled cylinder with ridges on its inner surface that would mate with the ridges on the shutter speed control. Another possibility would be to drill one or more off-center drive holes that would receive a driving pin mounted off-center on the drive shaft's end. The latter might be impossible, due to the complicated nature of the shutter speed control and its interaction with the film speed setting mechanisms.

*Figure 1: A timing diagram of an "ideal" 36 exposure sequence for a 138 second eclipse event, allowing two seconds before and after totality for transient phenomena. The horizontal axis of this figure is time in seconds, and the vertical axis is shutter speed of the camera, a discrete phenomenon with the steps shown. During an eclipse, aperture setting of the camera cannot be controlled if long focus telephoto or astronomical telescope lens equipment is used.*



The computer system which drives the camera during an eclipse is quite simple. The computer itself should be a dedicated 8 bit device with a suitable high level language program loaded into its local read only memory. Several 2708 or 2716 read only memory parts should suffice to store the systems software and the application program needed to control the telescope camera during a specific eclipse and to space out the 24 intermediate pictures during the estimated length of totality. Power requirements can be adequately handled by a single 6 or 12 V battery which also supplies power to the shutter speed control motor and the shutter tripping solenoid. The camera motor drive has its own dedicated NiCad battery pack which is totally independent of the computer. In a field situation it is assumed that batteries can be recharged through 110 V AC mains on the cruise ship or in land based hotels. If North American standard voltages are not available, the chargers can certainly be run through one of a number of standard converters available for world travelers.

*Photo 4: One of the most unusual occurrences was the appearance of several hot air balloons in the sky at the time of the eclipse. While the balloonists must certainly have been having fun, some of the people watching from the ground were, no doubt, perturbed. Here is a kind of man made "cloud" phenomenon totally unpredictable by any meteorologist.*

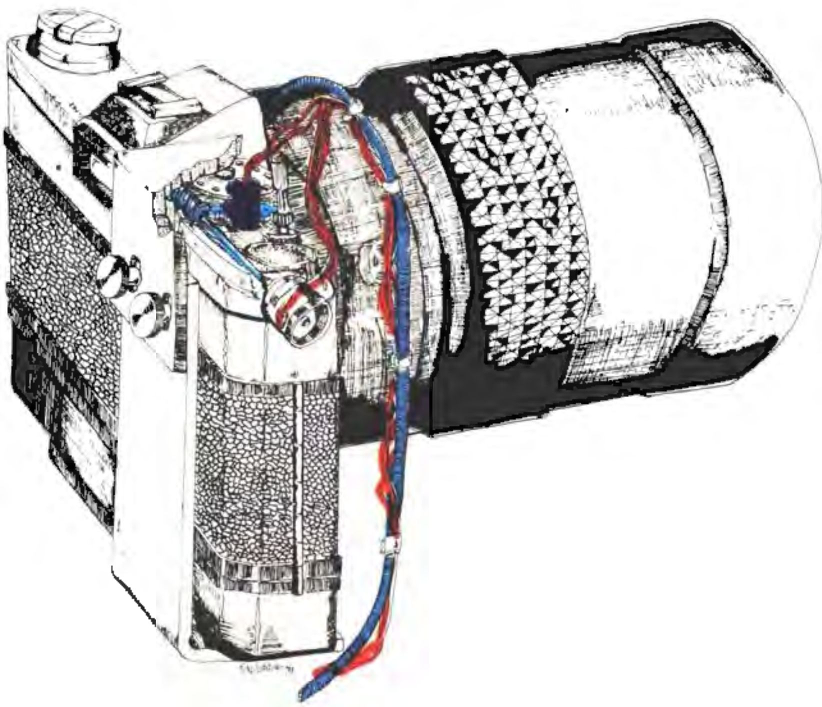
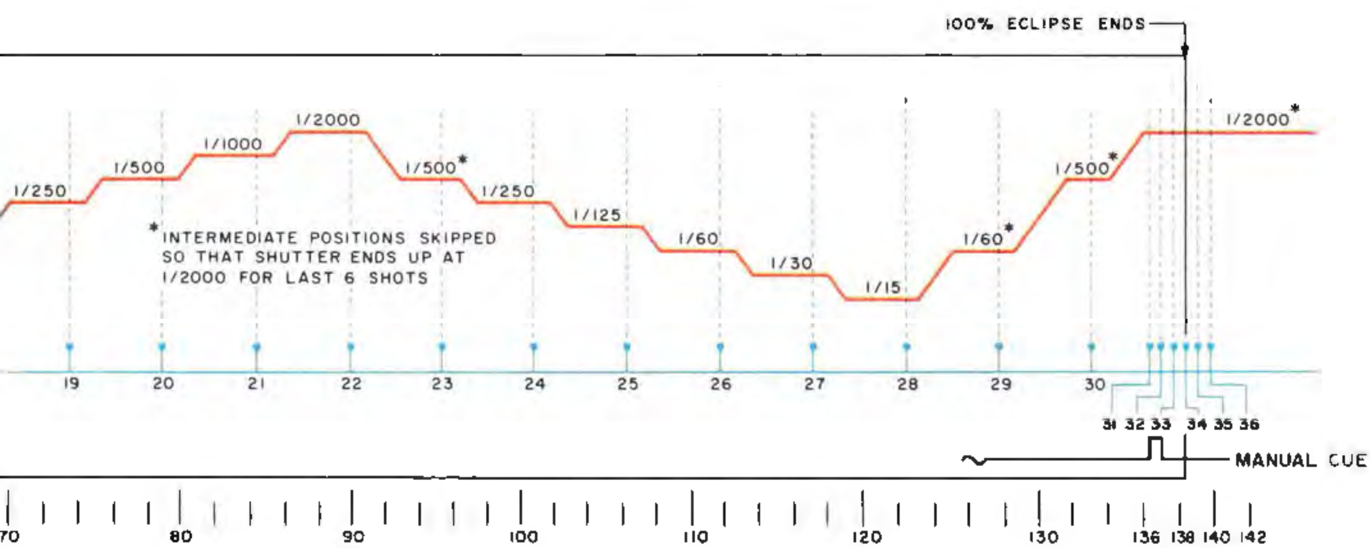


*Figure 2: A conceptual sketch of possible homebrew machinery adapted to a 35 mm camera with motor drive, to allow automation of eclipse photography according to the timing diagram in figure 1. This fantasy was created by artist Ken Lodding.*



**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 large (9 by 12 inch, 30.5 by 22.8 cm), self-addressed envelope, with 28 cents US postage affixed, 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.* ■

Will I ever build this? At this time I can't predict if and when I will get around to building this sort of system. If I do, readers can be certain that there will be photographic documentation of the system. My immediate deadline might be to get the system working for the 1980 eclipse which occurs over the equatorial Atlantic Ocean, Africa, Indian Ocean, India and China on February 16. However, as this is written, I don't even know if I will go to see that event.

This camera automation computer is one of those applications of a small computer system which is most appropriate. It has elements of the mechanical interfaces to electronics which are a necessary part of any practical robotic system, as well as elements of real time control akin to those needed for other practical uses of the small computer in home, laboratory and industry. It is the kind of system many of our readers are conceiving and building, whether it be for fun or for professional purposes. As time goes on, we can expect to see this kind of application documented in the form of articles with much greater detail than this editorial sketch. Conceiving, and then building this kind of application is when the fun of contemporary small computing reaches its highest level. ■

**The More Things Change, The More They Stay The Same. . . .**

On April 20, 1979, BYTE Publications Inc and onComputing Inc became a part of McGraw-Hill Publications Co. Thus, as we neared completion of our fourth year as an enterprise, BYTE, and *onComputing* magazines joined *Electronics*, *Aviation Week and Space Technology*, and *Data Communications* to become key parts in a group of high technology magazines published by McGraw-Hill.

BYTE will continue to be published from offices in Peterborough NH, with the same staff and the same dedication to quality. Aside from such detail changes as the notation "A McGraw-Hill Publication" on our cover, readers can expect the editorial and advertising content of BYTE to continue under the same philosophy which has established our reputation in the past. Indeed, a major factor in our decision to affiliate with McGraw-Hill is their commitment to the independence of individual magazines.

An interesting statistic is that at this exciting time, BYTE's paid circulation of about 156,000 readers (May 1979 issue) makes it second only to *Business Week* in paid circulation among the more than thirty magazines published by McGraw-Hill.

We look forward at this point to a long and flourishing relationship with the people who form the McGraw-Hill enterprise.

. . . .Carl Helmers

A Note About the Cover. . .

One of the interesting social phenomena of this eclipse was the appearance of a number of artificial clouds over the observation site: five or six different hot air balloons appeared over our site just at the time of totality. Photo 3 shows a wide angle shot that was intentionally overexposed during the partial phase of the eclipse just prior to totality. One of these artificial clouds is a dark object below and to the left of the sun in this picture. The telephoto shot shown in photo 4 captured one of these balloons in the sky to the west of the expedition site as they were drifting towards us, about 10 minutes before totality.

At the time of totality, I distinctly remember looking up and seeing two bright objects in the sky. One object was the eclipsed sun, and the second object, at about the same position as the dark balloon in photo 3, was one of the balloons with its propane flame shining a brilliant orange color. A man-made fire was complementing the eclipsed embers of the sun.

Combining the eclipse automation theme of this month's editorial with the hot air balloons actually observed, and the weather analysis and measurement themes of two of this month's articles, artist Robert Tinney has created a fantasy on eclipses, hot air balloons and weather for this month's cover. The dramatic effect of cumulonimbus thunderstorm clouds was used in place of the rather dull, high, thin cirrus cloud layer which partially obscured the 1979 eclipse as viewed from central Montana. And perhaps the hot air balloonists should have their heads examined for departing into this imagined thunderstorm, inexorable timing of an eclipse or not. But the resulting oil painting is an incomparable work of beauty, celebrating an uncommon event unique to our spaceship earth, its sister planet the Moon and a technological civilization.

# DOUBLE DENSITY


# SOLID SAVINGS!

Now you can put your S-100 system solidly into a full-size, single/double density, 600K bytes/side disk memory for just $1149 complete.

DISCUS/2D™ single/double density disk memory from Thinker Toys™ is fully equipped, fully assembled, and fully guaranteed to perform perfectly.

DISCUS/2D™ is a second generation disk memory system that's compatible with the new IBM System 34 format. The disk drive is a full-size Shugart 800R, the standard of reliability and performance in disk drives. It's delivered in a handsome cabinet with built-in power supply.

The S-100 controller utilizes the amazing Western Digital 1791 dual-density controller chip . . . plus power-on jump circuitry, 1K of RAM, 1K of ROM with built-in monitor, and a hardware UART to make I/O interfacing a snap.

The DISCUS/2D™ system is fully integrated with innovations by designer/inventor George Morrow. Software includes BASIC-V™ virtual disk BASIC, DOS, and DISK-ATE™ assembler/editor. Patches for CP/M* are also included. CP/M*, MicroSoft Disk BASIC and FORTRAN are also available at extra cost.

DISCUS/2D™ is the really solid single/double density disk system you've been waiting for. We can deliver it now for just $1149. And for just $795 apiece, you can add up to 3 additional Shugart drives to your system. Both the hardware and software are ready when you are.

Ask your local computer store to order the DISCUS/2D™ for you. Or, if unavailable locally, write Thinker Toys,™ 5221 Central Ave., Richmond, CA 94804. Or call (415) 524-2101 weekdays, 10-5 Pacific Time. (FOB Berkeley. Cal. res. add tax.)

*CP/M is a trademark of Digital Research.

*Morrow makes disk memory for*

**Thinker Toys™**

# Graphic Input of Weather Data

Stephen P Smith
106 E Clearview Av
State College PA 16801

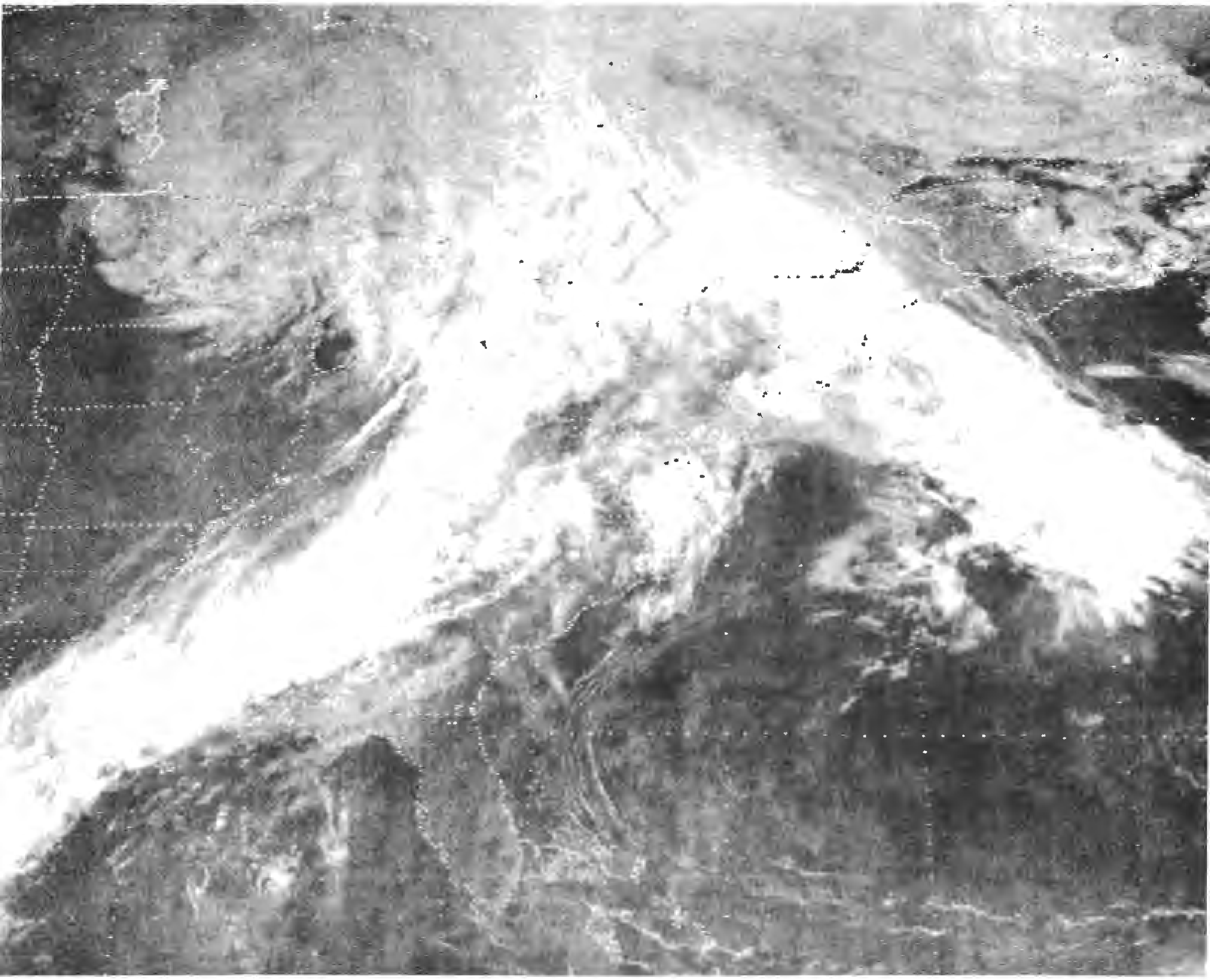


*Photo 1: Infrared and visible light photographs like this one are the primary tool of the satellite meteorologist. Cloud formations help locate rain. Temperature data from the infrared images indicates intensity. Photograph courtesy NOAA National Weather Service.*

The fact that everyone complains about the weather, but nobody does anything about it, is well-known. Weather forecasting is still more art than science.

Manual techniques still yield the best quantitative weather predictions when compared with the largest computer systems processing a wealth of satellite, radar, and ground station data. However even modest data processing equipment can be an important *tool* for the meteorologist. I'll show how the combination of a small personal computer and a Summagraphics Bit Pad graphics tablet simplifies the processing of rainfall estimates for a regional data base. The application is an interesting one, and the BASIC language software developed will be useful in any system employing a Bit Pad for data entry.

Locally, rainfall can be measured with simple gauges. The heavy showers common during the prime growing and flood seasons have irregular distributions, so local measurements may be inadequate for regional use. Agricultural planners need to know how much rain has fallen over a specific growing area. Hydrologists working on flood warning and control need to know how much has fallen within a given watershed. Both groups need this information broken down into relatively small elements of time and area, perhaps for each 24 hour period and for each 10 kilometer square. To achieve this detail, tools in addition to rain gauges must be used.

The first of these tools is ground based radar. Most of us have seen weather radars operating on television news broadcasts. Rainfall reflects the radar signal and provides a visual display for the operator, similar to figure 1. Showers can be located accurately, and relative intensity can be determined. Unfortunately, even highly calibrated radars have difficulty measuring actual amounts of rain, and most weather radars are not well calibrated for this application. Radar coverage is also not complete over all areas of the country.

A second tool, satellite imagery, has extended that coverage significantly (see figure 2). Geostationary satellites, which remain fixed over one point on the earth, provide pictures every half hour. Polar orbiting satellites, flying much closer to the earth, provide more detailed images several times a day. A trained meteorologist can identify cloud formations in pictures like photo 1. Several investigators primarily at the National Oceanic and Atmospheric Administration (NOAA, pronounced like Noah) have developed schemes to estimate the rainfall beneath these clouds. (see references 1 and 2).

*Text continued on page 20*

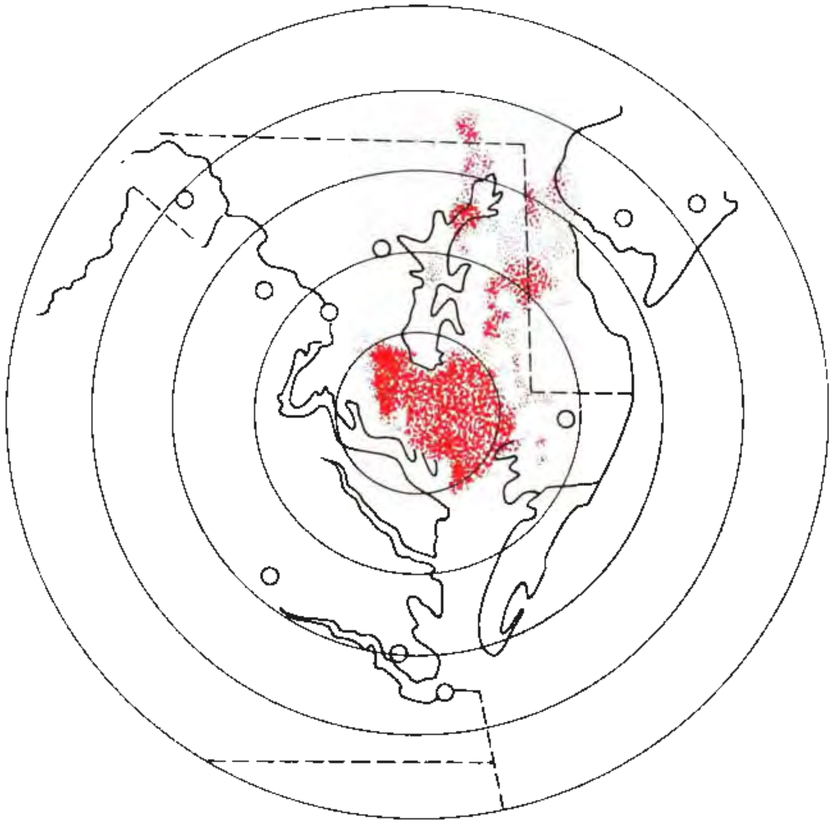


*Figure 1: Meteorologists can locate rainstorms using ground based radar. Images like this are familiar from television news. Unfortunately, they are not sufficient to gauge intensity, and radar coverage is not complete throughout the country.*

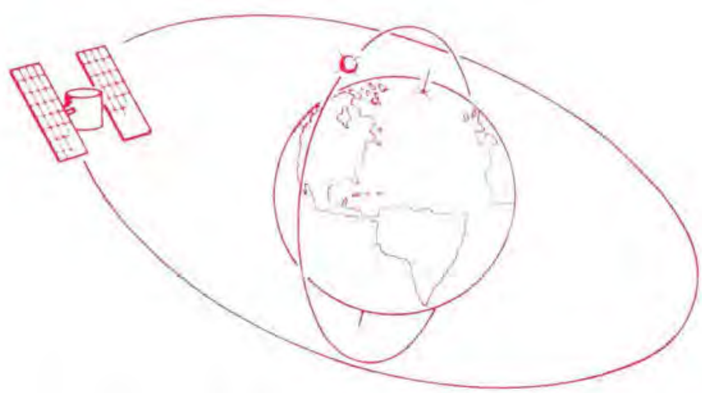


*Figure 2: Meteorologists can obtain imagery from two types of satellites. Geosynchronous satellites orbit at the same speed that the earth turns. They appear to be stationary, giving continuous coverage of one area of the Earth. Polar orbiting satellites fly a North to South pattern while the earth turns west to east below them. They provide frequent coverage of every point on the globe.*

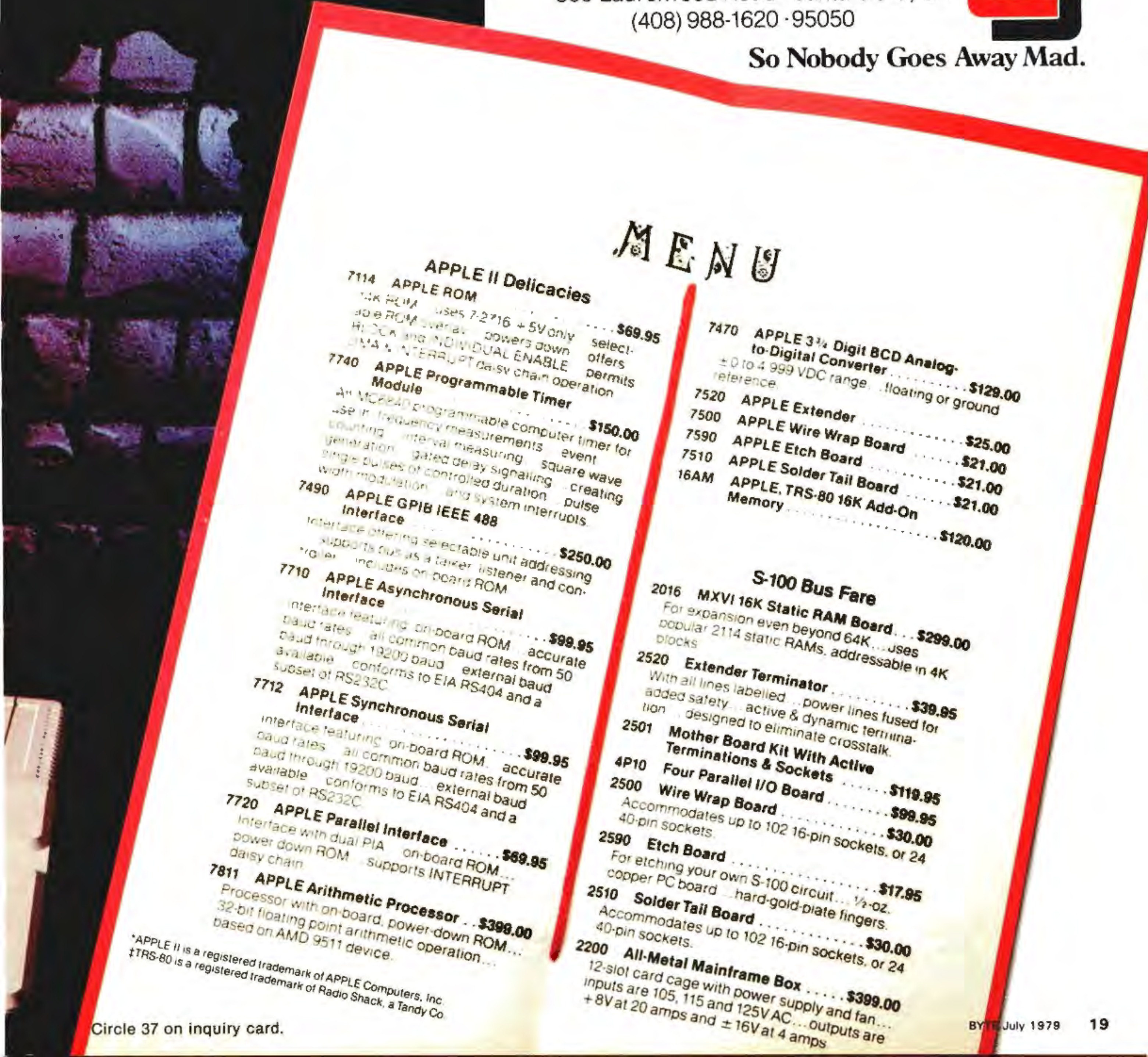
# We've worked like mad to

# cook up all your favorites.

Now, we're cooking. Our boys in the lab have turned circuit chefs these past three months to create a *smorgasbord* of deliciously assembled boards to support your APPLE II*, TRS-80‡ or S-100 bus systems. Feast your eyes on our monstrous selection in the menu below. Then, order enough to satisfy your hunger for experimentation for months to come. Contact your local computer store, or call us direct. *Bon appetit!*

**California Computer Systems**
309 Laurelwood Road · Santa Clara, CA
(408) 988-1620 · 95050

**So Nobody Goes Away Mad.**




Circle 37 on inquiry card.

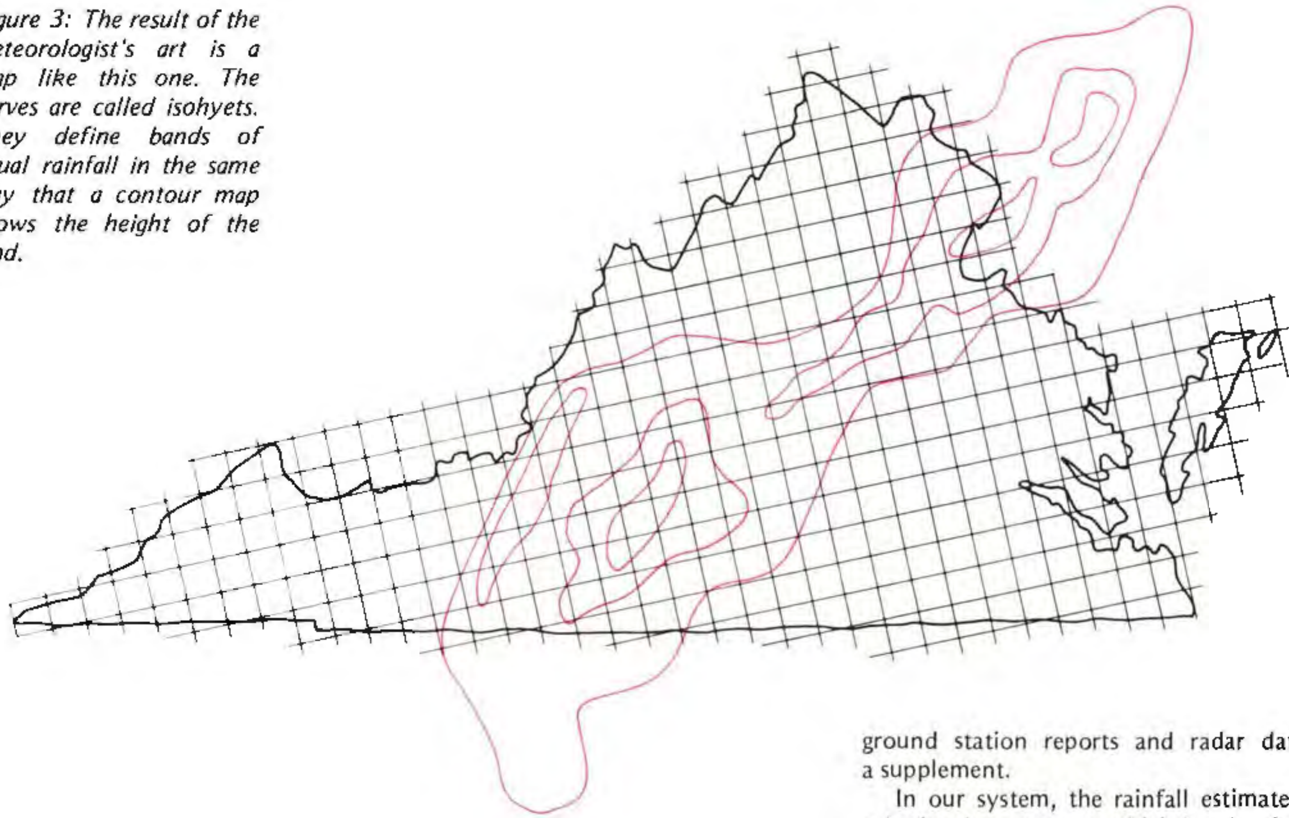


*Figure 3: The result of the meteorologist's art is a map like this one. The curves are called isohyets. They define bands of equal rainfall in the same way that a contour map shows the height of the land.*

*Text continued from page 17:*

The job of tying together the satellite information with ground station and radar data quickly falls to computers. The manual work should be made as simple as possible, with data provided to the computer in a format designed for the convenience of the meteorologist. The computer should accept this data, digitize it, manipulate it, store it, and produce reports in a format designed for agricultural planners, hydrologists, and other users. Programs of this type have typically been run on medium sized mainframe computers. For data entry and conversion, and limited report generation, however, there are a lot of good microcomputer applications.

Let's examine a specific example. We will accept rainfall estimates and process them to produce reports of accumulation in each 10 km square over a region of about 150,000 km$^2$ (ie: 30 by 50 squares). It would be nice if a digitized satellite photograph could be input to a computer to generate rainfall estimates directly, but that is well beyond the current state of the art. Attempts have been made to input manual interpretations of cloud formations seen in the photos and have the computer evaluate their rain potential (see reference 3). This, too, lacks the necessary precision. The best estimates come from the evaluations of an experienced meteorologist working with satellite photos, and using ground station reports and radar data as a supplement.

In our system, the rainfall estimates are submitted as maps on which bands of equal rainfall will be drawn. Figure 3 shows a sample. The bands are called *isohyets* and are similar to the *isobars,* or lines of equal barometeric pressure, also used by weather people.

A more familiar analogy might be a contour map. If locations on the ground are identified by XY coordinates, the rainfall rate can be thought of as a Z coordinate analogous to height at that point. Converting the isohyets to XYZ coordinates compatible with a grid of ten kilometer squares is the prime function of our processing system.

For a few locations, this could be done manually. The XY point could be located on the map and entered at a terminal along with the value of the isohyet in which it fell. When the number of points runs into the hundreds, however, and the data must be entered at half hour intervals, the manual approach becomes unworkable. A technique for rapidly entering the isohyets, automatically converting to XYZ format, and summing the entries over time must be devised.

Isohyets can be entered directly from the maps using a device called a graphic tablet. The map is placed on the tablet and a stylus is used to trace the outline of each isohyet. The tablet senses the position of the stylus, and the signals are electronically transmitted to a controller and converted to XY data. This type of equipment can resolve positions

| Next Byte | Byte Received | Mode 2 | Mode 1 | Rate 4 | Rate 2 | Rate 1 | Status Valid |
|---|---|---|---|---|---|---|---|

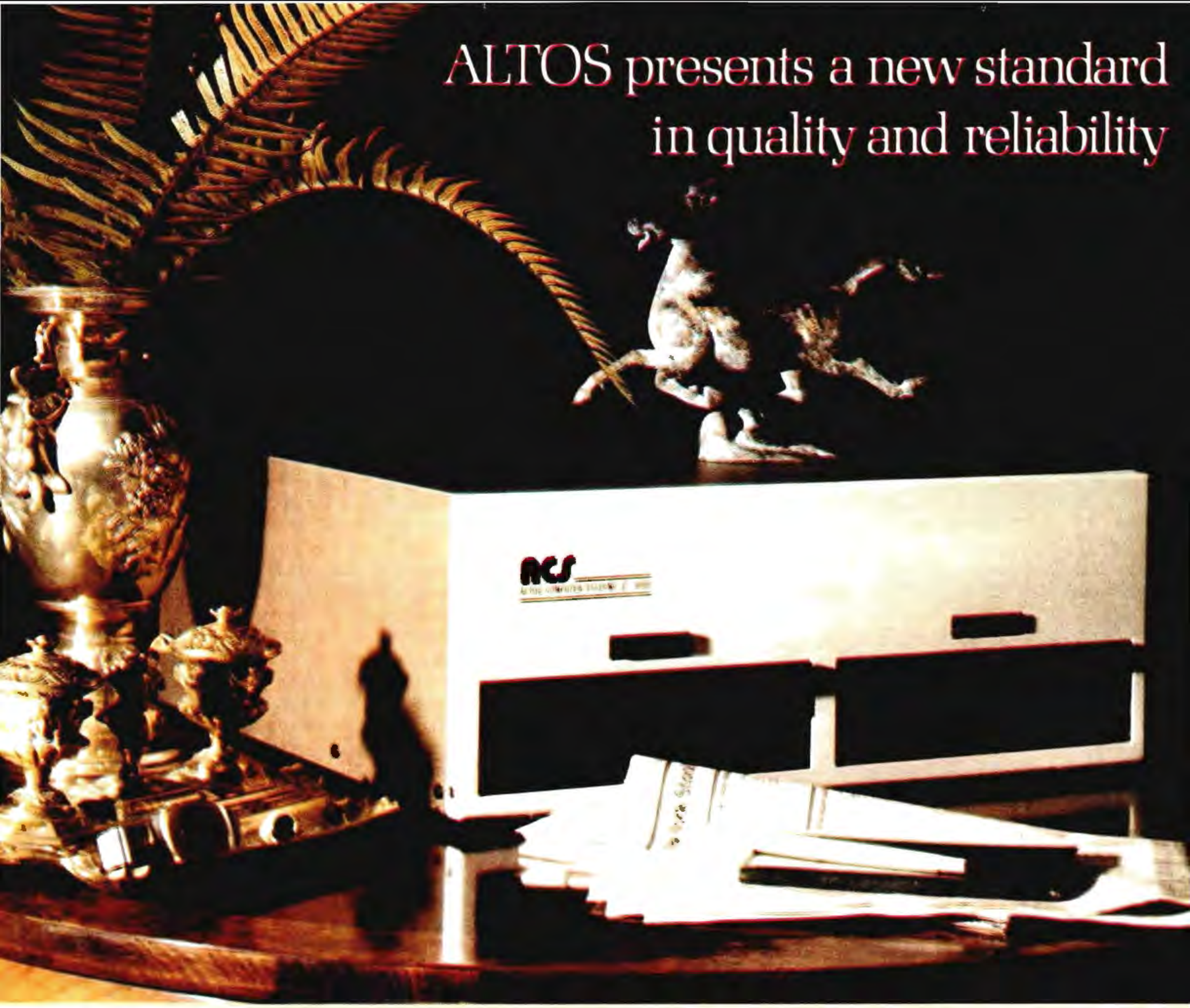
*Figure 4: The computer usually talks to the Bit Pad in parallel mode. One 8 bit word is used to control the Bit Pad. The two most significant bits are part of the communications handshake. The next five provide software control, duplicating switches on the controller. The last bit,* status valid, *should be set when transmitting a command.*

to 0.1 millimeter and can enter data as fast as the operator can trace the lines. Until recently, graphics tablets were expensive, typically $5000. With the introduction of the Bit Pad from Summagraphics, however, the price is now within the range of the personal computer experimenter for scientific and business applications. As of this writing, a complete unit with a tablet, stylus and controller can be purchased for $555.

Using the Bit Pad for our rainfall application involves a little hardware and some software to interface the graphics tablet to my Ohio Scientific computer. The hardware interface has largely been taken care of by Summagraphics. The software problem is a matter of scaling and accounting for misalignment of the maps during the digitizing process. The same techniques will apply to any Bit Pad application on an 8 bit microcomputer. Although BASIC may be too slow for some applications, I'll use it to illustrate this application, so the concepts will be available to the widest variety of users.

Before dealing with code, let's look at the hardware interface. The Bit Pad uses 8 bit parallel input and output as its standard. Serial communication at TTL (transistor-transistor logic) or RS-232 voltages is available as an option. With the option, it may be possible to place the Bit Pad between your serial terminal and computer in much the same way that a SwTPC cassette interface is installed. I have a parallel port based on a 6520 PIA (peripheral interface adaptor) on my central processor board, so I elected to use the parallel format. The physical connection consists of a cable with a DB-25P plug on each end. One end plugs into the Bit Pad's controller. The other connects to the PIA lines brought out to my computer's back panel. The installation could hardly be simpler.

The 8 input and 8 output signals flow along the cable. The Bit Pad receives commands from the computer in the format shown in figure 4. Each XY point is trans-




**WE'RE ALTOS COMPUTER SYSTEMS. Our SUN-SERIES ACS8000 business/scientific computer creates a new standard in quality and reliability in high technology computers.**

**HIGH TECHNOLOGY** The ACS8000 is a single board, Z80®* disk-based computer. It utilizes the ultra-reliable Shugart family of 8 inch, IBM compatible, disk drives. A choice of drives is available: single or double density, single or double sided. Select the disk capacity you need, when you need it: ½M, 1M, 2M, or 4M bytes. The ACS8000 features the ultimate in high technology hardware: a fast 4 MHz Z80 CPU, 64 kilobytes of 16K dynamic RAM, 1 kilobyte of 2708 EPROM, an AMD 9511 floating point processor, a Western Digital floppy disk controller, a Z80 direct memory access, Z80 Parallel and Serial I/O (two serial RS232 ports, 1 parallel port), and a Z80 CTC Programmable Counter/Timer (real time clock). In essence, the best in integrated circuit technology.

**BUILT-IN RELIABILITY** The ACS8000 is a true single board computer. This makes it inherently reliable and maintainable. The board and the two Shugart drives are easily accessible and can be removed in less than five minutes. All electronics are socketed for quick replacement. Altos provides complete diagnostic utility software for drives and memory.

**QUALITY SOFTWARE** Unlimited versatility. The ACS 8000 supports the widely accepted CP/M®** disk operating system and FOUR high level languages: BASIC, COBOL, PASCAL and FORTRAN IV. All available NOW.

**PRICE** ACS 8000-1, single density, single-sided [½ Mb] $3,840
ACS 8000-2, double density, single-sided [1 Mb] $4,500
ACS 8000-3, single density, double-sided [1 Mb] $4,800
ACS 8000-4, double density, double-sided [2 Mb] $5,300

Brackets show disk capacity per standard two drive system. All models come standard with 32 Kb RAM and two 8" disk drives as shown above. Expansion to 64 Kb is $363 per 16 Kb. FPP, DMA, software optional. Dealer/OEM discounts available. Delivery: 30 days ARO, all models.

*Z80 is a trademark of Zilog, Inc.
**CP/M is a trademark of Digital Research, Inc.

Circle 6 on inquiry card.

ALTOS
COMPUTER SYSTEMS 2338A WALSH AVENUE • SANTA CLARA • CA 95050 • (408) 244-5766

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Status (1) | Byte Available | F3 | F2 | F1 | F0 | 0 | 0 |
| Data (0) | Byte Available | X5 | X4 | X3 | X2 | X1 | X0 |
| Data (0) | Byte Available | X11 | X10 | X9 | X8 | X7 | X6 |
| Data (0) | Byte Available | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |
| Data (0) | Byte Available | Y11 | Y10 | Y9 | Y8 | Y7 | Y6 |

*Figure 5: The Bit Pad communicates with the computer in parallel mode. Five bytes are used to transmit data to the computer. The first two bits of each are handshaking signals. The first byte contains status information. The bit labeled F0 is set whenever the stylus is in contact with the tablet. F1, F2 and F3 are used only with an optional cursor. The remaining four bytes hold data. The second and third provide a 12 bit X coordinate and the fourth and fifth provide a 12 bit Y coordinate. Both are measured in absolute units (0.1mm or 0.005 inches) from the lower left corner of the tablet.*

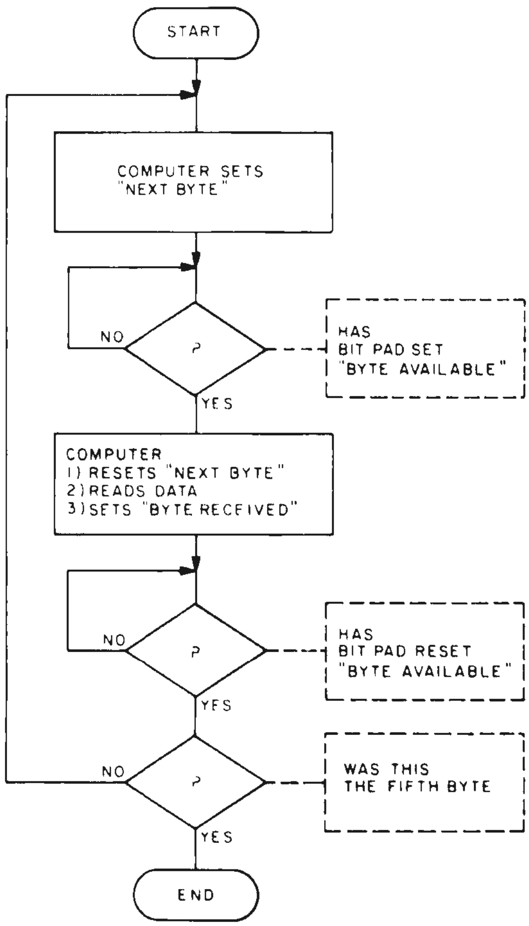



*Figure 6: A handshaking arrangement insures proper data transfer between the Bit Pad and your computer. The Bit Pad performs its part of the procedure automatically. A simple BASIC or machine language routine will handle the computer's end.*

mitted as a sequence of five bytes, as shown in figure 5. A ninth output bit is generated as a data strobe and can be used for interrupt driven software. Using this strobe, interaction between the Bit Pad and the computer could become largely a hardware function. In order to present that interface, however, a lengthy discussion of the PIA interrupt handling and control line features would be needed. That would be beyond the scope of this article and is of little practical use to experimenters with 8080 and Z-80 systems. The more general interface I'll develop here uses a software handshake illustrated in figure 6. It can be handled with a BASIC program and can be implemented as easily on the Motorola type 6820 and 6520 PIAs or the 8080/Z-80 compatible 8255 programmable peripheral interface or 8212 I/O (input/output) port.

The computer begins the handshake by setting the *next byte* bit of the command word. In BASIC, this is accomplished by using the POKE command to put the value 128 (binary 1000000) in the PIA output register. This is memory location 63488 in my system. When the Bit Pad sees the *next byte* command, it places a data word on the output lines, setting the *byte available* bit. The computer has been looking for this by executing an appropriate wait command.

Now it reads the data, sets *byte received* and resets *next byte*. The Bit Pad acknowledges by resetting *byte available*. The computer is now certain that a valid byte was read and that the Bit Pad is ready for a new sequence. If this was the fifth byte of a sequence, processing can continue. If not, *next byte* is set and the entire process is repeated. Should the computer somehow miss a byte or lose track of the count, it can recover by monitoring the *first byte* bit of each data word. It will be set (ie: have value 1) on the first of the five bytes and be reset (ie: value 0) on each of the others. The Bit Pad holds up its end of the handshaking automatically. The computer's end can be handled by the BASIC subroutine beginning at line 1100 in listing 1.

Using this subroutine, we end up with five values between 0 and 255 in array D. The status information in the first byte may be ignored for now. D(1) and D(2) contain the X location of the data point, but they also contain the *byte available* and *first byte* handshake signals (always set and reset respectively for data bytes). We must mask them out using a logical AND operation, or, in BASIC, by subtracting 64 (binary 01000000). The high order portion of X in D(2) is now multiplied by 64 ($2^6$) and added to D(1). The result is X, the absolute dis-

# MEET MINIMAX.™
## THE MOST ADVANCED, LEAST EXPENSIVE COMPUTER EVER CREATED.

Minimax has the largest memory (108k internal, 2.4 megabytes external) and the fastest data transfer rate (2 seconds per 30k characters) of any microcomputer in the world.

Minimax also has the most advanced full screen data entry and editing capabilities plus the highest resolution graphics (240x512).

The Minimax CPU features 64 user definable opcodes and ROM resident software complete with Microsoft Basic, DOS operating system, FIFTH language interpreter plus a complete machine language monitor with Tiny Assembler, Disassembler, Dump Facility and Debugging Aid.

And Minimax has a software library unmatched in the industry.

If all this sounds unbelievable, just check out the price: $4495 for the 800 kilobyte Minimax I and $5995 for the 2.4 megabyte Minimax II.

Now that's unbelievable.

Write or call for complete ordering information today. Dealer inquiries invited.

**MINIMAX**™

Compu/Think™
3260 Alpine Road, Menlo Park, CA 94025 (415) 854-2577

Circle 71 on inquiry card.

```
10    REM    DATA ENTRY ROUTINE USING SUMMAGRAPHICS BIT PAD
20    REM    WRITTEN IN MICROSOFT 6502 BASIC
30    REM    STEPHEN P. SMITH, STATE COLLEGE, PA
40    DIM D(4),EW(20,20),NS(20,20),M(10)
50    P1=63844:                        REM ADDRESS OF PARALLEL PORT
60    REM
1000  REM    GET A POINT FROM THE BIT PAD
1010  GOSUB  1100:                     REM FETCH A BYTE
1020  IF D1<128 GOTO 1010:             REM IS IT THE FIRST OF 5
1030  FOR I=1 TO 4:                    REM IF YES, READ NEXT 4
1040  GOSUB 1100:                      REM FETCH A BYTE
1050  D(I)=D1-64:                      REM STRIP THE HANDSHAKE BIT
1060  NEXT I
1070  X=D(2)*64+D(1):                  REM FIND THE ABSOLUTE X POSITION
1080  Y=D(4)*64+D(3):                  REM FIND ABSOLUTE Y POSITION
1090  RETURN
1100  REM    HANDSHAKE ROUTINE
1110  POKE P1, 128:                    REM SET "NEXT BYTE"
1120  WAIT P1+2,64,0:                  REM LOOK FOR '"BYTE AVAIL" SET
1130  D1=PEEK(P1+2):                   REM READ DATA
1140  POKE P1,64:                      REM RESET "NEXT BYTE", SET "BYTE RCVD"
1150  WAIT P1+2,64,64:                 REM LOOK FOR "BYTE AVAIL" RESET
1160  RETURN
3000  REM    SET UP AUTOSCALING
3010  GOSUB 1000:                      REM GET X,Y, POINT A
3020  AX=X: AY=Y
3030  GOSUB 1000:                      REM GET X,Y, POINT B
3040  BX=X: BY=Y
3050  REM
3060  DATA 0,0,30,50:                  REM A & B IN GRID COORDINATES
3070  READ AG,AH,BG,BH
3080  S1=SQR((BG-AG)↑2+(BH-AH)↑2):     REM DISTANCE AB IN GRID SYSTEM
3090  S2=SQR((BX-AX)↑2+(BY-AY)↑2):     REM DISTANCE AB IN BIT PAD SYSTEM
3100  S=S1/S2:                         REM SET SCALE FACTOR
3110  T1=ATN((BY-AY)/(BX-AX)):         REM ANGLE IN BIT PAD SYSTEM
3120  T2=ATN((BH-AH)/(BG-AG)):         REM ANGLE IN GRID SYSTEM
3130  DT=T2-T1:                        REM ROTATION ANGLE
3140  DX=AX-(AG*COS(DT)-AH*SIN(DT))/S: REM X TRANSLATION
3150  DY=AY-(AH*COS(DT)+AG*SIN(DT))/S: REM Y TRANSLATION
3160  RETURN
3190  REM
3200  REM    CONVERT ABSOLUTE X,Y TO GRID COORDINATES
3210  REM     J IS NUMBER OF ISOHYET
3230  REM     K IS NUMBER OF THIS POINT
3240  REM
3250  X=X-DX:                          REM TRANSLATE X
3260  Y=Y-DY:                          REM TRANSLATE Y
3270  X1=X*COS(DT)-Y*SIN(DT):          REM ROTATE X
3280  Y1=Y*COS(DT)+X*SIN(DT):          REM ROTATE Y
3290  EW(J,K)=X1*S:                    REM SCALE X AS EAST-WEST COORDINATE
3300  NS(J,K)=Y1*S:                    REM SCALE Y AS NORTH-SOUTH COORDINATE
3310  RETURN
4000  REM    MENU BOARD ROUTINE
4010  REM
4020  DATA 0.,.01,.02,.03,.04,.05,.06,.07,.08,.09,.10
4030  FOR I=1 TO 10:                   REM INITIALIZE 10
4040  READ M(I):                       REM VALUES FOR MENU
4050  NEXT I
4060  GOSUB 1000:                      REM FETCH AN X,Y POINT
4070  IF X>2794/20 GOTO 4060:          REM USE LEFTMOST 5% OF TABLET
4080  N=Y/2794*10:                     REM MENU HAD 10 ELEMENTS
4090  R=M(N):                          REM Y POSITION SELECTS ONE
4100  RETURN
```

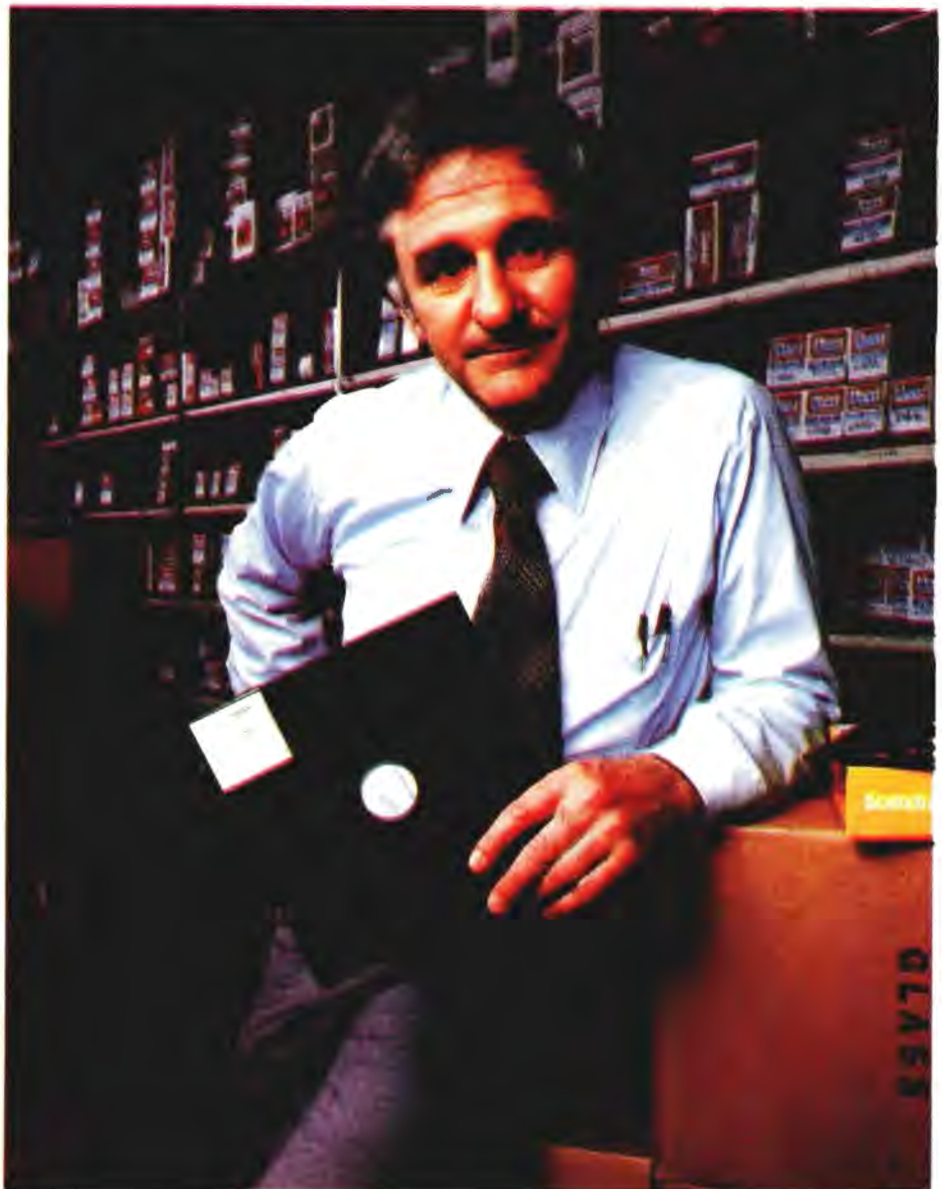
*Listing 1: BASIC program to use the Bit Pad for entering data. This program inputs a point using the discussed handshaking method, automatically scales and rotates the point, and then allows you to perform operations using that point.*

tance from the left side of the tablet. D(3) and D(4) are processed in the same way to calculate Y, the absolute distance from the lower edge of the tablet. The necessary subroutine begins at line 1000 of listing 1.

The absolute coordinates in the Bit Pad system run from 0 to 2794 for 0.1 millimeter resolution (0 to 2000 for .005 inch resolution). Given the scale of our map and assuming it is well aligned on the tablet, it should be a trivial matter to convert X and Y to our 10 kilometer grid coordinate system. Actually, a little software can be added to take care of the scale and map alignment, too. We begin each session by digitizing two known points on the map. The distance between them gives us the scale factor. Their relative orientation tells the system how the map is positioned on the tablet. Figure 7 and caption give a general presentation of coordinate transformations. The subroutine which sets up this transformation begins at line 3000 of listing 1. At line 3200 absolute X and Y values are converted into grid coordinates and stored in a pair of arrays.

Using BASIC, the data is entered by touching the stylus to a number of points around each isohyet. The smooth curves on the map are approximated in the computer by polygons as in figure 8. A larger number of points produces a better approxi-

# "Our inventory is our existence. Think we'd trust it to anything less than Scotch® Brand Diskettes?"



**Don Stone, Vice-President, Mass. Auto Supply Company, Inc., Boston, Mass.**

Scotch Diskettes are the diskettes you can depend upon with the information *your* business depends upon.

Each one is tested and certified error-free before it leaves our factory. Because we know nothing less than perfection is acceptable for *your* vital business data.

Scotch Diskettes are available in regular or mini sizes, compatible with almost any system.

To find out where you can find Scotch Diskettes, call toll free: 800-328-1300. (In Minnesota, call collect: 612-736-9625.) Ask for the Data Recording Products Division.

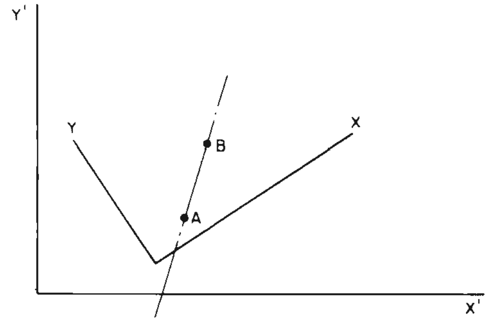
**If it's worth remembering, it's worth Scotch Data Recording Products.**

3M

Circle 368 on inquiry card.

*Figure 7: In the general coordinate transform there are two problems. The first is translation. Each point must be moved by the distance between the origins of the two coordinate systems. Because the distance is the same for every point, you need only know the coordinates of one point in both systems to compute the translation vector dX, dY. The second problem is rotation. The line between A and B makes a different angle with the X axis than with the X' axis. Each point must be rotated by the difference. To compute the transformation, each point is multiplied by a rotation matrix, then added to the translation vector. The equation looks like this:*

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos d & -\sin d \\ \sin d & \cos d \end{bmatrix} \cdot \begin{bmatrix} X' \\ Y' \end{bmatrix} + \begin{bmatrix} dX \\ dY \end{bmatrix}$$




**Would You Like to Participate in a Weather Reporting System?**

*A system is being installed in the state of Virginia that typifies what can be done with microprocessors. At each of seven remote stations, sensors collect data on wind, rainfall, temperature, etc; and store it in the memory of a dedicated microcomputer. About once a day the data is transmitted to a central minicomputer for processing and integration with other data sources. The microcomputers are nicknamed the Seven Dwarfs. The minicomputer, not surprisingly, is called Snow White. It should also be no surprise to those who work with small computers that the first station to be installed was called Grumpy.*

*Now suppose that instead of just seven stations, a large number of personal computer owners attached some simple sensors to their systems, and were linked in a personal computer network (see February 1978 BYTE). Such a network is being tried, also in Virginia. Sensors are manually read; touch tone pads are used for communication. The interest of involved citizens is producing a valuable new resource for meteorologists, hydrologists and agricultural planners; and an interesting new application for readers of BYTE.*■

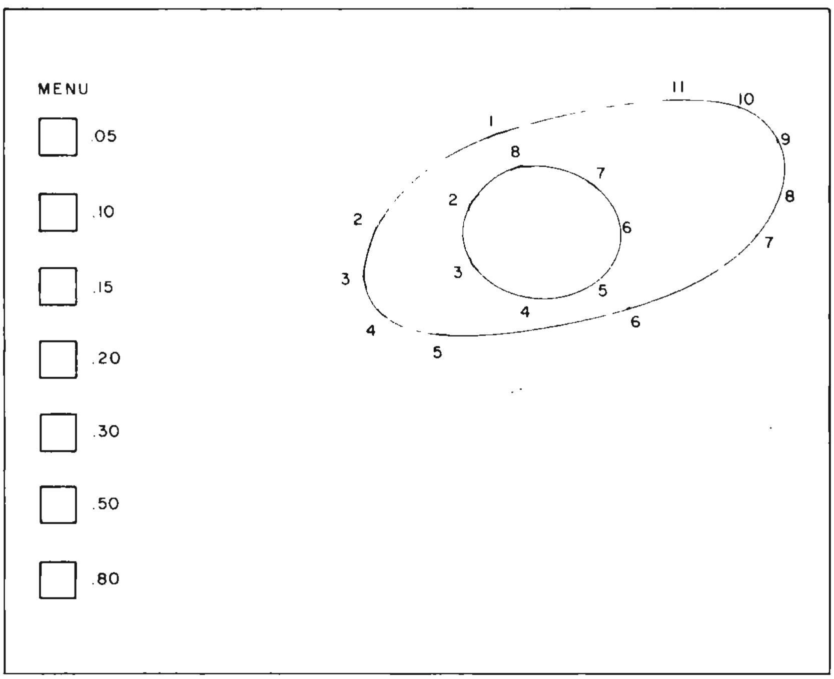



*Figure 8: After digitizing with the Bit Pad, simple closed curves are approximated by polygons. More corners in the polygon produce a better approximation. Unused areas of the tablet can be used for a menu board. Touching the stylus in the appropriate area enters the indicated value. A software handler is required.*

mation but takes longer to enter. If a faster, machine language routine is used, we can take advantage of the Bit Pad's stream mode of operation to enter a large number of points rapidly. As long as the stylus is touching the tablet, the Bit Pad transmits points continuously at a rate of between 1 and 64 points per second. The rate is set by the operator using switches on the control box, or by the computer setting the mode and rate bits of the command word.

Using either approach, the computer must be able to recognize when the entry of an isohyet is complete. In our system this occurs when the most recent point closely matches the first one. The computer now queries the operator for the rainfall rate corresponding to the isohyet just entered. The rate may be entered at a terminal, or by using a portion of the tablet itself as a menu board. In the latter approach, the computer associates a touch of the stylus in one of several small boxes with a preset rainfall rate. Figure 8 shows the menu board ready for operation. The subroutine which interprets the input begins at line 4000 of listing 1.

When all the isohyets have been entered, the computer can begin to assign a rainfall



Circle 180 on inquiry card. →

rate to each square in the grid. It begins with the area of heaviest rain, and determines which grid squares are surrounded by that isohyet (eg: does the center of a square fall within the approximating polygon?). The heaviest rate is assigned to each of these squares. Beyond this area are bands of successively lighter rain. The grid squares surrounded by these isohyets, which have not already been included in another isohyet, are assigned the corresponding lighter rates. The process continues until an area of zero rainfall is encountered. The inverse situation, an area of light rain surrounded by heavier precipitation, does not occur in the sudden, convective storms this program was designed to monitor.

When all the grid squares have been assigned, the computer then holds a record of the rainfall during a one half hour period. When the next map is processed, the storm will have moved, and the distribution of rain will be different. These half hour records can be totaled over periods of any length to provide accurate accumulation data. The meteorologist need only be concerned with instantaneous rates, however. The computer, with the aid of the Bit Pad, handles the motion of the storm and the subtle effects of its changing shape and intensity.

As you may well imagine, updating 1500 grid squares 48 times a day could easily overwhelm many small systems with data. Access to rapid, random storage such as floppy disks is mandatory. The work is still within the capabilities of personal sized computers, but it begins to involve specific operating systems and algorithms which would be useful to only a few readers. The BASIC routines developed here, however, apply to any Bit Pad application. They demonstrate how easy it is to use the Summagraphics graphics tablet for data entry. In doing so, I hope this article has also shown that a microcomputer, when teamed with some novel peripherals, can be used to process data for a challenging meteorological application. ■

## REFERENCES

1. Scofield, R A and V J Oliver, "A Scheme for Estimating Convective Rainfall From Satellite Imagery," NOAA Technical Memorandum NESS 86, April 1977.
2. Follansbee, W A, "Estimation of Daily Rainfall Over China and the USSR Using Satellite Imagery," NOAA Technical Memorandum NESS 81, September 1976.
3. *Earthsat Spring Wheat Yield System Test, Final Report,* Earth Satellite Corporation, prepared for Johnson Spaceflight Center, April 1976.
4. Cole, E W, *Introduction to Meteorology,* John Wiley and Sons Inc, New York, 1970.

# CAST A SPELL. WIN A SORCERER.

# If you've written software in Altair Basic, you've written "spells" for the Exidy Sorcerer.

# Now, make it pay off!

**There's never enough software.** Particularly good software.

That's why Exidy is sponsoring a software contest where nobody loses.

**Altair programs run on Sorcerer.** The Sorcerer computer's Standard Basic is compatible with Altair 4K and 8K Basic. So our contest is open to programs —we like to think of them as "spells" or "Sorcery"—written in all three of those Basic versions.

**Trade one of yours for one of ours.** Just for entering a program in our contest, we'll send you a new, professionally written and documented program. Free. It's a classic game of concentration that's a fun mind-stretcher for both kids and adults. Plus you'll get our new 20" by 24" color poster.

**And maybe 99 more good programs.** We'll publish a bound book of the best programs entered—up to 100 of them, with full credit to each author. If you enter you can have a copy for just the printing and mailing cost. And if your program is included, you get the book free.

**WIN THIS EXIDY SORCERER.**

**And maybe a free Exidy Sorcerer:** Submit one of the four programs judged "best," and win a free Sorcerer computer. (Or choose Sorcerer accessories of equal value.) There'll be one winner in each of the following categories: Business, Education, Fun & Games, and Home/Personal management.

**Test-run your entry free.** Take your program to any participating Sorcerer dealer if you want to give it a test run. At the same time, maybe you'll want to jazz up your program to take advantage of Sorcerer's state-of-the-art features. These include 512 by 240 high-resolution graphics; user-defined characters; and dual cassette I/O, among others.

You can turn in your entry right at the dealer's. And collect your poster and new program on the spot.

**Enter now.** Send us your entry with the coupon. Or visit your dealer. But cast your best spell at Exidy now. And see if you can't make a free computer appear on your doorstep.

## RULES:

1) Entries, including documentation, must be printed by computer or typed double spaced on 8½ by 11 paper, with your name on every page.
2) Enter as many times as you like. This coupon, or a copy of it, must be completed and attached to all entries.
3) Enter at any participating Exidy Sorcerer dealer, or mail entries postpaid to the address on this coupon.
4) Entries must be received by midnight, Aug. 31, 1979. Winners will be notified by Nov. 30, 1979. For a list of winners, send a self-addressed, stamped envelope marked "Winners List" to the coupon address.
5) You warrant, by your signature on this coupon, that all program and documentation material included in your entry is entirely your own original creation, and that no rights to it have been given or sold to any other party, and you agree to allow Exidy to use, publish, distribute, modify, and edit it as it sees fit.
6) All entries become the property of Exidy, Inc. No entries will be returned, nor any questions answered regarding individual entries. No royalties, payments or consideration beyond the items set forth in this advertisement will be given to any entrant.
7) Judging will be by a panel of experts chosen by, and including representatives of, Exidy, Inc. Judges may assign programs to whichever entry category they consider appropriate. Decision of the Judges is final.
8) Employees of Exidy, Inc., its dealers, distributors, advertising agencies and media not eligible. Void where prohibited, taxed or restricted by law.

Circle 133 on inquiry card.

Exidy inc.

EXIDY, INC.
969 W. Maude Ave.
Sunnyvale, CA 94086

Gentlemen:
Here's my "spell." Send me my free program and poster. If I win, send my Exidy Sorcerer computer to:

NAME ____________

ADDRESS ____________

CITY ____________ STATE ________ ZIP ______

DAYTIME PHONE ____________

TITLE OF PROGRAM ____________

CATEGORY ☐ Business ☐ Fun & Games ☐ Education ☐ Home/Personal Management

SIGNATURE ____________ DATE ______

SUPER LOW FREQUENCY OSCILLATOR

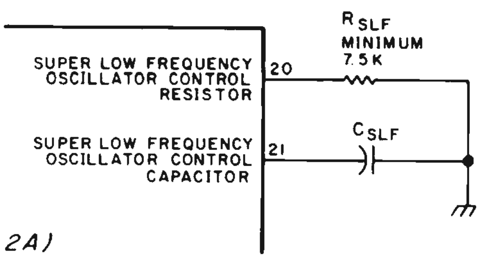
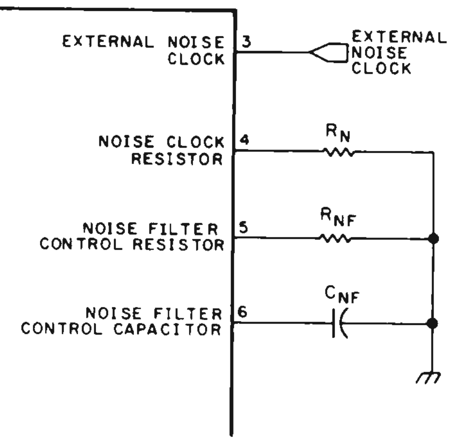
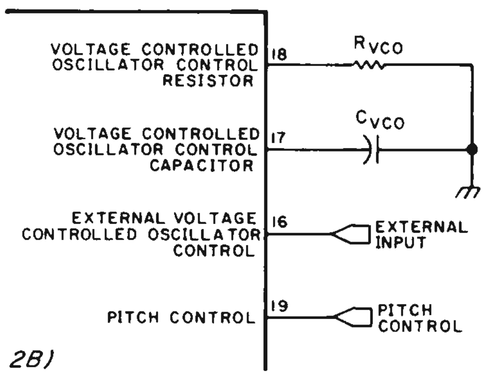



*Figure 2: The SLF (super low frequency) and VCO (voltage controlled oscillator) sections of the Texas Instruments SN-76477 complex sound generator. The desired frequency is selected by adjusting the resistor and capacitor circuits. The frequency is determined by the following formulas: super low frequency = $0.64/R_{SLF} \times C_{SLF}$ and voltage controlled oscillator $= 0.64/R_{VCO} \times C_{VCO}$.*

*Figure 3: The noise generator and filter section is composed of an external clock, two resistors, and a capacitor. The nominal value of $R_N$ is 47 k ohms and the minimum value for $R_{NF}$ is 7.5 k ohms.*




VOLTAGE CONTROLLED OSCILLATOR




## Oscillators

Figure 2 illustrates the two oscillator sections and equations for frequency selection. Figure 2a is an SLF (super low frequency generator) with a normal range of 0.1 Hz to 30 Hz. This super low frequency output is most often used to provide the input to the voltage controlled oscillator which runs at a higher frequency. Such a combination results in frequency modulated sound synthesis. A familiar example is a siren.

The voltage controlled oscillator can be externally controlled by grounding pin 22. The frequency is then governed by a 0 to 2.35 V signal applied to pin 16. Signals above 2.35 V will saturate oscillator output. As a further enhancement, the voltage controlled oscillator allows pitch control through a similarly ranged signal applied to pin 19.

*Figure 4: Outputs of the super low frequency oscillator, voltage controlled oscillator, and noise generator are digitally selected. The control table indicates the output produced for any particular input.*

The output of the voltage controlled oscillator and the super low frequency oscillator is a square wave which is supplied to the mixer and through the envelope selection logic to the envelope generator and modulator.

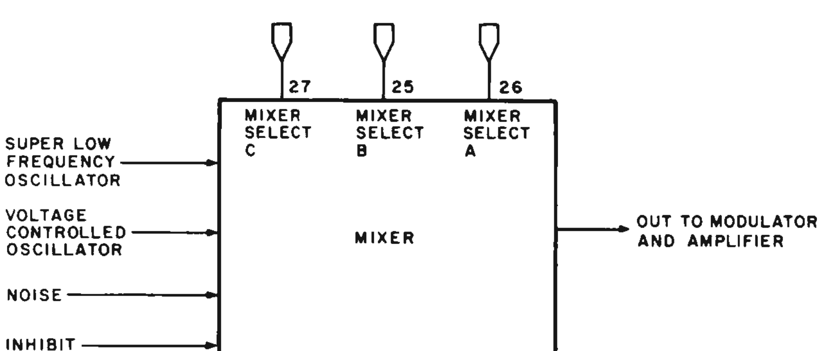
## Noise Generator and Filter

Since so many sounds incorporate noise as an integral component, the 76477 includes a noise generator which can be set to produce pink or white noise by selection of the proper components. (Pink noise has a spectral intensity inversely proportional to frequency over a specified range. White noise is random and has constant energy for a unit bandwidth.) Further refinement of the desired noise range is accommodated through an external clock input applied to pin 3. Figure 3 illustrates this hookup.

The noise generator output is sent to the mixer.

## The Mixer and Envelope Selection

Figure 4 shows how the mixer section of a sound generator works and specifically details the logic codes for the SN76477. The mixer is essentially a gating network which digitally combines the outputs from the super low frequency oscillator, voltage




| Mixer Select Inputs C (Pin 27) | B (Pin 25) | A (Pin 26) | Mixer Output |
|---|---|---|---|
| L | L | L | VCO |
| L | L | H | SLF |
| L | H | L | NOISE |
| L | H | H | VCO/NOISE |
| H | L | L | SLF/NOISE |
| H | L | H | SLF/VCO/NOISE |
| H | H | L | SLF/VCO |
| H | H | H | INHIBIT |

H = high level
L = low level or open

controlled oscillator, and noise generator through a 3 bit code applied to pins 25, 26, and 27. An additional inhibit state is added to shut off operation of the mixer when desired.

The individual outputs of the voltage controlled oscillator, super low frequency oscillator and noise generator are selected with codes of 000, 001, and 010 respectively, as shown in the chart accompanying figure 4. The true value of this device is demonstrated when complex sounds are produced by combining these three sources and utilizing the inhibit for emphasis.

Figure 5a shows how the voltage controlled oscillator can be modulated by the super low frequency oscillator. As mentioned, an example of this is a siren. If, on the other hand, the super low frequency oscillator were programmed as in figure 5b, and mixed with the noise generator, the mixer output would sound like the steam engine we previously discussed. For faster on/off pulsing of the noise generator, the voltage controlled oscillator could be selected, and would appear as in figure 5c.

The inhibit line, rather than being an actual sound source, controls the duration of the other three sections. The internal one shot, triggering a 100 ms burst of noise to a loud amplifier, will sound like a gun shot. This is detailed in figure 5d.

The combined mixer output then goes to the envelope generator and modulator where the amplitude (volume) of the output signal is tailored through proper attack and decay timing so that it will synthesize actual sounds accurately. A piano is most easily characterized by its sharp attack and very long decay. Figure 6 outlines the component calculations for these timed functions.

### Manual Sound Synthesizer

The SN76477 is essentially an independent sound generator. This means that with a few discrete components it can independently synthesize the sound of sirens, phasers, guns, etc. A computer is not required to program this device and, in fact, with the exception of the envelope, mixer and inhibit selection inputs, it is not directly controllable with a microprocessor. An example of a typical hardwired circuit using the SN76477 is shown in figure 7. This circuit simulates the sound of a steam engine and a whistle. The timing components were selected by using the equations outlined in figures 2 thru 6. This circuit produces two sounds by multiplexing the mixer between the voltage controlled oscillator frequency and the super low noise outputs. Normally, with the push button open the super low frequency oscillator pulses the noise generator on and off, producing a chug-chug sound. When the button is pushed, oscillator IC2 multiplexes the integrated circuit to the voltage

*Text continued on page 42*

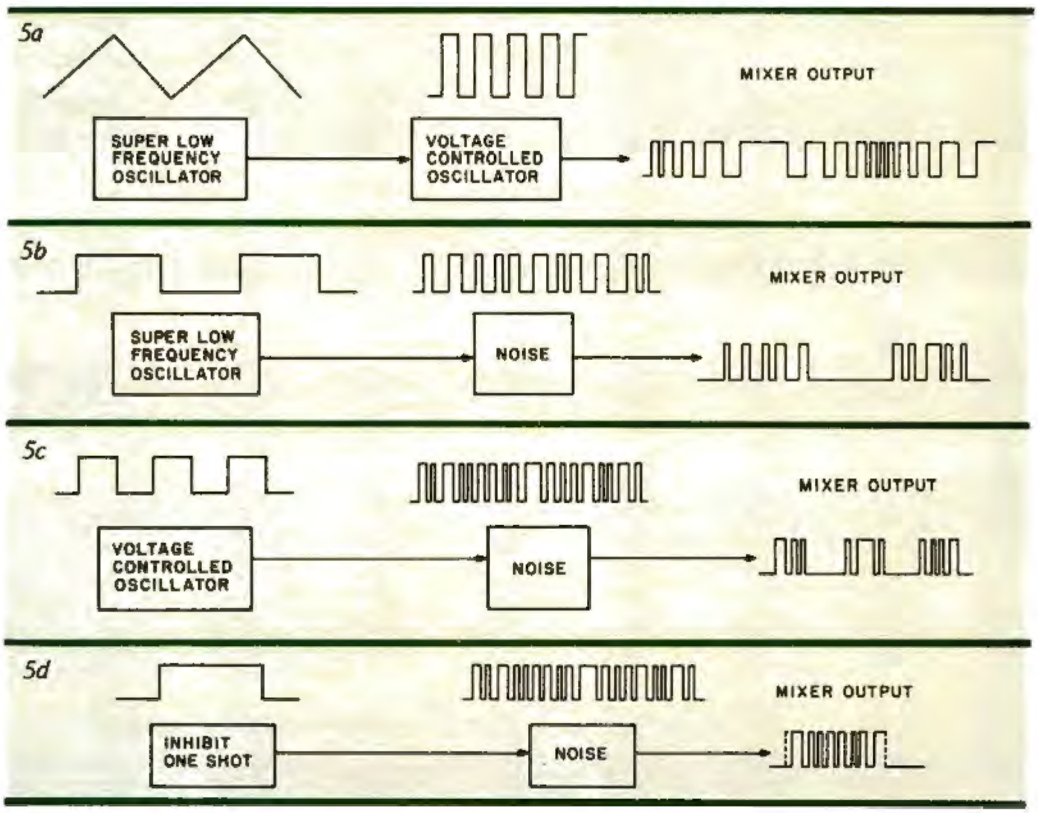
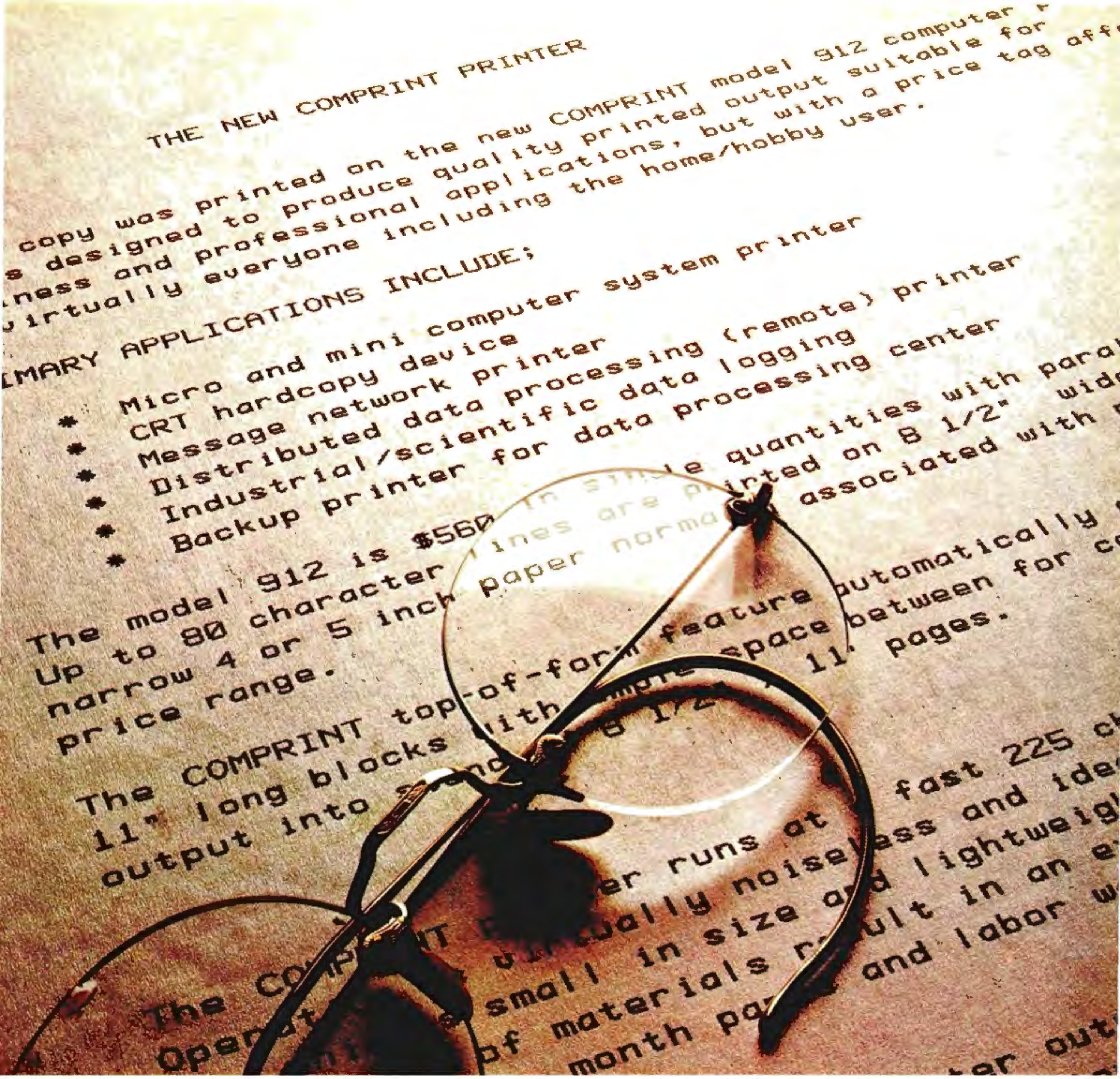



*Figure 5: By carefully choosing what signals are combined, a variety of different types of sounds can be produced. Figure 5a shows a combination of the super low frequency generator and the voltage controlled oscillator producing a sound such as a siren. Figure 5b combines the super low frequency oscillator and the noise generator to generate a sound such as a steam engine. In figure 5c, the voltage controlled oscillator and noise generator are mixed together to form a faster on and off pulsing than produced using the super low frequency generator. When the inhibit one shot is mixed with noise (figure 5d) the resulting sound would sound like a gun being fired.*




# This exceptional print quality for $560!

**The new Comprint model 912 printer for computers and terminals:**

- Fast—225 characters/second (170 lpm)
- 80 character lines on 8½" wide paper
- Quiet, non-impact operation
- 6 month warranty

**Available now at computer stores and industrial distributors.**

comprint

**Computer Printers International, Inc.** 340 E. Middlefield Rd., Mt. View, CA 94043 • 415 969-6161

Circle 48 on inquiry card.

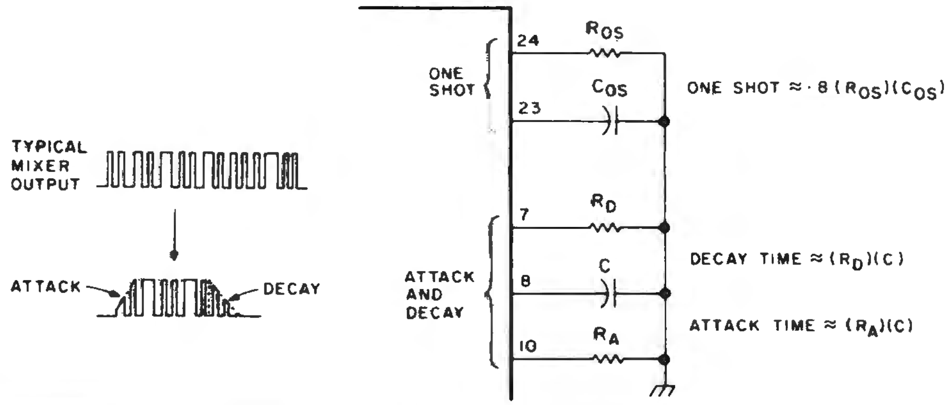
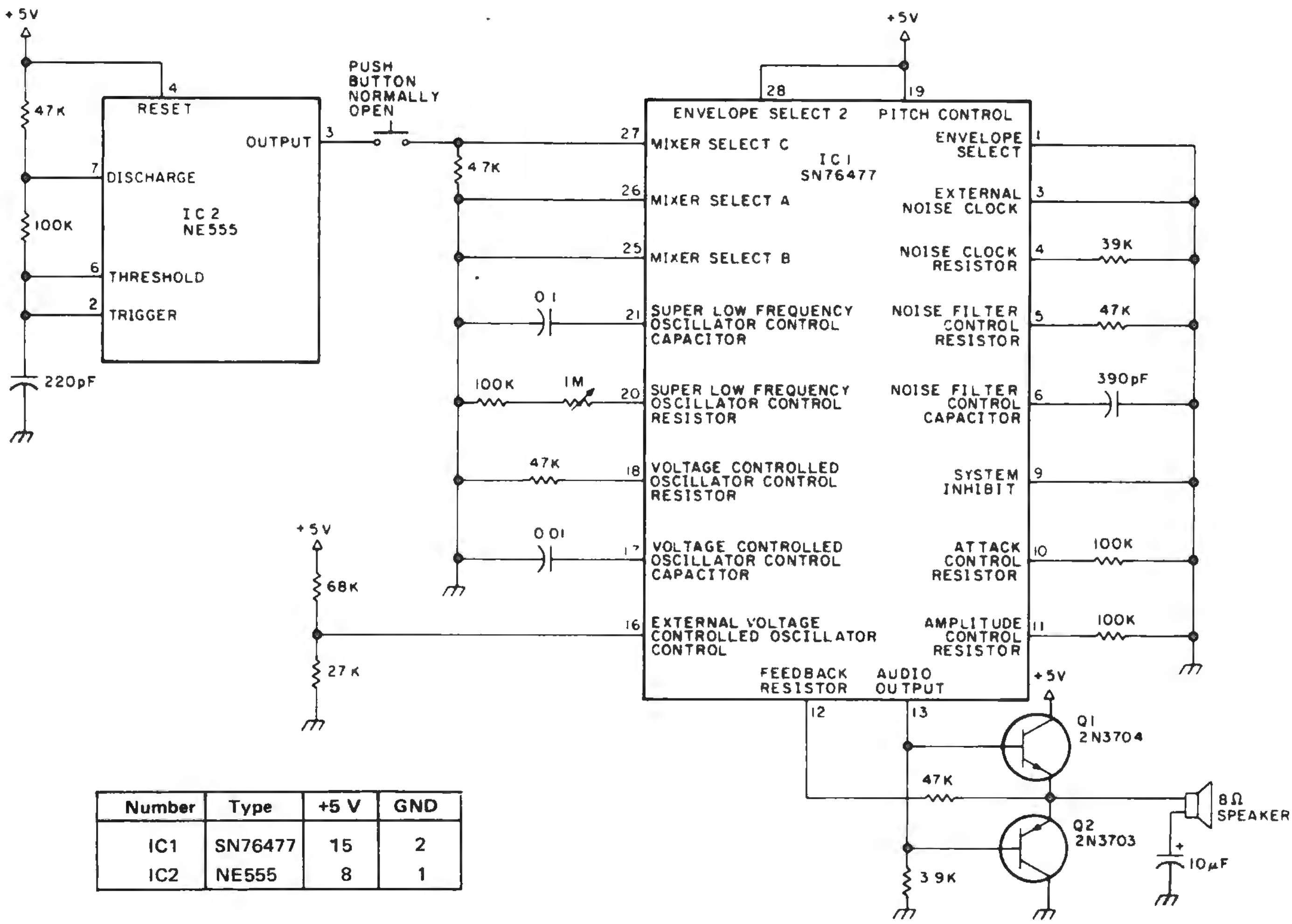



*Figure 6: The envelope selection (table) is determined by envelope select 1 and envelope select 2 (pins 1 and 28) as shown in the table. The attack and decay timing is determined by $R_{OS}$, $C_{OS}$, $R_D$, C, and $R_A$.*

| Envelope Select 1 (Pin 1) | Envelope Select 2 (Pin 28) | Function Selected |
|---|---|---|
| L | L | VCO |
| L | H | Mixer only |
| H | L | One shot |
| H | H | VCO with alternating cycles |

H = high level
L = low level or open




| Number | Type | +5 V | GND |
|---|---|---|---|
| IC1 | SN76477 | 15 | 2 |
| IC2 | NE555 | 8 | 1 |

*Figure 7: The Texas Instruments SN76477 is often used in a hardwired, dedicated device. One such use is simulated steam engine and whistle sound as shown here.*


The Computer Cookbook.™
The
COMPUTER
COOKBOOK
™
Fast Becoming the Reference Standard

| | I/O Port Bit Assignment | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Port 8 | Envelope Select 1 | Inhibit | Envelope Select 2 | Mixer C | Mixer A | Mixer B | VCO Select | External VCO Select |
| | Noise Filter Capacitor | | | Attack/Decay Capacitor | | | | SLF Capacitor |
| Port 9 | 150 pF | 0.001 $\mu$F | 0.01 $\mu$F | 0.1 $\mu$F | 1.0 $\mu$F | 4.7 $\mu$F | unused | 0.1 $\mu$F |
| | SLF Capacitor | | VCO Resistor | | | VCO Capacitor | | |
| Port 10 | 0.47 $\mu$F | 1.0 $\mu$F | 10 k$\Omega$ | 100 k$\Omega$ | 1 M$\Omega$ | 0.005 $\mu$F | 0.05 $\mu$F | 0.47 $\mu$F |

*Table 1: Designation of I/O (input/output) port assignments and associated component choices in the interface for the Texas Instruments SN76477 sound generator.*

| Sound Effect Desired | Hexadecimal Value Sent to Output Port | | |
|---|---|---|---|
| | Port 8 | Port 9 | Port 10 |
| Train | 32 | 80 | 80 |
| Phaser | B6 | 10 | 54 |
| Siren | 82 | 00 | 62 |

*Table 2: Values which are sent to the output ports connected to the SN-76477 interface to produce the indicated sound effects.*

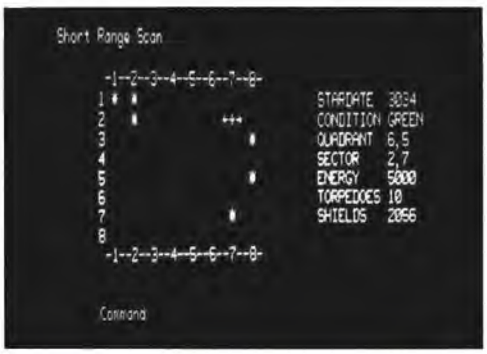



*Photo 2: A typical video based space exploration game could be enhanced by sound effects.*

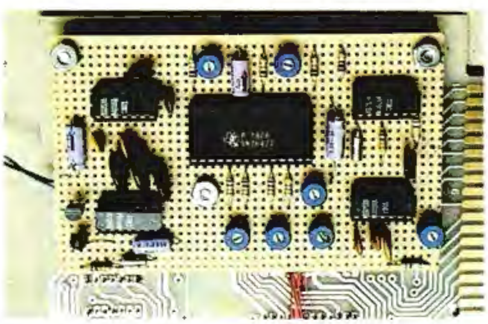


*Photo 3: A look at the prototype circuit of figure 7 attached to the back of an I/O (input/output) board.*

*Text continued from page 38:*
controlled oscillator only position approximately half the time. The voltage controlled oscillator is programmed to produce a whistle. Sufficient power to drive a speaker is facilitated by a two transistor complementary amplifier attached to pins 12 and 13.

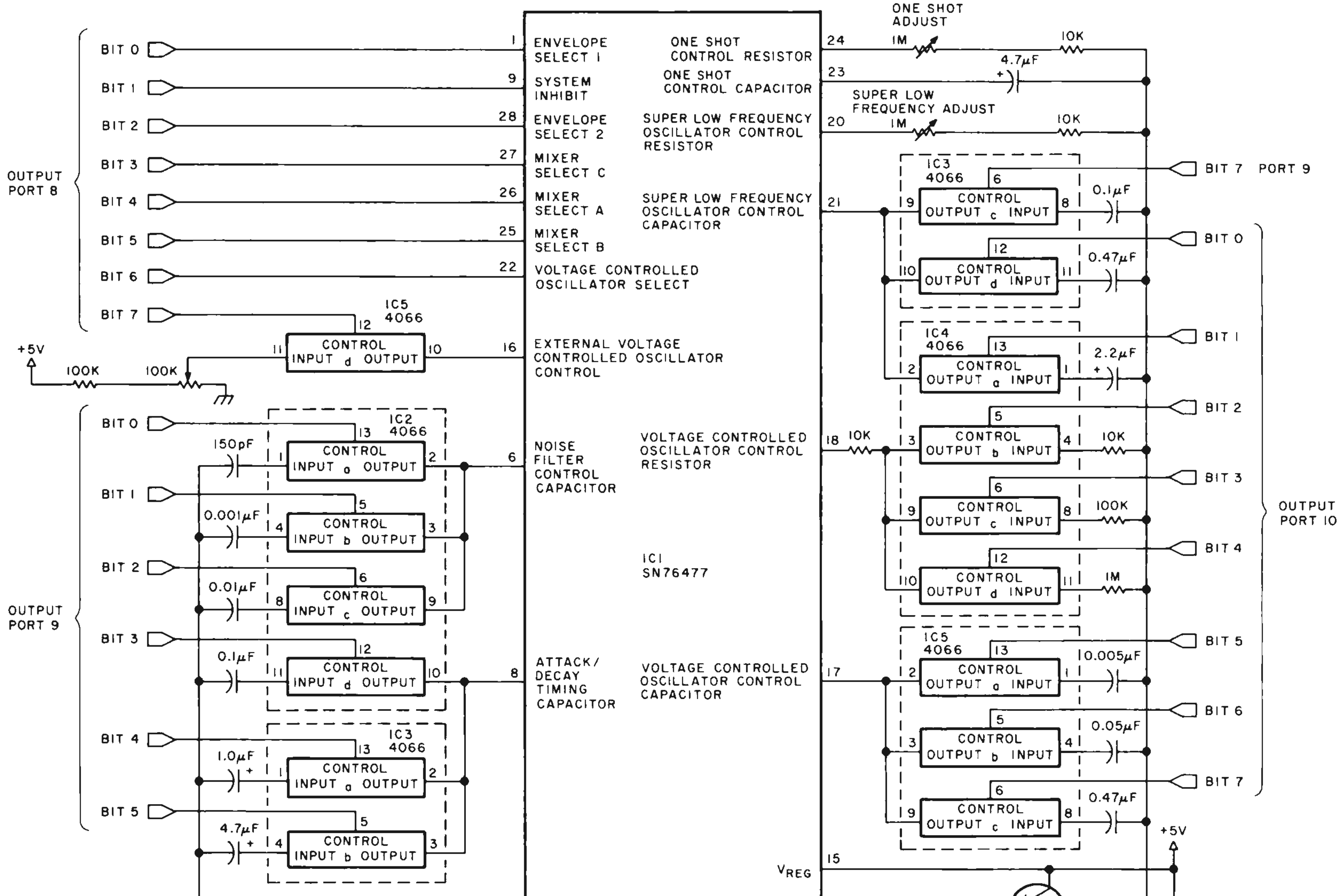
### Build a Computer Programmable Sound Generator Interface

While the SN76477 is not directly controllable by a computer as it exists, an interface between it and a computer can be designed which will give it some semblance of programmability. Figure 8 illustrates such an interface. Sound generation is programmed through three output ports, two of which control CMOS analog switches. These switches allow a variety of resistor and capacitor combinations to be selected. Total control requires three output commands from a BASIC or machine language program, and it is very easy to switch from a siren to a phaser gun sound when implemented as game sound effects. Photo 2 shows the video display of a typical spacewar game. Consider the sophistication that sound effects would add.

In the prototype, shown in photo 3, ports 8, 9, and 10 were chosen to drive the interface. Port 8 handles mixer and envelope selection; port 9 controls selection of components for the attack, decay and noise sections; and port 10 controls the SLF and VCO programming. The values chosen are nominal and will not allow unlimited sound synthesis. Potentiometers are added to facilitate fine tuning.

### A More Sophisticated Programmable Sound Generator

The SN76477 is attached to a microcomputer largely through brute force. A far more sophisticated device has been
*Text continued on page 45*




| Number | Type | +5 V | GND |
|---|---|---|---|
| IC1 | SN76477 | 15 | 2 |
| IC2 | 4066 | 14 | 7 |
| IC3 | 4066 | 14 | 7 |
| IC4 | 4066 | 14 | 7 |
| IC5 | 4066 | 14 | 7 |

*Figure 8: The SN76477 complete sound generator can be controlled by a computer. All capacitors are 100 V ceramic, except electrolytics which are 16 V or greater. All resistors are ¼W ±5%.*

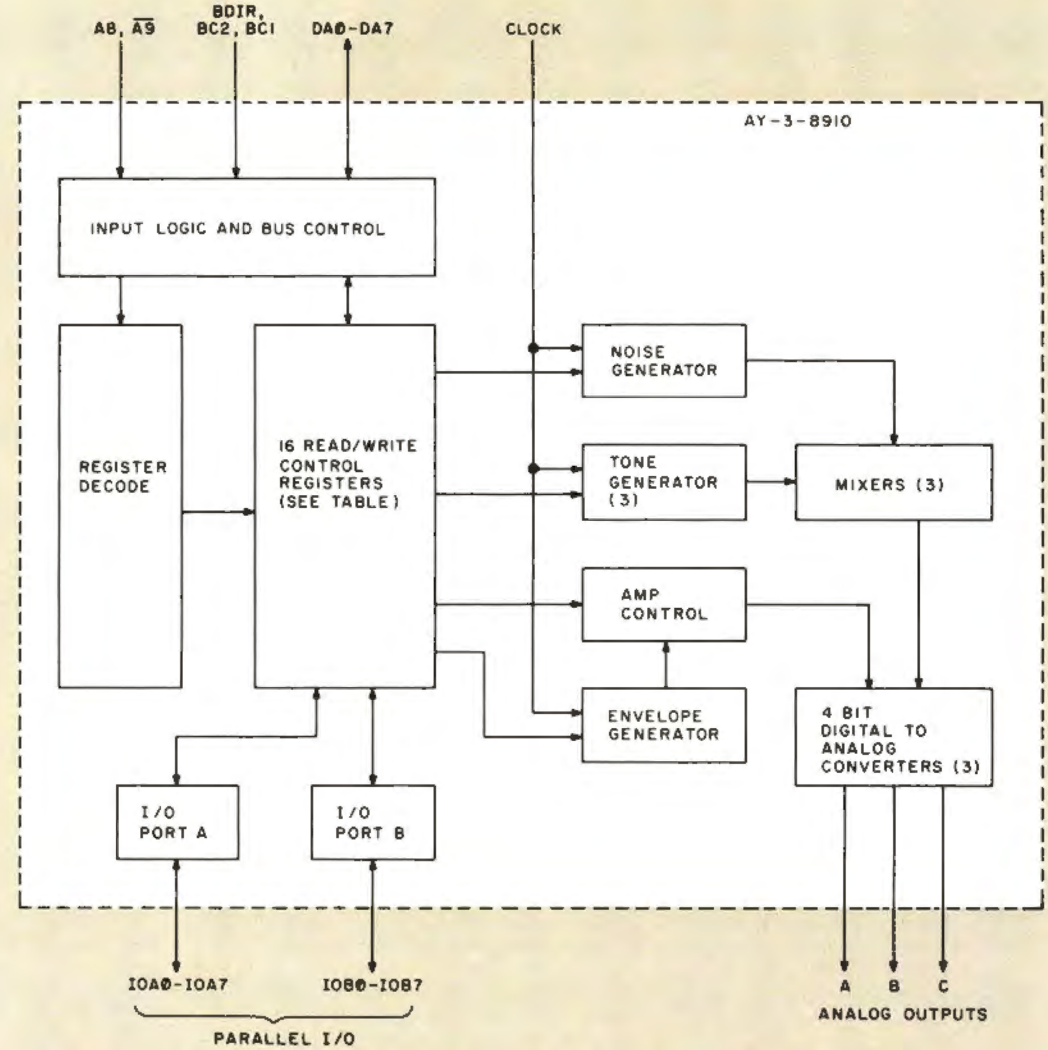
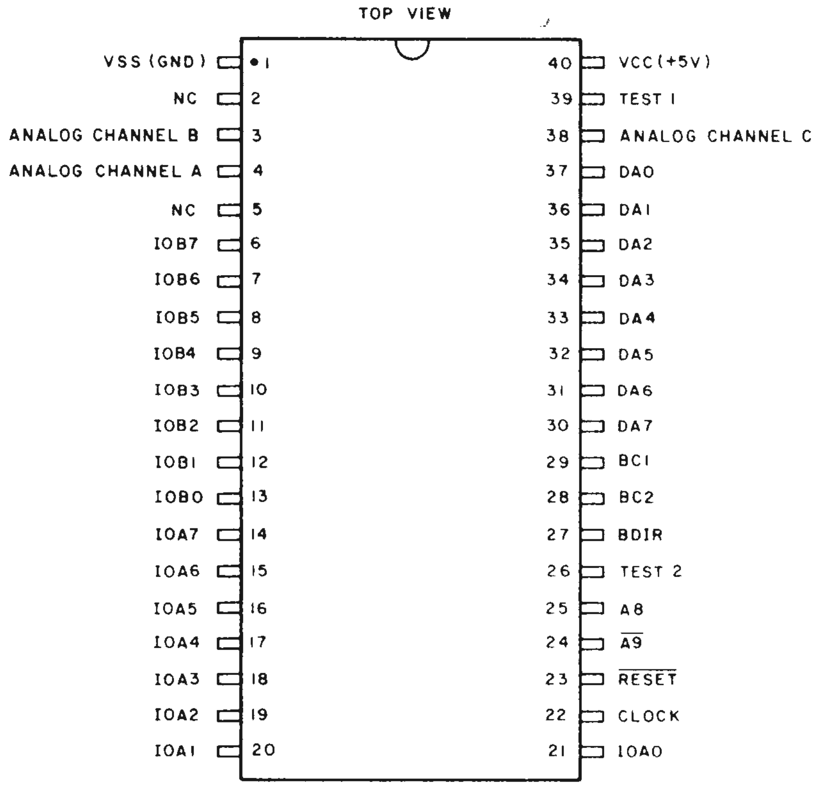



*Figure 9a: Functional block diagram of the General Instrument AY-3-8910 programmable sound generator. The device is made by General Instrument Corp, Microelectronics Division, 600 W John St, Hicksville NY 11802.*

| REGISTER | | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|---|---|---|---|
| R0 | CHANNEL A TONE PERIOD | 8 BIT FINE TUNE A | | | | | | | |
| R1 | | | | | | 4 BIT COARSE TUNE A | | | |
| R2 | CHANNEL B TONE PERIOD | 8 BIT FINE TUNE B | | | | | | | |
| R3 | | | | | | 4 BIT COARSE TUNE B | | | |
| R4 | CHANNEL C TONE PERIOD | 8 BIT FINE TUNE C | | | | | | | |
| R5 | | | | | | 4 BIT COARSE TUNE C | | | |
| R6 | NOISE PERIOD | | | | 5 BIT PERIOD CONTROL | | | | |
| R7 | $\overline{\text{ENABLE}}$ | $\overline{\text{IN}}$/OUT | | $\overline{\text{NOISE}}$ | | | $\overline{\text{TONE}}$ | | |
| | | IOB | IOA | C | B | A | C | B | A |
| R10 | CHANNEL A AMPLITUDE | | | | M | L3 | L2 | L1 | L0 |
| R11 | CHANNEL B AMPLITUDE | | | | M | L3 | L2 | L1 | L0 |
| R12 | CHANNEL C AMPLITUDE | | | | M | L3 | L2 | L1 | L0 |
| R13 | ENVELOPE PERIOD | 8 BIT FINE TUNE E | | | | | | | |
| R14 | | 8 BIT COARSE TUNE E | | | | | | | |
| R15 | ENVELOPE SHAPE/CYCLE | | | | | CONT | ATT | ALT | HOLD |
| R16 | I/O PORT A DATA STORE | 8 BIT PARALLEL I/O ON PORT A | | | | | | | |
| R17 | I/O PORT B DATA STORE | 8 BIT PARALLEL I/O PORT B | | | | | | | |

*Figure 9b: Map of the control registers of the AY-3-8910.*

*Figure 9c: Pin designations of the AY-3-8910 device.*




*Text continued from page 42:*
recently introduced and it is designed specifically as a bus controlled device. This new device is the AY-3-8910 from General Instrument. It uses no external components and synthesizes sounds totally by digital means. A functional block diagram is shown in figure 9a.

You'll notice a similarity between this programmable sound generator and the Texas Instruments device in that they both contain the same elemental sound synthesis components such as noise and tone generators. The real difference is that the General Instrument programmable sound generator is programmed through 16 read/write control registers rather than resistors and capacitors. These registers appear as 16 sequential memory mapped I/O (input/output) locations to the controlling processor.

The AY-3-8910 incorporates a noise generator, three tone generators, three mixers, an envelope generator and three digital to analog converters for amplitude control. An added benefit is the inclusion of two decoded I/O ports which are available for other external applications. All subsystems are controlled through the control register array.

The device is specifically designed to interface with the General Instrument CP1600 series of microprocessors but it can be easily accommodated by others. Figure 10 illustrates this simple attachment. A bidirectional address/data bus, DA0 thru DA7, provides the necessary communication path. Since there are 16 registers, only four bits of address are actually used, and A8 and $\overline{A9}$ serve more as device select lines by definition. BC1, BC2, and BDIR are the bus control lines and define bus direction, reading, and writing of register data. While an inexpensive circuit such as that shown in figure 11 can be used as the clock for both the processor and the programmable sound generator, they are basically independent and can be different rates. The programmable sound generator clock is primarily used for the sound synthesis. The reset line clears all registers.

For all practical purposes, signal line BC2 is unnecessary and can be tied to +5 V. The read/write control logic is shown in table 4.

The timing of BC1 and BDIR control lines are shown in figure 12. Data transfer is carried out by strobing these lines, while the

*Text continued on page 48*

**Whistling Bomb Sound Effect**

| Register Number | Hexadecimal Load Value | Explanation |
|---|---|---|
| Any not specified | 00 | |
| R7 | 3E | Enable tone only on channel A only. |
| R10 | 0F | Select maximum amplitude on channel A. |
| R0 | 30 (start) | Sweep effect for channel A tone period via a processor loop with approximately 25 ms wait time between each step from 30 to C0 (0.429 ms/2330 Hz to 1.72 ms/582 Hz). |
| R0 | C0 (end) | |
| R6 | 0F | Set noise period to midvalue. |
| R7 | 07 | Enable noise only on channels A,B,C. |
| R10 | 10 | Select full amplitude range under direct control of envelope generator. |
| R11 | 10 | |
| R12 | 10 | |
| R14 | 10 | Set envelope period to 0.586 seconds. |
| R15 | 00 | Select envelope decay, one cycle only. |

**Phaser Sound Effect**

| Register Number | Hexadecimal Load Value | Explanation |
|---|---|---|
| Any not specified | 00 | |
| R7 | 3E | Enable tone only on channel A only. |
| R10 | 0F | Select maximum amplitude on channel A. |
| R0 | 30 (start) | Sweep effect for channel A tone period via a processor loop with approximately 3 ms wait time between each step from 30 to 70 (0.429 ms/2330 Hz to 1.0 ms/1000 Hz). |
| R0 | 70 (end) | |
| R10 | 00 | Turn off channel A to end sound effect. |

*Table 3: Values which are loaded into the control registers of the General Instrument AY-3-8910 sound generator in order to produce the indicated sound effects.*

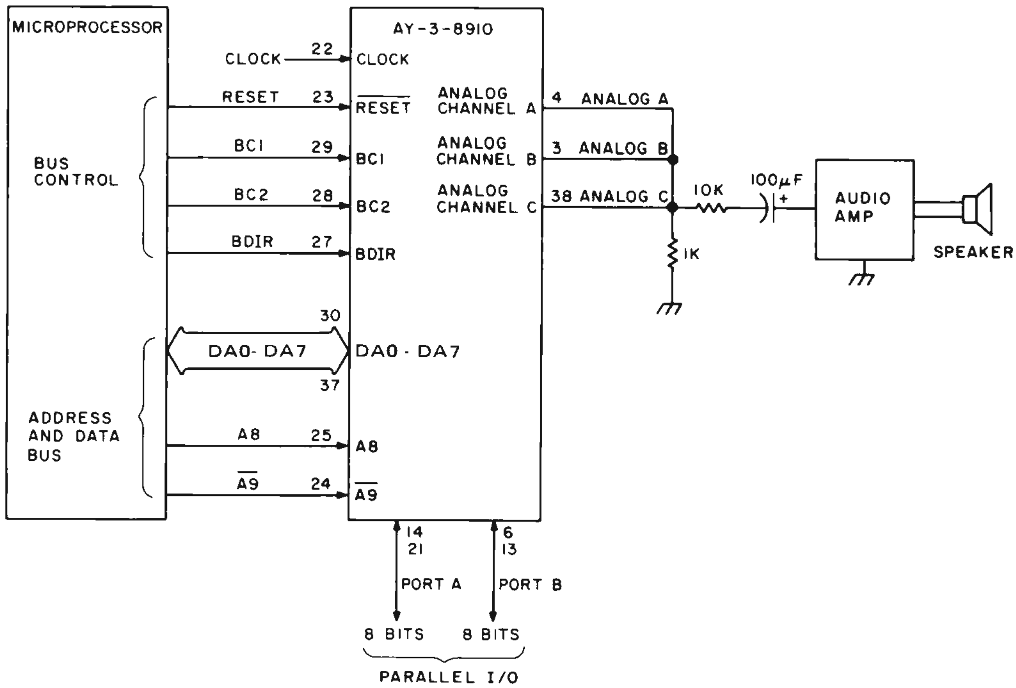



*Figure 10: Typical microprocessor to programmable sound generator interface.*

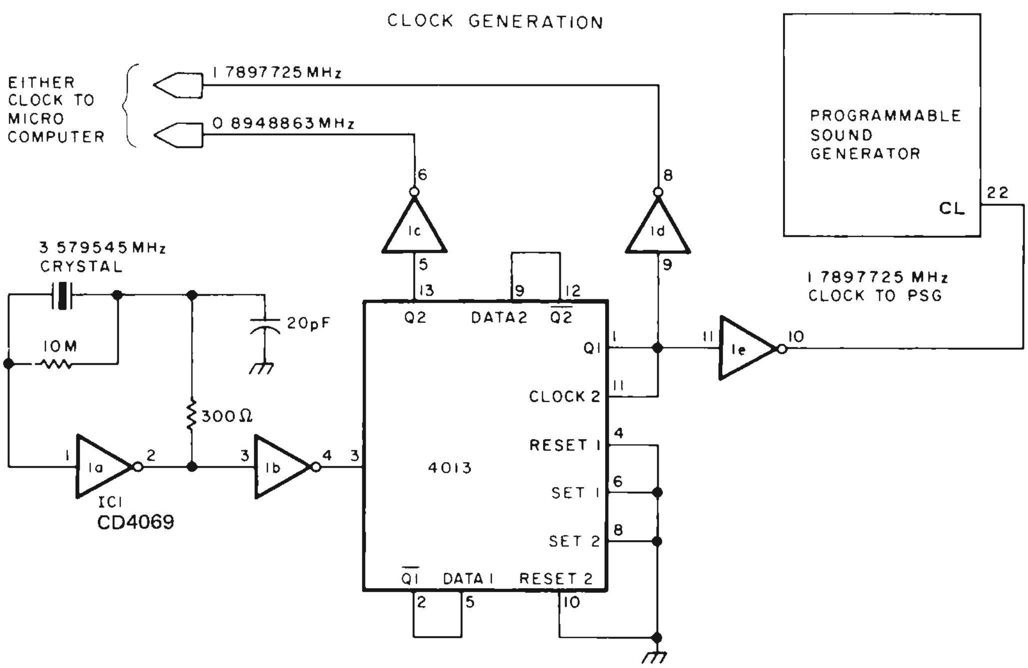
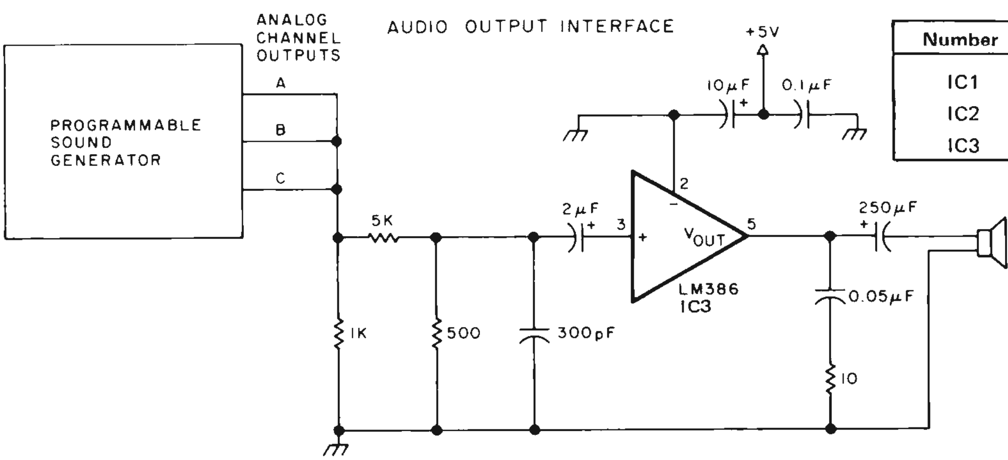



*Figure 11a: A simple clock generator which can be used as a clock for the processor and the programmable sound generator.*




| Number | Type | +5 V | GND |
|---|---|---|---|
| IC1 | CD4069 | 14 | 7 |
| IC2 | 4013 | 14 | 7 |
| IC3 | LM386 | 6 | 4 |

*Figure 11b: A typical audio output interface for driving a speaker from the programmable sound generator.*

# FUN & GAMES



# PLUS SERIOUS SYSTEM SOFTWARE


## AT YOUR COMPUTER RETAILER NOW

TRS-80, Apple II, Sorcerer, SOL and Southwest Tech 6800 owners: get more out of your personal computer.

Get into action with G2 Bullseye!, Sea Battle, Confrontation and Attack!

Sharpen your analytical abilities with G2's Outwit I, Outwit II and Mind Bender.

Take a chance with G2's Beat the House. Check out your health with Clinic.

And enjoy the challenging experience of two new G2 computer simulations: The Market and Wildcatting.

Or get serious with three powerful new languages. Level III Basic for the TRS-80. Extended Basic for the SOL. And Standard Basic for the Southwest Tech 6800. All written by Microsoft—the Basic wizards. Exclusively from G2.

Our software has more so your computer does more. Great programming. Highest quality cassette. The most comprehensive instruction manual available. Plus source listing print-out of every application program in Basic. You can learn how the programs were written. We even encourage you to do your own re-programming to improve your skills!

G2 software is available from computer retailers nationwide. If your local retailer doesn't have it, ask him to become a G2 dealer by calling us toll free: (800) 538-8540 (U.S.A.) or (800) 672-8691 (California).

**THE REASON YOU BOUGHT YOUR COMPUTER.**

GRT CORPORATION

A Product of GRT Corporation
Consumer Computer Group

| BDIR | BC2 | BC1 | Function | CP1600 Function Abbreviation |
|---|---|---|---|---|
| 0 | 1 | 0 | Inactive | NACT |
| 0 | 1 | 1 | Read from PSG | DTB |
| 1 | 1 | 0 | Write to PSG | DWS |
| 1 | 1 | 1 | Latch Address | INTAK |

*Table 4: Summary of the read/write control logic needed to control the AY-3-8910 sound generator.*

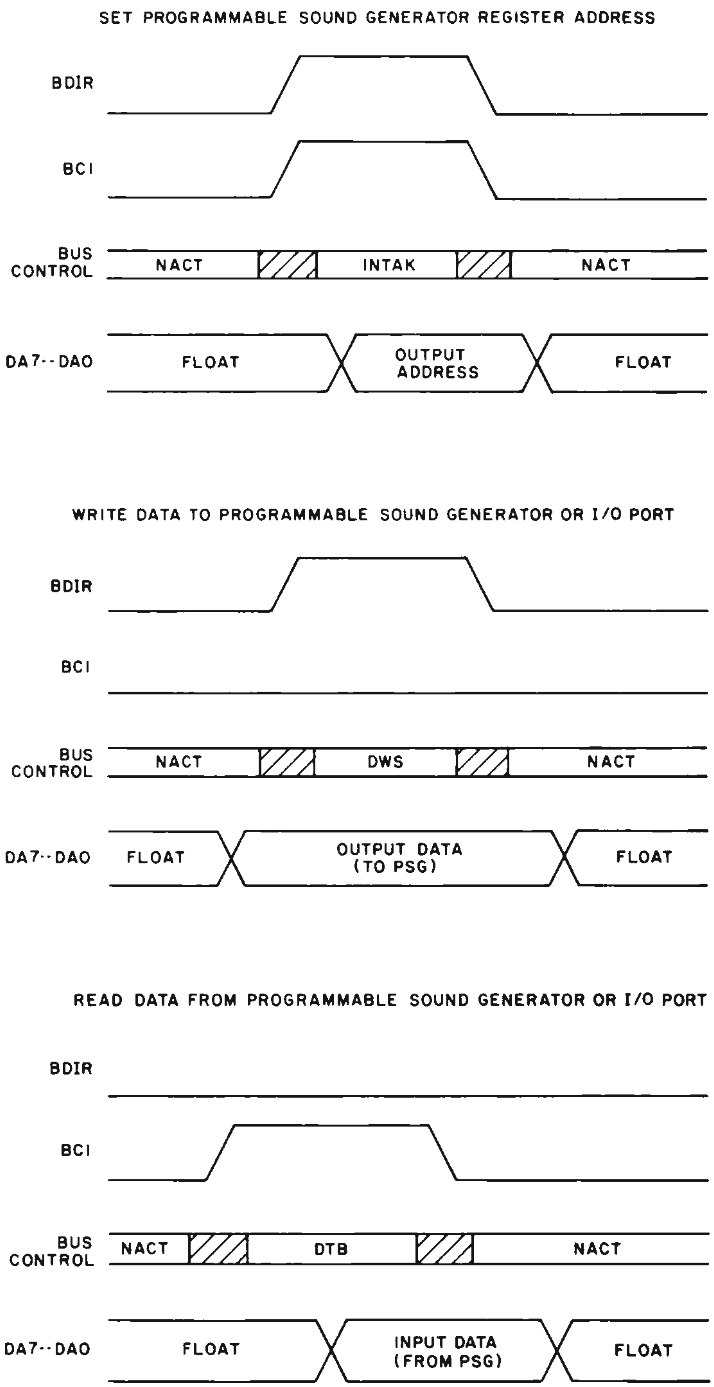



*Figure 12: Programmable sound generator bus timing logic.*

*Text continued from page 45:*
address/data bus contains the pertinent contents. These pulses should be short and one processor clock cycle should suffice.

### Tone Select

The registers are divided into six categories, and numbered in base eight:

| | |
|---|---|
| Tone generators | $R_0$ thru $R_5$ |
| Noise generator | $R_6$ |
| Mixer control | $R_7$ |
| Amplitude control | $R_{10}$ thru $R_{12}$ |
| Envelope control | $R_{13}$ thru $R_{15}$ |
| I/O ports | $R_{16}$ and $R_{17}$. |

Tones are square waves produced by dividing the input clock by 16, then counting that result down by a programmed 12 bit tone-period value. The 12 bit value, defined by the coarse and fine tune registers, is a combination of the two control registers. The 12 bits represent period and T = 1/frequency. The higher the register value, the lower the tone. Register contents range from 000000000001 (divide by 1) to 111111111111 (divide by 4095). With a 2 MHz clock the frequencies would be 125 kHz and 30.5 Hz respectively.

The other parameters, such as noise, mixers, amplitude, and envelope controls, are chosen in a similar manner. The actual programming technique is beyond the scope of this introduction to the AY-3-8910, and I suggest that interested readers send inquiries to General Instrument.

### Connecting the AY-3-8910 to the S-100 Bus

Figure 13 shows how an AY-3-8910 programmable sound generator can be connected as an I/O device on the S-100 8080 compatible bus. Switches SW1 through SW6 define the starting I/O address of the 16 programmable sound generator registers.

*Text continued on page 51*

| Number | Type | +5 V | GND |
|---|---|---|---|
| IC1 | 7485 | 16 | 8 |
| IC2 | 7485 | 16 | 8 |
| IC3 | 7404 | 14 | 7 |
| IC4 | 7402 | 14 | 7 |
| IC5 | 7400 | 14 | 7 |
| IC6 | 7400 | 14 | 7 |
| IC7 | 74148 | 16 | 8 |
| IC8 | 74LS367 | 16 | 8 |
| IC9 | 74LS367 | 16 | 8 |
| IC10 | 74LS367 | 16 | 8 |
| IC11 | 74LS367 | 16 | 8 |
| IC12 | AY-3-8910 | 40 | 1 |

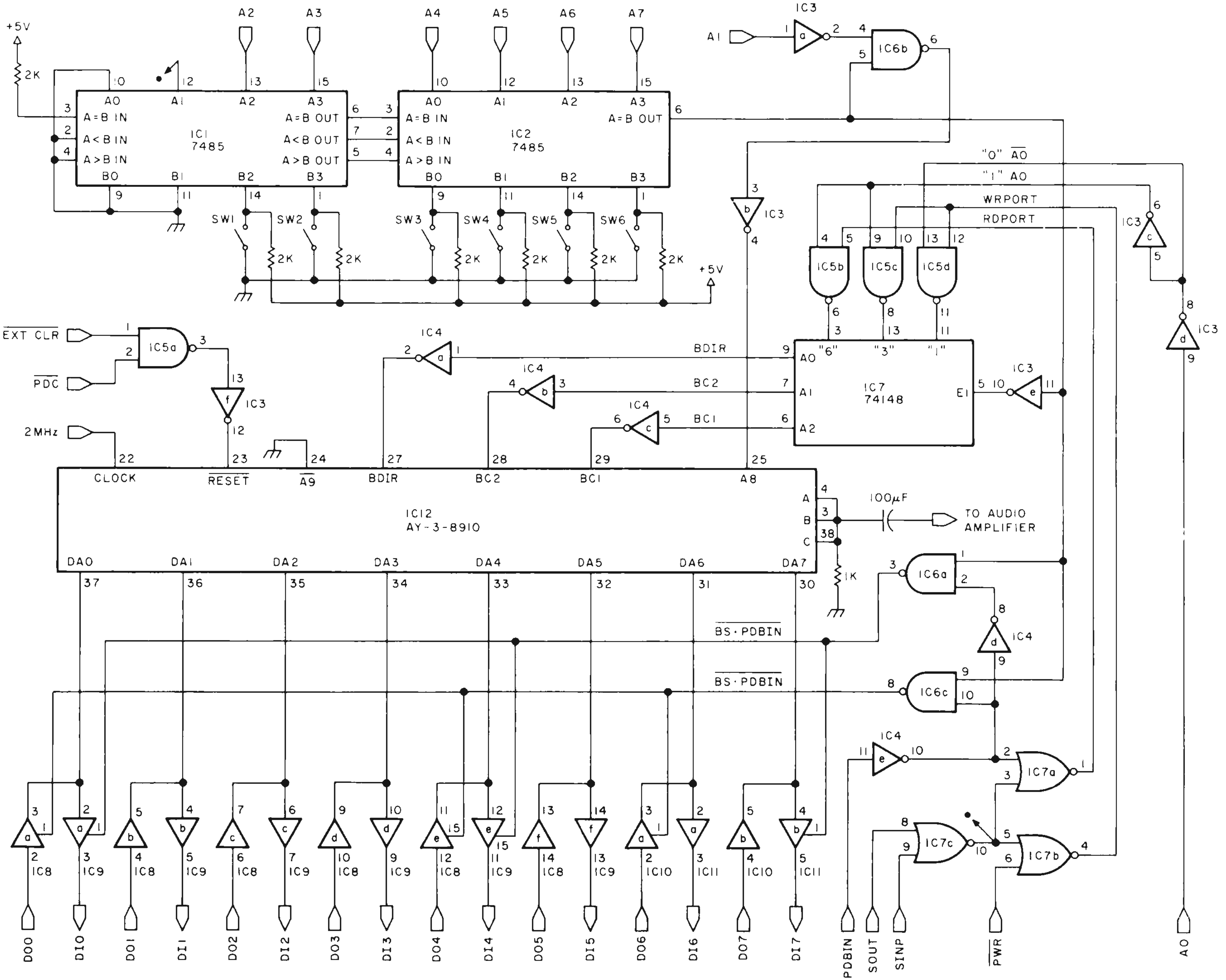



*Figure 13: The AY-3-8910 can be connected to the S-100 bus. SW1 thru SW6 define the starting address of the 16 control registers. The power pin assignment is shown in the table at left.*

```
                    LATCH ADDRESS ROUTINE

PORTADDR EQU 80H ; ADDRESS TRANSFER PORT ADDRESS
PORTDATA EQU 81H ; DATA TRANSFER PORT ADDRESS

; THIS ROUTINE WILL TRANSFER THE CONTENTS OF
; 8080 REGISTER C TO THE PSG ADDRESS REGISTER
PSGBAR     MOV   A,C ; GET C IN A FOR OUT
           OUT   PORTBAR ; SEND TO ADDRESS PORT
           RET

                    WRITE DATA ROUTINE

; ROUTINE TO WRITE THE CONTENTS OF 8080 REGISTER B
; TO THE PSG REGISTER SPECIFIED BY 8080 REGISTER C
;
PSGWRITE   CALL  PSGBAR ; GET ADDRESS LATCHED
           MOV   A,B ; GET VALUE IN A FOR TRANSFER
           OUT   PORTDATA ; PUT TO PSG REGISTER
           RET

                    READ DATA ROUTINE

; ROUTINE TO READ THE PSG REGISTER SPECIFIED
; BY THE 8080 REGISTER C AND RETURN THE DATA
; IN 8080 REGISTER B
;
PSGREAD    CALL  PSGBAR
           IN    PORTDATA ; GET REGISTER DATA
           MOV   B,A GET IN TRANSFER REGISTER
           RET
```

*Listing 1: Routines written for the 8080 microprocessor to operate the AY-3-8910 programmable sound generator.*

```
        LATCH ADDRESS ROUTINE

; AT ENTRY, B HAS ADDRESS VALUE
;
LATCH CLRA
   STAA 8005 ; GET D DIR A
   LDAA #FF
   STAA 8004 ; OUTPUTS
   LDAA #4
   STAA 8005 ; GET PERIPHERAL A
   STAB 8004 ; FORM ADDR
   STAA 8006
   CLRA
   STAA 8006 ; LATCH ADDRESS
   RTS ; RETURN

        WRITE DATA ROUTINE

; AT ENTRY, B HAD DATA VALUE
;
WRITE STAB 8004 ; FORM DATA
   LDAA #6 ; DWS
   STAA 8006
   CLRA
   STAA 8006 ; WRITE DATA
   RTS ; RETURN

        READ DATA ROUTINE

; AFTER READ, B HAS READ DATA
;
READ STA A 8005 ; GET D DIR
   STA A 8004 ; INPUTS
   LDAA #4
   STA A 8005 ; GET PERIPHERAL
   DECA
   STA A 8006 ; READ MODE
   LDA B 8004 ; READ DATA
   CLRA
   STA A 8006 ; REMOVE READ MODE
   RTS          ; RETURN
```

*Listing 2: Routines coded for the 6800 microprocessor to operate the AY-3-8910.*

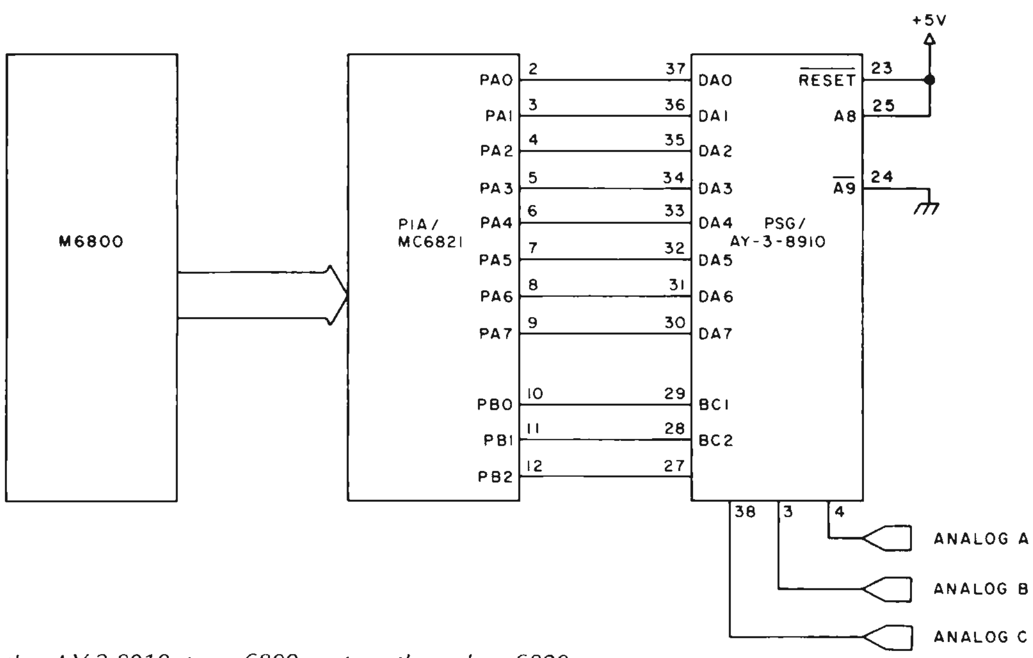



*Figure 14: Connecting the AY-3-8910 to a 6800 system through a 6820 programmable interface adapter is easier than interfacing the S-100 bus.*

*Text continued from page 48:*

Reading and writing from it is as illustrated.

A less complicated hardware interface is attained by using a peripheral interface adapter such as the 6820. Figure 14 demonstrates a technique which can be used for 6800 systems. The considerable difference in hardware complexity should in no way imply lack of ability using the 8080. If the S-100 bus is ignored and a 8255 programmable peripheral interface is used instead, it would result in a circuit similar to figure 14.

### In Conclusion

I have briefly presented two methods of sound synthesis. While both are simple to implement, it is easy to recognize that the Texas Instruments part is more applicable in dedicated designs while the General Instrument device is for general synthesizer applications. It is not inconceivable that the AY-3-8910 could produce almost any sound, and it is a natural for use with a music interpreter running on a microcomputer. Perhaps the next famous composer will not direct a 150 piece orchestra but, rather, a trio of microcomputers controlling a bank of AY-3-8910s.■

*Circuit diagrams and drawings pertaining to the AY-3-8910 were provided courtesy of General Instrument Corp.*

# Programming Quickies

## Apple Kaleidoscope

**Robert J Bishop**
**213 Jason Way**
**Mountain View CA 94043**

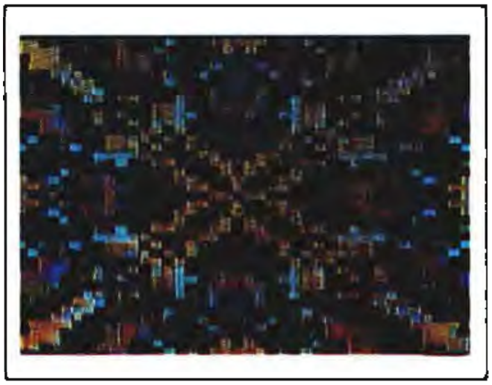


*Photo 1: Typical output of the Kaleidoscope program.*

Listing 1 is a short program that generates a fastmoving colorful display on your Apple II.

The program starts at hexadecimal location 0800 and resides in less than one page of memory. It may be entered in object form by use of the system monitor. To run the program, simply type: 800G. The display can be *frozen* by hitting any key on the keyboard. Hitting any key again will resume the action.

The program was written using Microproducts Editor/Assembler program for the Apple II.■

```
                 KALEIDOSCOPE

0800  4CA508          JMP STAR

                 PLOT (20+X,20+Y)

0803  18         DRAW CLC
0804  A914            LDA 14
0806  6510            ADC *X
0808  A8              TAY
0809  A914            LDA 14
080B  6511            ADC *Y
080D  2000F8          JSR PLOT

                 PLOT (20-X,20+Y)

0810  38              SEC
0811  A914            LDA 14
0813  E510            SBC *X
0815  A8              TAY
0816  200EF8          JSR PLT1

                 PLOT (20+Y,20+X)

0819  18              CLC
081A  A914            LDA 14
081C  6511            ADC *Y
081E  A8              TAY
081F  A914            LDA 14
0821  6510            ADC *X
0823  2000F8          JSR PLOT

                 PLOT (20-Y,20+X)

0826  38              SEC
0827  A914            LDA 14
0829  E511            SBC *Y
082B  A8              TAY
082C  200EF8          JSR PLT1

                 PLOT (20+X,20-Y)

082F  18              CLC
0830  A914            LDA 14
0832  6510            ADC *X
0834  A8              TAY
0835  38              SEC
0836  A914            LDA 14
0838  E511            SBC *Y
083A  2000F8          JSR PLOT

                 PLOT (20-X,20-Y)

083D  38              SEC
083E  A914            LDA 14
0840  E510            SBC *X
0842  A8              TAY
0843  200EF8          JSR PLT1

                 PLOT (20+Y,20-X)

0846  18              CLC
0847  A914            LDA 14
0849  6511            ADC *Y
084B  A8              TAY
084C  38              SEC
084D  A914            LDA 14
084F  E510            SBC *X
0851  2000F8          JSR PLOT

                 PLOT (20-Y,20-X)

0854  38              SEC
0855  A914            LDA 14
0857  E511            SBC *Y
0859  A8              TAY
085A  200EF8          JSR PLT1
085D  60              RTS
                 X    DL 0010
                 Y    DL 0011
                 Z    DL 0012
                 PLOT DL F800
```

```
                PLT1  .DL F80E
                :
                :"RANDOM" NUMBERS
                :
085E  E8        RND   INX
085F  D00D            BNE GRAB
0861  EE7008          INC GRAB+02
0864  AD7008          LDA GRAB+02
0867  09E0            ORA 0E0
0869  8D7008          STA GRAB+02
086C  E615            INC *ZZ
086E  BD00E0    GRAB  LDA BASE,X
0871  60              RTS
                :
                :MAIN SUBROUTINE
                :
0872  A510      GUTS  LDA *X
0874  208908          JSR STEP
0877  8510            STA *X
0879  A511            LDA *Y
087B  208908          JSR STEP
087E  8511            STA *Y
0880  A512            LDA *Z
0882  2064F8          JSR SETC
0885  200308          JSR DRAW
0888  60              RTS
0889  850F      STEP  STA *WORK
088B  205E08          JSR RND
088E  1002            BPL CONT
0890  E60F            INC *WORK
0892  6A        CONT  ROR
0893  9002            BCC TEST
0895  C60F            DEC *WORK
0897  A50F      TEST  LDA *WORK
0899  1003            BPL CMP
089B  A913            LDA 13
089D  60              RTS
089E  C914      CMP   CMP 14
08A0  D002            BNE EXIT
08A2  A900            LDA 00
08A4  60        EXIT  RTS
                WORK  .DL 000F
                :
                :MAIN PROGRAM
                :
08A5  2032F8    STAR  JSR GR
08A8  A900            LDA 00
08AA  8D50C0          STA $C050
08AD  8D52C0          STA $C052
08B0  8513            STA *XX
08B2  8514            STA *YY
08B4  8515            STA *ZZ
08B6  8516            STA *XXX
08B8  8517            STA *YYY
08BA  AA              TAX
08BB  A513      LOOP  LDA *XX
08BD  8510            STA *X
08BF  A514            LDA *YY
08C1  8511            STA *Y
08C3  A515            LDA *ZZ
08C5  8512            STA *Z
08C7  207208          JSR GUTS
08CA  A510            LDA *X
08CC  8513            STA *XX
08CE  A511            LDA *Y
08D0  8514            STA *YY
08D2  A516            LDA *XXX
08D4  8510            STA *X
08D6  A517            LDA *YYY
08D8  8511            STA *Y
08DA  A900            LDA 00
08DC  8512            STA *Z
08DE  207208          JSR GUTS
08E1  A510            LDA *X
08E3  8516            STA *XXX
08E5  A511            LDA *Y
08E7  8517            STA *YYY
08E9  AD00C0          LDA $C000
08EC  10CD            BPL LOOP
08EF  A900            LDA 00
08F0  8D10C0          STA $C010
08F3  AD00C0    WAIT  LDA $C000
08F6  10FB            BPL WAIT
08F8  A900            LDA 00
08FA  8D10C0          STA $C010
08FD  F0BC            BEQ LOOP
                XX    .DL 0013
                YY    .DL 0014
                ZZ    .DL 0015
                XXX   .DL 0016
                YYY   .DL 0017
                BASE  .DL E000
                SETC  .DL F864
                GR    .DL F832
                END   .EN
```

```
SYMBOL TABLE
DRAW    0803
X       0010
Y       0011
Z       0012
PLOT    F800
PLT1    F80E
RND     085E
GRAB    086E
GUTS    0872
STEP    0889
CONT    0892
TEST    0897
CMP     089E
EXIT    08A4
WORK    000F
STAR    08A5
LOOP    08BB
WAIT    08F3
XX      0013
YY      0014
ZZ      0015
XXX     0016
YYY     0017
BASE    E000
SETC    F864
GR      F832
END     08FF
```

# A Model of the Brain for Robot Control

## Part 2: A Neurological Model

James Albus
Project Manager
United States Dept of Commerce
National Bureau of Standards
Washington DC 20234

The ideas presented in this article represent the views of the author and not those of the Department of Commerce or the National Bureau of Standards.

In part 1 I described how sensory interactive, goal directed behavior can be generated and controlled by a multilevel hierarchy of computing modules. At each level of the hierarchy, input commands are decomposed into strings of output subcommands which form the input commands to the next lower level. Feedback from the external environment, or from internal sources, drives the decomposition process and steers the selection of subcommands so as to achieve successful performance of the task of reaching the goal. In this article I will address questions of what kind of neurological structures are believed to exist in the brain and what kind of computations, memory storage methods, and associative recall effects these structures seem to be performing.

Unfortunately, definitive experimental evidence about the structure and function of neurological circuitry in the brain is extremely difficult to obtain. Neurons, the brain's computing elements, are very tiny and delicate. It is hard to measure what is happening in them without damaging them or otherwise interfering with the flow of information related to their operation. Techniques do exist for measuring the activity of individual neurons and sometimes even observing the behavior of several neurons at the same time. There are also techniques which make it possible to monitor synchronized changes in the activity of large numbers of neurons.

However, the brain is such a complicated anatomical structure, with such a jumbled interconnection of different kinds of neurons being excited and inhibited by such a broad variety of chemical and electrical stimuli, that it is impossible to infer from these measurements any very sophisticated ideas about what mathematical functions are being computed or what procedures are being executed.

Neurons are as varied in size, shape, and type as trees and bushes in a tropical forest, and often are as closely intertwined and interconnected as a bramble patch overgrown with vines. Many of their most important information processing properties are statistical in nature, and these statistics may apply over ensembles of thousands of neurons.

The situation is further complicated by multiple feedback loops, some of which are confined to small, local clusters of neurons, and others which may thread through several entirely different regions of the brain. The result is that no one has yet been able to construct a clear picture of the overall information processing architecture in the brain. At present there exists no generally accepted theory which bridges the gap between hard neurophysiological measurements and psychological concepts such as perception and cognition.

Nevertheless, there is much that is known with certainty about the structure and function of at least some parts of the brain, particularly in the periphery of the sensory and motor systems. A great deal can be inferred from this knowledge. Furthermore, there is one area, the cerebellar cortex, where the geometry is sufficiently regular to enable researchers to positively identify a


September 1977


March 1979

number of important neurophysiological relationships.

The cerebellum, which is attached to the midbrain portion of the upper spinal cord and nestles up under the visual cortex, as shown in figure 1, is intimately involved with control of rapid, precise, coordinated movements of limbs, hands, and eyes. Injury to the cerebellum results in motor deficiencies, such as overshoot in reaching for objects, lack of coordination, and the inability to execute delicate tasks or track precisely with the eyes.

**The principal input to the cerebellar cortex arrives via mossy fibers.**

During the 1960s, advances in the technology of single cell recordings and electron microscopy made possible an elegant series of experiments by Sir John Eccles and a number of others. These experiments identified the functional interconnections between the principal components in the cerebellar cortex. A brief outline of the structure and function of the cerebellar cortex is shown in figure 2.

The principal input to the cerebellar cortex arrives via mossy fibers (so named because they looked like moss to the early workers who first observed them through a microscope). Mossy fibers carry information from a number of different sources such as the vestibular system (balance), the reticular formation (alerting, the cerebral cortex (sensory-motor activity), as well as from sensor organs which measure such quantities as position of joints, tension in tendons, velocity of contraction of muscles, pressure on skin, etc. It is possible to categorize mossy fibers into at least two classes based on their point of origin: one, those carrying information which may include commands from higher levels in the motor system; and two, those carrying feedback information about the results of motor outputs. Once these two sets of fibers enter the cerebellum, however, they intermingle and become virtually indistinguishable.

The feedback mossy fibers tend to exhibit a systematic regularity in the mapping from point of origin of their information to their termination in the cerebellum. It is thus possible to sketch a map of the body on the surface of the cerebellum corresponding to the origins of feedback mossy fiber information, as shown in figure 3. This map is not sharply defined, however, and has considerable overlap between regions due in part to extensive intermingling and multiple overlapping of terminations of the mossy fibers in the cerebellar granule cell layer. Each mossy fiber branches many times and makes excitatory (+) contact with several hundred granule cells spaced over a region several millimeters in diameter.

Granule cells are the most numerous cells in the brain. It is estimated that there are about $3 \times 10^{10}$ granule cells in the human cerebellum alone. There are 100 to 1000 times as many granule cells as mossy fibers. Each granule cell is contacted by 5 to 12 mossy fibers and gives off a single output axon which rises toward the surface of the cerebellum. When it nears the surface this axon splits into two parts which run about 1.5 mm in opposite directions along the folded ridges of the cerebellum, making contact with a number of different kinds of cells in passage. These axons from the granule cells thus run parallel to each other in a densely packed sheet (hence the name, parallel fibers).

One of the cell types contacted by parallel fibers are Golgi cells (named for their discoverer). These cells have a widely spread dendritic tree and are excited by parallel fibers over a region about 0.6 mm in diameter. Each Golgi cell puts out an axon which branches extensively, making inhibitory (–) contact with up to 100,000 granule cells in its immediate vicinity, including many of the same granule cells which excited it. The dendritic trees and axons of neighboring Golgi cells intermingle so as to blanket the entire granular layer with negative feedback. The general effect is that of an automatic gain control on the level of activity in the parallel fiber sheet.

It is thought that the Golgi cells operate such that only a small and controlled percentage (perhaps as little as 1 percent or

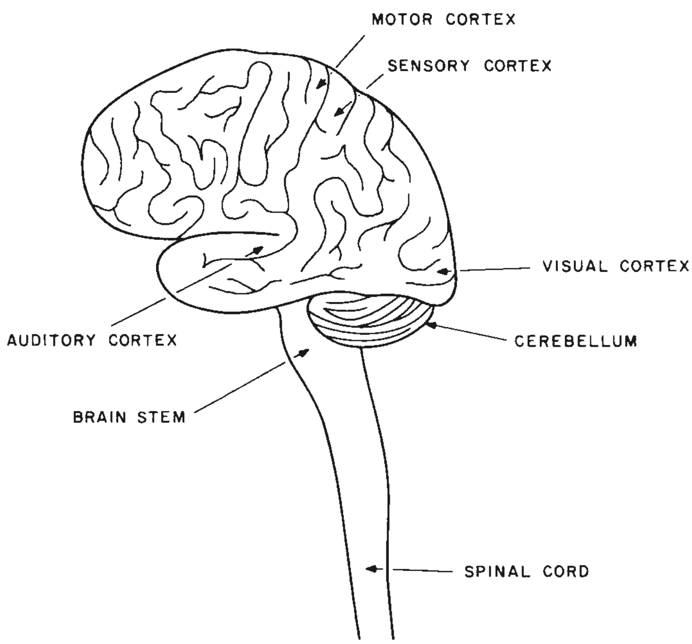



*Figure 1. Side view of human brain showing the cerebellum attached to the brain stem and partially hidden by the visual cortex.*

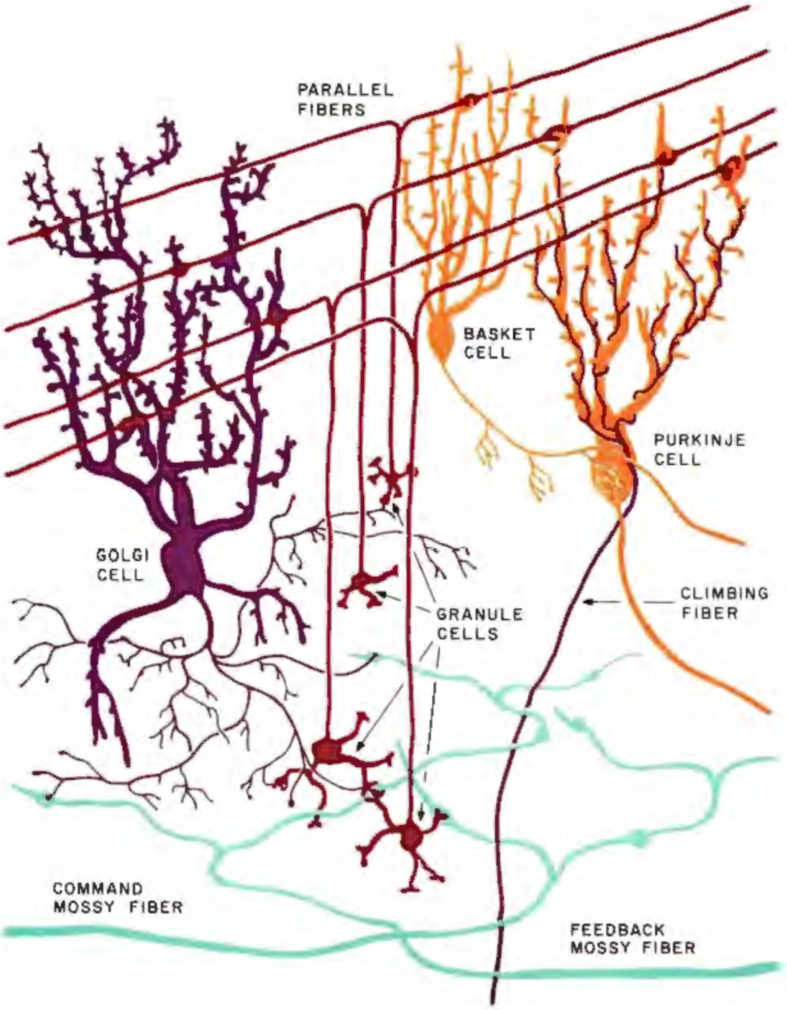



*Figure 2. The principal cells and fiber systems of the cerebellar cortex. Command and feedback information arrives via mossy fibers, each of which make excitatory (+) contact with several hundred granule cells. Golgi cells sample the response of the granule cells via the parallel fibers and suppress by inhibitory (–) contacts all but the most highly excited granule cells. Purkinje cells are the output of the cerebellar cortex. They sum the excitatory (+) effect of parallel fibers through weighted connections. They also receive inhibitory (–) input from parallel fibers via basket cell inverters. The strengths of these weights determine the transfer function of the cerebellar cortex. Climbing fibers are believed to adjust the strength of these weights so as to train the cerebellum.*

less) of the granule cells are allowed above threshold at any one time, regardless of the level of activity of the mossy fiber input. Any particular pattern of activity on the mossy fiber input will produce a few granule cells which are maximally excited, and a great many others which are less than maximally stimulated. The Golgi cells suppress the outputs of all but the few maximally stimulated granule cells. The result is that every input pattern (or vector) is transformed by the granule layer into a small, and relatively fixed percentage, or subset, of parallel fibers which are active.

These active parallel fibers not only contact Golgi cells, but make excitatory contact with Purkinje cells (named for their discoverer) and basket and stellate cells (named for their shapes) through weighted connections (synapses). Each Purkinje cell performs a summation over its inputs and produces an output which is the output of the cerebellar cortex. The basket and stellate cells are essentially inverters which provide the Purkinje with negative weights that are summed along with the positive weights from parallel fibers.

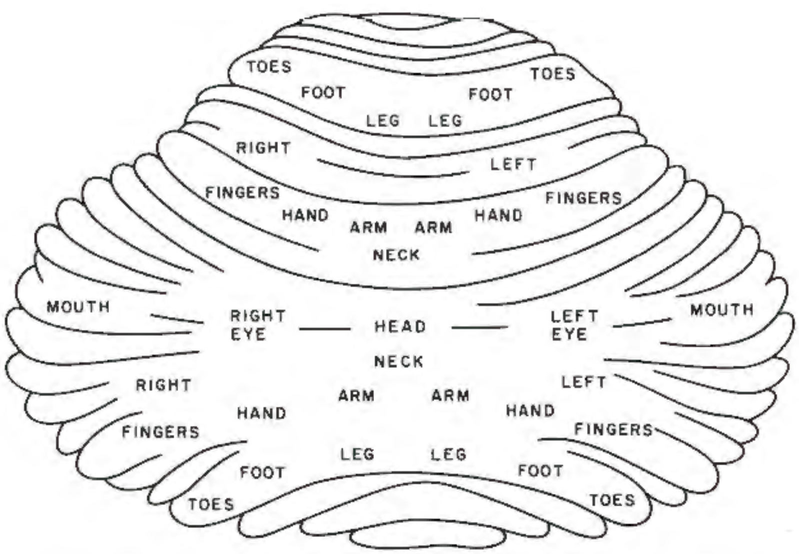



*Figure 3. A map of the surface of the cerebellar cortex showing the point of origin of mossy fiber feedback and ultimate destination of Purkinje cell output.*

The Golgi cells suppress the outputs of all but the few maximally stimulated granule cells.

A second set of fibers entering the cerebellar cortex are the climbing fibers, so named because they climb all over the Purkinje cells like ivy on a tree. There is typically one climbing fiber for each Purkinje cell. It is believed that these climbing fibers have some role in adjusting the strength of the weighted synaptic connections with the parallel fibers, so as to alter the Purkinje output. Climbing fibers are thus hypothesized to provide the information required for learning.

The availability of such detailed knowledge regarding the structure and function of the various cell and fiber types in the cerebellum has led a number of theoreticians to propose mathematical models to explain the information processing characteristics of the cerebellum. One model was developed independently in Great Britain by David Marr and in the United States by myself. The general outlines of this model are shown in figure 4. My further work has produced the more abstract version illustrated in figure 5, as well as a mathematical formalism called the CMAC (Cerebellar Model Arithmetic Computer).

CMAC is defined by a series of mappings:

$$S \rightarrow M \rightarrow A \rightarrow p$$

where:

S is an input vector;
M is the set of mossy fibers used to encode S;
A is the set of granule cells contacted by M;
p is an output value.

The overall mapping:

$$S \rightarrow p$$

has all of the properties of a function:

$$p = h\ (S)$$

as described in part 1. A set of L CMACs operating on the same input produces a mapping:

$$S \rightarrow P$$

which has the properties of the function:

$$P = H\ (S).$$

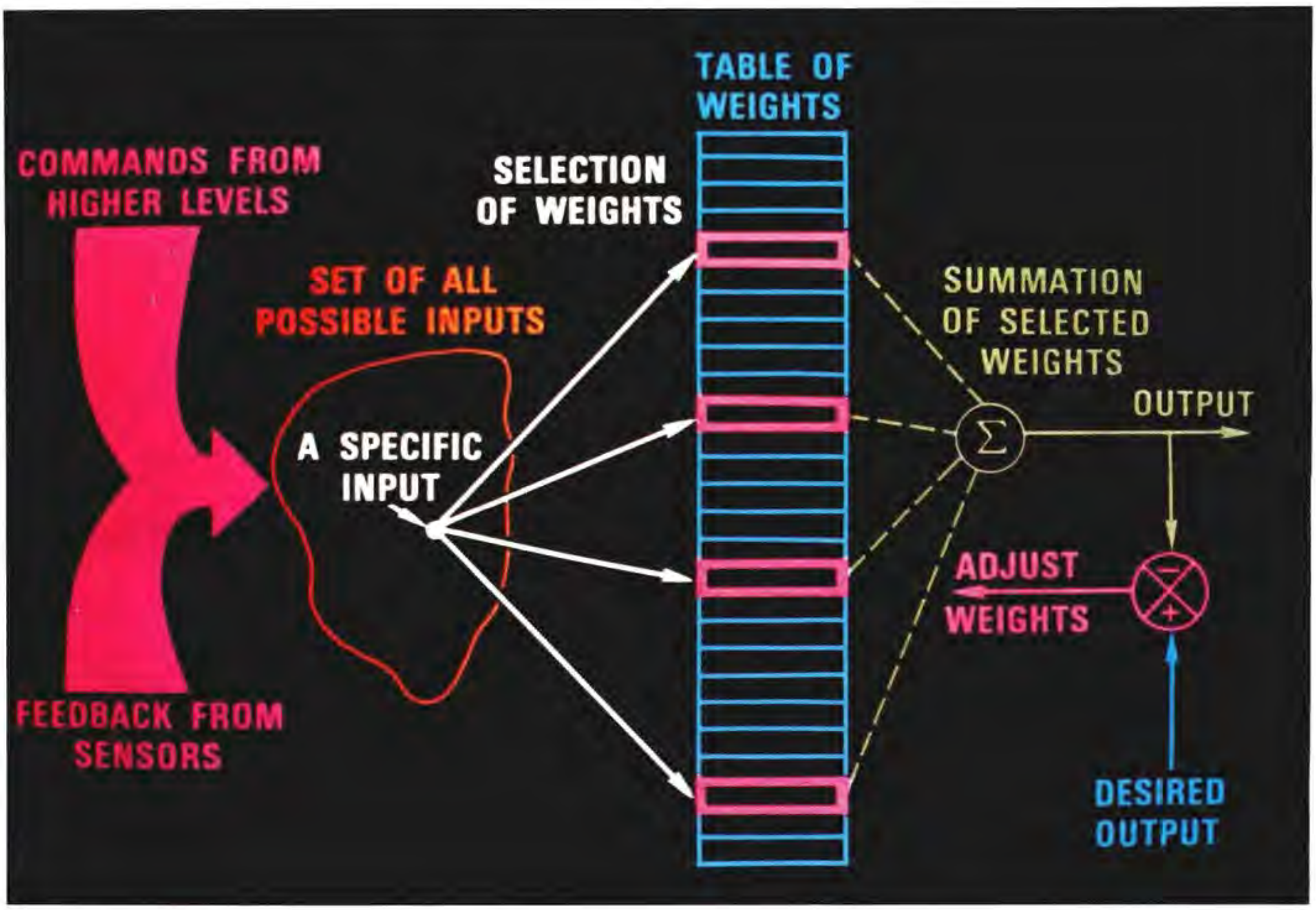
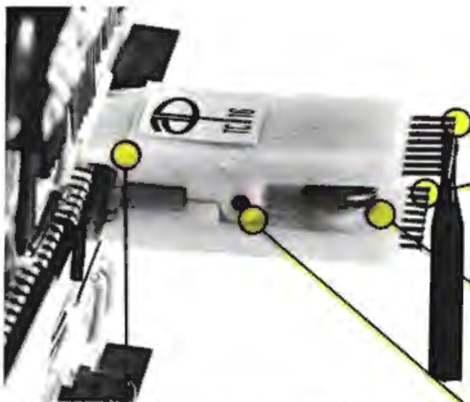



*Figure 4. A theoretical model of the cerebellum.*

# NEW.

## Meet Super Grip II, the great new test clip from A P Products.



New narrow-nose design makes it easy to attach on high-density boards. And now you can test ICs with only .040" between opposing legs.

New "duck bill" contacts are flat, won't roll off IC leads.

Open-nose construction enables probe at IC leg.

Pin rows are offset for easy attachment of probes.

Contacts are gold-plated phosphor bronze. "Contact comb" construction separates contacts with precision. No shorts.

Heavy-duty, industrial-grade springs for firm contact pressure—and a good grip when pulling ICs. They'll keep their spring indefinitely. No intermittents.

Steel pivot pin. Engineering-grade thermoplastic body molded around contact pins. Made to last!


New button-head pins keep probes from sliding off. (Straight pin models for logical connections.)

The new A P Super Grip II is, without question, the best way there is to troubleshoot DIP ICs.

You get positive contacts. No intermittents. No shorts. Ever.

So it's endlessly useful to you—and it's built to stay useful indefinitely.

Try one. You'll find 8, 14, 16, 16 LSI, 18, 20, 22, 24, 28, 36 and 40-pin models at your nearby A P store. (Make sure it's your A P store.)

Need the address? Call (toll-free) 800-321-9668. And ask for our complete A P catalog, The Faster and Easier Book.

A P PRODUCTS INCORPORATED
Box 110D • 72 Corwin Drive
Painesville, Ohio 44077
Tel. 216/354 2101
TWX: 810-425-2250

**Faster and Easier is what we're all about.**

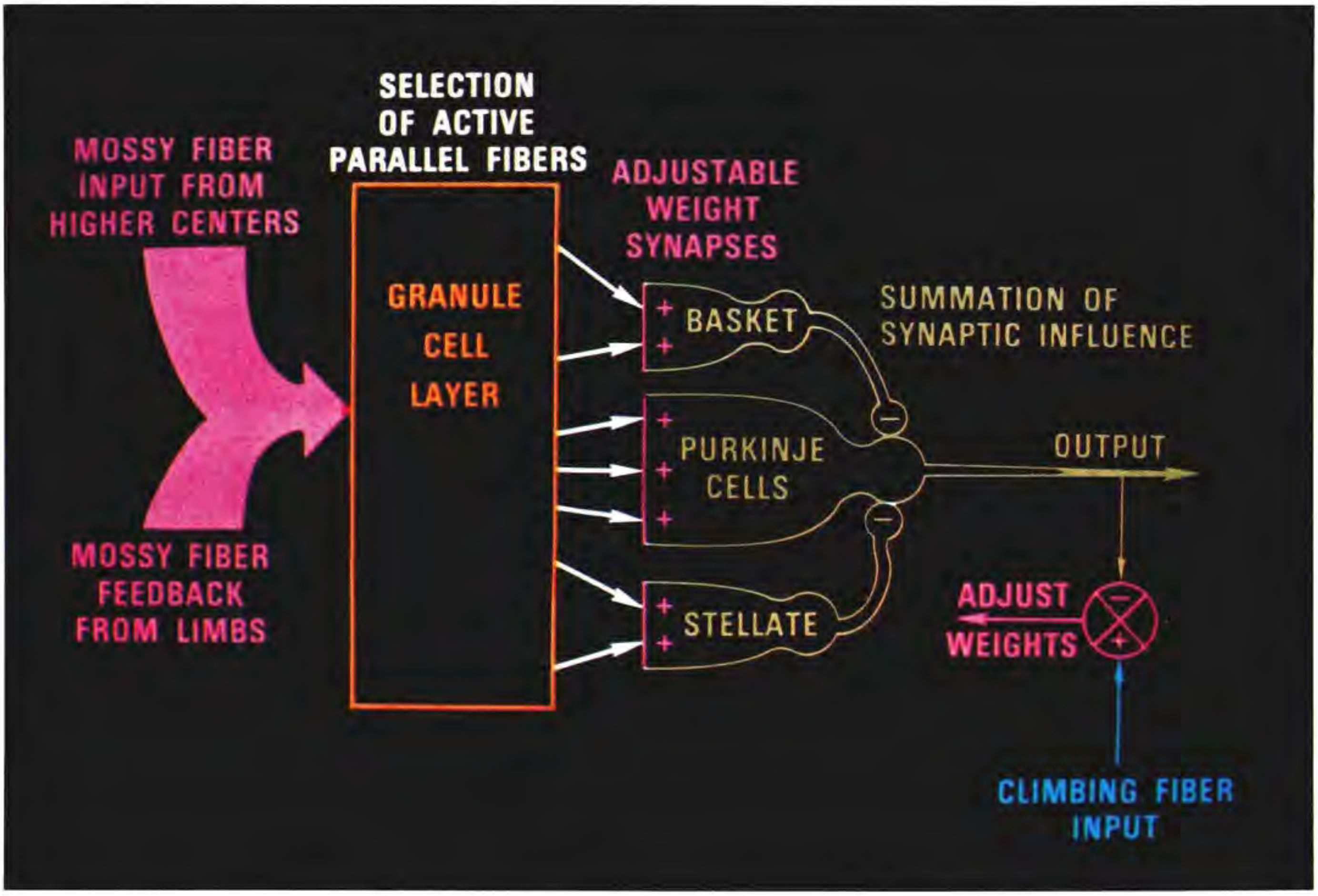



*Figure 5. A schematic representation of CMAC (Cerebellar Model Arithmetic Computer).*

**Pulse frequency and phase modulation are subject to quantization noise and bandwidth limitations.**

We may describe the information encoded by mossy fibers as a vector $\mathbf{S} = \mathbf{C} + \mathbf{F}$ where:

$\mathbf{C} = (s_1, s_2, \ldots, s_i)$ is a vector, or list, of command variables;

and

$\mathbf{F} = (s_{i+1}, \ldots, s_N)$ is a vector, or list, of feedback variables.

+ is an operator denoting the combination of two vectors defined by two lists of variables into a single vector or list of variables.

That is:

$$\mathbf{S} = \mathbf{C} + \mathbf{F}$$

means that $\mathbf{S} = (s_1, s_2, \ldots, s_i, s_{i+1}, \ldots, s_N)$. Some of the elements of the command vector **C** may define symbolic motor commands such as <REACH>, <PULL BACK>, <PUSH>, etc. The remainder of the elements in **C** define arguments, or modifiers, such as the velocity of motion desired, the force required, the position of the terminal point of a motion, etc. Elements of the feedback vector **F** may represent physical parameters such as the position of a particular joint, the tension in a tendon, the velocity of contraction of a muscle, the pressure on a patch of skin, and so on.

### Mapping S → M

The vector components of **S** must be transmitted from their various points of origin to their destination in the cerebellar granular layer. Distances may range from a few inches to over a foot. This presents a serious engineering problem because mossy fibers, like all nerve axons, are noisy, unreliable, and imprecise information channels with limited dynamic range. Pulse frequency and pulse phase modulation (which the brain uses for data transmission over long distances) are subject to quantization noise and are bandwidth limited. Nerve axons typically cannot transmit pulse rates above two or three hundred pulses per second. Nevertheless, high resolution high bandwidth data is required for precise control of skilled actions.

The brain solves this problem by encoding each of the high precision variables to be transmitted so that it can be carried on a large number of low precision channels. Many mossy fibers are assigned to each input variable such that any one fiber conveys only a small portion of the information content of a single variable.

The nature of this encoding is that any particular mossy fiber will be maximally active over some limited range of the variable that it encodes, and less than maximally active over the rest of its variable's range. For example, the output of the mossy fiber labeled a in figure 6 is maximally active whenever the elbow joint is between 90° and 120° and is less than maximally active for all other elbow positions. The mossy fiber labeled b in figure 6 is maximally active whenever the elbow angle is greater than 160°. Now if there exists a large number of mossy fibers whose responses have a single maximum but which are maximally active over different intervals, it is then possible to tell the position of the elbow quite precisely by knowing which mossy fibers are maximally active. For example, in figure 7 the fact that mossy fibers a, b, and c are maximally active indicates that the elbow joint is between 118° and 120°.

The CMAC models this encoding scheme in the following way: define $m_i$ to be the set of mossy fibers assigned to convey the value of the variable $s_i$; define $m_i{}^*$ to be the mossy fibers in $m_i$ which are maximally stimulated by a particular value of $s_i$. If for every value of $s_i$ over its range there exists a unique set $m_i{}^*$ of maximally active mossy fibers, then there is a mapping $s_i \rightarrow m_i{}^*$ such that knowing $m_i{}^*$ (ie: which fibers in $m_i$ are maximally active) tells us what is the value of $s_i$. If such a mapping is defined for every component $s_i$ in the vector **S** then we have a mapping:

$$S \rightarrow M = \begin{cases} s_1 \rightarrow m_1{}^* \\ s_2 \rightarrow m_2{}^* \\ \cdot \\ \cdot \\ \cdot \\ s_N \rightarrow m_N{}^* \end{cases}$$

where M is the set of all mossy fibers which encode the variables in the vector **S**.

In CMAC each of the $s_i \rightarrow m_i{}^*$ mappings may be defined by a set of K quantizing functions ${}^iC_1$, ${}^1C_2, \ldots, {}^iC_k$ each of which is offset by a value of 1/K times the quantizing interval. An example of this is given in figure 8 where K = 4 and N = 2. Component $s_1$ is represented along the horizontal axis, and the range of $s_1$ is covered by four quantizing functions:

$${}^1C_1 \ \{A, B, C, D, E\}$$
$${}^1C_2 \ \{F, G, H, J, K\}$$
$${}^1C_3 \ \{M, N, P, Q, R\}$$
$${}^1C_4 = \{S, T, V, W, X\} \ .$$

Each quantizing function is offset from the previous one by one resolution element. For every possible value of $s_i$ there exists a unique set $m_1{}^*$ consisting of the set of values produced by the K quantizing functions. For example (in figure 8), the value $s_1 = 7$ maps into the set $m_1{}^* = \{B, H, P, V\}$.

A similar mapping is also performed on $s_2$ by the set of quantizing functions:

$${}^2C_1 = \{a, b, c, d, e\}$$
$${}^2C_2 = \{f, g, h, j, k\}$$
$${}^2C_3 = \{m, n, p, q, r\}$$
$${}^2C_4 = \{s, t, v, w, x\} \ .$$

For example, the value $s_2 = 10$ maps into the set $m_2{}^* = \{c, j, q, v\}$. Now, if the $s_1$ component in figure 8 corresponds to the position of the elbow joint, the mossy fiber labeled B will be maximally active whenever

*Text continued on page 64*

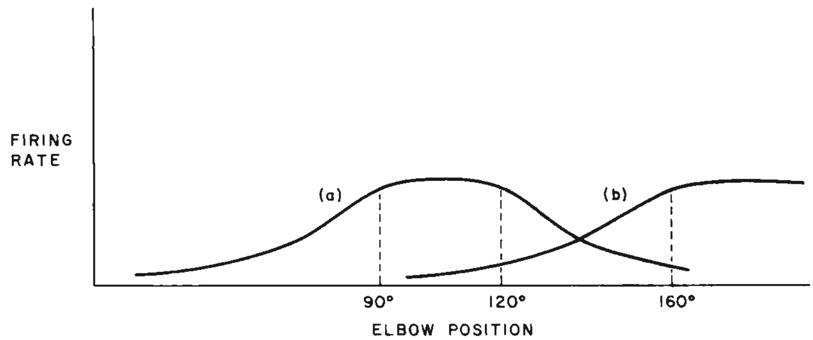



*Figure 6. Typical responses of mossy fibers to the sensory variable they encode.*

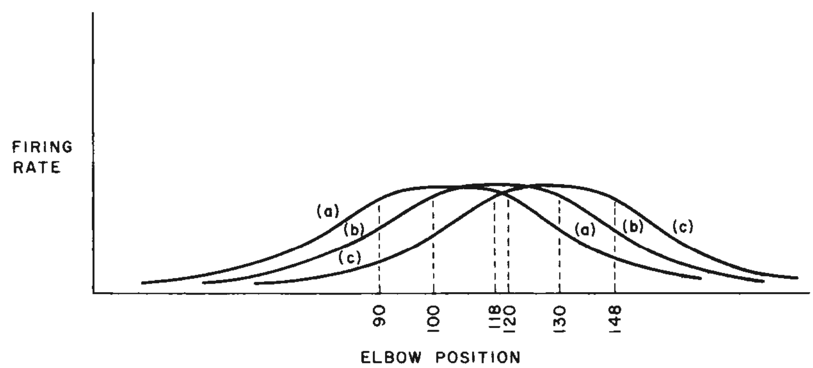



*Figure 7: Three different mossy fibers encoding a single sensory variable (elbow position). All three fibers maximally active simultaneously indicates that the elbow lies between 118° and 120°.*

Circle 281 on inquiry card.

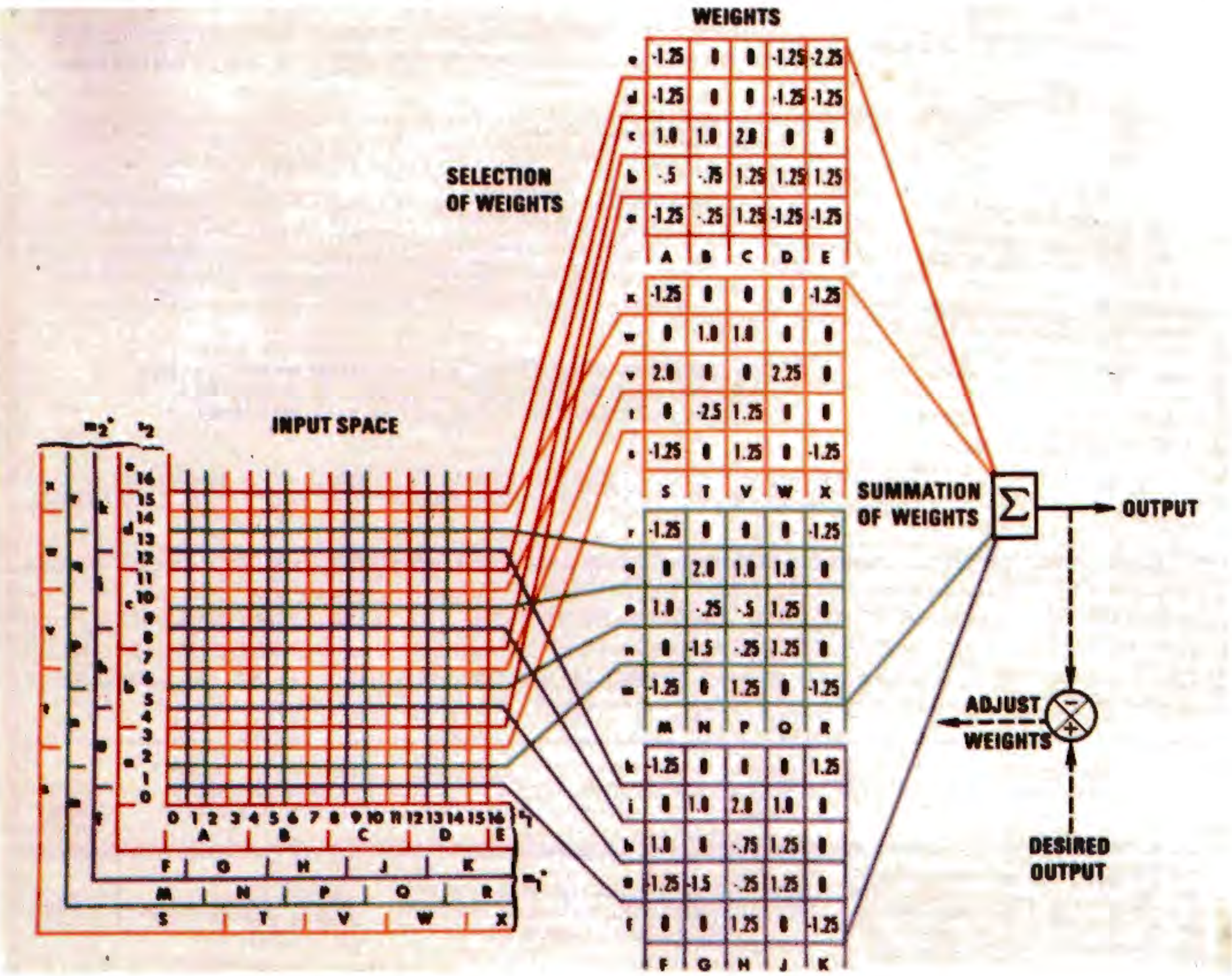
*Text continued from page 61:*
the elbow is between 4 and 7, and less than maximally active whenever the elbow position is outside that region. Similarly, the mossy fiber labeled H is maximally active when the elbow is between 5 and 8, the fiber P maximally active between 6 and 9, and V between 7 and 10, etc. The combination of mossy fibers in the set $m_1{}^* = \{B, H, P, V\}$ thus indicates that the variable $s_1 = 7$. If $s_1$ changes one position from (from 7 to 8, for example), the mossy fiber labeled B will drop out of the maximally active set $m_1{}^*$ to be replaced by another, labeled C.

### Encoding Advantages

This encoding scheme has a number of advantages. The most obvious is that a single precise variable can be transmitted reliably over a multitude of imprecise information channels. The resolution (or information content) of the transmitted variable depends on the number of channels. The *more* mossy fibers dedicated to a particular variable, the greater the *precision* with which it is represented.

A second equally important result is that small changes in the value of the input variable $s_i$ have no effect on most of the elements in $m_i{}^*$. This leads to a property known as *generalization*, which is crucial for learning and recall in a world where no two situations are ever exactly the same. In CMAC the extent of the neighborhood of generalization along each variable axis depends on the resolution of the CMAC quantizing functions. In the brain this corresponds to the width of the maximally active region of the mossy fibers.

*Figure 8. A simple two variable CMAC with four quantizing functions on each variable. A detailed explanation is in the text.*



# NOW **YOU** CAN AFFORD
## THE WORLD FAMOUS
## Diablo Printer Terminal

IPSI PROUDLY ANNOUNCES A PRICE BREAKTHROUGH ON DAISY WHEEL PRINTER TERMINALS

**Featuring: RS 232** compatibility at 110, 150, 300, and 1200 baud: **Print speed** of 45 char/sec: **Graphics** capability: Over 30 **print wheels** from 10 to 15 char/inch: 5 ribbons in color, cloth, film or two color: **Form length** adjustable: **Spacing** adjustable from 120 pitch to .1 pitch: Line spacing adjustable from 48 lines/in. to .4 line/in.: Identical in every respect to it's high priced sister the **Diablo HyTerm II*** model 1620.

**Price:** $2,595.00 each in single units, quantity discounts available.

**Service:** Major cities in the U.S. and Canada.

You have probably waited a long time to purchase your computer printer at a reasonable price. Wait no longer! The Diablo 1620 lists for $3,240.00 we sell the IPSI 1622 for $2,595.00; save well over $600.00. For immediate action phone in your order collect and we will ship your terminal the same day.

We also Stock Diablo 1345A (Hytype II*) mechanisms for immediate shipment at the low price of $1,150.00. Order one now and build your own terminal using your own CPU, or do-it-yourself by buying our 1622 kit and assemble your own terminal within 10 hours for $2,295.00

If you've always wanted to turn your computer onto word processing, this is the only way to go. Ask for our free word processing software.

# IPSI
INTERNATIONAL PERIPHERAL SYSTEMS, INC.
1849 N. HELM
FRESNO, CALIFORNIA 93727
CALL (209) 252-3635

***Hyterm and HyType are registered trademarks of Xerox Corp.**

Circle 174 on inquiry card

## M → A Mapping

**The number of mossy fibers dedicated to a variable determines precision of its representation.**

Just as we can identify (or name) mossy fibers by the input variables they encode, so we can identify granule cells by the mossy fibers which provide them with input. Each granule cell receives input from several different mossy fibers, and no two granule cells receive input from the same combination of mossy fibers. This means that we can compute a unique name (or address) for each granule cell by simply listing the mossy fibers which contact it. For example, a granule cell contacted by two mossy fibers B and c can be named (or addressed) Bc.

In the CMAC example in figure 8, 25 granule cells are identified by their contacts with mossy fibers from the quantizing functions ${}^1C_1$ and ${}^2C_1$. 25 other granule cells are identified by ${}^1C_2$ and ${}^2C_2$, 25 by ${}^1C_3$ and ${}^2C_3$, and 25 more by ${}^1C_4$ and ${}^2C_4$. There are, of course, many other possible combinations of mossy fiber names which might be used to identify a much larger number of granule cells. For this simple example, however, we will limit our selection to the permutation of corresponding quantizing functions along each of the coordinate axes. This provides a large and representative sample which uniformly spans the input space. Furthermore, this particular naming algorithm is simple to implement either in software or hardware.

We can define A to be the set of all granule cells identified by their mossy fiber inputs. Of course, all of the granule cells in A are not active at the same time. As was previously noted, most granule cells are inhibited from firing by Golgi cell gain control feedback. Only the small percentage of granule cells whose input mossy fibers are *all* maximally active can rise above threshold. We will define the set of active granule cells as A*.

Since we already know which mossy fibers are maximally active (ie: those mossy fibers in the sets $m_i$*), we can compute names of granule cells in A*. For example, in figures 8 and 10, if $s_1 = 7$ and $s_2 = 10$, then $m_1^* = \{B, H, P, V\}$ and $m_2^* = \{c, j, q, v\}$. The active granule cells in A* can now be computed directly as $A^* = \{Bc, Hj, Pq, Vv\}$. All other granule cell names in the larger set A involve at least one mossy fiber which is not maximally active (ie: not in $m_1$* or $m_2$*).

Note that, as illustrated in figure 9, the granule cell Bc will be active as long as the input vector remains in the region of input space $4 \leqslant s_1 \leqslant 7$ and $8 \leqslant s_2 \leqslant 11$. Thus, the generalizing property introduced by the S → M mapping carries through to the naming of active granule cells. A particular granule cell is active whenever the input vector **S** lies within some extended region, or neighborhood, of input space. Other granule cells are active over other neighborhoods. These neighborhoods overlap, but each is offset from the others so that for any particular input S, the neighborhoods in A* all overlap at only one point, namely the point defined by the input vector. This is illustrated in figure 10. If the input vector moves one resolution element in any direction, for example, from (7, 10) to (8, 10), one active granule cell (Bc) drops out of A* to be replaced by another (Cc).

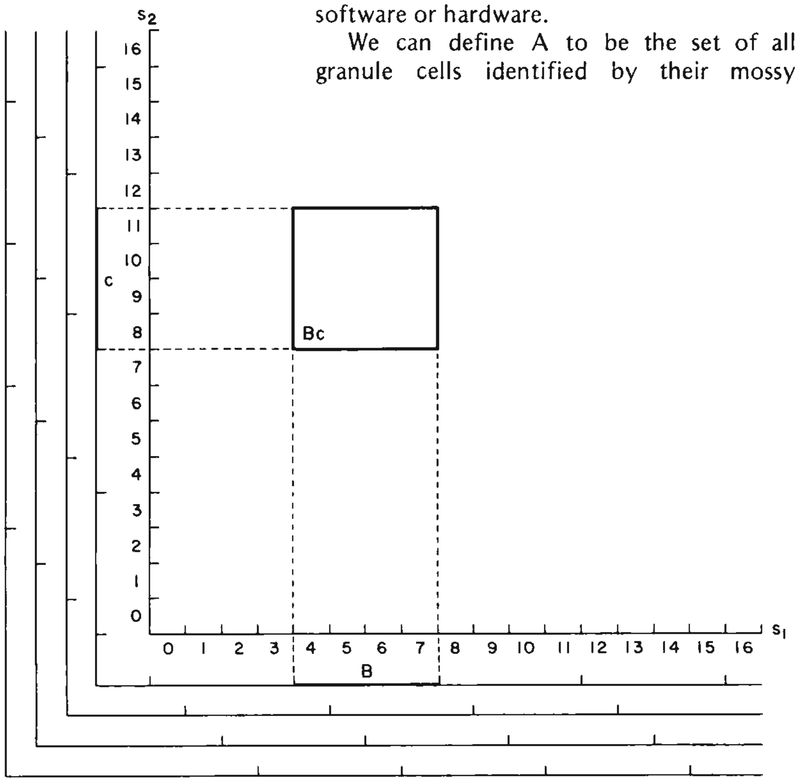



*Figure 9. The weight Bc will be selected as long as the CMAC input vector lies in the region bounded by $4 \leqslant s_1 \leqslant 7$, $8 \leqslant s_2 \leqslant 11$.*

## A → p Mapping

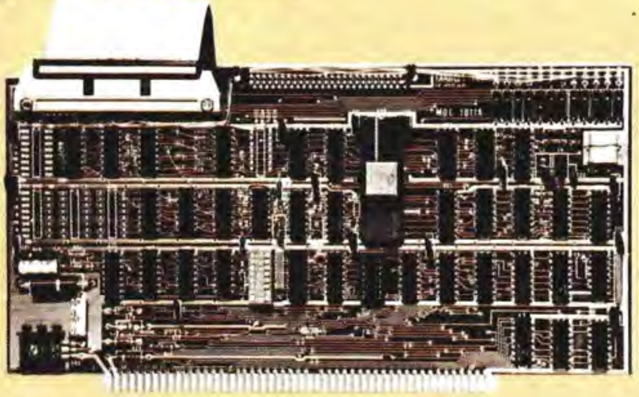
Granule cells give rise to parallel fibers which act through weighted connections on the Purkinje output cell, varying its firing rate. Each cell in A is associated with a weight which may be positive or negative. Only the cells in A* have any effect on the Purkinje output cell. Thus, the Purkinje output sums only the weights selected (or addressed) by A*. This sum is the CMAC output scalar variable p. For example, in figure 8, **S** = (7, 10) maps into $A^* = \{Bc, Hj, Pq, Vv\}$ which selects the weights:

$$W_{Bc} = 1.0$$
$$W_{Hj} = 2.0$$
$$W_{Pq} = 1.0$$
$$W_{Vv} = 0.0.$$

# The TARBELL Connection

In an effort to offer products that meet the continually changing demands of the microcomputer industry, TARBELL ELECTRONICS is pleased to offer immediate delivery of these quality components and operating software. All TARBELL products are available from computer store dealers everywhere.

## Tarbell Floppy Disk Interface



- Plugs directly into your IMSAI or ALTAIR and handles up to 4 standard single drives in daisy-chain.
- Operates at standard 250K per second on normal disk format capacity of 256K bytes.
- Works with modified CP/M Operating System and BASIC-E Compiler.
- Hardware includes 4 extra IC slots, built-in phantom bootstrap and on-board crystal clock. Uses WD 1771 LSI chip.
- Full 6-month warranty and extensive documentation.
- Kit $190 . . . . . . . Assembled $265.

Specify drive for assembled units. Complete disk subsystems with operating software available. Please inquire for details.

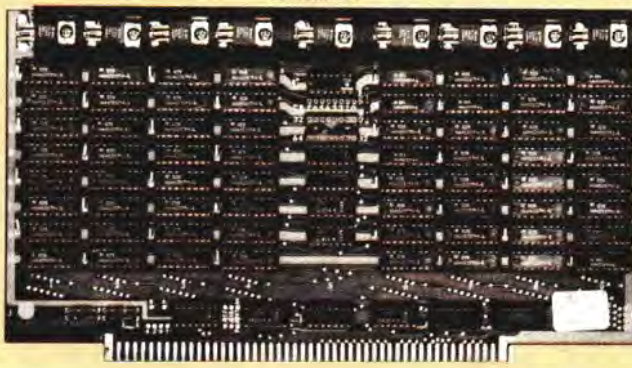
## Tarbell Disk BASIC

- Runs on 8080, 8085 or Z80
- Searches a file quickly for a string.
- Up to 64 files open at once.
- Random Access.
- Assignment of I/O.
- Alphanumeric line labels allowed.
- Read and Write string or numeric data.
- Unlimited length of variable names and strings.
- Procedures with independent variables.
- Number system 10 digits BCD integer or floating point.
- Chain to another program.
- Cause programs to be appended onto programs already in memory.
- Cause interpreter to enter edit mode using 15 single character edit commands.

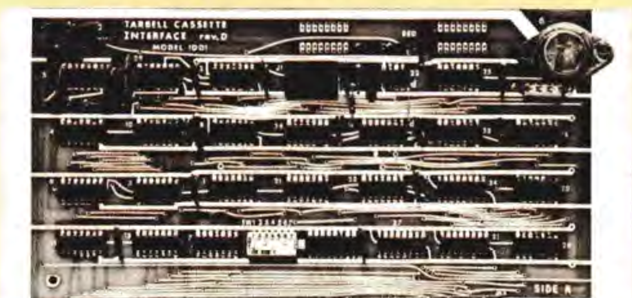
Occupies 24K of RAM. Tarbell BASIC on CP/M Disk . . . . . . . . . . . . . .$48.
Source on paper or CP/M Disk . . .$25.
CP/M and BASIC-E on disk with manuals. . . . . . . . . . . . . . . . .$100.

## Tarbell 32K RAM Memory

- 32K Static Memory
- S-100 Bus Connector
- 9 regulators provide excellent heat distribution.
- Extended addressing (bank switching.)
- Phantom line.
- Low power requirement.
- 20-Page operating manual.
- Full 1-year warranty.
- Assembled and tested full price only $625
- 16K version also available, assembled and tested only $390.



## Tarbell Cassette Interface



- Plugs directly into your IMSAI or ALTAIR.
- Fastest transfer rate: 187 (standard) to 540 bytes/second.
- Extremely reliable—Phase encoded (self-clocking).
- 4 extra status lines, and 4 extra control lines.
- 37-page manual included.
- Device code selectable by DIP-switch.
- Capable of generating Kansas City tapes.
- No modification required on audio cassette recorder.
- Complete kit $120 . . . . . . . . . . . . . . . Assembled $175.
- Manual may be purchased separately . . . . . . . . . . . . . $8.
- Full 6-month warranty on kit and assembled units.

## Tarbell Cassette BASIC

Includes most features of ALTAIR Extended BASIC, plus these added features:

- Assignment of I/O.
- Alphanumeric line labels.
- Unlimited length of variable names and strings.
- Number system 10 digits BCD integer or floating point.
- Procedures with independent variables.
- Read and Write string data.
- Multi-file capability.

Full price with complete documentation . . . . . . . . . . . .$48.

Prepaid, COD, or cash only. California residents please add 6% sales tax.

*ALTAIR is a trademark/tradename of Pertec Computer Corporation*
*CP/M is a trademark/tradename of Digital Research*

Tarbell Electronics

950 DOVLEN PLACE • SUITE B
CARSON, CALIFORNIA 90746
(213)538-4251 • (213) 538-2254

These weights are summed to produce the output:

$$p = 4.0.$$

Thus the input $\mathbf{S} = (7, 10)$ produces the output $h(\mathbf{S}) = 4$.

In figure 8 four weights are selected for every $\mathbf{S}$ vector in input space. Their sum is the value of the output p. As the input vector moves from any point in input space to an adjacent point one weight drops out to be replaced by another. The difference in value of the new weight minus the old is the difference in value of the output at the two adjacent points. Thus, the difference in adjacent weights is the partial derivative (or partial difference) of the function at that point. As the input vector $\mathbf{S}$ moves over the input space, a value p is output at each point. We can therefore say that the CMAC computes the function:

$$p = h(\mathbf{S}).$$

The particular function h computed depends on the particular set of values stored in the table of weights. For example, the set of weights shown in figure 8 computes the function shown in figure 11.

In the cerebellum there are many Purkinje cells which receive input from essentially the same mossy fibers. Thus, there are many CMACs all computing on the same input vector $\mathbf{S}$. We can therefore say that a set of L CMACs computing on the same input vector produces a vector mapping:

$$\mathbf{P} = H(\mathbf{S}).$$

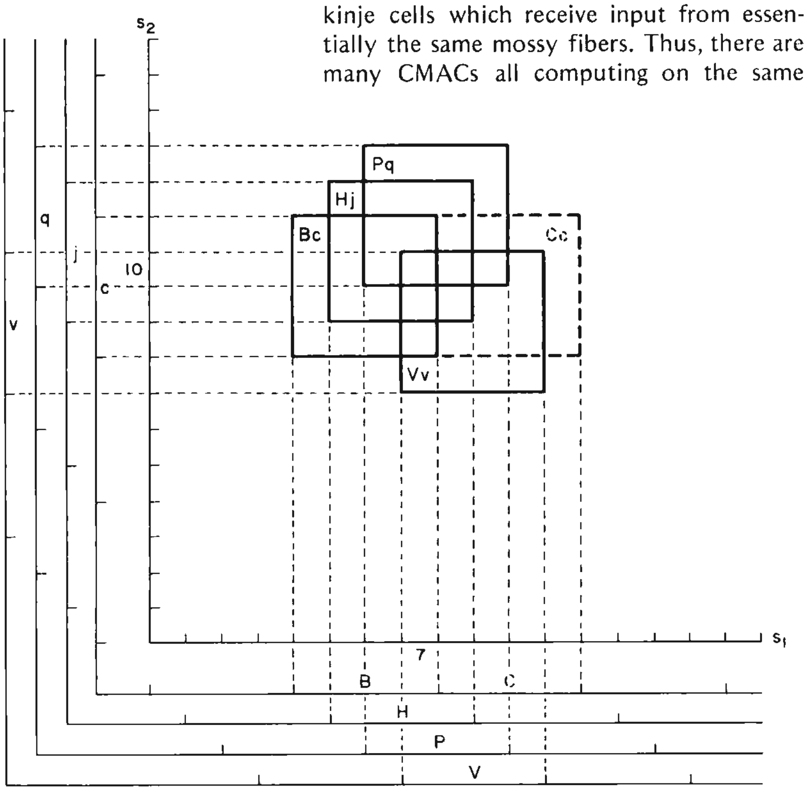



*Figure 10. The input vector $(s_1, s_2) = (7, 10)$ selects weights Bc, Hj, Pq, and Vv. These all overlap only at the point (7, 10). If the input vector $(s_1, s_2)$ moves to (8, 10) the weight Bc will drop out to be replaced by Cc.*

**Data Storage in CMAC**

One of the most fascinating, intensively studied, and least understood features of the brain is memory, and how data is stored in memory. In the cerebellum each Purkinje cell has a unique fiber, a climbing fiber, which is believed to be related to learning. Fibers from an area called the *locus coeruleous* have recently been discovered which appear to be related to learning. In addition, a number of hormones have been shown to have profound effects on learning and retention of learned experiences.

While the exact mechanism (or mechanisms) for memory storage are as yet unknown, the cerebellar model upon which CMAC is based hypothesizes that climbing fibers carry error correction information which "punishes" synapses that participate in erroneous firing of the Purkinje cell. The amount of error correction that occurs at any one experience may depend on factors such as the state of arousal or emotional importance attached by the brain's evaluation centers to the data being stored during the learning process.

Cerebellar learning is modeled in CMAC by the following procedure:

- Assume that $\hat{H}$ is the function we want CMAC to compute. Then $\hat{\mathbf{P}} = \hat{H}(\mathbf{S})$ is the desired value of the output vector for each point in the input space.
- Select a point $\mathbf{S}$ in input space where $\hat{\mathbf{P}}$ is to be stored. Compute the current value of the function at that point $\mathbf{P} = H(\mathbf{S})$.
- For every element in:

$$\mathbf{P} = (p_1, p_2, \ldots, p_L)$$

and in:

$$\hat{\mathbf{P}} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L)$$

if:

$$|\hat{p}_i - p_i| \leq \xi_i$$

where $\xi$ is an acceptable error, then do nothing; the desired value is already stored. ($|\hat{p}_i - p_i|$ is the absolute value of $\hat{p}_i - p_i$.)

However, if $|\hat{p}_i - p_i| > \xi_i$ then add $\Delta_i$

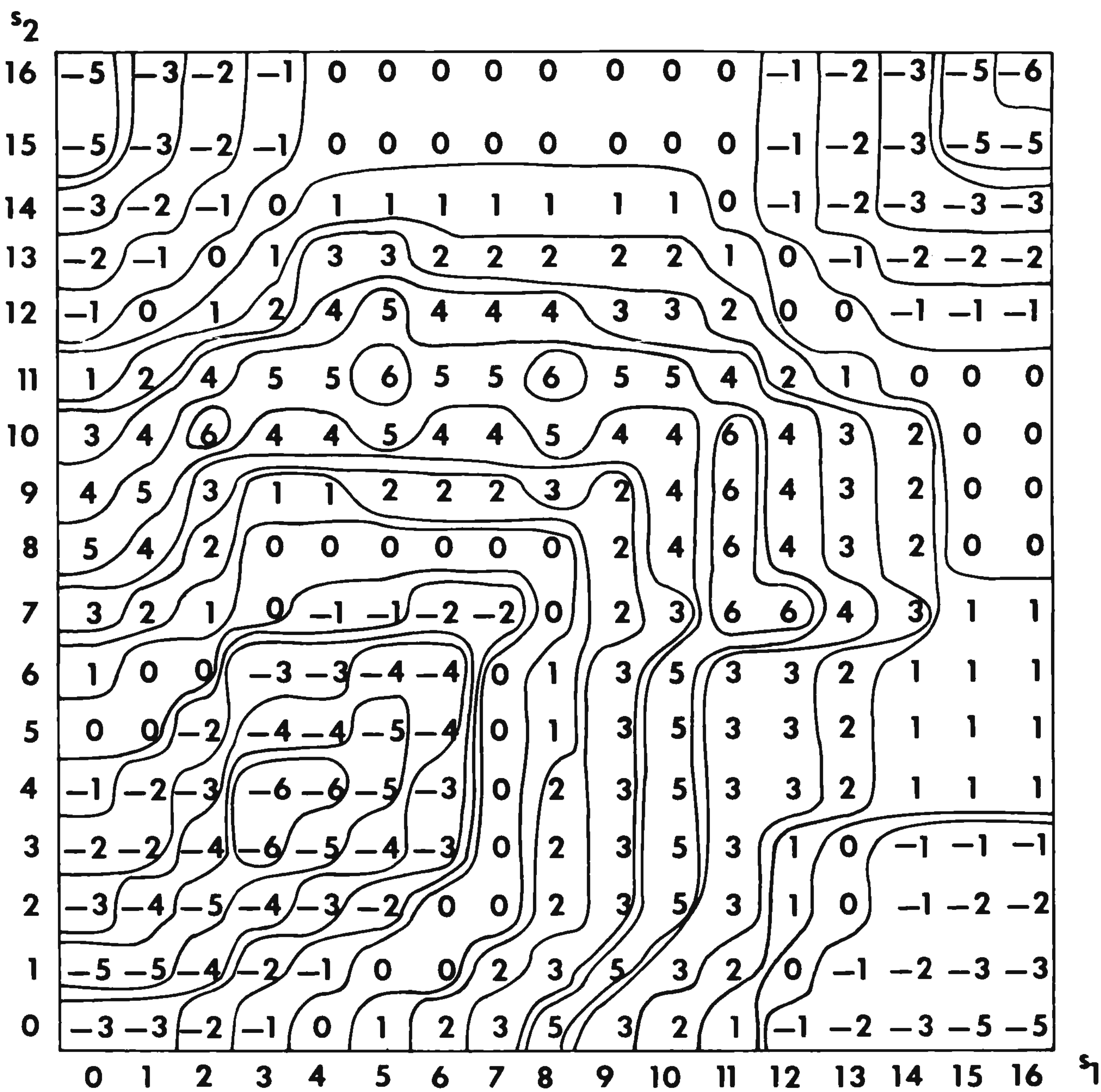



*Figure 11. The particular set of weights shown in Figure 8 will compute the function shown here.*

to every weight which was summed to produce $p_i$ where:

$$\Delta_i = g\left(\frac{\hat{p}_i - p_i}{|A^*|}\right) \qquad (1)$$

$|A^*|$ is the number of weights in the set $A^*$ which contributed to p, and g is a gain factor which controls the amount of error correction produced by one learning experience.

If $g = 1$, then CMAC produces oneshot learning which fully corrects the observed error in one data storage operation. If $0 < g < 1$, then each learning experience moves the output $p_i$ only in the direction of the desired value $\hat{p}_i$. More than one memory storage operation is then required to achieve correct performance.

An example of how an arbitrary function such as:

$$\hat{p} = (\sin x)(\sin y)$$

where:

$$x = 2\pi\, s_1/360$$

and:

$$y = 2\pi\, s_1/360$$

12a

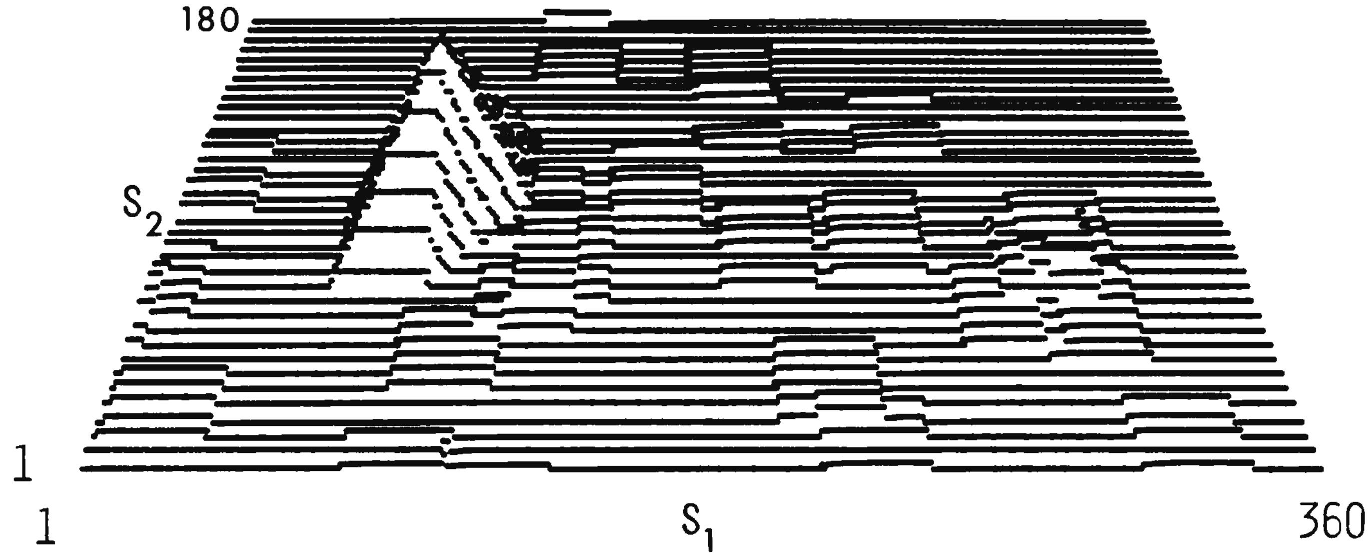



*Figure 12. The effect of training CMAC on the function $\hat{p}$ = sin ($2\pi$ $s_1$ /360) sin ($2\pi$ $s_2$ / 360). (a) One training at ($s_1$, $s_2$) = (90, 90). (b) A second training at ($s_1$, $s_2$) = (270, 90). (c) Training at 16 points along a trajectory defined by $s_1$ = 90. (d) Training at 175 selected points scattered over the input space.*

can be stored in CMAC is shown in figure 12. In this example the input is defined with unity resolution over the space $0 < s_1 \leqslant 360$ and $0 < s_2 \leqslant 180$, and the number of weights selected by each input is $|A^*| = 32$.

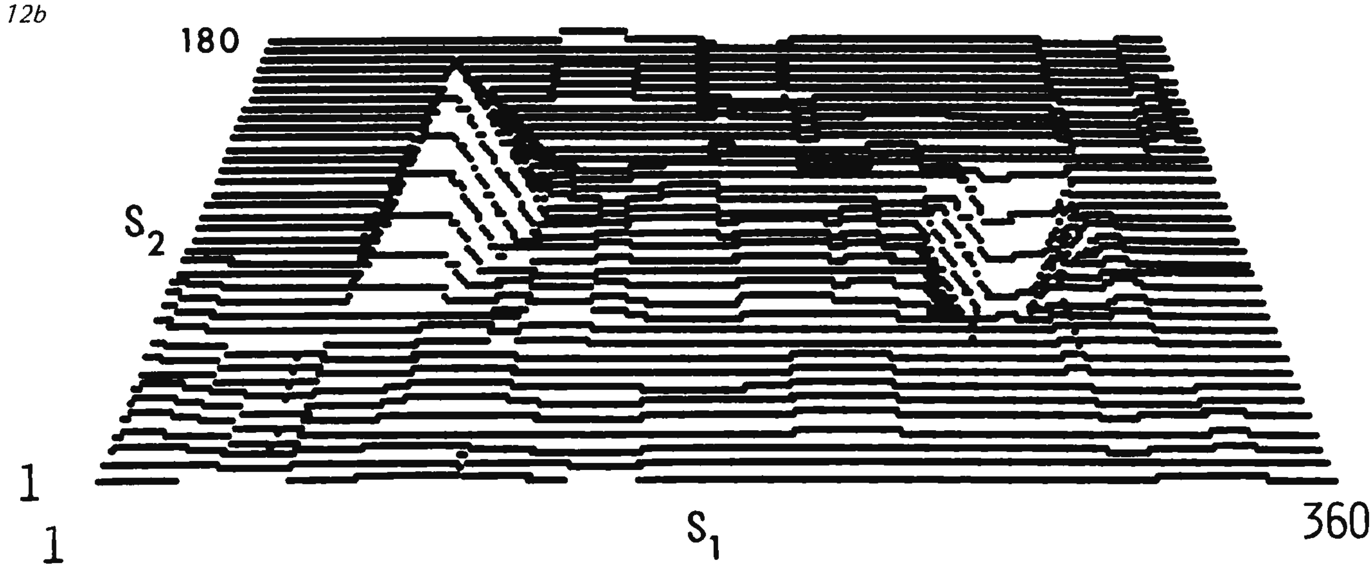
Initially all the weights were equal to 0. The point $\mathbf{S}_1$ = (90, 90) was chosen for the first data entry. The value of the desired function $\hat{p} = \hat{h}$ (90, 90) is 1. By formula (1) (where g = 1) each of the weights selected by **S** = (90, 90) is set to 1/32, causing the proper value to be stored at **S** = (90,90) as shown in figure 12a. After two data storage operations, one at (90, 90), the other at (270, 90), the contents of the CMAC memory are as shown in figure 12b. After 16 storage operations along the $s_2$ = 90 axis the results are as shown in figure 12c. After 175 storage operations scattered over the entire input space, the contents of the CMAC memory are as shown in figure 12d.

**CMAC Memory Requirements**

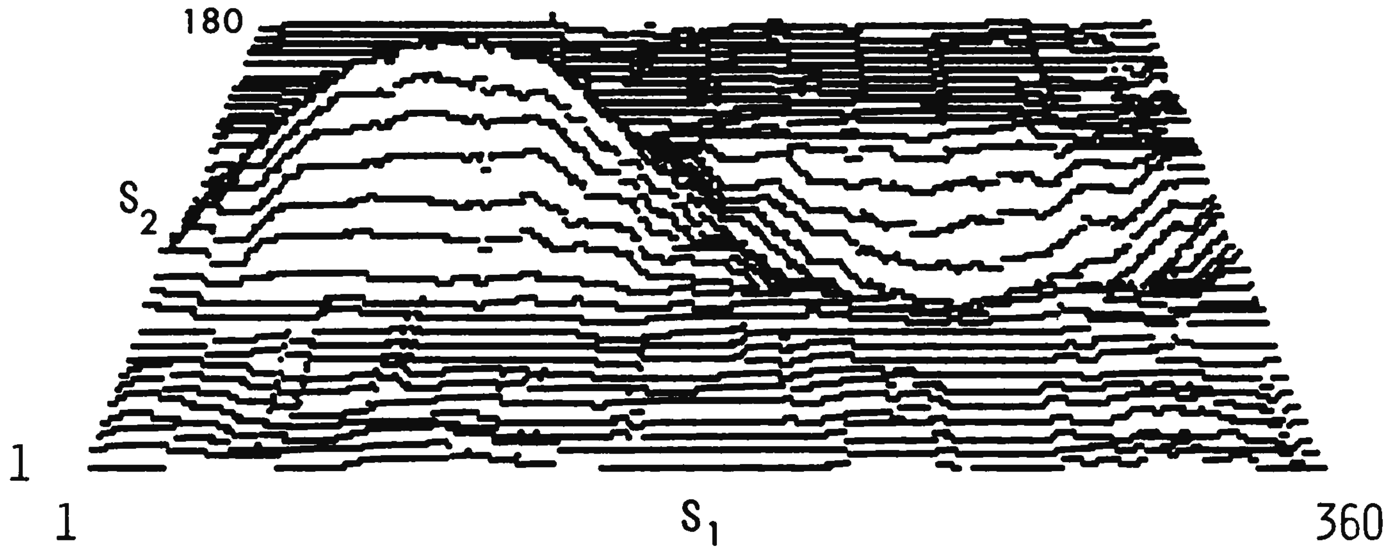
The CMAC **S** → A* mapping corresponds to an address decoder wherein **S** is the input address and the active granule cells in A* are select lines. These access weights whose sum can be interpreted as the contents of the address **S**. In a conventional memory, each possible input address selects a unique single location wherein is stored the contents of that address, as illustrated in figure 13a. In CMAC each possible input address selects a unique *set* of memory locations, the sum of whose contents is the contents of the input address, as shown in figure 13b.

This suggests that the Cerebellar Model

12b



*12c*




Arithmetic Computer might require considerably less memory than a conventional lookup table in storing certain functions. The reason is that the number of ways that x elements can be selected from a table of y entries always exceeds y and, in some cases, it does so by orders of magnitude.

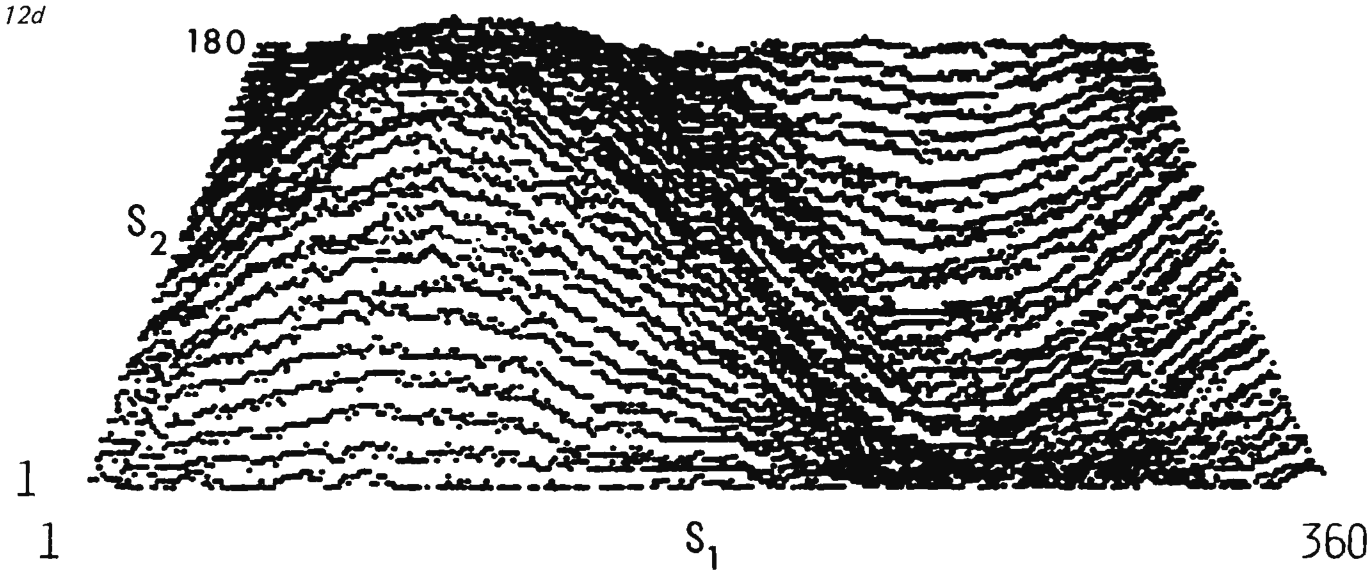
A conventional memory requires $R^N$ memory locations to store a function of N variables, where R is the number of resolution elements on each variable. CMAC requires at most $K \times Q^N$ memory locations, when K is the number of quantizing functions and Q the number of resolution elements on each quantizing function.

A modest example of CMACs reduced memory requirements can be seen in figure 8 where N = 2 and R = 17. Here then are $17^2$, or 289, possible input vectors. The CMAC shown has only 100 weights since K = 4 and Q = 5. Thus $K \times Q^N$ = 100. This savings in memory size becomes increasingly significant for large N. It allows CMAC to store a large class of low resolution functions of up to 12 variables over the entire input space with computer memory of practical size (less than 100 K bytes), whereas conventional table lookup becomes impractical for similar functions of more than four variables.

An even greater savings in memory requirements can be achieved by the use of hash coding techniques in the selection of addresses for the elements in A*. Hash coding allows CMAC to store functions of many variables, so long as the information content of the portion of the function stored does not exceed the number of bits in the CMAC

**Hash coding, a memory addressing technique, compresses sparse address space.**

*12d*



memory. For example in figure 12, the 360 by 180 (over 64,000) element input space is represented in a 1024 location CMAC memory by hash coding.

Hash coding is a commonly used memory addressing technique for compressing a large but sparsely populated address space into a smaller, more densely populated one. (See "Making Hash with Tables" by Terry Dolhoff in *Programming Techniques: Program Design*, BYTE Books, 1979.) Many addresses in the larger space are mapped

*13a*

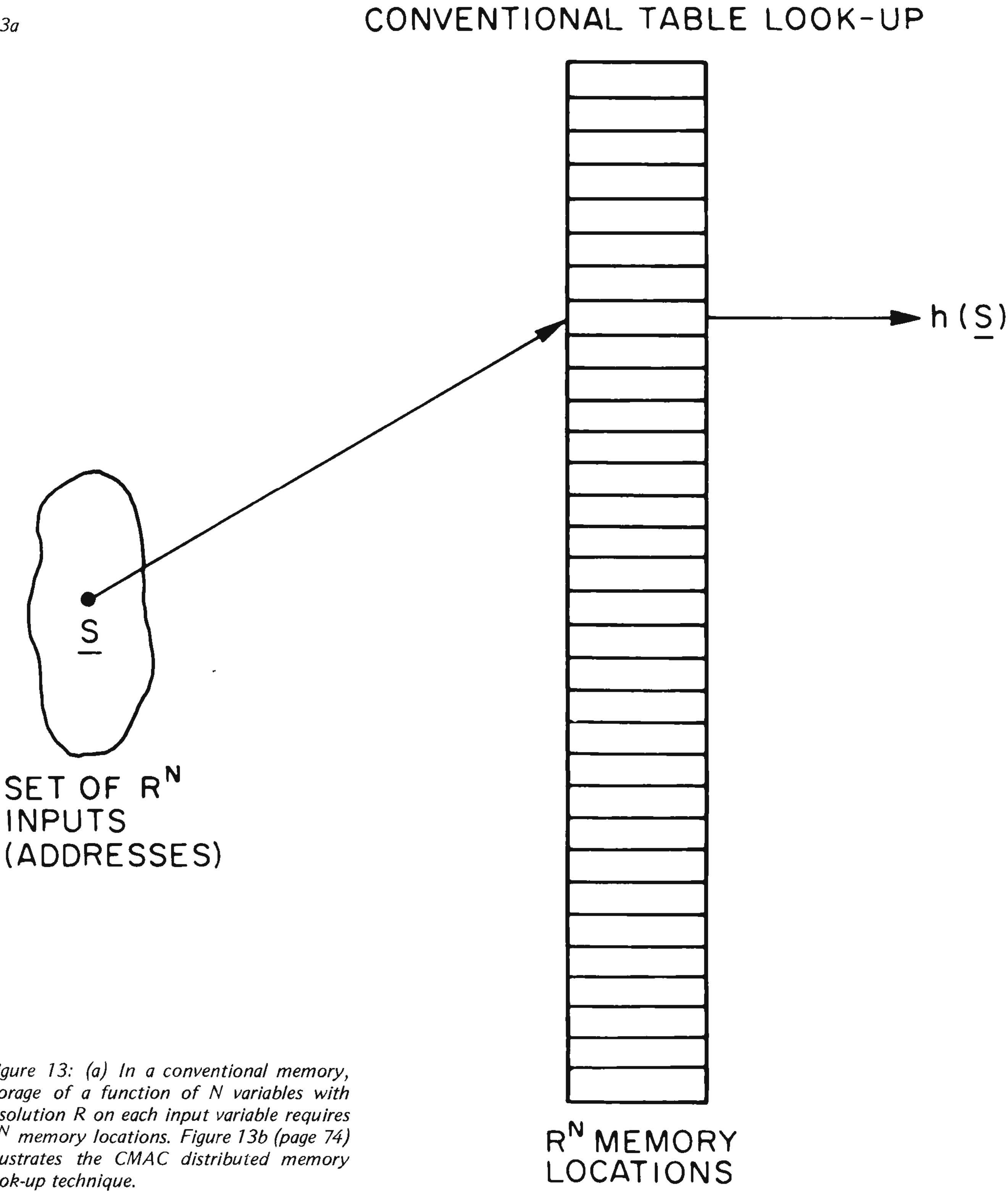



*Figure 13: (a) In a conventional memory, storage of a function of $N$ variables with resolution $R$ on each input variable requires $R^N$ memory locations. Figure 13b (page 74) illustrates the CMAC distributed memory look-up technique.*

# THE SEARCH FOR A SMALL COMPUTER SYSTEM STARTS HERE

## It's the 3rd Annual National Small Computer Show

**New York Coliseum, August 23-26, 1979**

presenting the state-of-the-art showcase for micro-and mini-systems technology and software. Here you can survey virtually all makes and models of small computers, whether your interest runs to a no-nonsense micro priced in the hundreds of dollars or a powerful mini costing $20,000 or more. They're all here.

The world of small computers is quite large, extending to business and professional offices, scientific research, medicine and bionics, education, the home and hobbyist, therapeutic applications for the handicapped, design and engineering. A full selection of lectures is presented to provide a grasp of small systems technology, so that you know what to consider when buying a computer or word processor. It's the first step in discovering what a system can really do for you!

NSCS lectures include sessions on system selection, computer languages, word processing functions, artificial intelligence, software applications, and a dozen more topics for people of all interests.

Plan now to attend. There will be about 30,000 square feet of exhibits, and more than 40 hours of lectures from which to choose. Registration fee is only $5.00 per day, including lectures.

**LECTURES:** (Program subject to change)

**Thursday, August 23**

- **1 p.m.** The Peril of Becoming a Machine-Oriented Business User
- **1 p.m.** Introduction to Small Business Systems
- **2 p.m.** Selecting a Word Processing System
- **2 p.m.** Distributed Data Processing
- **3 p.m.** Accounts Receivable/General Ledger/Accounts Payable
- **3 p.m.** Is There a Computer in Your Educational Future
- **4 p.m.** Mailing Lists: Load, Time and Cost
- **4 p.m.** Word Processing Systems in the Law Office
- **5 p.m.** Basic BASIC
- **5 p.m.** Achieving Quality Control in Word Processing

**Friday, August 24**

- **1 p.m.** The Peril of Becoming a Machine-Oriented Business User
- **1 p.m.** Introduction to Small Business Users
- **2 p.m.** Selecting a Word Processing System
- **2 p.m.** Distributed Data Processing
- **3 p.m.** Unassigned at press time
- **3 p.m.** How to Write a User-Oriented Program
- **4 p.m.** Efficient Expansion of a Small System
- **4 p.m.** Investment Analysis
- **5 p.m.** Accounts Receivable/General Ledger/Accounts Payable
- **5 p.m.** Exploiting the Apple/Dow Jones Computer Link

**Saturday, August 25**

- **11 a.m.** Introduction to Personal Computing
- **11 a.m.** Unassigned at press time
- **12 p.m.** Computer Music Update
- **12 p.m.** Unassigned at press time
- **1 p.m.** Introduction to PASCAL
- **1 p.m.** Computer Art Forms
- **2 p.m.** Household Applications
- **2 p.m.** Artificial Intelligence
- **3 p.m.** How to Write a User-Oriented Program
- **3 p.m.** Investment Analysis
- **4 p.m.** Basic BASIC
- **4 p.m.** Unassigned at press time

**Sunday, August 26**

- **11 a.m.** Introduction to Personal Computing
- **11 a.m.** Computer Music Update
- **12 p.m.** Household Applications
- **12 p.m.** Unassigned at press time
- **1 p.m.** Efficient Expansion of a Small System
- **1 p.m.** Computer Art Forms
- **2 p.m.** Unassigned at press time
- **2 p.m.** Unassigned at press time
- **3 p.m.** Microcomputers for the Handicapped: Update
- **3 p.m.** Exploiting the Apple/Dow Jones Computer Link
- **4 p.m.** Mailing Lists: Load, Time and Cost
- **4 p.m.** Introduction to PASCAL

## REGISTRATION FOR AMERICA'S BIGGEST SMALL COMPUTER SHOW

**Please register me for the 3rd Annual National Small Computer Show, Aug. 23-26, 1979.**

NAME ______________________ BUSINESS TITLE (If Any) ______________

COMPANY (If Any) ______________________ TELEPHONE ______________

ADDRESS ______________________

______________________ ZIP ________

**(Check main interest)**

- ☐ Banking/Insurance
- ☐ Business office
- ☐ Communications
- ☐ Educational
- ☐ Government
- ☐ Hobby
- ☐ Industrial/Manufacturing
- ☐ Military
- ☐ Professional
- ☐ Stock Brokerage
- ☐ Transportation
- ☐ Utility
- ☐ Wholesale/Retail
- ☐ Other ________

**(Check main job function)**

- ☐ Accountant
- ☐ Administrator (Business)
- ☐ Architect/Builder
- ☐ Art Director
- ☐ Banker
- ☐ Computer technician
- ☐ Consultant
- ☐ Controller
- ☐ Engineer
- ☐ Industrial Designer
- ☐ Lawyer/law office mgr.
- ☐ Marketing manager
- ☐ Medical doctor
- ☐ Medical technician
- ☐ Office manager
- ☐ Programmer
- ☐ Public Servant
- ☐ Research/Development
- ☐ Teacher
- ☐ DP manager
- ☐ WP manager
- ☐ WP operator
- ☐ Stock Broker
- ☐ Systems Analyst
- ☐ Student
- ☐ Other ________

☐ ONE DAY $5 ☐ TWO DAYS $10
☐ THREE DAYS $15 ☐ FOUR DAYS $20

Mail with payment of $5 for each day you wish to attend. Use one form per person. Registration badge will be sent by mail in early August. Check or money order only.

**Mail prior to Aug. 10 to:**
National Small Computer Show,
110 Charlotte Place,
Englewood Cliffs, N.J. 07632.

Circle 258 on inquiry card.

13b

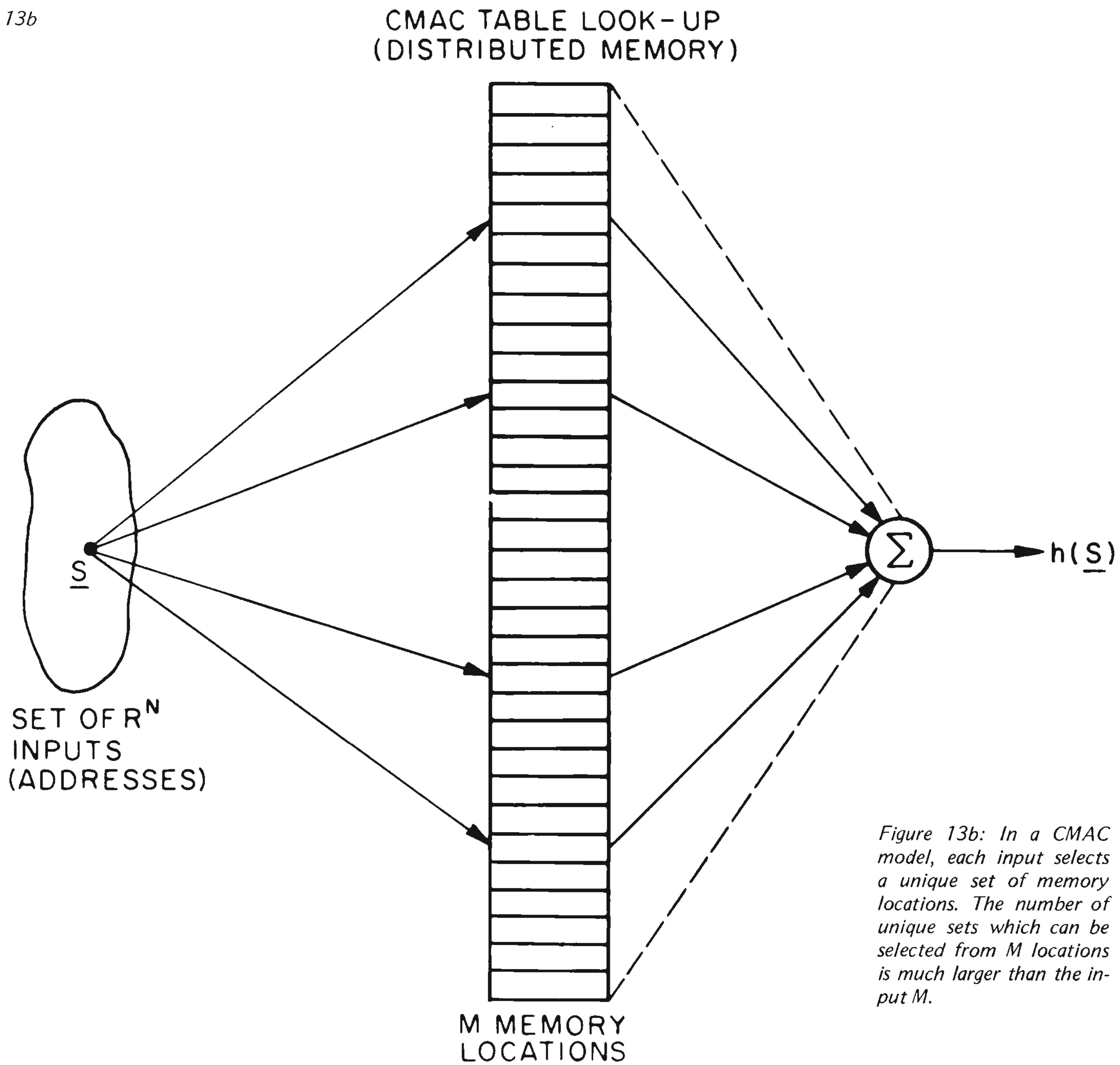



*Figure 13b: In a CMAC model, each input selects a unique set of memory locations. The number of unique sets which can be selected from M locations is much larger than the input M.*

onto each of the addresses in the smaller space. One method is simply to overlay pages. Hashing works when the probability of a collision (ie: more than one filled location in the large memory mapping into the same address in the small memory) is low.

CMAC can tolerate a fairly high incidence of collisions because of its distributed memory (ie: its output is the sum of many locations). Thus a collision (which in a conventional memory would make the output completely incorrect) in CMAC introduces only a small amount of noise into the output. Hash coding noise can be seen in the base plane in figure 12a, b, c.

In CMAC, hashing noise is randomly scattered over the input space each time new data is stored. Thus each new data storage operation degrades previously stored data somewhat. The effect is that the contents of a CMAC memory are most accurately defined in the regions where it is most recently stored. Old data tends to gradually fade, or be "forgotten", due to being hashed over.

**CMAC Memory Generalization**

The fact that each possible CMAC input vector selects a unique *set* of memory locations rather than a single location implies that any particular location may be selected by more than one input vector. In fact, the

(14a)

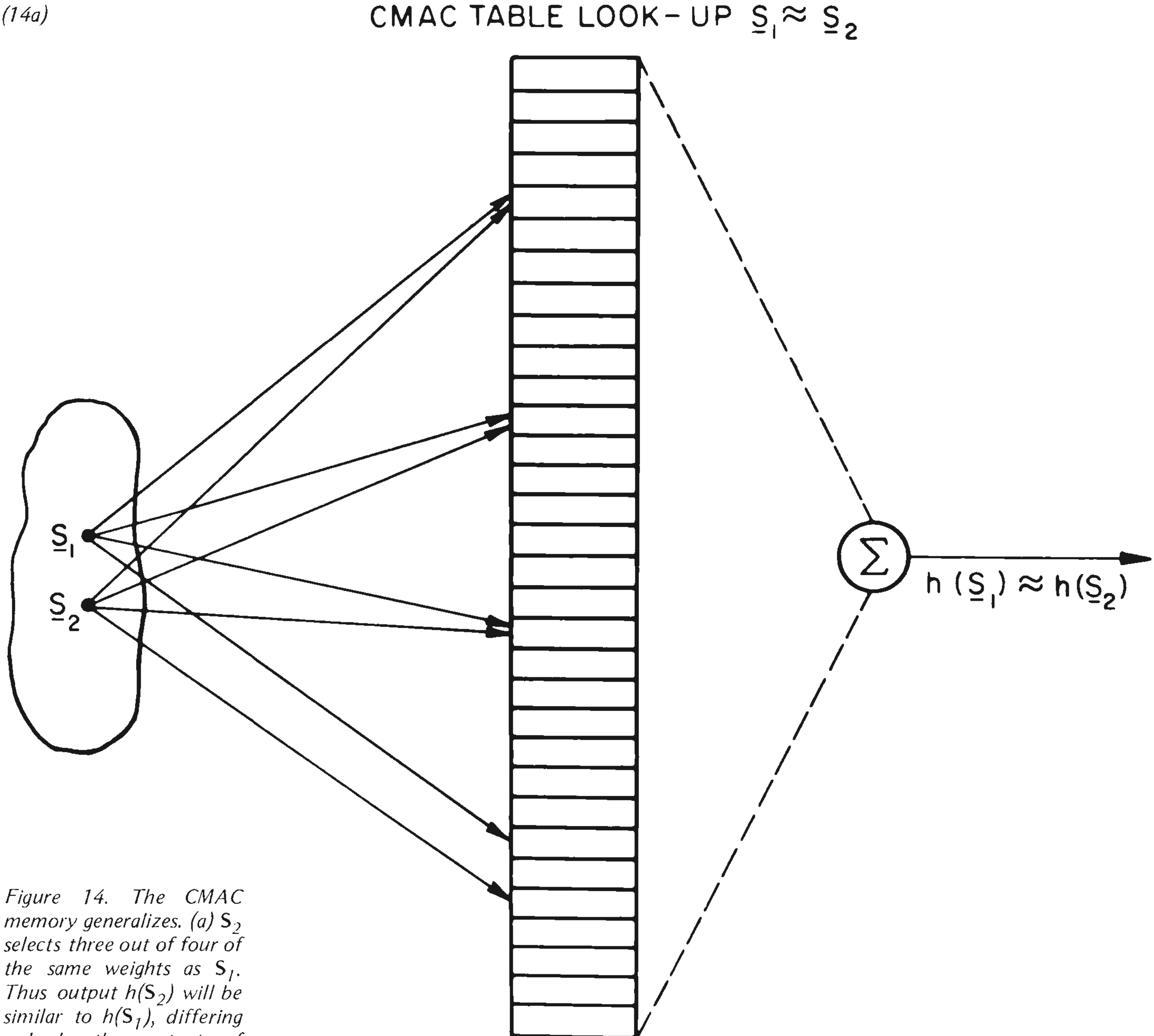



*Figure 14. The CMAC memory generalizes. (a) $S_2$ selects three out of four of the same weights as $S_1$. Thus output $h(S_2)$ will be similar to $h(S_1)$, differing only by the contents of the location not in common. (b) When $S_2$ is outside of the neighborhood of generalization of $S_1$ the overlap goes to 0 (except for random hashing collisions).*

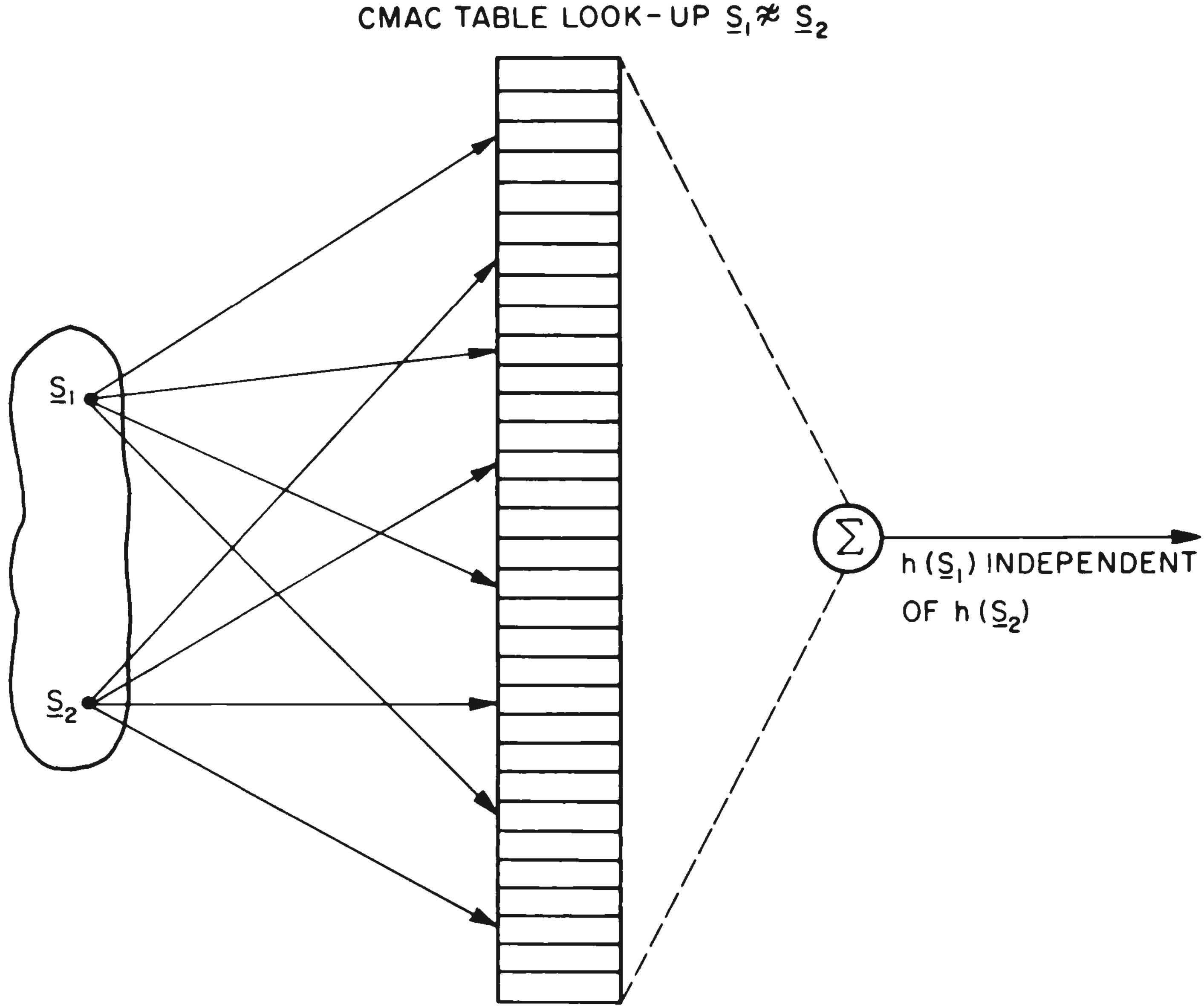
$S \rightarrow A^*$ mapping insures that any two input vectors which are similar (ie: close together in input space) will activate many of the same granule cells, and hence select many of the same weights. This is the property of CMAC which causes it to generalize.

In figure 14a the input vector $S_2$ selects three out of four of the same memory locations as $S_1$. Thus, the output $h(S_2)$ will be similar to $h(S_1)$, differing only by the contents of the single location which is not in common. The $S \rightarrow A^*$ mapping controls the amount of overlap between sets of selected memory locations such that, as the input space distance between two input vectors increases, the amount of overlap decreases. Finally, at some distance the overlap becomes 0 (except for random hashing collisions), as in figure 14b, and the sets of selected memory locations are disjoint. At that point input $S_2$ can be said to be outside the neighborhood of generalization of $S_1$. The value of the output $h(S_2)$ is thus independent of $h(S_1)$.

The extent of the neighborhood of generalization depends on both the number of elements in the set $A^*$ and the resolution of the $s_i \rightarrow m_i^*$ mappings. It is possible in CMAC to make the neighborhood of generalization broad along some variable axes and limited along others by using different resolution quantizing functions for different input variables. This corresponds to the effect in the cerebellum where some input variables are resolved finely by many mossy fibers and others resolved more coarsely by fewer mossy fibers.

A good example of generalization can be

(14b)




seen in figure 12a. Following a single data storage operation at $S_1$ = (90, 90) we find that an input vector $S_2$ = (91, 90) will produce the output p = 31/32 even though nothing had ever been explicitly stored at (91, 90). This occurs because $S_2$ selects 31 of the same weights as $S_1$. A third vector $S_3$ = (92, 90) or a fourth $S_4$ = (90, 92), will produce p = 30/32 because of sharing 30 weights with $S_1$. Not until two input vectors are more than 32 resolution elements apart do they map into disjoint sets of weights.

As a result of generalization, CMAC memory addresses in the same neighborhood are not independent. Data storage at any point alters the values stored at neighboring points. Pulling one point to a particular value as in figure 12a produces the effect of stretching a rubber sheet.

Generalization has the advantage that data storage (or training) is not required at every point in the input space in order for an approximately correct response to be obtained. This means that a good first approximation to the correct H function can be stored for a sizable envelope around a $T_s$ trajectory by training at only a few points along that trajectory. For example, figure 12c demonstrates that training at only 16 points along the trajectory defined by $s_2$ = 90 generalizes to approximately the correct function for all 360 points along that trajectory plus a great many more points in an envelope around that trajectory. Further training at 175 points scattered over the entire space generalizes to approximately the correct response for all 360 by 180 (over 64,000) points in the input space as shown in figure 12d.

Generalization enables CMAC to predict


# Low Cost Add-On Storage for Your TRS-80*. In the Size You Want.

## When you're ready for add-on disk storage, we're ready for you. Ready with six mini-disk storage systems — 102K bytes to 591K bytes of additional *on-line* storage for your TRS-80*.

- **Choose either 40-track TFD-100™ drives or 77-track TFD-200™ drives.**
- **One-, two- and three-drive systems immediately available.**
- **Systems include Percom PATCH PAK #1™, on disk, at no extra charge. PATCH PAK #1™ de-glitches and upgrades TRSDOS* for 40- and 77-track operation.**
- **TFD-100™ drives accommodate "flippy disks." Store 205K bytes per mini-disk.**
- **Low prices. A single-drive TFD-100™ costs just $399. Price includes PATCH PAK #1™ disk.**
- **Enclosures are finished in system-compatible "Tandy-silver" enamel.**

Whether you need a single, 40-track TFD-100™ add-on or a three-drive add-on with 77-track TFD-200™s, you get more data storage for less money from Percom.

Our TFD-100™ drive, for example, lets you store 102.4K bytes of data on one side of a disk — compared to 80K bytes on a TRS-80* mini-disk drive — and 102.4K bytes on the other side, too. Something you can't do with a TRS-80* drive. That's almost 205K bytes per mini-disk.

And the TFD-200™ drives provide 197K bytes of on-line storage per drive — 197K, 394K and 591K bytes for one-, two and three-drive systems.

PATCH PAK #1™, our upgrade program for your TRSDOS*, not only extends TRSDOS* to accommodate 40- and 77-track drives, it enhances TRSDOS* in other ways as well. PATCH PAK #1™ is supplied with each drive system at no additional charge.

The reason you get more for less from Percom is simple. Peripherals are not a sideline at Percom. Selling disk systems and other peripherals is our main business — the reason you get more engineering, more reliability and more back up support for less money.

**In the Product Development Queue . . . a *printer interface* for using your TRS-80* with any serial printer, and . . . the *Electric Crayon*™ to map your computer memory onto your color TV screen — for games, animated shows, business displays, graphs, etc. Coming PDQ!**

™ TFD-100, TFD-200, PATCH PAK and Electric Crayon are trademarks of PERCOM DATA COMPANY.
*TRS-80 and TRSDOS are trademarks of Tandy Corporation and Radio Shack which have no relationship to PERCOM DATA COMPANY.

PERCOM

PERCOM DATA COMPANY, INC.
DEPT. B • 211 N. KIRBY • GARLAND, TX. 75042

**To order add-on mini-disk storage for your TRS-80*, or request additional literature, call Percom's toll-free number: 1-800-527-1592. For detailed Technical information call (214) 272-3421.**

Orders may be paid by check or money order, or charged to Visa or Master Charge credit accounts. Texas residents must add 5% sales tax.

**Percom *'peripherals for personal computing'***



Circle 305 on inquiry card.

**ABOUT THE AUTHOR:**

*Dr James S Albus worked for NASA from 1957 to 1972 designing optical and electronic subsystems for over 15 spacecraft, and for one year managed the NASA Artificial Intelligence Program. Since 1973 he has been with the National Bureau of Standards where he has received several awards for his work in advanced computer control systems for industrial robots. He has written a survey article on robot systems for* Scientific American *(February 1976) and his Cerebellar Model Arithmetic Computer won the* Industrial Research Magazine *IR-100 Award as one of the 100 most significant new products of 1975.*

on the basis of a few representative learning experiences what the appropriate behavioral response should be for similar situations. This is essential in order to cope with the complexities of real world environments where identical $T_s$ trajectories seldom, if ever, reoccur.

An example of how CMAC uses generalization to learn trajectories in a high-dimensional space is shown in figure 15. A seven degree of freedom manipulator arm was controlled by seven CMACs, one for each joint actuator, such that the output vector **P** = H(**S**) had seven components. The input vector **S** to each CMAC contained 18 variables corresponding to position and velocity feedback from each of the seven joints of the arm, plus four binary bits defining the Elemental Move Command. The resolution on the feedback variables was different for each of the seven CMACs, being highest resolution from the joint driven by the output $p_1$ and lower for other joints in inverse proportion to their distance along the arm from the controlled joint.

The desired output trajectory $T\hat{p}_a$ is shown as the set of solid curves marked (a) in figure 16. This trajectory corresponds to the Elemental Movement <SLAP> which is a motion an arm might make in swatting a mosquito.

The (i) curve in figure 17 shows the learning performance with no previous learning over twenty complete $T\hat{p}_a$ "slap" motions. At the beginning of each motion the arm was positioned at the correct starting point and driven from there by the **P** output computed by the CMAC H function. Differences between **P** and $\hat{\mathbf{P}}$ at 20 points along the slap trajectory were corrected by formula 1 (with g set to 1/20). Each point on the curve in figure 17 represents the sum of all the errors for all the joints during an entire slap motion. Note that learning is rapid despite the high dimensional input space in which no two $T_s$ trajectories were ever exactly the same. This is due to CMAC's ability to generalize from a relatively small number of specific teaching experiences to a large number of similar but not identical trajectories.

The (ii) curve in figure 17 shows the learning performance on the same twenty $T\hat{p}_a$ trajectories when preceded by twenty training sessions on the $T\hat{p}_b$ trajectory indicated by the dotted set of curves marked (b) in figure 16. Note that performance on $T\hat{p}_a$ is consistently better following prior learning on a similar trajectory $T\hat{p}_b$. The learning on $T\hat{p}_b$ generalizes to the similar trajectory $T\hat{p}_a$.

Needless to say, predictions based on generalization are not always correct and sometimes need to be refined by further learning. The ability of CMAC to discriminate (ie: to produce different outputs for different inputs, ($\mathbf{S}_1$ and $\mathbf{S}_2$) depends upon how many weights selected by $\mathbf{S}_1$ are *not* also selected by $\mathbf{S}_2$, and how different in value those weights are. If two inputs which are close together in input space are desired to produce significantly *different* outputs,

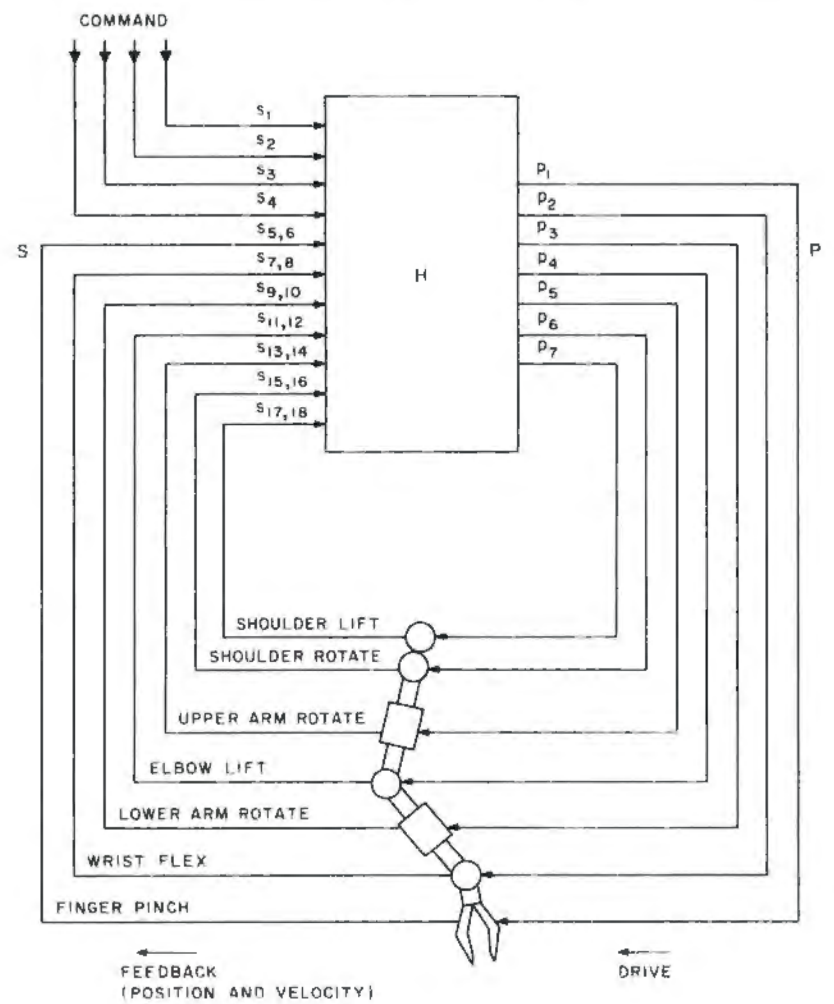



*Figure 15. Information flow diagram for a robot arm controlled by seven CMACs.*

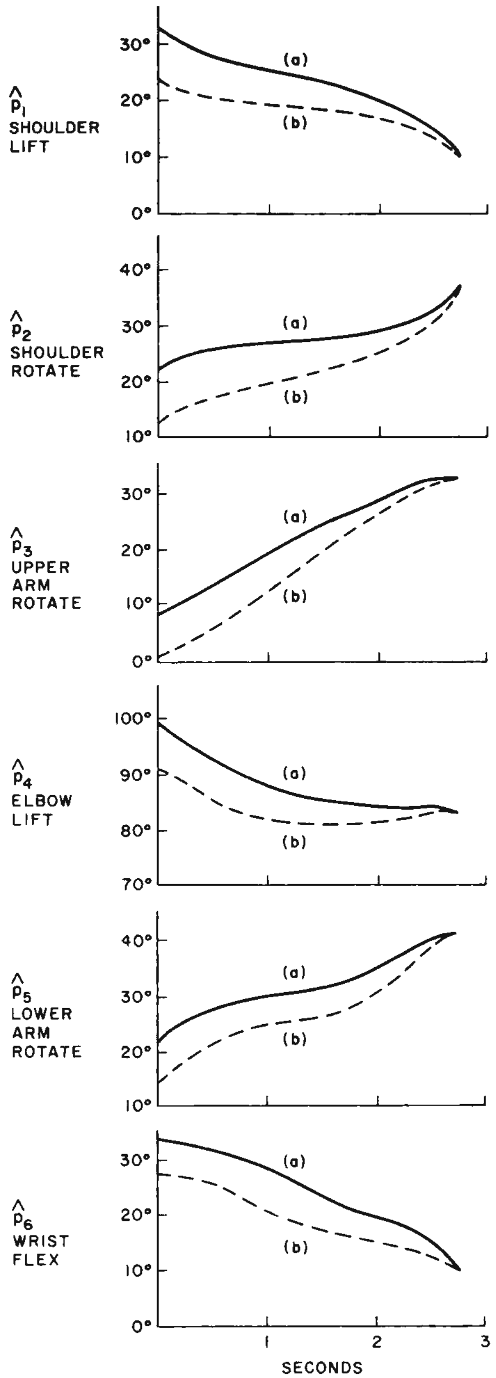



*Figure 16. Two similar trajectories $T\hat{p}_a$ and $T\hat{p}_b$ which have different starting points but the same endpoint. Both trajectories define a version of an Elemental Movement (SLAP) which was taught to the CMACs of figure 15.*

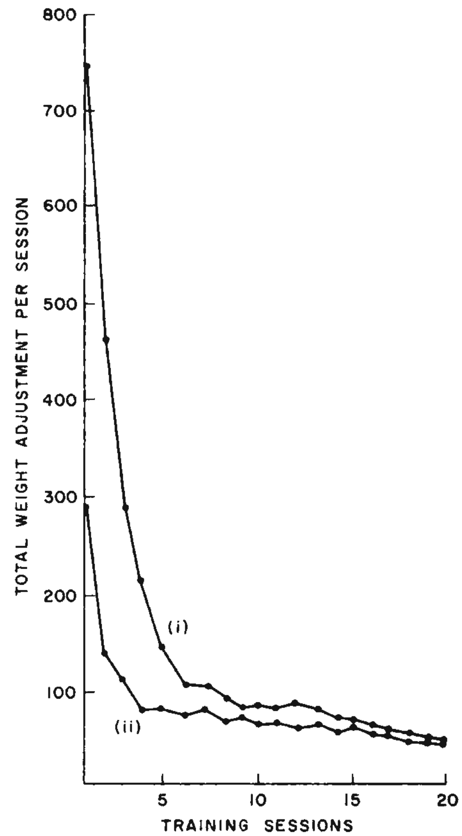



*Figure 17. CMAC learning and generalization performance on the SLAP motion $T\hat{p}_a$. Curve i is with no previous training. Curve ii is after 20 training sessions on the similar trajectory $T\hat{p}_b$. The improvement of ii over i is due to generalization.*

then repeated training may be required to overcome the (in this case erroneous) tendency of CMAC to generalize by building up large differences in the few weights which are not in common.

In most behavioral control situations, sharp discontinuities requiring radically different outputs for highly similar inputs do not occur. Indeed most servocontrol functions have simple S shaped characteristics along each variable axis. The complexity in control computation in multivarient servosystems typically derives from cross-products which affect the slope of the function, or produce skewness, and nonsymetrical hills and valleys in various corners of the N dimensional space. As can be seen from figure 11 these are the type of functions CMAC can readily store, and hence compute. Nevertheless, even on smooth functions generalization may sometimes introduce errors by altering values stored at neighboring locations which were already correct. This type of error corresponds to what psychologists call learning interference, or *retroactive inhibition.*

For example, in the learning of the two similar trajectories in figure 16, training on $T\hat{p}_a$ causes degradation or interference with what was previously learned on $T\hat{p}_b$. This can be seen in figure 18 where, after 20 training sessions on $T\hat{p}_b$, the CMAC is trained 20 sessions on $T\hat{p}_a$. Following this the performance on $T\hat{p}_b$ is degraded. However, the error rate on $T\hat{p}_b$ quickly improves

# SYNCHRO-SOUND

## The ORIGINAL Computer People Who KNOW computers and offer EVERYTHING you need in Small Computer Systems

**Compare PRICE, QUALITY, DELIVERY, SERVICE and you'll see why you don't have to look anyplace else!**

**TEXAS INSTRUMENTS**
810 Multi-Copy Impact Printer
ONLY **$1599.00**
SINGLE QUANTITY PRICE

We carry a full line of Texas Instruments products

### TERMINALS

LEAR-SIEGLER
ADM-3A . . . .
ADM-31 . . . .
ADM-42 . . . .
**CALL*** for prices

HAZELTINE
1400 . . . .
1410 . . . .
1500 . . . .
Mod 1 Edit . . . .
**CALL*** for prices

### SOFTWARE

Here are nine packages that enable you to make full use of the potential in your small computer system:

- ACCOUNTS RECEIVABLE
- INVENTORY
- ORDER ENTRY
- WORD PROCESSING
- PAYROLL
- REAL ESTATE MGT.
- MEDICAL-DENTAL
- MAILING PROGRAM
- SHIPPING & RECEIVING
- and MORE!

Microsoft Basic, Fortran, Cobol available

**CALL*** for prices

### COMPUTERS

NORTHSTAR
Horizon II (kit) . . . .
**CALL*** for price

DIGITAL SYSTEMS
DSC-2 . . . . . . . . . . .
Dble Density
Dual Drive Disk . . . . .
**CALL*** for prices

CROMEMCO
System 3 . . . .
**CALL*** for price

### PRINTERS

OKIDATA
SL125 . . . .
110 . . . .
**CALL*** for prices

TELETYPE
43 . . . .
**CALL*** for price

CENTRONICS
Micro . . . .
779-2 . . . .
700-2 . . . .
703-0 . . . .
**CALL*** for prices

### MORE SPECIALS

| | |
|---|---|
| Decwriter II | $1595.00 |
| Qume Sprint 5/45 RO | 2795.00 |
| Persci 277 Dble Density | 1395.00 |
| Imsai PCS 80/15 | 599.00 |
| Anadex DP 8000 Printer | 995.00 |
| 44 Column Printer | $295.00 |
| Livermore Accoustic Coupler | 249.00 |
| Javelin 9" Monitor | 159.95 |
| Centronics Micro P-1 Printer | 495.00 |

*** Our Prices are too low to advertise. Please call or write. We now carry a full line of Alpha–Micro Products**

We have a full staff of Programmers and Computer Consultants to design, configure and deliver a Turnkey Computer System to meet your specific requirements.

**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

BankAmericard Master Charge

Circle 355 on inquiry card.

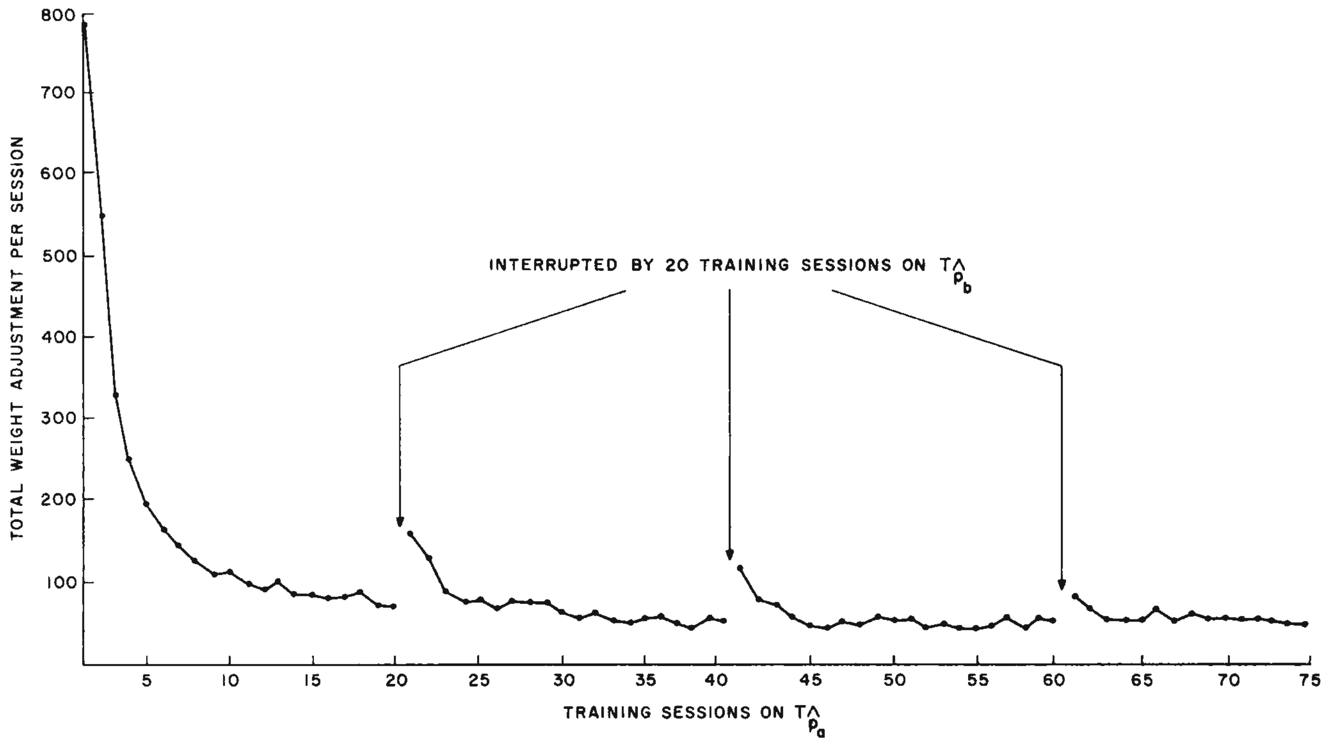



*Figure 18. The effect of learning interference. If training on $T\hat{p}_a$ is interrupted by training on the similar trajectory $T\hat{p}_b$, a degradation in performance on $T\hat{p}_a$ is observed. Repeated iterations gradually overcome this learning interference.*

over another 20 training sessions. Following this another 20 training sessions are conducted on $T\hat{p}_a$. Again degradation in $T\hat{p}_b$ due to learning interference occurs, but not as severely as before. Another set of 20 training sessions on $T\hat{p}_b$ followed by another 20 on $T\hat{p}_a$ shows that the amount of learning interference is declining due to the buildup of values in the few weights which are not common to both $Ts_a$ and $Ts_b$. Thus, learning interference, or retroactive inhibition, is overcome by repetition of the learning process.

## CMAC as a Computer

The ability of CMAC to store and recall (and hence compute) a general class of multivarient mathematical functions of the form **P** = H(**S**) demonstrates how a relatively small cluster of neurons can calculate the type of mathematical functions required for multivarient servomechanisms, coordinate transformations, conditional branches, task decomposition operators, and IF/THEN production rules. These are the types of functions that we showed in part 1. They are required for generating goal-directed behavior (ie: the purposive strings of behavior patterns such as running, jumping, flying, hunting, fleeing, fighting, and mating, which are routinely accomplished with apparent ease by the tiniest rodents, birds, and even insects).

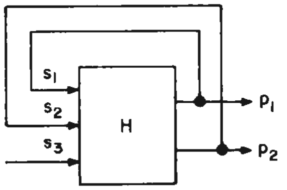
In the case of multivarient servomechanisms the **S** vector corresponds to commands plus feedback (ie: **S** = **C** + **F**). For coordinate transformations the **S** vector contains the arguments as well as the variables in the transformation matrix.

In the case of conditional branches, one or more of the input variables in **S** can be used to select different regions in input space where entirely different functions are stored. Assume, for example, that in figure 12 a third variable $s_3$ had been included in the function being stored. Assume that $s_3$ is held constant at $s_3 = 0$ while storing the function $p = (\sin x)(\sin y)$. Following that, an entirely different function, say $p = 3x + 5y^2$, could be stored with $s_3$ held constant at $s_3 = 50$. Since every point in the input space for $s_3 = 0$ is outside the neighborhood of generalization of the input space for $s_3 = 50$, there would be no interference except for random hashing collisions. The stored function would then be:

$$\begin{aligned} p &= (\sin x)(\sin y) && \text{if } s_3 = 0 \\ p &= 3x + 5y^2 && \text{if } s_3 = 50 \quad . \end{aligned}$$

In the interval $0 < s_3 < 50$ the function would change smoothly from $p = (\sin x)(\sin y)$ to $p = 3x + 5Y^2$. Additional func-

a)




| H | | | | |
|---|---|---|---|---|
| S | | | P | |
| $s_1$ | $s_2$ | $s_3$ | $p_1$ | $p_2$ |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

b)

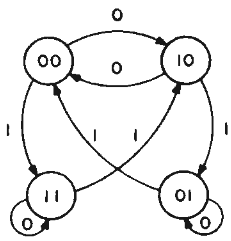



*Figure 19. A CMAC with feedback directly from output to input behaves like a finite state automaton for binary inputs and outputs. It behaves like a "fuzzy state automaton" for nonbinary s and p variables.*

tions could be stored for other values of $s_3$, or other conditional variables $s_4$, $s_5$, and so on might be used for additional branching capabilities. If these conditional variables are part of a command vector, then each different input command can select a different subgoal generator. If they are part of the feedback, then different environmental conditions can trigger different behavioral patterns for accomplishing the subgoals.

**It is possible to construct a CMAC equivalent of any finite state automaton.**

If some of the variables in the P output vector loop directly back to become part of the S input vector (as frequently happens in the cerebellum as well as in other parts of the brain), then CMAC becomes a type of finite state automaton, string generator, or task decomposition operator. For example, the CMAC in figure 19a behaves like the finite state automaton in 19b. The loopback inputs $s_1$ and $s_2$ define the state of the machine, and $s_3$ is the input. The H function defines the state transition table. In general it is possible to construct a CMAC equivalent of any finite state automaton. Of course, CMAC can accept inputs and produce outputs which are nonbinary. Furthermore, the outputs generalize. Thus, CMAC is a sort of "fuzzy state automaton."

A Cerebellar Model Arithmetic Computer with direct feedback from output to input demonstrates how a neural cluster can generate a string of outputs (subgoals) in response to a single input, or unchanging string of inputs. Additional variables added to F from an external source increase the dimensionality of the input space and can thus alter the output string (task decomposition) in response to environmental conditions.

The different possible feedback pathways to a CMAC control module cast light on a long standing controversy in neurophysiology regarding whether behavior patterns are generated by "stimulus-response chaining" (ie: a sequence of actions in which feedback from sensory organs is required to step from one action to the next) or by "central-patterning" (ie: a sequence which is generated by internal means alone). A CMAC hierarchy may include tight feedback loops from the output of one level back to its own input to generate central patterns, longer internal loops from one level to another to cycle through a sequence of central patterns, as well as feedback from the environment to select or modify central patterns or their sequence in accordance with environmental conditions.

The above discussion makes it obvious that CMAC can also implement IF/THEN production rules by the simple mechanism of making the S vector (or the $T_s$ trajectory) correspond to an IF premise. The P vector output (or $T_p$ trajectory) becomes the THEN consequent.

The capability of CMAC to simulate a finite state automaton, to execute the equivalent of a conditional branch, and to compute a broad class of multivarient functions makes it possible to construct the CMAC equivalent of a computer program. Conversely it is possible to construct a hierarchy of computing modules, perhaps implemented on a network of microprocessors, which is the equivalent of a CMAC hierarchy. This has profound implications regarding the type of computing architecture which might be used to build a model of the brain for robot control.

Note in this regard that CMAC produces nothing comparable to a DO loop or an interrupt. Each CMAC is a state machine which samples (or polls) a set of input variables and computes a set of output variables. There is no way that it can be instructed to DO something N times. CMAC can, of course, perform a DO-UNTIL in the sense that if the input is constant, the output will remain constant until the input changes. Thus for a constant input $S_1$, CMAC will DO $P_1 = H(S_1)$ UNTIL $S_1$ changes to $S_2$. But this is not a DO loop in the customary sense.

Similarly, one or more of the CMAC input variables can be used to "interrupt" an ongoing trajectory by causing a branch to a new trajectory. A hierarchy of CMACs can return to the interrupt trajectory after a deviation, if the higher level goals remain


# FOR FAST IMPLEMENTATION OF BUSINESS APPLICATIONS

**Compact Compiler requires only 20K bytes.** Our compact Version 3 Compiler requires only 20K bytes but supports a powerful subset of standard COBOL plus the CIS COBOL language extensions for conversational mode working, providing full CRT screen formatting with built-in cursor control and numeric field validation.

Other features include CIS COBOL Indexed Sequential and Interactive Debug packages, linkage to run time subroutines to CHAIN programs together, PEEK & POKE memory locations and GET & PUT to special peripherals.

**Forms Generator speeds program development.** Forms is a powerful utility which can be used with either the Standard or Compact COBOL Compiler. Forms generates record descriptions for Data Entry and Enquiry Programs. The operator creates and edits the required data entry form on the CRT by means of a question and answer session, and then automatically creates the required COBOL source text.

This text is held on diskette as COPY files which the programmer can then simply COPY into his program in the normal COBOL way.

**Standard Compiler conforms to ANSI 74 Specifications.** And now our ANSI 74 standard Version 4 Compiler implements level 1 of Nucleus, Table Handling, Sequential I/O, Relative I/O, Indexed I/O, Library, Segmentation and Inter Program Communication plus many level 2 features and the CIS COBOL language extensions employed in Version 3. This compiler runs in 30K bytes and is ideal for implementing or converting large systems using modular programming. Features include dynamic program loading and fast program development turnround.

CIS COBOL **products run on the 8080 or Z80 microprocessors under the CP/M operating system.** They are distributed on a variety of diskette formats and have a CONFIG utilility supplied as standard, which enables you to drive many different types of CRT. All are themselves written in CIS COBOL and are therefore readily transferred to different operating environments including new operating systems and new microprocessors, making CIS COBOL a very attractive proposition for OEMs.

MICRO FOCUS

**MICRO FOCUS LTD.** *58 Acacia Rd, St. Johns Wood, London NW8 6AG*
*Telephone: 01-722 8843 Telex: 28536 MICROF G*
**CP/M is a trademark of Digital Research*

**Circle 221 on inquiry card.**

unchanged throughout the lower level trajectory deviation. This, however, is quite a different mechanism from the interrupt circuitry in the normal computer where a program counter is stored so that program execution can continue after the interrupt has been serviced.

The implication here is that a set of robot control programs modeled after a CMAC hierarchy will include *no* DO-loops and will *not* be interrupt driven. Every computing module will implement a simple state mapping function of the form **P** = H(**S**).

Note also that in a CMAC hierarchy, a deviation in a higher level trajectory changes the command string, and hence the program, of all the levels below it. This implies real time modification of program statements and thus makes the use of a compiler based programming language somewhat cumbersome. A robot control system modeled after a CMAC hierarchy should use some form of an interpretive language where program statements are translated into machine code at execution time. A language similar to FORTH seems ideal. (An interpretive language can, of course, be written in a compiler based language. Also, languages can be devised which are partially compiled and partially interpreted.) We will return to these and other practical issues of computing architecture for robot control at a later time.

## CMAC as a Pattern Recognizer

As was discussed in part 1, any spatial pattern can be represented as a vector. For example, a picture can be represented as an array, or ordered list, of brightness or color values. A symbolic character can be represented as an ordered list of features (or arbitrary numbers, as in the ASCII convention). Any temporal pattern can be represented as a trajectory through an N-dimensional space. For example, an audio pattern is a sequence of pressure or voltage values (ie: a one-dimensional trajectory). A moving picture or television scene corresponds to a sequence of picture vectors (ie: an N-dimensional trajectory where N is the number of picture resolution elements or pixels).

The fundamental problem of pattern recognition is to name the patterns. All the patterns with the same name are in the same class. When a pattern has been given a name we say it has been recognized. For example, when the image of a familiar face falls on my retina and I say to myself "That's George," I have recognized the visual pattern by naming it.

At this point we need to introduce some new notation to clearly distinguish between

MicroPro International Corporation

The Producers of WORD-MASTER T.M. and SUPER-SORT T.M.

Proudly Present

# WORD-STAR T.M.

Here it is! The most complete, totally integrated, word processing system software you've ever seen on a microcomputer. WORD-STAR T.M. really shines in ease of use, with its unique, dynamically activated menu scheme that allows typists to become word processing operators in minutes instead of weeks. For only $495, your 8080/8085/Z-80 *with any CRT device* becomes a word processing system with the features of a Lanier, Wang, Vydec, Xerox, etc, for far less cost.

Features like ON-SCREEN VISUAL TEXT COMPOSITION, with word wrap, plus dynamic justification and remargination. Imagine being able to type without hitting RETURN, and as WORD-STAR T.M. moves you to the next line, the preceding line is redisplayed, justified to your left and right margins! Center a line with a keystroke! Set **BOLDFACE** and/or underline even in mid paragraph! Rejustify text to new margins, on-screen, at will!

DYNAMIC PAGINATION shows the printer page breaks during text entry, correction, or review. Dot commands control vertical page layout, CONDITIONAL PAGINATION, page HEADINGS, page NUMBERS, etc. The integrated PRINT COMMAND can print selected pages only, pause between pages for letterhead loading, and drive any CP/M* list device. WORD-STAR T.M. runs with the CP/M interfaced printer and terminal of your choice.

SPECIFICATIONS: status line showing page, line, column of cursor; variable menu suppression for experienced operators; dynamic display of text during entry or on call permits CRT preview of printed copy; full disk buffering (document sizes up to diskette capacity); multiple file/disk edits in a session; CP/M file compatible — also useful for data entry, editing programs, etc. Commands include: cursor up/down/left/right; scroll line/screenful, up/down, once/variable rate; insert/overtype; delete character/word/line left/right/entire; variable margin set/release; set/clear tab stops; justify/ragged right; block move/copy/delete; search/replace once/n times/global/selective/whole words only; write to/read from additional files; set/go to any of 10 place markers. Additional support provided for CRT's with inverse video/dual intensity, line insert/delete, and erase to end of line. Dialog interface for easy installation. AND MORE!

Registered WORD-MASTER T.M. users can get a $100 discount from us or your participating dealer when upgrading to WORD-STAR T.M. Offer good for a limited time only.

WORD-STAR T.M. operates with any CP/M* operating system, 32K, and either a memory mapped video board or any CRT TERMINAL with cursor control and clear screen (Hazeltine, Lear-Siegler, Soroc, Microterm, Hewlett Packard, Infoton, Beehive, Processor Tech VDM, IMSAI VIO, SD Systems, Altos, Dynabyte, GNAT, etc). We ship on 8" IBM, Micropolis II, or Northstar diskette.

Other fine MicroPro Products include:

| | Software/Manual | | Software/Manual |
|---|---|---|---|
| Word-Star T.M. | $495/25 | Super-Sort T.M. I | $250/25 |
| Word-Master T.M. | $150/25 | Super-Sort T.M. II | $200/25 |
| Tex-Writer T.M. | $ 75/15 | Super-Sort T.M. III | $150/25 |

**MICROPRO INTERNATIONAL CORPORATION**
**1299 4th Street, San Rafael, California 94901**
**Telephone (415) 457-8990 Telex 340388 Dealer/OEM inquiries invited**

***"PROFESSIONAL QUALITY SOFTWARE YOU CAN COUNT ON, NOW"***

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

Circle 228 on inquiry card.

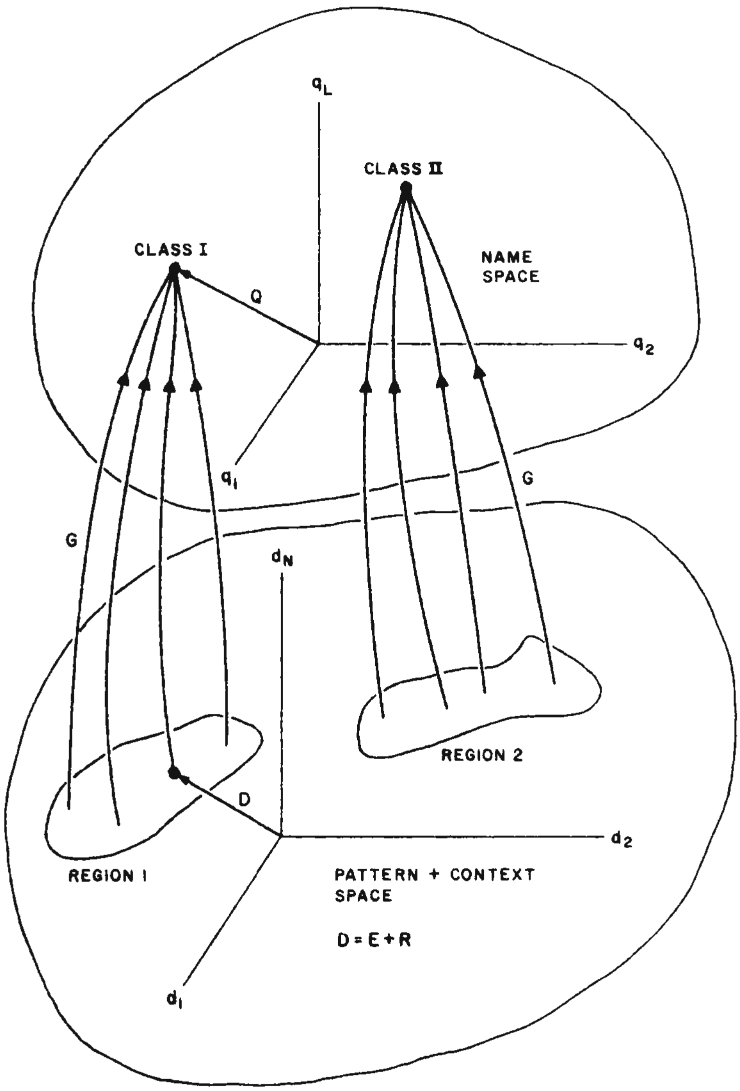



*Figure 20. The* **D** *vector is composed of sensory variables* **E** *and context variables* **R**. *The function G recognizes the existence of a* **D** *vector in a particular region of pattern plus context space by outputting a* **Q** *vector which is the name of that region.*

vectors in the sensory processing hierarchy and those in the behavior-generating hierarchy. Thus we will define the input vector to a CMAC pattern recognizer as:

$$\mathbf{D} = \mathbf{E} + \mathbf{R}$$

where:

$$\mathbf{E} = (d_1, d_2, \ldots, d_i)$$

is a vector, or list, of data variables derived from sensory input from the external environment, and:

$$\mathbf{R} = (d_{i+1}, \ldots, d_N)$$

is a vector of data variables derived from recalled experiences, or internal context. The CMAC mapping operator in the sensory processing hierarchy will be denoted G and the output **Q** such that:

$$\mathbf{Q} = G(\mathbf{D})$$

We can now define a CMAC **D** vector to represent a sensory pattern plus context such that each component $d_i$ represents a data point or feature of the pattern plus context. The existence of the **D** vector within a particular region of space therefore corresponds to the occurrence of a particular set of features or a particular pattern in a particular context. The recognition problem is then to find a set of CMAC weights such that the G function computes an output vector:

$$\mathbf{Q} = G(\mathbf{D})$$

such that **Q** is the name of the pattern plus context **D** as shown in figure 20.

In other words G can recognize the existence of a particular pattern and context (ie: the existence of **D** in a particular region of input space) by outputting the name **Q**. For example,

**Q** = Class I whenever **D** is in Region 1
**Q** = Class II whenever **D** is in Region 2
.
.
.
etc.

The **D** → **A** mapping in the sensory processing CMAC can be chosen so as to define the size of the neighborhood of generalization on the input space. This means that, as long as the regions of input space corresponding to pattern classes are reasonably well separated, the G function can reliably distinquish one region of input space from another and hence classify the corresponding sensory patterns correctly.

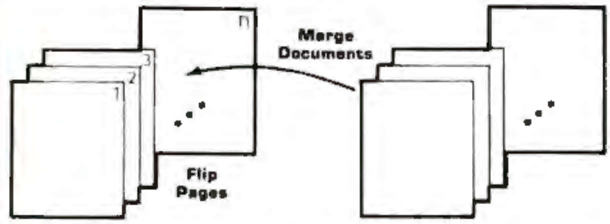
In the case where the **D** vector is time dependent, an extended portion of a trajectory $T_D$ may map into a single name **Q** as shown in figure 21. It then is possible by integrating **Q** over time and thresholding the integral to detect, or recognize, a temporal pattern $T_D$ such as a sound or a visual movement.

Note that the recognition, or naming, of a temporal pattern (as illustrated in figure 21) is the inverse of the decomposition of a task as illustrated in figures 14 thru 17 in the previous article in this series. In task decomposition a slowly varying command **C** is decomposed into a rapidly changing output **P**. In pattern recognition a rapidly changing sensory experience **E** is recognized by a slowly varying name **Q**.

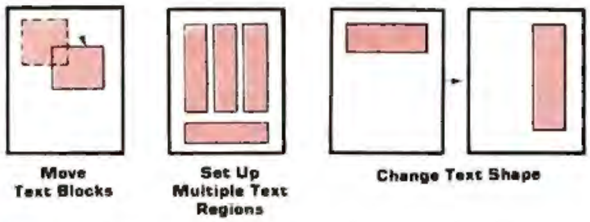
**Wordsmith** is the video text editing system you've been waiting for. Its power, flexibility and simplicity help you carve any text editing task down to size—in a way you can understand. We wanted a system that allows you to think in traditional ways about text layout, yet at the same time makes the traditionally tedious operations such as cut and paste simple and fast. We think we've done it. We want you to decide for yourself.

## Power

- **Page Oriented Philosophy.** A document is a collection of pages. The screen displays one entire page at a time. Simple random access page flipping commands take you quickly to any page in the document. Equally efficient commands allow you to insert, delete, copy and move pages both within one document and across documents.



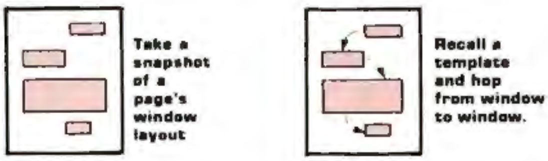

- **Extensive Block Manipulation Capabilities.** Using "windows", portions of text, charts, etc., can be quickly and effortlessly moved around on the current page, or across pages. The shape and size of any window can be changed in real time, with the contained text automatically reformatting itself (heeding word and paragraph boundaries) to conform to the new shape.




- **Instantaneous Formatting.** Compacting (extraneous blank deletion) and right justifying are simple commands that tidy up a full page or window's worth of text in the blink of an eye. Random access cursor movement, line and character insert and delete, line and page split and join, and a host of other line and character level commands help you put text in its place quickly and accurately.

## Flexibility

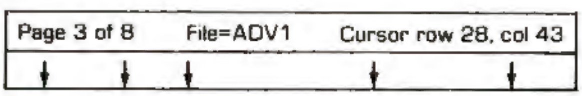
- **Logical/Physical Page Distinction.** Define your own hardcopy size. Wordsmith remembers the difference between the screen size and the hardcopy page size.
- **Modular Hardcopy Driver.** Drive a Qume® Sprint-5 or TTY-like device directly now, Diablo, NEC and other hardcopy devices soon.
- **Pure Text.** Wordsmith files are pure text with no control characters mixed in. This universal format keeps you as compatible with the world as possible. What you see on the screen is what you get as hardcopy.
- **Page Templates.** Snapshots of the block layout of a page can be saved as named disk files, then later recalled and superimposed on the current page. Use such "templates" for standard multicolumn layouts, common letter formats, and fixed-field forms. A single keystroke dispatches you quickly from block to block as you fill in your page.




- **File Switching.** Moving from document to document to examine, copy, move and change text is like rolling off a log. You're not confined to one disk file at a time anymore.

## Simplicity

- **Auto Word Break.** Forget the right margin. Wordsmith notices when you won't be able to complete the current word and moves it to the next line for you as you continue typing.
- **Understandable Commands.** The most frequently used commands are single keystrokes. The rest are easily remembered abbreviations.
- **Informative Status Lines.** The top two screen lines constantly display page number information, document name, cursor position, tab stops and status/error phrases. You're always in touch with your document.




- **Protection Against Catastrophic Errors.** It's nearly impossible to ruin your document with a single bad command. Wordsmith's page oriented design and double-checking user interface help you do what you mean!

# The Wordsmith™

## TEXT EDITOR

Defining the New Generation of Text Editing

**from Micro Diversions, Inc.**
**8455-D Tyco Rd.**
**Vienna, Va. 22180**
**(703) 827-0888**

- Direct CP/M® and North Star DOS compatibility
- Available for 40x86, 24x80 and 16x64 memory-mapped video boards
- Fully reentrant for efficient multi-programming environments (6K program space, 5K data area)
- 8080 and Z80 compatibility

**Ordering Information:**
$200
(Screensplitter™ Owners $80)
Manual only $15
Check, VISA, Mastercharge

1. CP/M or North Star DOS version?
2. TTY or QUME interface?
3. Brand and memory address of video display board?
4. Ship on single or double density, 5" or 8" diskette?

**Inquire about our custom keyboard.**

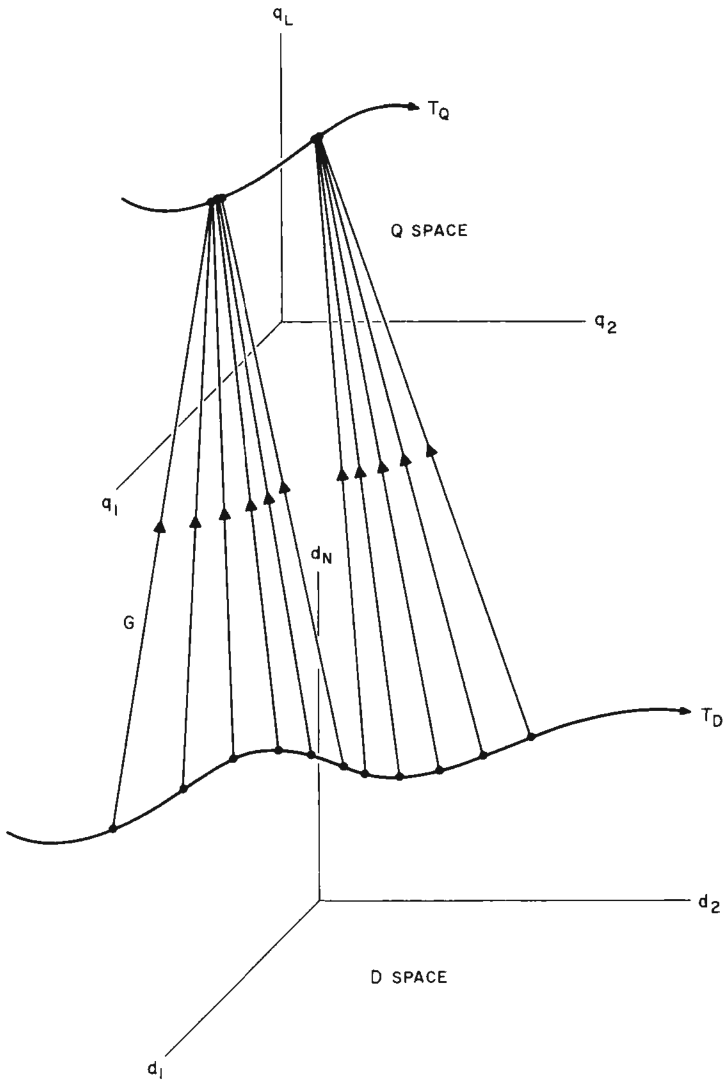



*Figure 21. A time varying* D *vector traces out a trajectory $T_D$ which represents a sensory experience $T_E$ taking place in the context $T_R$. A section of a $T_D$ trajectory which maps into a small region of Q space corresponds to the recognition of an extended temporal pattern as a single event.*

## The Use of Context

It frequently occurs in pattern recognition or signal detection that the instantaneous value of the sensory input vector **E** is ambiguous or misleading. This is particularly true in noisy environments or in situations where data dropouts are likely to occur. In such cases the ambiguity can often be resolved or the missing data filled in if the context can be taken into account, or if the classification decision can make use of some additional knowledge or well founded prediction regarding what patterns are expected.

In CMAC the addition of context or prediction variables **R** to the sensory input **E** such that **D** = **E** + **R** increases the dimensionality of the pattern input space. The context variables thus can shift the total input (pattern) vector **D** to different parts of input space depending on the context. Thus, as shown in figure 22, the ambiguous patterns $\mathbf{E}_1$ and $\mathbf{E}_2$, which are too similar to be reliably recognized as being in separate classes, can easily be distinguished when accompanied by context $\mathbf{R}_1$ and $\mathbf{R}_2$.

In the brain, many variables can serve as context variables. In fact, any fiber carrying information about anything occurring simultaneously with the input pattern can be regarded as context. Thus context can be data from other sensory modalities as well as information regarding what is happening in the behavior-generating hierarchy. In many cases, data from this latter source is particularly relevant to the pattern recognition task, because the sensory input at any instant of time depends heavily upon what action is currently being executed. For example, information from the behavior-generating hierarchy provides contextual information necessary for the visual processing hierarchy to distinguish between motion *of* the eyes and motion of the room *about* the eyes.

In a classic experiment, von Holst and Mittelstaedt demonstrated that this kind of contextual data pathway actually exists in insects. They observed that a fly placed in a chamber with rotating walls will tend to turn in the direction of rotation so as to null the visual motion. They then rotated the fly's head 180° around its body axis (a procedure which for some reason is not fatal to the fly) and observed that the fly now circled endlessly. By attempting to null the visual motion it was now actually increasing it.

Later experiments with motion perception in humans showed that the perception of a stationary environment despite motion of the retinal image caused by moving the eyes is dependent on contextual information derived from the behavior-generating hierarchy. The fact that the context is actually derived from the behavior-generating hierarchy rather than from sensory feedback can be demonstrated by anesthetizing the eye muscles and observing that the effect depends on the *intent* to move the eyes, and not the physical act of movement. The perceptual correction occurs even when the eye muscles are paralyzed so that no motion actually results from the conscious intent to move.

## CMAC as a Predictive Memory

Contextual information can also provide predictions of what sensory data to expect. This allows the sensory processing modules to do predictive filtering, to compare incoming data with predicted data, and to "flywheel" through noisy data or data dropouts.

The mechanism by which such predictions, or expectations, can be generated is illustrated in figure 23. Here contextual input for the sensory processing hierarchy is shown as being processed through a CMAC M module before being presented to the sensory pattern recognition G modules at each level. Inputs to the M modules derive from the P vector of the corresponding behavior-generating hierarchy at the same level, as well as an X vector which includes context derived from other areas of the brain, such as other sensory modalities or other behavior-generating hierarchies. These M modules compute R = M(P + X). Their position in the links from the behavior-generating to the sensory processing hierarchies allows them to function as a predictive memory.

They are in a position to store and recall (or remember) sensory experiences (E vector trajectories) which occur simultaneously with P and X vector trajectories in the behavior-generating hierarchy and other locations within the brain. For example, data may be stored in each $M_i$ module by setting the desired output $\hat{R}_i$ equal to the sensory experience vector $E_i$. At each instant of time t = k, sensory data represented by $E_i^k$ will then be stored on the set of weights selected by the $P_i^k + X_i^k$ vector. The result will be that the sensory experience represented by the sensory data trajectory $T_{E_i}$ will be stored in association with the context trajectory $T_{P_i+X_i}$.

Any time afterwards, t = k + j, a reoccurrence of the same context vector $P_i^{k+j} + X_i^{k+j} = P_i^k + X_i^k$ will produce an output $R_i^{k+j}$ equal to the $E_i^k$ stored at time t = k. Thus a reoccurrence of the same context trajectory $T_{P_i} + X_i$ will produce a recall trajectory $T_{R_i}$ equal to the earlier sensory

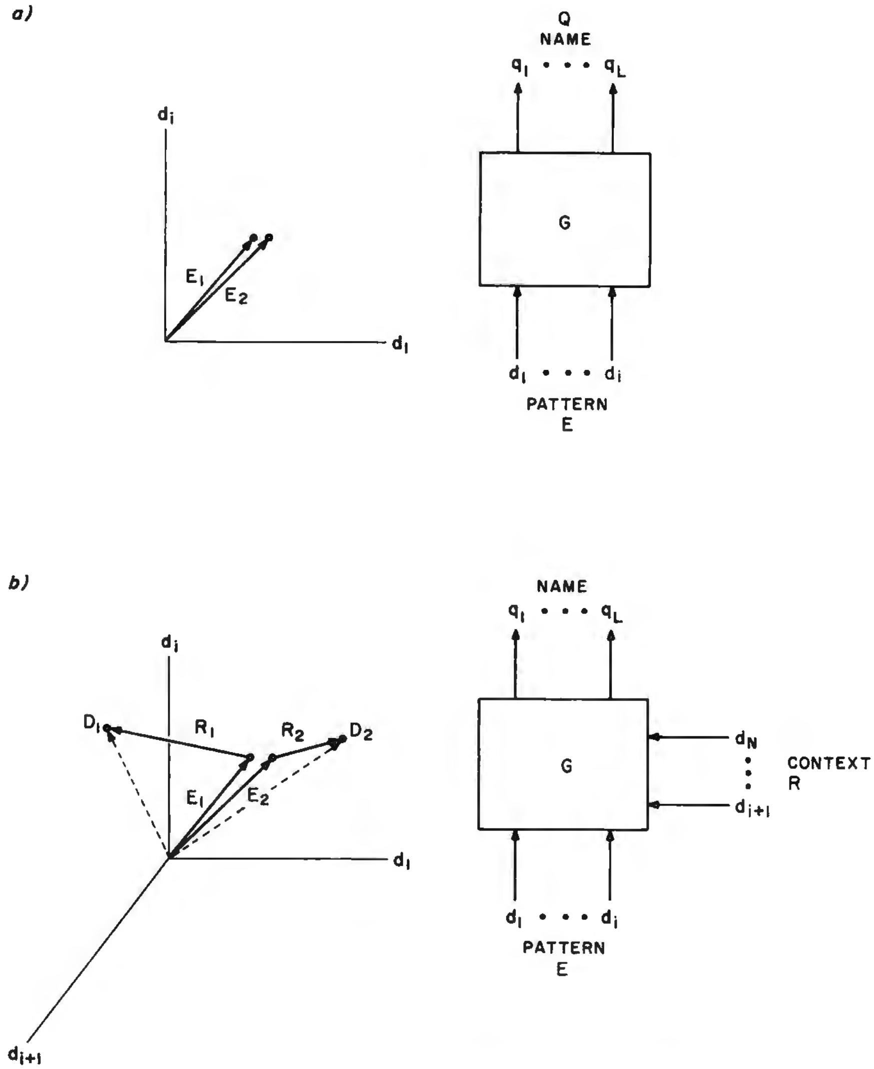



*Figure 22. In (a) the two pattern vectors $E_1$ and $E_2$ are too close together in pattern space to be reliably recognized (ie: named) as being in different classes. In (b) the addition of context $R_1$ to $E_1$ and $R_2$ to $E_2$ makes the vectors $D_1$ and $D_2$ far enough apart in pattern plus context space to be easily recognized as in separate classes.*

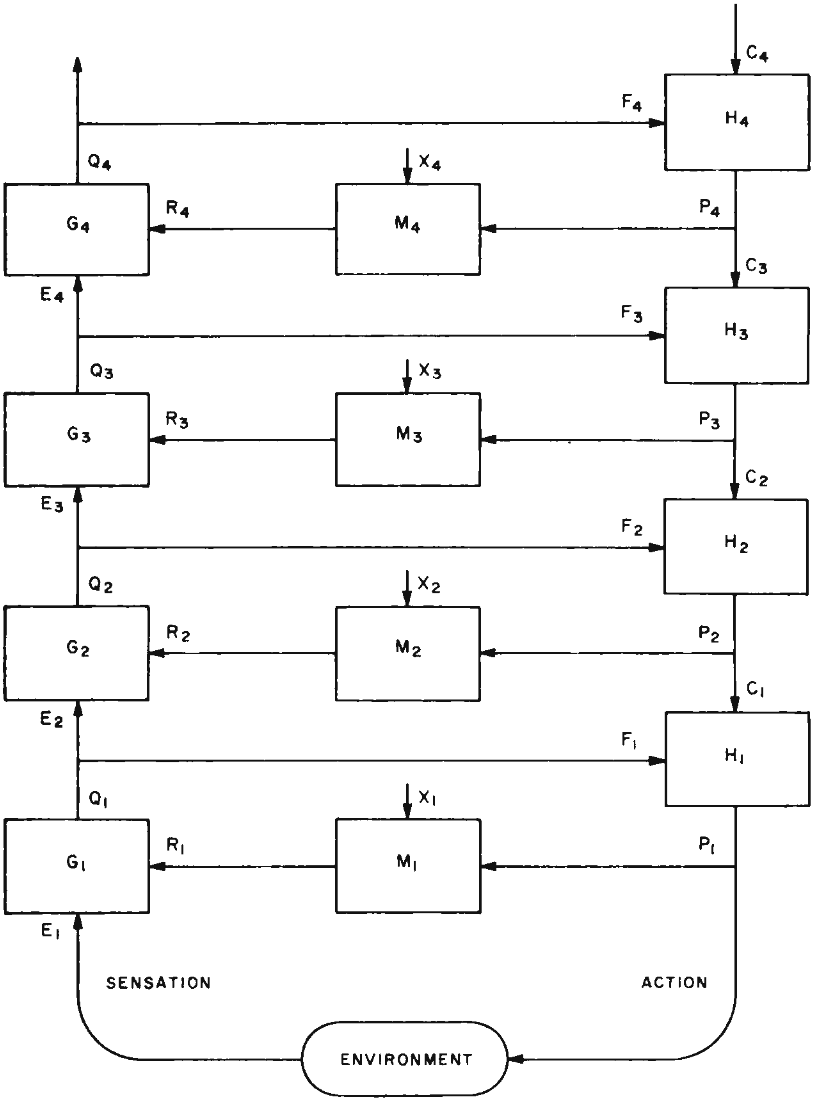



*Figure 23. A cross-coupled, processing-generating hierarchy. The $M_i$ modules remember sensory experiences which occur in association with specific activity in the generating hierarchy ($P_i$) and other sensory modalities ($X_i$). The $M_i$ modules thus learn a set of internal expectations (ie: a predictive model) of the external world as seen through the sensory input channels.*

experience $T_{E_i}$. These predictive memory modules thus provide the sensory processing hierarchy with a memory trace of what sensory data occurred on previous occasions when the motor generating hierarchy (and other parts of the brain) were in similar states along similar trajectories. This provides the sensory processing system with a prediction of what sensory data to expect. What is expected is whatever was experienced during similar activities in the past.

In the ideal case, the predictive memory modules $M_i$ will generate an expected sensory data stream $T_{R_i}$ which exactly duplicates the observed sensory data stream $T_{E_i}$. To the extent that this occurs in practice it enables the $G_i$ modules to apply very powerful mathematical techniques to the sensory data. For example, the $G_i$ modules can use the expected data $T_{R_i}$ to:

- Perform cross-correlation or convolution algorithms to detect sync patterns and information bearing sequences buried in noise.
- Flywheel through data dropouts and noise bursts.
- Detect (or recognize) deviations or even omissions from an expected pattern as well as the occurrence of the pattern in its expected form.

If we assume, as shown in figure 23, that predictive recall modules exist at all levels of the processing-generating hierarchy, then it is clear that the memory trace itself is multileveled. In order to recall an experience precisely at all levels, it is necessary to generate the same context (ie: $P_i + X_i$ address) at all levels as existed when the experience was recorded.

**Internal World Model**

We can say that the predictive memory

modules $M_i$ define the brain's internal model of the external world. They provide answers to the question, "If I do this and that, what will happen?" The answer is that whatever happened before when this and that was done will probably happen again. In short, IF I do Y, THEN Z will happen when Z is whatever was stored in predictive memory the last time (or some statistical average over N last times) that I did Y, and Y is some action such as performing a task or pursuing a goal in a particular environment or situation, which is represented internally by the **P** vectors at the various different levels of the behavior-generating hierarchy and the **X** vectors describing the states of various other sensory processing behavior-generating hierarchies.

The $M_i$ modules (as all CMAC modules) can be thought of as storing knowledge in the form of IF/THEN rules. The CMAC property of generalization produces a recall vector $R_i$ (a THEN consequent) which is similar to the stored experience so long as the context vector $P_i + X_i$ (the IF premise) is within some neighborhood of the context vector during storage.

Much of the best and most exciting work now going on in the field of artificial intelligence revolves around IF/THEN production rules, and how to represent knowledge in large computer programs based on production rules. Practically any kind of knowledge, or set of beliefs, or rules of behavior can be represented as a set of production rules. The CMAC hierarchy shown in figure 23 illustrates how such computational mechanisms can arise in the neurological structure of the brain.

## Conclusion

We have now completed the second step in our development. I have described a neurological model which can store and recall (and hence compute) a broad class of mathematical functions. I have shown how a hierarchical network of such models can execute tasks, seek goals, recognize patterns, remember experiences, and generate expectations. The final part of this series will include a brief overview of evidence that such networks actually exist in the brain. Also, this part will describe how a CMAC hierarchy can create plans, solve problems, and produce language. Finally I will discuss the design of robot control systems incorporating these properties and offer some suggestions as to how brain-like computing networks might be constructed and trained.■

**Practically any kind of knowledge can be represented as a set of production rules.**

### ADDITIONAL READING


1. Albus, J S, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *Journal of Dynamic Systems, Measurement, and Control*, September 1975, pages 220 thru 233.
2. Albus, J S, "A Theory of Cerebellar Function," *Mathematical Biosciences*, 10, 1971, pages 25 thru 61.
3. Eccles, J C, M. Ito, and J. Szentagothai, *The Cerebellum as a Neuronal Machine*, Springer, Berlin, 1967.
4. Stelmach, G E, (editor) *Motor Control: Issues and Trends*, Academic Press, New York, 1976.




Circle 33 on inquiry card.

# Event Queue

*In order to gain optimum coverage of your organization's computer conferences, seminars, workshops, courses, etc, notice should reach our office at least three months in advance of the date of the event. Entries should be sent to: Event Queue, BYTE Publications, 70 Main St, Peterborough NH 03458. Each month we publish the current contents of the queue for the month of the cover date and the two following calendar months. Thus a given event may appear as many as three times in this section if it is sent to us far enough in advance.*

**July 2-13, Applications of Microcomputers to Life Science Education,** Michigan Technological University, Houghton MI. The emphasis of this workshop will be on the use and development of educational computer models and simulations. The workshop will be set up to provide a maximum interaction with the microcomputer. Discussions will focus on ways of employing computer modeling techniques in undergraduate course work. Contact Dept of Biological Sciences, Michigan Technological University, Houghton MI 49931.

**July 9-20, Computing Systems Reliability,** University of California, Santa Cruz CA. Contact Institute in Computer Science, University of California Extension, Santa Cruz CA 95064.

**July 11-13, Microcomputer Applications,** Southern Technical Institute, Marietta GA. The emphasis of this seminar will be on the applications of microcomputers in industry. Software, hardware and interfacing techniques will be discussed. Contact Dr Richard L Castellucis, Southern Technical Institute, Electrical Engineering Technology Dept, 534 Clay St, Marietta GA 30060.

**July 16-18, Software Engineering,** Crystal City Marriott, Arlington VA. This seminar is intended to familiarize the project manager, the system analyst, and the application programmer with techniques of developing software to meet user needs. Contact Information Technology Inc, POB 10129, Austin TX 78766.

**July 16-18, Data Base Management,** Crystal City Marriott, Arlington VA 22202. This seminar is intended to familiarize the applications programmer, data processing manager, and software system engineer with the latest techniques for the implementation and use of data base management. Contact Information Technology, POB 10129, Austin TX 78766.

**July 16-27, Introduction to Digital Electronics and Microcomputer Interfacing,** Lexington VA. This hands-on laboratory course is for academic and industrial personnel. There will be approximately 60 hours of laboratory instruction with one microcomputer laboratory station for each two participants. Contact Prof Philip Peters, Dept of Physics, Virginia Military Institute, Lexington VA 24450.

**July 19-20, Project Management,** Crystal City Marriott, Arlington VA. The purpose of this seminar is to provide a basic understanding of the methodologies, tools, techniques and skills of software management. Contact Information Technology Inc, POB 10129, Austin TX 78766.

**July 19-20, Structured Programming,** Crystal City Marriott, Arlington VA. This course is aimed at both programmers and managers. It will cover an integrated set of software development techniques that can be scaled up for any size project development. It supports the development of error free programs by providing the programmer with effective means of controlling the design and code through continual validation checks. Contact Information Technology Inc, POB 10129, Austin TX 78766.

**July 19-20, BASIC: A Computer Language for Executives,** New York NY. Executive computing, problem solving, planning, forecasting and database systems will be discussed. Also to be covered are programming fundamentals, the mindless computer, sequence, decision and iteration, computer languages and BASIC. Contact American Management Associations, 135 W 50th St, New York NY 10020.

**July 23-27 Finite Element Method in Mechanical Design,** University of Michigan, Ann Arbor MI. This course is intended for engineers working in mechanical design where knowledge of stresses, displacements, or vibratory motion is important. No previous experience with finite elements is assumed. The course will familiarize the attendee with finite element modeling concepts and will review the fundamentals on which the method is based. Contact Engineering Summer Conferences, 400 Chrysler Ctr, North Campus, University of Michigan, Ann Arbor MI 48109.





Circle 294 on inquiry card

**August 1-3, Microcomputer Applications,** Southern Technical Institute, Marietta GA. The emphasis of this seminar will be on the applications of microcomputers in industry. Software, hardware and interfacing techniques will be discussed. Contact Dr Richard L Castellucis, Southern Technical Institute, Electrical Engineering Technology Dept, 534 Clay St, Marietta GA 30060

**August 6-8, Pattern Recognition and Image Processing,** Hyatt Regency Chicago O'Hare, Chicago IL. This conference is sponsored by the Machine Intelligence and Pattern Analysis Committee of the IEEE Computer Society. The program will consist of submitted and invited papers and a large trade show of graphics and image processing equipment. Contact PRIP 79, POB 639, Silver Spring MD 20901.

**August 6-10, SIGGRAPH '79,** Chicago IL. This sixth annual conference on computer graphics will feature tutorials, technical sessions and an exposition of state-of-the art computer graphics and image processing equipment. Contact Maxine D Brown, SIGGRAPH '79 Exposition, Hewlett-Packard, 19400 Homestead Rd, Cupertino CA 95014.

**August 6-10, Modern Communication Systems: Analysis and Design,** University of Southern California, Los Angeles CA. This course is devoted to the analysis and design of modern communication systems, with emphasis on the derivation of practical design equations useful for trade-off studies and overall synthesis. Contact University of Southern California, Continuing Engineering Education, Los Angeles CA 90007.

**August 6-10, Advanced Microcomputer System Development: High Level Languages, Technology Trends, and Hands-On Experience,** University of Southern California, Los Angeles CA. This course is intended to present the participants with a clear picture of the microcomputer revolution, provide hands-on programming experience using Extended BASIC and FORTRAN, analyze technology trends in the microcomputer field, and assess the impact of VHSI/VLSI. Contact University of Southern California, Continuing Engineering Education, Los Angeles CA 90007.

**August 8-10, SIGPLAN Symposium on Compiler Construction,** Boulder CO. This symposium will consider methods of, and experience with, constructing compilers. The emphasis will be less on theoretical methods, and more on techniques applied to real compilers. Contact Professor Leon Osterweil, Dept of Computer Science, University of Colorado, Boulder CO 80309.

**August 8-10, First Annual Conference on Research and Development in Personal Computing,** Hyatt Regency O'Hare, Chicago IL. This conference is sponsored by the Association for Computing Machinery (ACM) Special Interest Group on Personal Computing (SIGPC). A large trade show of personal computer and graphics equipment is planned to accompany an assortment of papers, panels, user group meetings, workshops, and person-to-person poster booths. Contact Bob Gammill, Computer Science Division, Dept of Mathematical Sciences, 300 Minard Hall, North Dakota State University, Fargo ND 58102.

**August 13-15, Conference on Simulation, Measurement and Modeling of Computer Systems,** Boulder CO. This conference will feature performance prediction techniques employed during the design, procurement and maintenance of computer systems. It will provide a forum for both applied and theoretical work in the disciplines of performance monitoring, modeling, and simulation of computer systems. Contact Gary Nutt, Xerox PARC, 3333 Coyote Hill Rd, Palo Alto CA 94304.

**August 13-16, Q-GERT Network Modeling and Analysis,** Ramada Inn, Lafayette IN 47905. This course will provide the attendee with the information necessary to model complex systems using Q-GERT. Emphasis will be on the procedures for modeling and analysis. Contact Pritsker and Associates Inc, POB 2413, W Lafayette IN 47906.

**August 13-17, High Speed Computation: Vector Processing,** The University of Michigan, Ann Arbor MI. In this course, the architectural, software, and algorithmic issues of vector architecture are coordinated by the discussion of concepts in computer architecture, and by detailed study of current vector processors and their use. Contact Engineering Summer Conferences, 400 Chrysler Ctr, North Campus, The University of Michigan, Ann Arbor MI 48109.

**August 19-22, International Conference on Computing in the Humanities,** Dartmouth College, Hanover NH. This conference is intended to foster computer research and technique in all areas of humanistic study; to promote international cooperation in the development of programs, data banks, and equipment; and to make the results of research available. The program will include a plenary session each evening and shorter sessions during the day. Contact Stephen V F Waite, Kiewit Computation Ctr, Dartmouth College, Hanover NH 03755.

**August 19-24, 1979 Symposium for Innovation in Measurement Science,** Hobart and William Smith Colleges, Geneva NY. Sponsored by the Scientific Instrumentation and Research Division of the Instrument Society of America, scheduled sessions at this symposium include innovation in computers and electronics, mass flow measurement, chemical analysis, applied analysis in instrument control, physical analysis, medical instrumentation, and advances in industrial measurement. Contact Instrument Society of America, 400 Stanwix St, Pittsburgh, PA 15222.

**August 23-26, National Small Computer Show,** New York Coliseum, New York NY. Exhibitions will include those of major manufacturers, distributors, and publications in the small computer field. A lecture series will include topics of interest to business and professional people, hobbyists and the general public. Contact National Small Computer Show, 74 E 56th St, New York NY 10022.

**September 4-6, International Conference and Exhibition on Engineering Software,** University of Southampton, England. The aim of this conference is to provide a forum for the presentation and discussion of recent advances in engineering software and the state-of-the-art in this field. The exhibition, held in conjunction with the conference, will cover all software products, services and equipment related to engineering software. Contact Dr R Adey, Engsoft, 6 Cranbury Place, Southampton SO2 OLG, ENGLAND.

**September 4-7, Compcon Fall '79,** Capital Hilton Hotel, Washington DC. This 18th IEEE Computer Society International conference will present the latest developments in microprocessor architecture, support software, operating systems, and peripheral devices. Contact IEEE Computer Society, POB 639, Silver Spring MD 20901.

**September 5-8, Info/Asia,** Ryutsu Center, Tokyo. This exposition will be devoted to information management, computers, word processing and advanced business equipment. The exposition will be accompanied by a four day conference. Contact Clapp and Poliak Inc, 245 Park Av, New York NY 10017.

**September 18-20, Wescon/79,** St Francis Hotel, San Francisco CA. Contact Electronic Conventions Inc, 999 N Sepulveda Blvd, El Segundo CA 90245.

**September 25-27, WPOE '79,** San Jose Convention Ctr, San Jose CA. This show will be dedicated to word processing and office/business equipment, services and materials. Complementing the exhibit will be a three day executive conference program that focuses on emerging technologies and their applications in the office. Contact Cartlidge and Associates Inc, 491 Macara Av, Suite 1014, Sunnyvale CA 94086.

**September 26-29, MIMI '79,** Queen Elizabeth Hotel, Montreal, Canada. This symposium is intended as a forum for the presentation and discussion of recent advances in mini and microcomputers and their applications. Special emphasis will be given to the theme of the conference, "The Evolving Role of Minis and Micros Within Distributed Processing." Contact The Secretary, MIMI '79 Montreal, POB 2481, Anaheim CA 92804.

**September 28-30, Northeast Personal and Business Computer Show,** Hynes Auditorium, Boston MA. Displays and exhibits will showcase microcomputers and small computer systems of interest to businesspeople, hobbyists, professionals, etc. Lectures and seminars will be presented for all categories and levels of enthusiasts, including introductory classes for novices. Contact Northeast Exposition, POB 678, Brookline MA 02147. ■

# Letters

*Text continued from page 6:*
written) would only perform garbage collection.

There are more facilities which could be added to this simple data base structure, but it would probably be better to stop at this point.

**Jack L Warner**
**Bell Laboratories**
**600 Mountain Ave**
**Murray Hill NJ 07974**

## HAMMING ERROR CORRECTING CODE HAS PROBLEMS?

Michael Wimble recently described a method for storing coded data using a Hamming error correcting code which will correct a single bit error and detect double bit errors (February 1979 BYTE, page 180). It is very similar to a scheme I have used successfully for several years and recently published (*Computer Design*, September 1978).

Mr Wimble's scheme, however, will cause havoc with some data recording devices as some of his coded bytes are exactly the same as some of the common control characters, ie: his data 3, coded as a hexadecimal 93, is identical to the ASCII Device Control 3 character, and data E, coded as a hexadecimal IE, is the ASCII Record Separator. The latter should not cause trouble, but the former will automatically activate or deactivate some kinds of papertape terminals.

If this is likely to be a problem on the equipment you are using, one simple way to overcome it is to omit the P4 parity bit altogether and use strictly 7 bit codes. The media channel on which the omitted bit would have been stored is then placed so that no objectionable control characters are ever generated (except, of course, by error). Programs to generate and decode such schemes are presented in the *Computer Design* paper cited above.

You are not able to detect double errors using only 7 bits, but if your equipment is that bad it's probably time to pack it in anyway.

**George White**
**Institut de Recherche**
**d'Informatique et d'Automatique**
**Domaine de Voluceau**
**Rocquencourt BP 105**
**78150 Le Chesnay FRANCE**

## COMMENTS FROM A CHESS MASTER

I was most flattered to read the story about my chess match with CHESS 4.7 ("Chess 4.7 versus David Levy" December 1978 BYTE, page 84) and I am delighted that you have been giving chess such excellent coverage in the pages of your magazine.

Since your article appeared I have been plagued by people writing to ask whether I have collected the $2,500 that I won in the bet. Professors Donald Michie (Edinburgh University), John McCarthy (Stanford University) and Seymour Papert (MIT) paid promptly and with great sportsmanship, just as I would have done had I lost the bet. Edward Kozdrowicki (Aerospace Corporation, El Segundo CA) has refused all attempts to persuade him to pay.

I hope that this will answer any further readers who might be curious about the bet.

David N L Levy
**104 Hamilton Terrace**
**London NW8 9UP**
**ENGLAND** ■

# BYTE's Bugs

### Community Bulletin Board Correction

In the BYTE News for April 1979 (page 195) we mentioned that there was a PCNET run by the Chicago Area Computer Hobbyist Exchange. We should have said that a Community Bulletin Board is privately run by Ward Christensen and Randy Suess.

### Correction

In May's "What's New" on page 254 we listed Semionics Associates' REM S-100 board as having a capacity of 8 K bytes and priced at $525. This *should* have read "The REM S-100 add-in recognition memory board has a capacity of 4 K bytes and is priced at $345."

### Trap Door Trap

For shame! The National Bureau of Standards standard data encryption algorithm is *not* a trap-door algorithm. That term refers to one-way or public key systems.

**Don McClimans**
**Computer Systems Consultant**
**41 Washburn Pk**
**Rochester NY 14620**

*Oops! You, along with several other people, caught us with that one.* [RGAC] ■

# BYTE News . . . .

FCC TRYING TO CRACK DOWN ON TV INTERFERENCE. The Federal Communications Commission (FCC) has asked Atari, Apple, Commodore, Heath, Southwest Technical Products, and Radio Shack to submit their personal computer systems for TV interference testing. The systems made by these companies are presently exempted from FCC regulations since they are not directly connected to a TV set. However, there have been complaints regarding radio frequency (RF) interference from personal computer systems, and the FCC has decided to develop regulations regarding permissible RF radiation levels.

The computer manufacturers involved have indicated a willingness to cooperate with the FCC's effort. The regulations could fine noncomplying manufacturers and permit the issuance of cease and desist orders. Some industry experts feel that a few manufacturers' computer systems would not pass the FCC regulations.

INTEL ENHANCES 8086 FAMILY WITH I/O PROCESSOR. Intel continues to lead the way in microprocessor and microcomputer systems. Recently they announced the 8089, an I/O (input/output) processor to work with the 8086 16 bit microprocessor. This processor can more than double the performance of the 8086 by relieving it of I/O operations, much like the communications channel on an IBM 370.

$200 DISK SYSTEM EXPECTED BY YEAR END. Shugart and Matsushita Electric of Japan have signed an agreement whereby Matsushita will manufacture a low cost version of Shugart's popular minifloppy disk drive. The drive is expected to sell for $50 in large OEM quantities and retail at about $125. Add to this the interface/controller circuitry, and the total retail cost should work out to a little over $200. This is less than a third of the price of current minifloppy systems. Matsushita expects to be making 100 drives per hour by year end.

The drive will store 70 K bytes, use a new head design, and be housed in sheet metal rather than cast aluminum. It will be only 2 inches high, half the height of the current drive. An industrial version with heavy duty components will be sold at $65 OEM.

Nippon Electric (NEC) is also rumored to be developing a low cost 5 inch disk drive.

14 MILLION MICROPROCESSORS SOLD LAST YEAR. That's right, 14 million microprocessors were manufactured in 1978. One million 8 bit microprocessors and 13 million 4 bit microprocessors were made. If you didn't realize it already, most were used in games. The most manufactured microprocessors were the 8 bit 6502 and the 4 bit TMS-1000. However, sales of electronic games using microprocessors have recently taken a sharp drop. Hence, the probability exists that there may be a slight decrease in microprocessor production in 1979.

16 BIT MICROPROCESSOR PICTURE STILL FUZZY. It is beginning to look as if Intel may have taken the right approach with the 8086 by designing a part which could be placed in production far ahead of the Zilog Z-8000 or Motorola 68000. They have over a year's head start compared to the Z-8000 and possibly another half year's lead over the 68000.

The 8086 part is far simpler than the Z-8000 or 68000 parts, and as a result it is closer to the earlier generations of microprocessors. Support parts for the 8086 such as the new 8087 floating point coprocessor also give the 8086 a commanding availability lead over the other two contenders at this time. All three machines are aimed at the high end of microcomputer application, providing significant computational power equivalent to traditional mini and main frame computers.

In the meantime, the traditional minicomputer manufacturers are not sitting still. Digital Equipment Corp (DEC), the largest of the "old-time" minimakers, has created an integrated circuit manufacturing facility to make its own 16 bit microprocessor (the LSI-11). This fall they will be making a super-micro mainframe, called the LSI-11/23, that will have almost all the power of a standard minicomputer at a fraction of the price ($6,800 compared to $12,000).

THE JAPANESE ARE COMING. Although last year a few Japanese electronics manufacturers introduced personal computer systems on their own home ground, none, so far, has ventured into the US market. This is probably due to the competition that already exists and the lack of sufficient price markup on personal computer systems. If the Japanese enter the personal computer market, it will probably be in the peripherals area.

However, the first major Japanese manufacturer has entered the small business computer market. NEC has introduced its ASTRA series of 16 bit microcomputer systems that start at $13,000 and range up to $130,000. The video terminal employs a Z-80 processor.

NEW BUBBLE MEMORY TECHNOLOGY. In a paper delivered by a Bell Laboratory researcher at a recent conference, it was disclosed that Bell Labs has made a major breakthrough in bubble memory technology. This breakthrough will mean a four times increase in storage size, a substantial decrease in cost and ten times faster operating speed. Although Texas Instruments and Rockwell have been in production on bubble memory devices for nearly a year, their high cost and small storage capability have prohibited their wide use. This new development, which will still take a few years to reach the market, should have a large impact on the mass storage area, particularly floppy disks.

The new device replaces the drive coils used in present bubble memories with wafer-thin conductive layers of gold or aluminum overlaid on the garnet structure. A current flows through these layers forming tiny magnetic fields around holes etched into the surface. The polarity of these fields controls the bubble movements.

By eliminating the costly and bulky coil structure, a new pathway design became available which provides a fourfold increase in storage capacity, is easier and less costly to produce, and reduces integrated circuit size, thereby reducing travel time.

IBM also announced that it has fabricated bubble memory devices that are 1 square inch in size and contain 25 M bits. These devices were made at the IBM Research Center in Yorktown Heights NY.

In the meantime, TI and Rockwell are currently sending out samples of their 256 K bit bubble devices and expect to be in production on these units by the end of the year. They expect to be sampling 1 M bit devices by the end of 1980, with production beginning in 1981.

On the whole, it does not appear that bubbles will provide any meaningful competition to floppies until the mid 1980s.

PERSONAL COMPUTER/CABLE TELEVISION SYSTEM PLANNED. Six Star Cablevision, a Los Angeles cable television outfit, will soon begin test marketing a personal computer system designed for use with a closed circuit TV system. Six Star will allocate 3 of 42 available channels to transmit data from data banks to subscribers. They claim to have 50 applications programs already prepared, which would be regenerated every 7 seconds. They plan to use a Mattel personal computer system with a printer, and charge $4 to $6 above the regular $7.50 monthly fee.

**Sol Libes**
**ACGNJ**
**1776 Raritan Rd**
**Scotch Plains NJ 07076**

# ...And the future

**THE BYTE BOOK OF COMPUTER MUSIC** combines the best computer music articles from past issues of BYTE Magazine with exciting new material—all written for the computer experimenter interested in this fascinating field.

You will enjoy Hal Chamberlin's "A Sampling of Techniques for Computer Performance of Music", which shows how you can create four-part melodies on your computer. For the budget minded, "A $19 Music Interface" contains practical tutorial information—and organ fans will enjoy reading "Electronic Organ Chips For Use in Computer Music Synthesis".

New material includes "Polyphony Made Easy" and "A Terrain Reader". The first describes a handy circuit that allows you to enter more than one note at a time into your computer from a musical keyboard. The "Terrain Reader" is a remarkable program that creates random music based on land terrain maps.

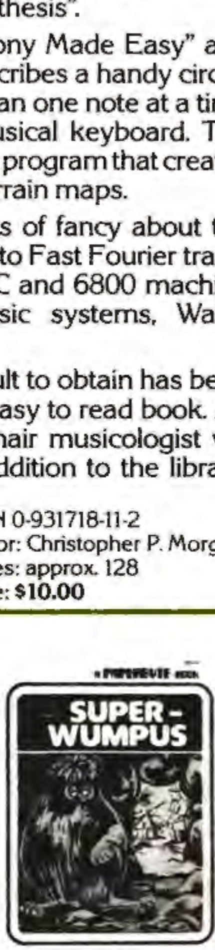
Other articles range from flights of fancy about the reproductive systems of pianos to Fast Fourier transform programs written in BASIC and 6800 machine language, multi-computer music systems, Walsh Functions, and much more.

For the first time, material difficult to obtain has been collected into one convenient, easy to read book. An ardent do-it-yourselfer or armchair musicologist will find this book to be a useful addition to the library.

NEW

ISBN 0-931718-11-2
Editor: Christopher P. Morgan
Pages: approx. 128
Price: **$10.00**

**SUPERWUMPUS** is an exciting computer game incorporating the original structure of the WUMPUS game along with added features to make it even more fascinating. The original game was described in the book **What To Do After You Hit Return,** published by the People's Computer Company. Programmed in both 6800 assembly language and BASIC, SUPERWUMPUS is not only addictively fun, but also provides a splendid tutorial on setting up unusual data structures (the tunnel and cave system of SUPERWUMPUS forms a dodecahedron). This is a **PAPERBYTE**™ book.




ISBN 0-931718-03-1
Author: Jack Emmerichs
Pages: 56
Price: **$6.00**

**TINY ASSEMBLER 6800, Version 3.1** is an enhancement of Jack Emmerichs' successful Tiny Assembler. The original version (3.0) was described first in the April and May 1977 issues of BYTE magazine, and later in the **PAPERBYTE**™ book **TINY ASSEMBLER 6800 Version 3.0.**


In September 1977, BYTE magazine published an article entitled, "Expanding The Tiny Assembler". This provided a detailed description of the enhancements incorporated into Version 3.1, such as the addition of a "begin" statement, a "virtual symbol table", and a larger subset of the Motorola 6800 assembly language.

All the above articles, plus an updated version of the user's guide, the source, object and PAPERBYTE™ bar code formats of both Version 3.0 and 3.1 make this book the most complete documentation possible for Jack Emmerichs' Tiny Assembler.

ISBN 0-931718-08-2
Author: Jack Emmerichs
Pages: 80
Price: **$9.00**

A walk through this book brings you into **Ciarcia's Circuit Cellar** for a detailed look at the marvelous projects which let you do useful things with your microcomputer. A collection of more than a year's worth of the popular series in BYTE magazine, **Ciarcia's Circuit Cellar** includes the six winners of BYTE's On-going Monitor Box (BOMB) award, voted by the readers themselves as the best articles of the month: **Control the World** (September 1977), **Memory Mapped IO** (November 1977), **Program Your Next EROM in BASIC** (March 1978), **Tune In and Turn On** (April 1978), **Talk To Me** (June 1978), and **Let Your Fingers Do the Talking** (August 1978).

Each article is a complete tutorial giving all the details needed to construct each project. Using amusing anecdotes to introduce the articles and an easy-going style, Steve presents each project so that even a neophyte need not be afraid to try it.

NEW

ISBN 0-931718-07-4
Author: Steve Ciarcia
Pages: approx. 128
Price: **$8.00**

# is right now!

**BASEX,** a new compact, compiled language for microcomputers, has many of the best features of BASIC and the 8080 assembly language—and it can be run on any of the 8080 style microprocessors: 8080, Z-80, or 8085. This is a **PAPERBYTE™** book.

Subroutines in the **BASEX** operating system typically execute programs up to five times faster than equivalent programs in a BASIC interpreter—while requiring about half the memory space. In addition, **BASEX** has most of the powerful features of good BASIC interpreters including array variables, text strings, arithmetic operations on signed 16 bit integers, and versatile IO communication functions. And since the two languages, BASEX and BASIC, are so similar, it is possible to easily translate programs using integer arithmetic data from BASIC into BASEX.

The author, Paul Warme, has also included a BASEX Loader program which is capable of relocating programs anywhere in memory.

NEW

ISBN 0-931718-05-8
Author: Paul Warme
Pages: 88
Price: **$8.00**

**PROGRAMMING TECHNIQUES** is a series of **BYTE BOOKS** concerned with the art and science of computer programming. It is a collection of the best articles from BYTE magazine and new material collected just for this series. Each volume of the series provides the personal computer user with background information to write and maintain programs effectively.


The first volume in the Programming Techniques series is entitled **PROGRAM DESIGN.** It discusses in detail the theory of program design. The purpose of the book is to provide the personal computer user with the techniques needed to design efficient, effective, maintainable programs. Included is information concerning structured program design, modular programming techniques, program logic design, and examples of some of the more common traps the casual as well as the experienced programmer may fall into. In addition, details on various aspects of the actual program functions, such as hashed tables and binary tree processing, are included.

ISBN 0-931718-12-0
Editor: Blaise W. Liffick
Pages: 96
Price: **$6.00**

**SIMULATION** is the second volume in the Programming Techniques series. The chapters deal with various aspects of specific types of simulation. Both theoretical and practical applications are included. Particularly stressed is simulation of motion, including wave motion and flying objects. The realm of artificial intelligence is explored, along with simulating robot motion with the microcomputer. Finally, tips on how to simulate electronic circuits on the computer are detailed.

ISBN 0-931718-13-9
Editor: Blaise W. Liffick
Pages: approx. 80
Price: **$6.00**
Publication: Winter 1979

**RA6800ML: AN M6800 RELOCATABLE MACRO ASSEMBLER** is a two pass assembler for the Motorola 6800 microprocessor. It is designed to run on a minimum system of 16 K bytes of memory, a system console (such as a Teletype terminal), a system monitor (such as Motorola MIKBUG read only memory program or the ICOM Floppy Disk Operating System), and some form of mass file storage (dual cassette recorders or a floppy disk).

The Assembler can produce a program listing, a sorted Symbol Table listing and relocatable object code. The object code is loaded and linked with other assembled modules using the **Linking Loader LINK68.** (Refer to **PAPERBYTE™** publication **LINK68: AN M6800 LINKING LOADER** for details.)

There is a complete description of the 6800 Assembly language and its components, including outlines of the instruction and address formats, pseudo instructions and macro facilities. Each major routine of the Assembler is described in detail, complete with flow charts and a cross reference showing all calling and called-by routines, pointers, flags, and temporary variables.

In addition, details on interfacing and using the Assembler, error messages generated by the Assembler, the Assembler and sample IO driver source code listings, and **PAPERBYTE™** bar code representation of the Assembler's relocatable object file are all included.

This book provides the necessary background for coding programs in the 6800 assembly language, and for understanding the innermost operations of the Assembler.

ISBN 0-931718-10-4
Author: Jack E. Hemenway
Pages: 184
Price: **$25.00**

to order books see next page .....

**LINK68: AN M6800 LINKING LOADER** is a one pass linking loader which allows separately translated relocatable object modules to be loaded and linked together to form a single executable load module, and to relocate modules in memory. It produces a load map and a load module in Motorola MIKBUG loader format. The Linking Loader requires 2 K bytes of memory, a system console (such as a Teletype terminal), a system monitor (for instance, Motorola MIKBUG read only memory program or the ICOM Floppy Disk Operating System), and some form of mass file storage (dual cassette recorders or a floppy disk).

It was the express purpose of the authors of this book to provide everything necessary for the user to easily learn about the system. In addition to the source code and **PAPERBYTE**™ bar code listings, there is a detailed description of the major routines of the Linking Loader, including flow charts. While implementing the system, the user has an opportunity to learn about the nature of linking loader design as well as simply acquiring a useful software tool.

ISBN 0-931718-09-0
Authors: Robert D. Grappel
& Jack E. Hemenway
Pages: 72
Price: **$8.00**
Winter 1979

**TRACER: A 6800 DEBUGGING PROGRAM** is for the programmer looking for good debugging software. TRACER features single step execution using dynamic break points, register examination and modification, and memory examination and modification. This book includes a reprint of "Jack and the Machine Debug" (from the December 1977 issue of BYTE magazine). **TRACER** program notes, complete assembly and source listing in 6800 assembly language, object program listing, and machine readable **PAPERBYTE**™ bar codes of the object code.

ISBN 0-931718-02-3
Authors: Robert D. Grappel
& Jack E. Hemenway
Pages: 24
Price: **$6.00**

**MONDEB: AN ADVANCED M6800 MONITOR-DEBUGGER** has all the general features of Motorola's MIKBUG monitor as well as numerous other capabilities. Ease of use was a prime design consideration. The other goal was to achieve minimum memory requirements while retaining maximum versatility. The result is an extremely versatile program. The size of the entire MONDEB is less than 3 K.

Some of the command capabilities of MONDEB include displaying and setting the contents of registers, setting interrupts for debugging, testing a programmable memory range for bad memory locations, changing the display and input base of numbers, displaying the contents of memory, searching for a specified string, copying a range of bytes from one location in memory to another, and defining the location to which control will transfer upon receipt of an interrupt. This is a **PAPERBYTE**™ book.

ISBN 0-931718-06-6
Author: Don Peters
Pages: 88
Price: **$5.00**

**BAR CODE LOADER.** The purpose of this pamphlet is to present the decoding algorithm which was designed by Ken Budnick of Micro-Scan Associates at the request of BYTE Publications, Inc., for the **PAPERBYTE**™ bar code representation of executable code. The text of this pamphlet was written by Ken, and contains the general algorithm description in flow chart form plus detailed assemblies of program code for 6800, 6502 and 8080 processors. Individuals with computers based on these processors can use the software directly. Individuals with other processors can use the provided functional specifications and detail examples to create equivalent programs.

ISBN 0-931718-01-5
Author: Ken Budnick
Pages: 32
Price: **$2.00**

BYTE BOOKS Division • 70 Main Street • Peterborough, New Hampshire 03458

Please send the books I have checked.

- ☐ Computer Music $10.00
- ☐ SUPERWUMPUS $6.00
- ☐ Tiny Assembler (3.1) $9.00
- ☐ Circuit Cellar $8.00
- ☐ BASEX $8.00
- ☐ Program Design $6.00
- ☐ Simulation $6.00
- ☐ RA6800ML $25.00
- ☐ Link68 $8.00
- ☐ TRACER $6.00
- ☐ Mondeb $5.00
- ☐ Bar Code Loader $2.00

$ _____ Total Books

Add 60 per book

$ _____ Postage/Handling

$ _____ Grand Total

☐ Check enclosed ☐ Bill Visa ☐ Bill Master Charge

Card No. ____________________ Exp. Date ________

Name ______ Title ______ Company ______

Street ______ City ______ State/Province ______ Code ______



Circle 36 on inquiry card.

Kurt Schmucker
Mathematician
Dept of Defense
Washington DC
20755

# The Mathematics of Computer Art

*Figure 1: "Sine Curve Man" by Charles Csuri and James Shaffer, a trigonometrically based plot. Reprinted with permission from* Computers and Automation, *August 1967, Copyright 1967 and published by Berkeley Enterprises Inc, 815 Washington St, Newtonville MA 02160.*

## Introduction

Computer scientists and personal computer enthusiasts have a great appreciation of the beauty and form of art. They often use the tools of their trade, the computer and its associated peripheral devices, to create works of art. These works express particular, somewhat algorithmic and mathematical tastes in art forms. Since the late 1960s the use of computers and computer controlled devices for the generation of this artwork (often in three dimensions) has been firmly established. (See references 3, 4, 8, 9, and 14.) A great portion of this artwork has relied heavily on the computer's ability to precisely manipulate numerical quantities to produce drawings or sculptures that express complex mathematical relationships. Drawings in this category include figures which show the relationships between the phase, amplitude, and periods of different trigonometric functions; graphs of functions of two or more variables; and moiré patterns that can express complex relationships by interaction between families of similar simple curves (see reference 13).

This is not to say that all or even the majority of computer art is inherently mathematical. Two of the latest crazes in computer art, the recreation of natural scenes and the randomly drawn picture (called "controlled serendipity" by one artist in reference 11), are in essence nonmathematical. This article, however, will be concerned only with those figures which have mathematics as their basis.

Among figures which rely heavily on mathematics, two classes can easily be separated. One class is distinguished by the fact that it is precisely the equations themselves which give the figures beauty and appeal. While even the mathematically uninitiated can perceive the beauty of these forms, only those who understand the underlying mathematics can fully appreciate the plots. Some examples of this class are the endless varieties of lissajous figures (see references 2 and 6), and two other famous trigonometrically based plots, "Sine Curve Man" (shown in figure 1; see reference 15) and "Christmas Wreath" (see reference 1). The beauty of "Sine Curve Man" is in part due to the undulating sine curves, differences in the phase

### About the Author

*Kurt J Schmucker has been employed as a mathematician at the Department of Defense in Washington DC since 1974. He has masters degrees from both Michigan State University and Johns Hopkins University. He is now an advanced special student in the Computer Science Department at the University of Maryland and an assistant professorial lecturer in computer science at George Washington University. Mr Schmucker's current interests are in natural language processing and computer graphics.*

*Mr Schmucker is the author of "The Computers of Star Trek," which appeared in the December 1977 BYTE.*

between the different curves, and the variation in the amplitudes. These form the mathematical base for the figure.

The other class of figures relies on mathematics not for the positioning of the actual lines but for the meaning or the importance of the resulting total plot. For these figures, the actual equations which are plotted are not as important as the relationships which are revealed. Some examples of this class are moiré patterns and projection plots of multidimensional figures (see reference 12).

An example of a moiré figure is shown in figure 2. Notice that the lines in this figure are nothing more than regularly spaced radii of two circles — lines whose equations are easily determined. What *is* fascinating is the complex interference pattern, a pattern which can express complex relationships between those lines. In this article, these two classes of figures will be discussed by examining in detail one example of each.

**Crest**

An example of a computer generated figure which relies on complex mathematical relationships for its beauty is the crest,

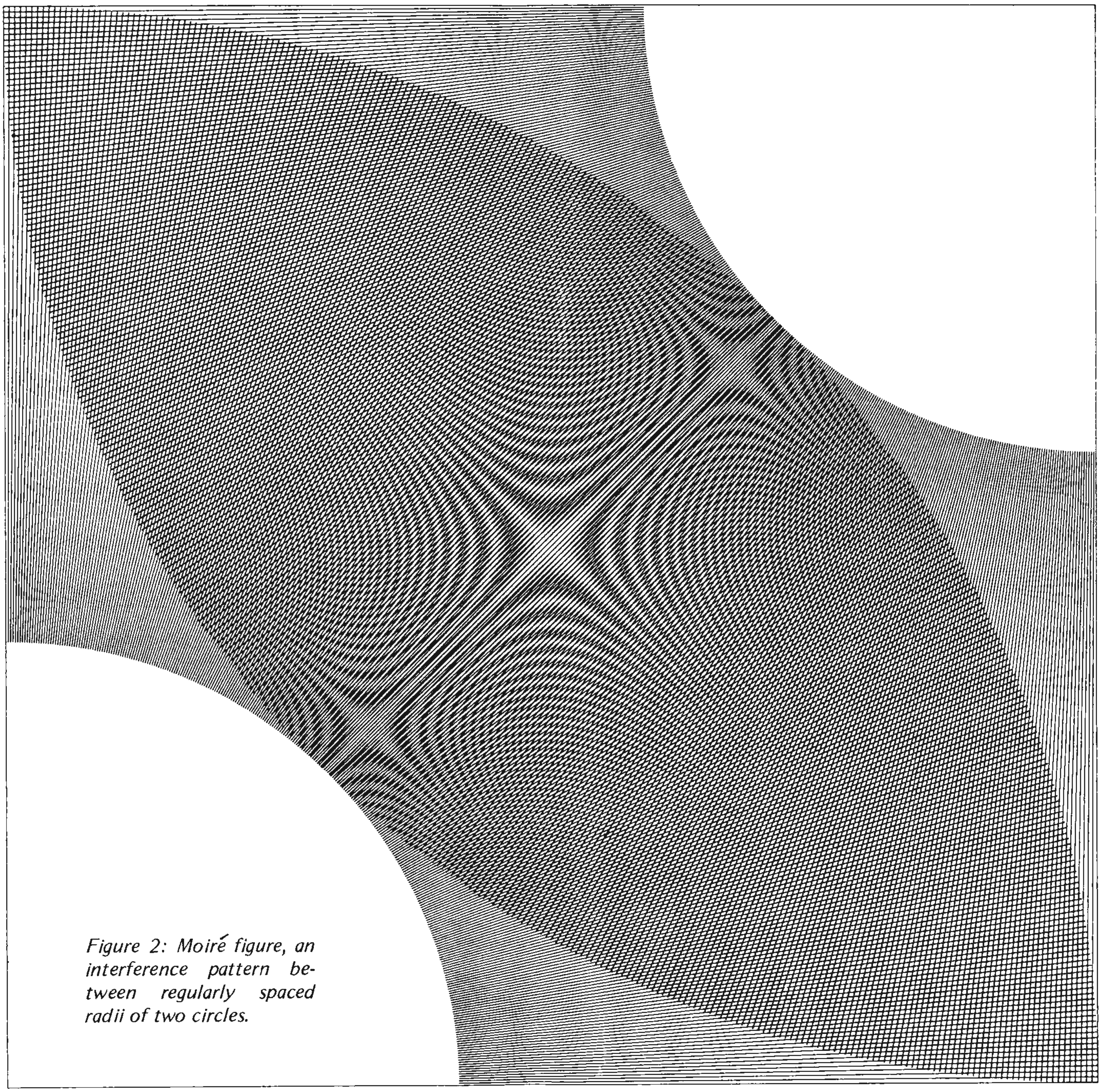


*Figure 2: Moiré figure, an interference pattern between regularly spaced radii of two circles.*

shown in figure 3 (see reference 5). While the beauty of this figure can be appreciated without examining its mathematics, a more complete understanding is necessary in order to reproduce it on a different computer or to fully comprehend the complexity of the figure. One can easily examine figure 3 and determine by its symmetry the decomposition which is shown in figure 4.

The basic unit of figure 3 is shown in figure 5. If the equations which generate the basic unit can be found, then the entire figure can be generated by appropriately manipulating these equations. In an analysis of the unit in figure 5, one can see that the equation of the outer envelope of lines is the only portion of real importance. An examination of this curve brings to mind the spirals studied when one first encounters the use of polar coordinates. There are a number of different kinds of such spirals, most notably the spiral of Archimedes, the parabolic spiral, and the logarithmic spiral. By comparing the graphs of these spirals to figure 5 it can be seen that the logarithmic spiral closely approximates the desired curve. Recall that a logarithmic spiral (shown in figure 6) has an equation of the form $r = ae^{-\theta/b}$, where $a$ and $b$ are positive real numbers. By a suitable choice of the constants $a$ and $b$, along with some transformations applied to the equations of two such spirals, we will be able to obtain the equation of the desired envelope.

To find the equation of this envelope, the graph of the logarithmic spiral must be rotated, translated, and reflected. The fact that the curve is usually expressed in polar form simplifies this task considerably. All three of these transformations can be expressed much more easily in that system than in Cartesian coordinates. Figure 7 shows the resulting graphs and their equations as the graph of the spiral is progressively reflected about the y axis, rotated clockwise by 60°, and translated.

Superimposing the graph of:

$$x = -ae^{-\theta/b}\cos(\theta - \pi/3) + a\cos\pi/3$$

$$y = ae^{-\theta/b}\sin(\theta - \pi/3) + a\sin\pi/3$$

upon the last portion of figure 7, the graph in figure 8 is obtained, which is precisely the desired envelope.

*Text continued on page 110*

**Except where noted Illustrations by:**
**Alexander A Ames**
**Systems Analyst**
**Dept of Defense**
**Washington DC**
**20755**

*Software for most popular 8080/Z80 computer disk systems, including*
**NORTH STAR, MICROPOLIS, iCOM, SD SYSTEMS, DYNABYTE DB8/2, HELIOS, ALTAIR, TRS-80, 8" IBM** *and* **OHIO SCIENTIFIC** *formats.*

Software with Manual / Manual Alone

**GRAHAM-DORIAN SOFTWARE SYSTEMS**

☐ **PAYROLL SYSTEM** — Maintains employee master file. Computes payroll withholding for FICA, Federal and State taxes. Prints payroll register, checks, quarterly reports and W-2 forms. Can generate ad hoc reports and employee form letters with mail labels. Requires CBASIC. Supplied in source code. .......... **$605/$35**

☐ **APARTMENT MANAGEMENT SYSTEM** — Financial management system for receipts, disbursements and security deposits of apartment projects. Captures data on vacancies, revenues, etc. for annual trend analysis. Daily report shows late rents, vacancy notices, vacancies, income lost through vacancies, etc. Requires CBASIC. Supplied in source code. .......... **$605/$35**

☐ **INVENTORY SYSTEM** — Captures stock levels, costs, sources, sales, ages, turnover, markup, etc. Transaction information may be entered for reporting by salesman, type of sale, date of sale, etc. Reports available both for accounting and decision making. Requires CBASIC. Supplied in source code. .......... **$605/$35**

**OTHER**

☐ **Z80 Development Package** — Consists of: (1) disk file line editor, with global inter and intra-line facilities; (2) Z80 relocating assembler, Zilog/Mostek mnemonics, conditional assembly and cross reference table capabilities; (3) linking loader producing absolute Intel hex disk file for CP/M LOAD, DDT or SID facilities. Standard CP/M and TRS-80 CP/M versions available .......... **$95/$15**

☐ **TEXTWRITER II** — Text formatter to justify and paginate letters and other documents. Special features include insertion of text during execution from other disk files or console, permitting recipe documents to be created from linked fragments on other files. Ideal for contracts, manuals, etc. .......... **$75/$5**

☐ **DISINTEL** — Disk based disassembler to Intel 8080 or TDL/Xitan Z80 source code, listing and cross reference files. Intel or TDL/Xitan pseudo ops optional. Runs on 8080. Standard CP/M and TRS-80 CP/M versions available .......... **$65/$10**

☐ **DISZILOG** — As DISINTEL to Zilog/Mostek mnemonic files. Runs on Z80 only. Standard CP/M and TRS-80 CP/M versions available .......... **$65/$10**

Software with Manual / Manual Alone

☐ **WHATSIT?** — Interactive data-base system using associative tags to retrieve information by subject. Hashing and random access used for fast response. Requires CBASIC .......... **$125/$25**

☐ **XYBASIC** Interactive Process Control BASIC — Full disk BASIC features plus unique commands to handle bytes, rotate and shift, and to test and set bits. Available in Integer, Extended and ROMable versions.
Integer Disk or Integer ROMable .......... **$295/$25**
Extended Disk or Extended ROMable .......... **$395/$25**

☐ **SMAL/80** Structured Macro Assembled Language — Package of powerful general purpose text macro processor and SMAL structured language compiler. SMAL is an assembler language with IF-THEN-ELSE, LOOP-REPEAT-WHILE, DO-END, BEGIN-END constructs .......... **$75/$15**

☐ **Selector II** — Data Base Processor to create and maintain single Key data bases. Prints formatted, sorted reports with numerical summaries. Available for Microsoft and CBASIC (state which). Supplied in source code .......... **$195/$20**

☐ **Selector III** — Multi (i.e., up to 24) Key version of Selector II. Comes with applications programs including Sales Activity, Inventory, Payables, Receivables, Check Register, Expenses, Appointments, and Client/Patient. Requires CBASIC Supplied in source code .......... **$295/$20**
Enhanced version for CBASIC-2 .......... **$345/$20**

☐ **CPM/374X Utility Package** — has full range of functions to create or re-name an IBM 3741 volume, display directory information and edit the data set contents. Provides full file transfer facilities between 3741 volume data sets and CP/M files .......... **$195/$10**

☐ **Flippy Disk Kit** — Template and instructions to modify single sided 5¼" diskettes for use of second side in singled sided drives .......... **$9.75**

☐ **BASIC Comparison** — A comprehensive features and performance analysis of five 8080 disk BASIC languages — CBASIC, BASIC-E, XYBASIC, Microsoft Disk Extended BASIC, and Xitan's Disk BASIC. Itemizes results of 21 different benchmark tests for speed and accuracy and lists instructions and features of each BASIC ......(send 20¢ S.A.S.E.) **FREE**

☐ **TRS-80 FORTRAN PACKAGE** — Professional disk-based language and utility package written by Microsoft, creators of Level II BASIC, the package runs on a TRS-80 system with 32K RAM, one or more drives and TRSDOS. The software is supplied on diskettes and consists of a relocatable machine code FORTRAN Compiler, Macro Assembler, a Linking Loader, Subroutine Library, Text Editor .......... **$325**
Macro assembler, loader and editor alone .......... **$165**

# Fill it in.

**Lifeboat Associates,** 2248 Broadway, New York, N.Y. 10024 **(212) 580-0082**

| Software | Price |
|---|---|
| ☐ manual alone | |
| ☐ manual alone | |
| ☐ Check ☐ U.P.S. COD ☐ Visa ☐ Master Charge — Shipping | |
| Account # Exp. Date — $1.00 for C.O.D. | |
| Signature — Total | |

My computer configuration (specifying disk system):

Name

Address (No P.O. Box)

City State Zip

EFFECTIVE APRIL 15, 1979

™ *The Software Supermarket is a trademark of Lifeboat Associates*

All software requires CP/M for operation except TRS-80 FORTRAN

*Disk systems and formats: North Star single or double density, IBM single or 2D/256, Altair, Helios II, Micropolis Mod I or II, 5¼" soft sector (Micro iCOM/SD Sales/Dynabyte), etc.*

*Add $1/item shipping ($2 min.). Add $1 additional for UPS C.O.D.*

*Manual cost applicable against price of subsequent software purchase.*

*The sale of each proprietary software package conveys a license for use on one system only.*


TM

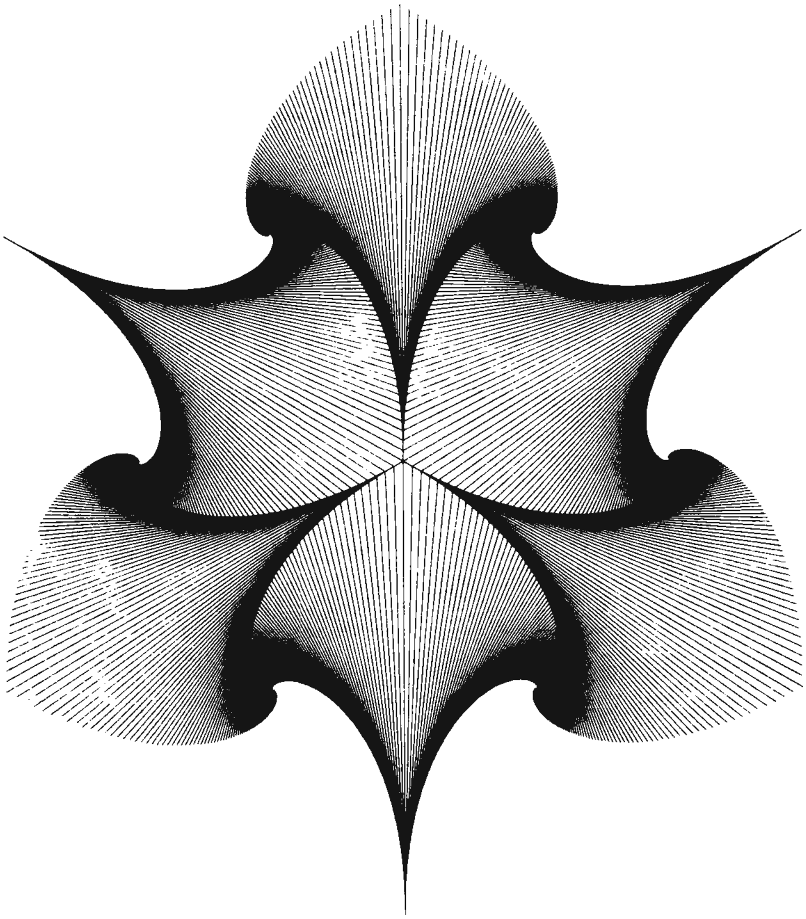


*Figure 3: "Crest." This figure is composed of fundamental units shown in figure 5, and the author's algorithm is explained in the text.*

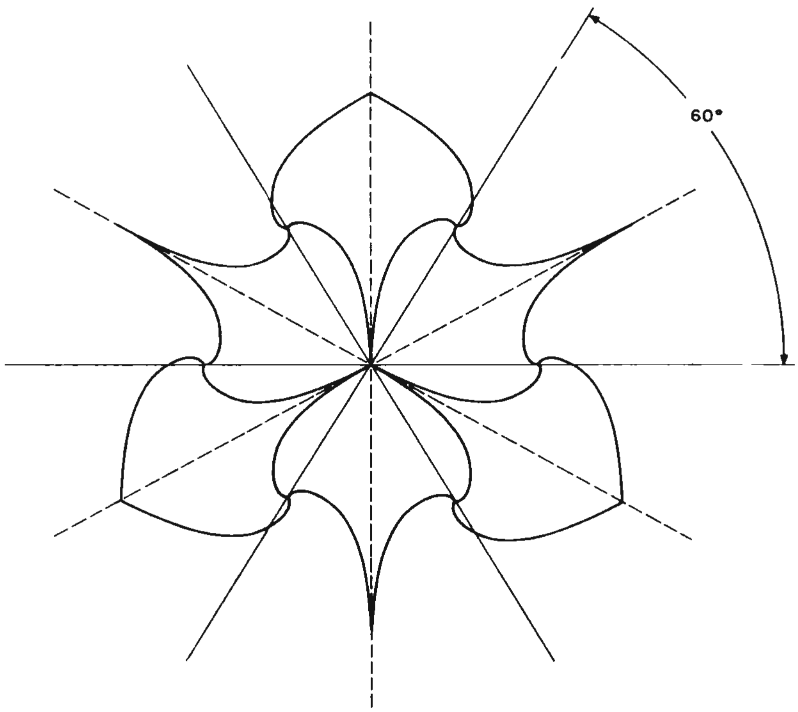



*Figure 4: Decomposition of crest in figure 3.*

*Text continued from page 107:*

By redoing this work in rectangular coordinates, we can see how much easier it is to manipulate these equations in the polar coordinate system. The reflected, rotated, and translated coordinates of a point (X,Y) can be calculated with the matrix equation which is called *a* in table 1.

Substituting the specific values needed to repeat the previous work and multiplying the three 3 by 3 matrices together, we obtain equation *b* in table 1. This is the same result obtained earlier.

It is now a trivial matter to obtain the lines in figure 5 by drawing chords between points selected equiangularly along each of the two curves. One can extend this by similar modifications to the equation $r = ae^{-\theta/b}$ to obtain the crest in figure 3. The constants *a* and *b* determine the size of the resulting plot and the curvature of each of the six "leaves" respectively.

### The Dissected Square

The plot in figure 9 is not too difficult to understand at first glance (see reference 7). In essence it is a set of concentric squares with the area between the squares divided into smaller squares. Postponing the detailed discussion until later, the figure can be constructed in the following manner: given a square with a side of length X, construct a concentric square with a smaller side of length Y. The value of Y is determined by X in a manner to be explained later, but note that Y < X. Extend the sides of the smaller square until they meet the edges of the square of side X. The intermediate result is shown in figure 10. Divide the shaded regions into squares. (It will be shown that this is always possible when X and Y are chosen carefully.) At this point, consider the square of side Y to be the outer square and

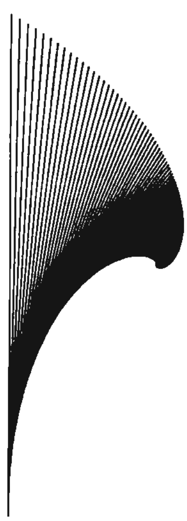


*Figure 5: Fundamental building block of the crest figure. The form of the curves resembles a logarithmic spiral.*

# The Software Company

# Technical Systems Consultants, Inc.

## 6800 & 8080 Text Processing

TSC's Text Editing System and Text Processing packages, available for both 6800 and 8080, can turn your micro into a very powerful word processing system. Two steps are involved. First you edit a file to say what you want without worrying about justification, pagination, etc. The TSC editor is a very powerful line and content oriented editor which may be used for this purpose.

The second step is to "process' the edited file. The TSC text processor reads the edited file and outputs it to a printer with right justification, pagination, page numbering and titling, indenting, margin control, title centering, and much more. Macros may be written in the text processor commands which allow complex operations such as footnotes, form letters, or almost anything imaginable.

Price includes manual with printed source listing and object code on tape or disk.

**6800 Text Editing System**

| | | |
|---|---|---|
| **SL68-24C** | **w/cassette** | **$30.45** |
| **SL68-24D** | **w/mini FLEX™ disk** | **$31.50** |
| **SL68-24F** | **w/8" FLEX™ 1.0 disk** | **$50.00** |

**8080 Text Editing System**

| | | |
|---|---|---|
| **SL80-10P** | **w/paper tape** | **$37.50** |
| **SL80-10F** | **w/8" CP/M™ disk** | **$40.00** |
| **SL80-10D** | **w/5" HS CP/M™ disk** | **$40.00** |

**6800 Text Processing System**

| | | |
|---|---|---|
| **SL68-29C** | **w/cassette** | **$38.95** |
| **SL68-29D** | **w/mini FLEX™ disk** | **$40.00** |
| **SL68-29F** | **w/8" FLEX™ 1.0 disk** | **$75.00** |
| **SL68-29F2** | **w/5" FLEX™ 2.0 disk** | **$65.00** |

**8080 Text Processing System**

| | | |
|---|---|---|
| **SL80-11P** | **w/paper tape** | **$41.00** |
| **SL80-11F** | **w/8" CP/M™ disk** | **$50.00** |
| **SL80-11D** | **w/5" HS CP/M™ disk** | **$50.00** |

## FLEX™ Software

The FLEX™ disk operating system is rapidly becoming the standard of 6800 disk users. It's powerful capabilities and ease of use make it a natural choice for any application from the most minimal hobbyist system to the most complete industrial development system.

**FLEX™ for SWTPC and SSB**

SWTPc mini floppy users can now upgrade to FLEX™ 2.0, the same version of FLEX™ as found on the SWTPc 8" disk system. Simply add 8K of memory at $A000 and boot up from a FLEX™ 2.0 disk. Also available for Smoke Signal disk users in a mini floppy version (FLEX™ 2.0) and a full sized version (FLEX™ 1.0). Includes TSC editor and assembler on disk. All manuals also supplied (no source listings).

| | | |
|---|---|---|
| **FX02-SWT** | **FLEX™ 2.0 for 5" SWTPc** | **$75.00** |
| **FX02-SSB** | **FLEX™ 2.0 for 5" SSB** | **$90.00** |
| **FX01-SSB** | **FLEX™ 1.0 for 8" SSB** | **$100.00** |

**Sort/Merge Package**

Sorts any type file according to parameters you supply. Files too large to fit in memory are automatically split up, individually sorted, and merged into one output file. Accepts fixed or variable length records, up to 20 ascending or descending keys, non-ASCII sequence sorts, and much more. No source listing. Specify mini FLEX™, FLEX™ 1.0, or FLEX™ 2.0

| | | |
|---|---|---|
| **AP68-10** | **6800 Sort/Merge** | **$75.00** |
| | **Manual Only** | **$15.00** |

**FLEX™ Utilities**

Complete set of 36 additional utilities ranging from memory dump to disk diagnostics. Includes printed manuals and source listings with object code AND source listing on disk.

| | | |
|---|---|---|
| **UV1-6D** | **w/mini FLEX™ disk** | **$99.95** |
| **UV1-6F** | **w/8" FLEX™ 1.0 disk** | **$124.95** |
| **UV1-6F2** | **w/5" FLEX™ 2.0 disk** | **$119.95** |

## TSC BASIC for 6800

The fastest floating point BASIC interpreter for any 8 bit microprocessor. Features include six digit floating point math, full transcendental functions, unlimited string length, if/then/else construct, logical operators, and two-dimensional arrays including string arrays.

FLEX™ disk versions support random as well as sequential access files (the mini FLEX™ version has only sequential). Random files accessed by true record I/O or by virtual arrays. Disk version also allows TRACE, a renumber facility, ON ERROR GOTO, PTR for obtaining the address of a variable, CHAIN, and a very powerful method of printing to any type and number of output devices. SAVE and LOAD commands produce FLEX™ compatible files which can be externally edited while a COMPILE command places an unreadable, compressed form of the source on disk which can only be executed by RUN. This feature permits the distribution of proprietary BASIC programs.

Cassette version runs in a minimum of 12K, disk version should have at least 16K. No source listing is included.

| | | |
|---|---|---|
| **AP68-11C** | **w/casette** | **$39.95** |
| **AP68-11D** | **w/mini FLEX™ disk** | **$49.95** |
| **AP68-11F** | **w/8" FLEX™ 1.0 disk** | **$59.95** |
| **AP68-11F2** | **w/5" FLEX™ 2.0 disk** | **$54.95** |

All orders should include 3% for postage and handling (8% on foreign orders). Send 25¢ for a complete software catalog.

## Technical Systems Consultants, Inc.

Box 2574 W. Lafayette, IN 47906

*CP/M™ is a registered trademark of Digital Research
FLEX™ is a trademark of Technical Systems Consultants, Inc

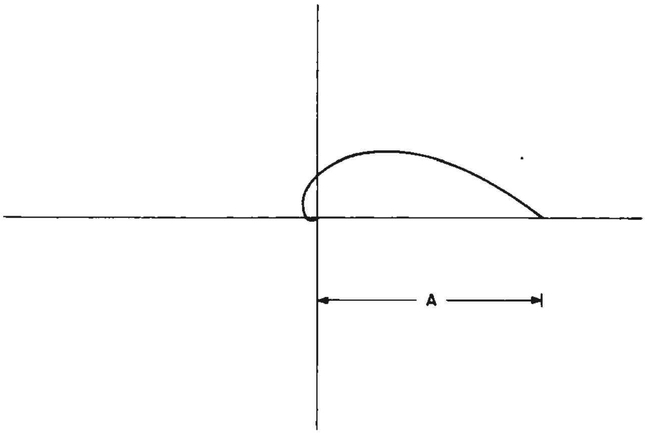



*Figure 6: Logarithmic spiral with equation of form $r = ae^{-\theta/b}$ using polar coordinates.*

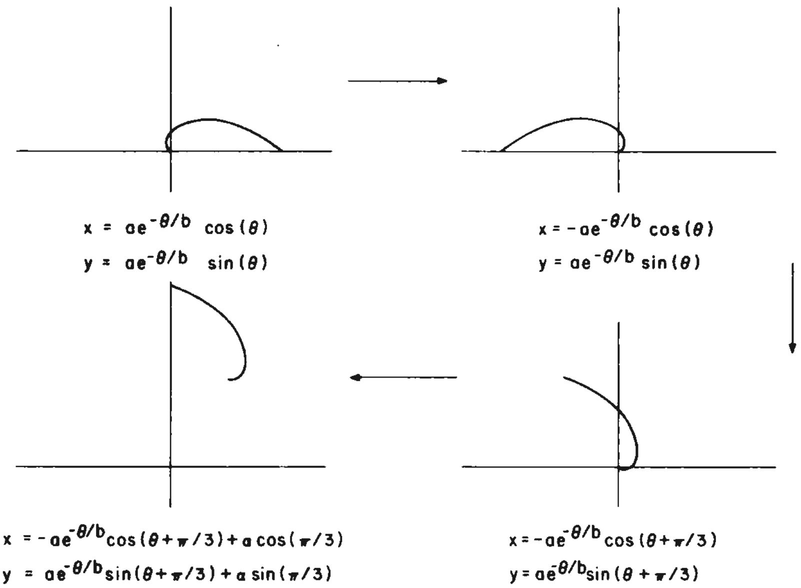



*Figure 7: Graphs and equations of a logarithmic spiral as it is reflected about the y axis, rotated clockwise by 60°, and translated.*

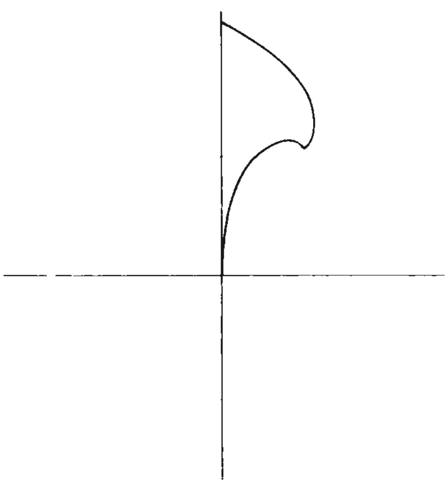


*Figure 8: Superimposition of graph of: $x = -ae^{-\theta/b}\cos(\theta - \pi/3) + a\cos\pi/3$; and $y = ae^{-\theta/b}\sin(\theta - \pi/3) + a\sin\pi/3$; which yields the desired envelope shape.*

begin again by choosing a suitable Y′ where Y′ < Y. This process is terminated when Y′ assumes a certain specified value. What is significant about this plot, however, is the mathematics that it represents. This figure proves the following theorem:

$$\sum_{i=1}^{n} i^3 = \left( \sum_{i=1}^{n} i \right)^2$$

for all positive integers, $n$ (an offshoot of the theorem of Nicomachus) for the case $n = 26$.

To see this, it is easier to examine the associated figure for a smaller $n$ than 26, say $n = 6$ (see figure 11). If the smallest squares in the center of the figure are taken as unit squares, then the area of the large square can be calculated in two different ways. In the first way, the lengths of two sides can be multiplied. Since we are dealing with squares, any two sides can be used. The left side is of length 6(6 + 1) or in general $n(n + 1)$, as can be seen by considering the shaded squares which lie along the left side. The length of the opposite side can be calculated by considering the shaded squares which extend diagonally from the center to the right side to obtain:

$$b = 2(6 + 5 + 4 + 3 + 2 + 1)$$

or in general:

$$b = 2 \sum_{i=1}^{n} i$$

Therefore the area of the square is:

$$ab = 6(6 + 1) \times 2(6 + 5 + 4 + 3 + 2 + 1)$$

or in general:

$$ab = n(n + 1) \times 2 \sum_{i=1}^{n} i = 4 \left( \sum_{i=1}^{n} i \right)^2$$

However, the area of the square can also be calculated by summing the areas of all the component squares. There are four squares of area 1, eight squares of area 4, twelve of area 9, etc. Therefore the area of the large square is:

$$4\times1\times1^2 + 4\times2\times2^2 + 4\times3\times3^2 + 4\times4\times4^2 + 4\times5\times5^2 + 4\times6\times6^2$$

or in general:

$$4 \sum_{i=1}^{n} i^3$$

By equating these computations of area, the desired theorem is obtained.

In drawing this figure, one need only choose an X of the form $n(n+1)$ for the side of the largest square, where $n$ is an integer greater than 3. The sides of the inner squares

$$a \quad \underbrace{[x' \; y' \; 1]}_{\text{new point}} = \underbrace{[x \; y \; 1]}_{\text{old point}} \underbrace{\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{reflection}} \underbrace{\begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{rotation by } \gamma} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}}_{\text{translation}}$$

$$b \quad [x' \; y' \; 1] = [r\cos\theta \;\; r\sin\theta \;\; 1] \begin{bmatrix} -\cos\pi/3 & \sin\pi/3 & 0 \\ \sin\pi/3 & \cos\pi/3 & 0 \\ a\cos\pi/3 & a\sin\pi/3 & 1 \end{bmatrix}$$

$$= [-r\cos(\theta+\pi/3) + a\cos\pi/3 \quad r\sin(\theta+\pi/3) + a\sin\pi/3 \quad 1]$$

$$= [-ae^{-\theta/b}\cos(\theta+\pi/3) + a\cos\pi/3 \quad ae^{-\theta/b}\sin(\theta+\pi/3) + a\sin\pi/3 \quad 1]$$

*Table 1: Matrix equation a reflects, rotates, and translates coordinates of a point (X, Y). Matrix equation b has substituted in it the specific values needed to repeat the earlier equations. We obtain the same result.*

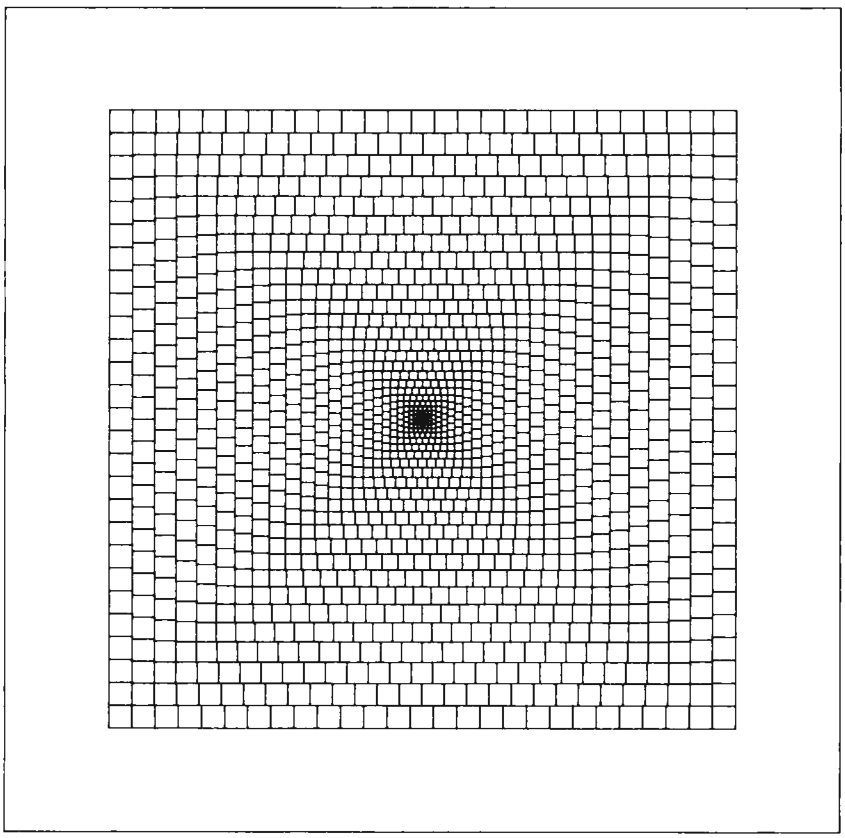


*Figure 9: Dissected square, a set of concentric squares with the area between the squares divided into smaller squares.*

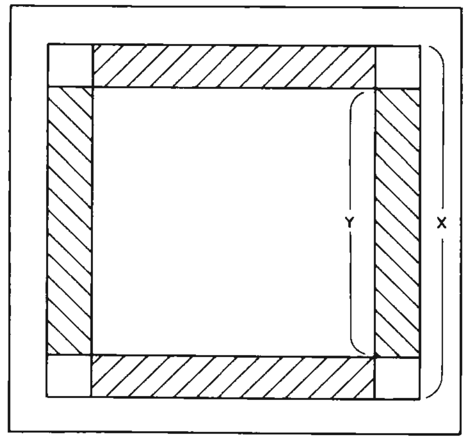



*Figure 10: A square with sides of length X has constructed within it a concentric square with sides of length Y. The sides of the smaller square are extended until they meet the edges of the square of side X. The shaded regions are next divided into squares.*

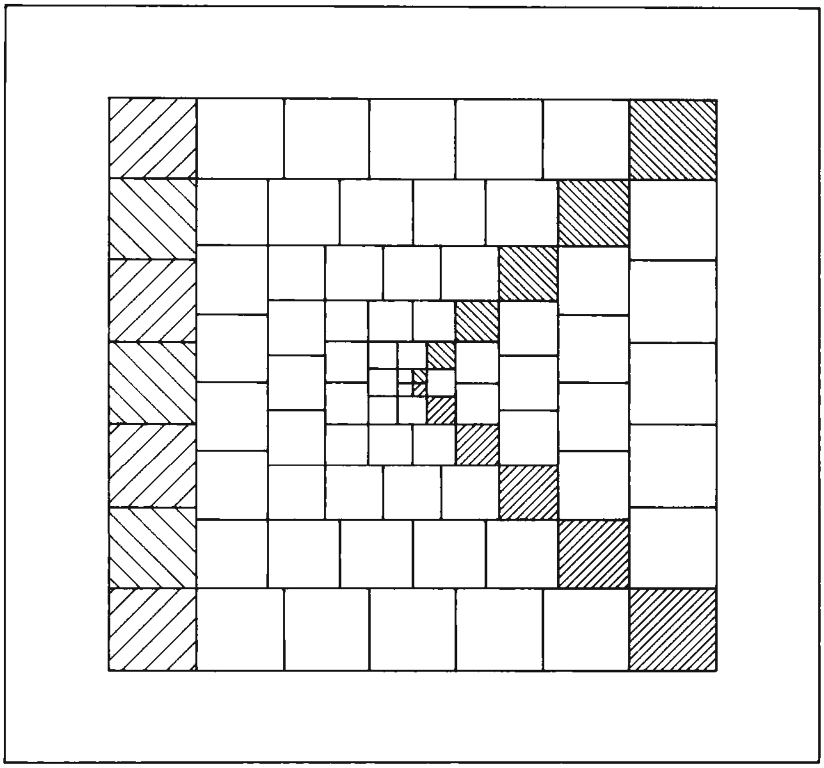


*Figure 11: Dissected square for* n = *6. The left side is of length 6(6+1).*

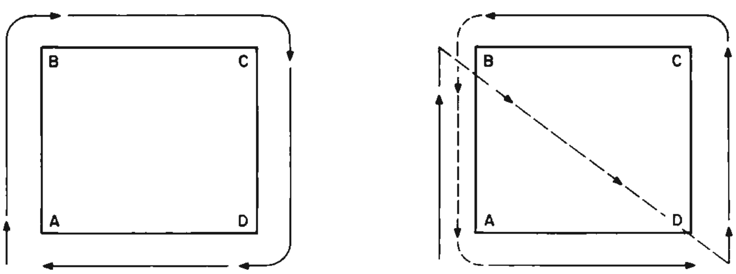



*Figure 12: Two plotting procedures for a square with vertices ABCD. The smallest amount of pen motion occurs at left when plotting begins at point A with consecutive drawing movements to B, C, D, and then back to A. In this method the length of nondrawing moves is 0.*

*At right is seen a nonoptimal plotting scheme. Starting at A, the pen draws to B, a nondrawing move is made to point D, the pen draws from D to C and then to B, a move is made to A, and then pen draws from A to D.*

are also numbers of this form, obtained by decrementing $n$ by 1 for each successive new square. When $Y' = 2$, the last two lines are drawn, completing the figure. This choice of X and Y always allows the shaded areas of figure 10 to be decomposed into squares, as they are all rectangles with one side of length $n$ and the other of $n(n-1)$. A rectangle with these proportions is dissectible into $(n-1)$ squares of side $n$.

**Plotting Considerations and Implementation**

In developing the software to produce these drawings, the logic used to understand the generation of the figures was extended into the implementation of the code. Although this solution to the problem works, it turns out to be grossly inefficient in construction and plotting time.

These figures are best plotted on a high speed incremental plotter using ink rather than a ballpoint pen. The use of ink in plotting immediately causes a 50% reduction in plotting speed in order to avoid smears on the final plot. This and the high density of lines required to produce an aesthetically pleasing picture resulted in an average plot time of two hours per figure.

With these two considerations in mind, it became desirable to optimize the required plot time by minimizing pen movement. In the plotting of figures like those above, the total pen movement is comprised of the movement used to reposition the pen prior to the drawing of a new line (ie: when the pen tip is in the up position) and the actual drawing of the line (ie: when the pen tip is in the down position, that is, is in contact with the plotting surface and is drawing). While the total length of the "draws" (ie: when the pen tip is down and drawing) is fixed for any given figure, the length of the "moves" is variable. The total plot time can be diminished by minimizing these moves.

Consider the plotting of a square whose vertices are ABCD (see figure 12). Let us

# We have what you want... when you want it.

You want speed? Except for the Heath H8 board, *every* Econoram* runs trouble-free with 4 MHz systems. You want low power? We use low power static parts exclusively, which means our *maximum* guaranteed current specs are often lower than the *typical* current specs of our competitors. You want low cost? Our volume purchasing and technical know-how keeps our products price-competitive with boards that do a whole lot less. You want it now? Not only do we stock what we advertise, but many computer stores both nationwide *and* overseas stock much or all of the CompuKit™ line.

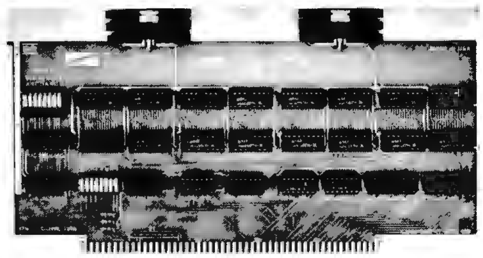
You want a choice of configurations? For lowest price buy an "unkit", with sockets and bypass caps pre-soldered in place for easy assembly; specify assembled and tested boards when you can't wait to get up and running; and if you're dealing with critical applications—like accounting or medical applications—check into the boards qualified under our high-reliability Certified System Component(CSC) program. These boards are burned in for 200 hours minimum and are immediately replaced in the event of failure within 1 year of invoice date.

You want reliable, cost-effective operation? That's how we got our reputation...ask any Econoram owner.

## FEATURED BOARD OF THE MONTH:

## ECONORAM IIA $149 UNKIT, $164 ASSM

The 8K X 8 **Econoram II** has been the workhorse of many an S-100 computer system...and now it's improved, with features like new 4K static chips, guaranteed 4 MHz operation, simpler layout, and low power (draws 8W *maximum* from the 8V buss —and that's a guaranteed spec). We've retained all the popular features of the original, such as deselect switch for 4K operation, dual 4K block configuration with independent addressing, and switchable memory protect for each block. **The oldest board on the block is now one of the newest, and we think you're really going to like it.**



## OTHER ECONORAM* FULLY *STATIC* MEMORY BOARDS:

| Name | Storage | Buss | Speed | Configuration | Unkit | Assm | CSC |
|---|---|---|---|---|---|---|---|
| Econoram IV | 16K X 8 | S-100 | 4 MHz | 1-16K block | $295 | $329 | $429 |
| Econoram VI | 12K X 8 | H8 | 2 MHz | 1-8K, 1-4K | $200 | $270 | n/a |
| Econoram VII | 24K X 8 | S-100 | 4 MHz | 2-4K, 2-8K | $445 | $485 | $605 |
| Econoram IX | 32K X 8 | Dig Grp | 4 MHz | 2-4K, 1-8K, 1-16K | $649 | n/a | n/a |
| Econoram X | 32K X 8 | S-100 | 4 MHz | 2-8K, 1-16K | $599 | $649 | $789 |
| Econoram XI | 32K X 8 | SBC | 4 MHz | 2-8K, 1-16K | n/a | n/a | $1050 |

## BANK SELECT MEMORIES (for Alpha Micro Systems, Marinchip, etc.)

| Name | Storage | Buss | Speed | Configuration | Unkit | Assm | CSC |
|---|---|---|---|---|---|---|---|
| Econoram XII-16 | 16K X 8 | S-100 | 4 MHz | 2 indep. banks** | $369 | $419 | $519 |
| Econoram XII-24 | 24K X 8 | S-100 | 4 MHz | 2 indep. banks** | $479 | $539 | $649 |
| Econoram XIII | 32K X 8 | S-100 | 4 MHz | 2 indep. banks** | $629 | $699 | $849 |

***Econoram** is a trademark of **Bill Godbout Electronics.**

**Econoram XII-16 and -24 have 2 independent banks addressable on 8K boundaries; Econoram XIII has 2 independent banks addressable on 16K boundaries.

## OTHER PRODUCTS:

### NEW! 2708 EROM BOARD KIT... at a special introductory price.

4 independently addressable 4K blocks, with selective disable for each block. Built to Econoram standards (dipswitch addressing, top quality board, sockets wave-soldered in place), and includes dipswitch selectable jump start built right into the board. Includes all support chips and manual, but does *not* include EROMs. Special introductory price through August 1, 1979: **$69.95**. After that, the price goes up to the normal $85... don't say we didn't warn you.

### POPULAR ICs AT POPULAR PRICES

Low power **2102s** for 2 MHz systems on special: **10/$9.90. 1791 MOS LSI** dual density disc controller from Western Digital: **$59** with pinout and data. **1771** single density controller: **$22.50**. All parts are offered on a while-they-last basis.

### NEW MULLEN COMPUTER PRODUCTS: H8 EXTENDER BOARD KIT $39

We're happy to distribute a kit that really takes the hassle out of troubleshooting or testing the popular H8 computer. Includes jumper links in the power supply lines for insertion of fuses, Ammeters, current limiters, etc.

### ACTIVE TERMINATOR KIT $34.50

As written up in the April '79 issue of ***Kilobaud Microcomputing.*** Our much imitated design plugs into any S-100 motherboard to reduce ringing, crosstalk, noise, and other buss-related problems.

### S-100 MOTHERBOARDS

**11 slot unkit: $90. 18 slot unkit: $124.** Each motherboard includes all edge connectors wave-soldered in place for easy assembly, integral active termination circuitry, extra wide power and ground traces, and much more.

**TERMS:** Allow 5% shipping, excess refunded. Cal res add tax. VISA®/Mastercharge® call our 24 hour order desk at **(415) 562-0636**. COD OK with street address for UPS (COD charge applies). Prices good through cover month of magazine. Orders under $15 add $1 handling.

CompuPro™ from GODBOUT ELECTRONICS

**FREE FLYER: We'll be glad to tell you more than the space of this ad permits.** Just send your name and address, we'll take care of the rest. If you're in a hurry, enclose 41¢ in stamps for 1st class delivery.

Circle 150 on inquiry card.

assume that vertex A is the origin of the plot. Clearly, the smallest amount of pen movement possible is $4s$, where $s$ is the length of the side of the square. The value of $4s$ is obtained when the plotting begins at point A (ie: the origin) with consecutive draws to B, C, D, and then a final draw to A. In this case the length of the moves is 0. A nonoptimal plotting scheme for this figure would be to start at A, and then draw to B, move to D and draw to C and then to B, move to A and then draw to D. The total pen movement for this scheme is $5s + \sqrt{2}s$, where again $s$ is the length of the square. It should be clear that there is no upper limit on the total pen movement, as the moves have no effect on the resulting plot and can be increased without bound.

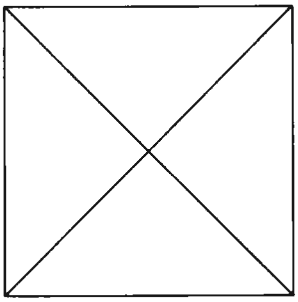


*Figure 13: A simple plot for which the best possible plotting scheme includes some nondrawing pen moves.*

Unfortunately, it is not always possible to find a plotting scheme in which the length of the pen moves is 0. A simple plot for which the best possible plotting scheme includes some moves is shown in figure 13. If $s$ is the side of the square, the best possible plotting scheme has a total pen movement of $5s + 2\sqrt{2}s$ (see reference 10).

Of the two figures discussed in detail, the crest and the dissected square, only the crest can be drawn with zero moves. The plotting scheme which obtains this optimal solution is shown in figure 14. Using this strategy resulted in a substantial savings in total plotting time.

Unfortunately, no plotting scheme for the dissected square which has zero moves is possible. In fact, no scheme was found which significantly reduced the total plot time from that obtained by using the notions explained in detail above. It is felt that this is because all the plotting schemes we tried involved decomposing long line segments into a number of smaller such segments which were not drawn consecutively. With an on line incremental plotter this requires the processor controlling the pen to issue a much larger number of plot commands. In a multiprocessing environment, any advantage gained in the total length of the moves was completely eliminated by the increased processing time with its associated overhead.■

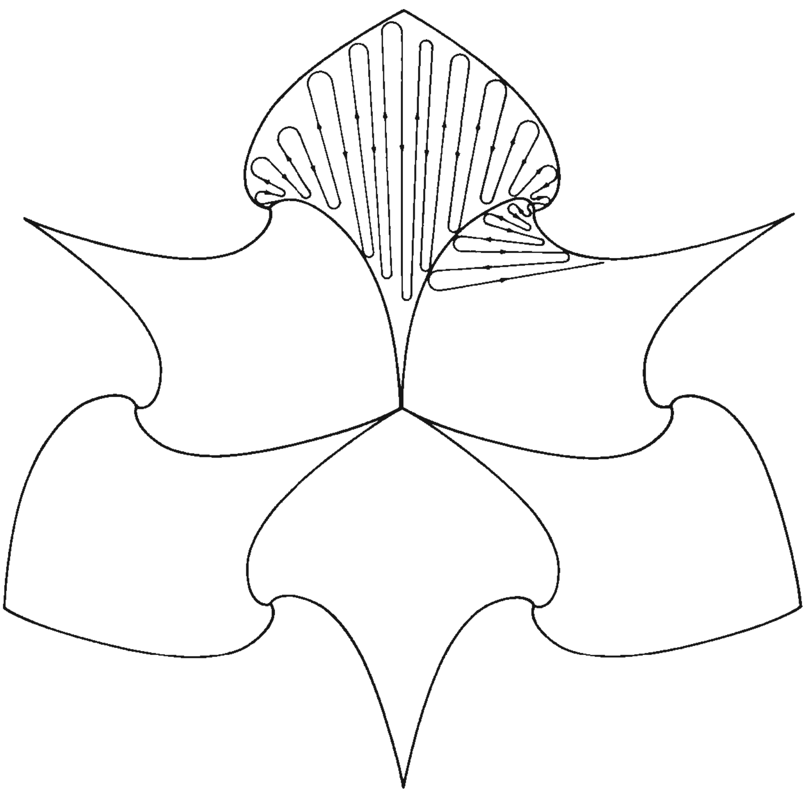


*Figure 14: Optimal plotting scheme for drawing the crest with no wasted pen movement.*

## REFERENCES


1. "Christmas Wreath," Entry in the 6th Annual Computer Art Contest, *Computers and Automation,* volume 17, number 8, August 1968, pages 8 thru 27.
2. "Circus," First prize winner in the 8th Annual Computer Art Contest, *Computers and Automation,* volume 18, number 9, August 1969, pages 12 thru 32.
3. Csuri, Charles, "Computer Graphics and Art," *Proceedings of the IEEE,* volume 62, number 4, April 1974, pages 503 thru 515.
4. Franke, Herbert W, *Computer Graphics — Computer Art,* Phaidon, 1972.
5. Franke, Herbert W, *op cit,* page 18.
6. Franke, Herbert W, *op cit,* pages 20 thru 22.
7. Gardner, Martin, *Scientific American,* volume 229, number 10, October 1973, pages 114 thru 118.
8. Hertlein, Grace C, "Computer Art for Computer People — A Syllabus," *SIGGRAPH '77 Proceedings, Computer Graphics,* volume 11, number 2, Summer 1977, pages 249 thru 254.
9. Ives, Roger, "Computer-Aided Sculpture," *Computers and Automation,* volume 18, number 9, August 1969, page 33.
10. Lewis, Harry R, and Papadimitriou, Christos H, "The Efficiency of Algorithms," *Scientific American,* volume 238, number 1, January 1978, pages 96 thru 109.
11. Mueller, Robert E, "Idols of Computer Art," *Art in America,* May-June 1972; reprinted in *Creative Computing,* May-June 1978, pages 100 thru 106.
12. Noll, A Michael, "Computer Animation and the Fourth Dimension," *AFIPS Fall Joint Computer Conference 1968,* pages 1279 thru 1283.
13. Oster, Gerald, and Nishijima, Yasunori, "Moiré Patterns," *Scientific American,* volume 208, number 5, May 1963, pages 54 thru 63.
14. Prueitt, Melvin L, *Computer Graphics,* Dover Publications Inc, New York, 1975.
15. "Sine Curve Man," First prize winner in the 5th Annual Computer Art Contest, *Computers and Automation,* volume 16, number 8, August 1967, pages 8 thru 21.

# ANOTHER FIRST FROM MOUNTAIN HARDWARE. SUPERTALKER.

**[illegible] YOUR APPLE.**
SuperTalker allows you to add the dimension of human speech output in your computer programs. Add voice to games. Program verbal prompting for the operator of your business system. Use verbal warnings under program control as an enunciator in commercial security or control rooms. Create educational programs that verbally coach the student.

**THE SUPERTALKER SYSTEM.**
SuperTalker is a new Mountain Hardware peripheral system which allows the Apple II computer to output exceptionally high quality human speech through a loudspeaker under program control. Output may also be directed through any P.A. or stereo system. Initially, spoken words are digitized into RAM memory through the system microphone. Speech data in RAM may then be manipulated like any other stored data.

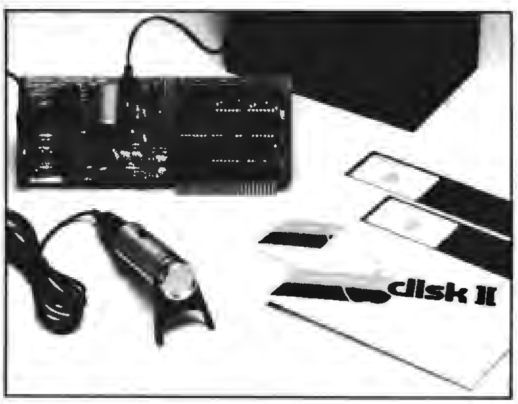
**A COMPLETE PACKAGE.**
The SuperTalker peripheral system consists of: The SuperTalker peripheral card which plugs into [illegible] a loudspeaker; easy-to-use operating software and documentation; plus, two ready-to-run SuperTalker programs.

**OPERATING SYSTEMS.**
In order to achieve maximum utility using SuperTalker, the SuperTalker Disk Operating System permits output of human speech under program control with direct I/O routines. It also provides a preparation program which permits the creation of voice files on diskette. BASIC program routines are provided which require only one-line statements to output a word or phrase. Routines also support cassette storage.

## FOR YOUR APPLE II

**TEACH YOUR COMPUTER TO TALK.**
For $279 assembled and tested, SuperTalker gives your Apple II a voice in the matter.

**AVAILABLE NOW.**
Mountain Hardware's SuperTalker, Apple Clock and 100,000 Day Clock™ (for S-100 bus computers) are available through computer dealers worldwide.




**Mountain Hardware, Inc.**
**LEADERSHIP IN COMPUTER PERIPHERALS**
300 Harvey West Blvd., Santa Cruz, CA 95060
(408) 429-8600

Sounds super.

☐ Send me everything I need to know about SuperTalker.
☐ Also information on your Real-Time clocks for Apple II and S-100.

Name ____________________

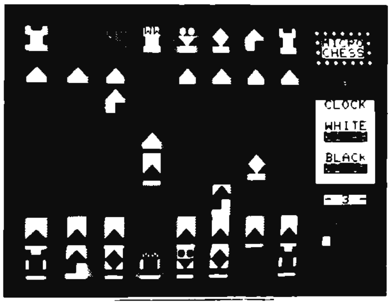
Address ____________________

City ____________ State ________ Zip ________

Apple II is a trademark of Apple Computer, Inc.

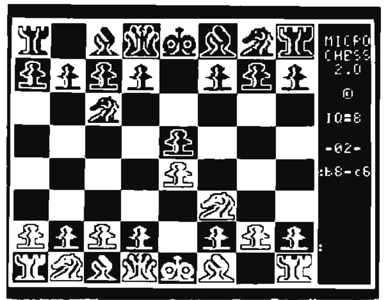
Circle 257 on inquiry card.

# PET / TRS-80 / APPLE: Personal Software brings you the finest!

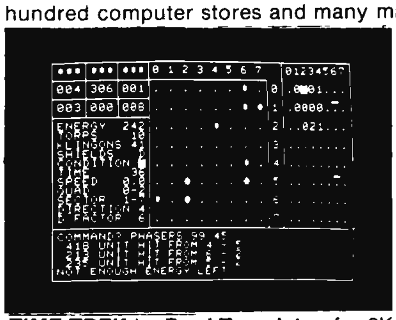



## MICRO CHESS

### The Industry's First Gold Cassette Over 50,000 Sold

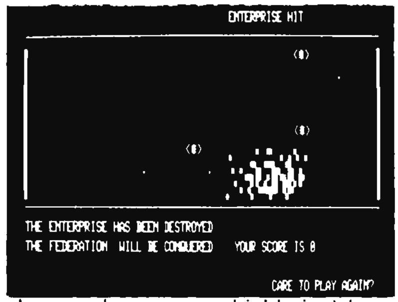



**MICROCHESS** is the industry's best selling computer game. And no wonder—because MICROCHESS gives you more than just a chessplaying program: A convenient, foolproof set of commands and error checks ... complete instructions in a 5½" by 8½" booklet ... a cassette that's guaranteed to load, with disk versions coming soon ... and several levels of difficulty to challenge you not just once, but time after time. It's available through well over three hundred computer stores and many mail order sources ... always originating from Personal Software. What's more, *every* Personal Software product is selected to give you these same benefits of easy availability, reliable cassettes, readable documentation, a carefully thought out user interface ... and most important, continuing challenge and enjoyment, not just once but time after time. If you haven't already, order your own gold cassette: MICROCHESS, by **Peter Jennings**, for 8K **PET**s, 16K **APPLE**s, and 4K Level I and II **TRS-80**s ............................ **$19.95**

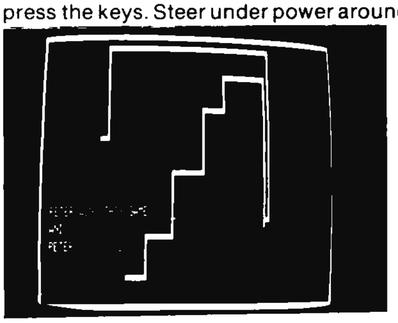


## TIME TREK

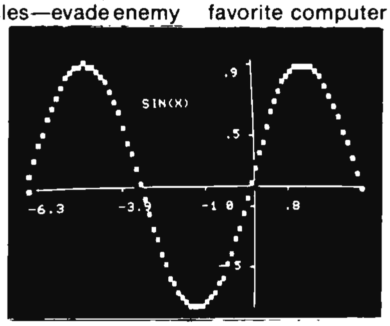
### A Tour De Force In Real Time Action Strategy Games



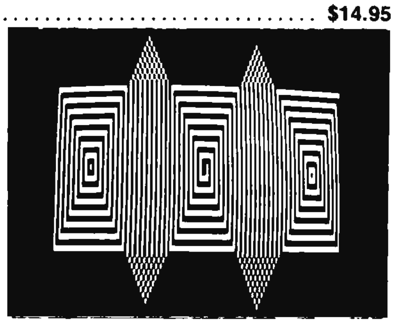

**TIME TREK** by **Brad Templeton** for 8K **PET**s and **Joshua Lavinsky** for 4K Level I and II **TRS-80**s adds a dramatic new dimension to the classic Star Trek type strategy game: REAL TIME ACTION! You'll need fast reflexes as well as sharp wits to win in this constantly changing game. Be prepared—the Klingons will fire at you as you move, and will move themselves at the same time, even from quadrant to quadrant—but with practice you can change course and speed, aim and fire in one smooth motion, as fast as you can press the keys. Steer under power around obstacles—evade enemy shots as they come towards you—lower your shields just long enough to fire your phasers, betting that you can get them back up in time! With nine levels of difficulty, this challenging game is easy to learn, yet takes most users months of play to master. ADD SOUND EFFECTS with a simple two-wire hookup to any audio amplifier; the TRS-80 also produces sound effects directly through the keyboard case, to accompany spectacular graphics explosions! You won't want to miss this memorable version of a favorite computer game ............................ **$14.95**



**BLOCKADE** by **Ken Anderson** for 4K Level I and II **TRS-80**s is a real time action game for two players, with high speed graphics in machine language. Each player uses four keys to control the direction of a moving wall. Try to force your opponent into a collision without running into a wall yourself! A strategy game at lower speeds, BLOCKADE turns into a tense game of reflexes and coordination at faster rates. Play on a flat or spherical course at any of ten different speeds. You can hear SOUND EFFECTS through a nearby AM radio—expect some razzing if you lose! ............ **14.95**




**GRAPHICS PACKAGE** by **Dan Fylstra** for 8K **PET**s includes programs for the most common 'practical' graphics applications: PLOTTER graphs both functions and data to a resolution of 80 by 50 points, with automatic scaling and labeling of the axes; BARPLOT produces horizontal and vertical, segmented and labeled bar graphs; LETTER displays messages in large block letters, using any alphanumeric or special character on the PET keyboard; and DOODLER can be used to create arbitrary screen patterns and save them on cassette or in a BASIC program .................... **$14.95**



**ELECTRIC PAINTBRUSH** by **Ken Anderson** for 4K Level I and II **TRS-80**s: Create dazzling real time graphics displays at speeds far beyond BASIC, by writing 'programs' consisting of simple graphics commands for a machine language interpreter. Commands let you draw lines, turn corners, change white to black, repeat previous steps, or call other programs. The ELECTRIC PAINTBRUSH manual shows you how to create a variety of fascinating artistic patterns including the one pictured. Show your friends some special effects they've never seen on a TV screen! ......... **$14.95**

**WHERE TO GET IT:** *Look for the **Personal Software™** display rack at your local computer store. If you can't find the product you want, you can order direct with your VISA/Master Charge card by dialing **1-800-325-6400** toll free (24 hours, 7 days; in Missouri, dial 1-800-3426600). If you have questions, please call **617-783-0694**. Or you can mail your order to one of the addresses below, as of the dates shown.*

Until July 1: P.O. Box 136
Cambridge, Mass. 02138

**Personal Software™**

After July 1: 592 Weddell Dr.
Sunnyvale, Calif. 94086



Circle 302 on inquiry card.

# Sonic Anemometry for the Hobbyist

**Neil Dvorak**
**2562 S Newport**
**Denver CO 80224**

Meteorological measurement generally concerns itself with five variables: air pressure, humidity, temperature, wind speed, and wind direction. A single *sonic anemometer* can sense each of the last three variables. Accuracy and linearity are excellent. Additionally, the actual air temperature is detected in a manner which is insensitive to solar radiation, which can easily heat up conventional thermometers.

In a sonic anemometer, wind vane and rotating cups are replaced with transducers which measure the speed of sound as a function of wind velocity and temperature. Commercially available research grade instruments cost upwards of $10,000, and until recently employed analog computational circuitry.

The arrival of the microcomputer and its associated display techniques makes such a scientific instrument economically feasible as an experimenter's project. As a bonus, data can be logged into memory over time, averaged, and displayed as desired. A tantalizing option involves the attachment of a fast but inexpensive 4 bit analog to digital converter which enables the instrument to double as an ultrasonic echo radar device. At this time, however, such investigations have progressed only to echoing observations on the time base of a triggered oscilloscope.

In operation the instrument uses a pair of pulse travel times in the North-South direction and a corresponding pair for the East-West direction. These vector components are easily processed into a resultant wind vector with magnitude and direction. Physically, two sets of ultrasonic transducers face each other at opposite ends of a path. Simultaneous sound fronts and eventual reception yield two travel times whose difference is a measure of wind speed along the path.

## Fundamental Relationships

The following derivation yields wind speed:

$$\Delta t = t_2 - t_1 = \frac{D}{C-W} - \frac{D}{C+W}$$

$$= \frac{2DW}{C^2-W^2} \approx \frac{2DW}{C^2} \quad . \tag{1}$$

$$\text{Therefore } W = \left(\frac{C^2}{2D}\right)\left(\Delta t\right) \quad , \tag{2}$$

where C = speed of sound
D = path length
W = wind speed
t = difference of travel times.

The resultant wind speed, $W_r$, being the sum of two orthogonal vectors, is simply expressed as:

$$W_r = \sqrt{W_{NS}^2 + W_{EW}^2} \quad . \tag{3}$$

Temperature is found by adding a pair of travel times:

$$t_1 + t_2 = \frac{D}{C+W} + \frac{D}{C-W} = \frac{2DC}{C^2-W^2}$$

$$\approx \frac{2DC}{C^2} = \frac{2D}{C} \quad . \tag{4}$$

If $C = 20\sqrt{T_k}$ is substituted in the above relationship:

$$T_k = \left[\frac{2D}{20\,(t_1 + t_2)}\right]^2 \quad , \tag{5}$$

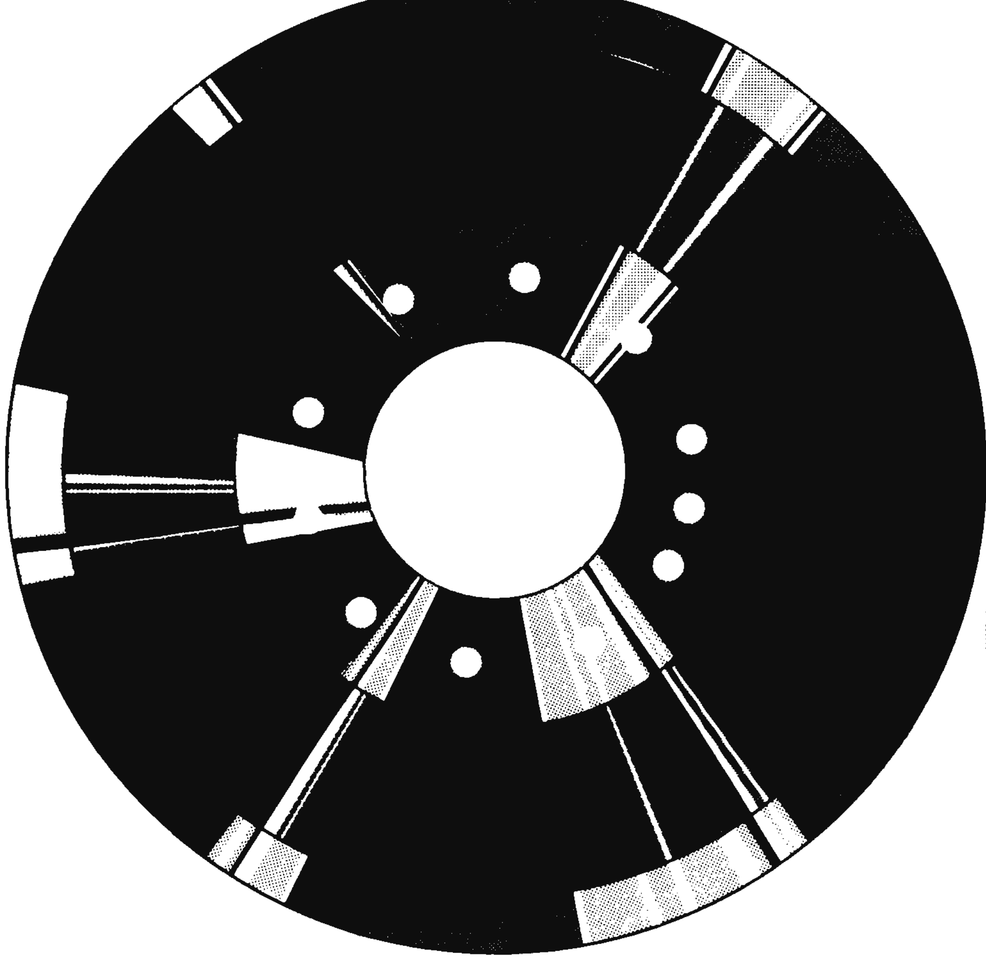
where $T_k$ is degrees Kelvin.

Since the velocity calibration of the instrument varies about 3.5% over a 0°C to 30°C range, the temperature measurement can be used to correct the velocity readings. Using equations (2) and (4) above:

$$W = \frac{2D\,(t_1 - t_2)}{(t_1 + t_2)^2} \quad . \tag{6}$$

Wind speed measuring resolution can be determined if the computer's input cycle time and anemometer path length are known. Recall that:

$$W = \Delta t\left(\frac{C^2}{2D}\right)$$



# It isn't in the hole it's on the doughnut!

## INTRODUCING — LOUMAR MANAGEMENT SYSTEMS SOFTWARE

**4** programs to give you a comprehensive package for business management

The Loumar General Accounting System is a versatile, fully integrated software package designed for small and medium sized businesses. It is also suitable for CPA's and bookkeeping service films.

The complete software system is composed of four main modules: GENERAL LEDGER, ACCOUNTS RECEIVABLE, ACCOUNTS PAYABLE AND PAYROLL. Each module may be used separately or in combination with any other module. Supplied on disk as run-time modules. Source not available.

All software is written in CBASIC II and utilizes the powerful CP/M operating system.

General system features include:

Automatic posting to general journal • Strict error detection • Report production on demand • Consistent operating procedures • User oriented. No previous computer knowledge required • Designed by accounting professionals • Comprehensive, well presented reports and manuals • Single or multiple client capabilities.

**HARDWARE SPECIFICATIONS**

The end user's microcomputer must satisfy the following requirements:

48k RAM
Dual floppy disk system
Printer with tractor. All printing is done in 80 col. format
CRT with at least a 64 character by 16 line display
CP/M and CBASIC II

Write for our brochure — Dealerships still available
Contact: Distributor

**PAYROLL:** Up to 500 employees — **$550.**

**ACCOUNTS RECEIVABLE:** Up to 1000 customers and 1000 monthly transactions — **$550**

**ACCOUNTS PAYABLE:** Up to 1000 vendors and 1300 monthly transactions — **$400**

**GENERAL LEDGER:** Up to 200 accounts with 2000 entries — **$450**

**MISSION CONTROL** • 2008 WILSHIRE BOULEVARD, SANTA MONICA, CA 90403 • (213) 829-5137

Circle 237 on inquiry card.

Assume a path length of 4.84 feet and C = 1100 feet per second. A Z-80 based computer operating at a system clock frequency of 2.5 MHz, for example, can accept input no faster than 8 $\mu$s per byte. Under these conditions, resolution is:

$$\frac{8 \times 10^{-6} \text{ seconds} \times (1100 \text{ feet/second})^2}{2 \times 4.84 \text{ feet}}$$

$$= 1 \text{ foot/second.}$$

For physical construction convenience I chose a three foot path length. This gives a resolution of nearly 1 mile per hour.

### Construction Details

Rigid support should be used for the transducer mounting. I chose 3/4 inch plastic pipe for low cost and ease of assembly. See figure 3 for more detail. The angle $\alpha$ is not critical. Just keep in mind that the height (H) must be high enough to prevent off-axis energy from the transducers from bouncing off any hardware in the bottom center of the assembly (such as a printed circuit card). These reflections can return to the point of origin before arrival of the desired pulse. The plastic pipe for both axes can easily be fastened to a wooden base frame. If you need a permanent installation, you can investigate better support arrangements.

The ultrasonic transducers (Model MK-109) range from one to two dollars apiece on the surplus market, and they can be conveniently supported by pieces of 3/4 inch thin wall plastic pipe. Use shielded wire between the transducers and the interface electronics. The electronic circuit card should be sheltered from the elements. (A plastic sandwich bag will work for the short term.) Unshielded wire, such as a ribbon cable, can be used between the computer and the interface.

Mechanical adjustment, besides the obvious line-of-sight alignment, consists of physically moving one or more transducers in their holders so that both *component vectors* are zero in still air. A program such as the demonstrator routine in listing 1 should be used for this adjustment.

In wiring the preamplifier section of the receiver, note that all the 74C04 integrated circuits are connected to a separate 5 V zener regulated supply. The shields of the receiver-transducer coaxial cables connect to the negative side of this zener diode. This preamplifier common connects to ordinary digital ground at only *one* location — a lead from the negative side of the zener to supply ground.

### Interface Electronics

Figures 1 and 2 may seem to indicate that considerable effort was wasted on obtaining an enormous signal to noise ratio. Not so. The barium titanate transducers (commonly used in intrusion alarms), having inherently high Q (ratio of inductance to resistance) and self resonance, are efficient only after many oscillations have built up. As impulse generators they are only marginally acceptable; I used them for their low cost and availability.

Complementary metal oxide semiconductor inverters, biased in their linear region, perform as stable high gain preamplifiers. The logic state edge detectors, formed by the comparator and type D flip flop combination, respond to the first negative or positive cycle received that exceeds a preset noise threshold.

The triacs are used to switch the output of the step-up transformer to either pair of transmitting transducers. Exclusive OR gate IC7c generates a delayed start strobe to the pulse generating circuitry. This delay allows the steering triacs to settle and permits only the desired set of transducers to activate. The monostable multivibrator IC10a, sensitive to transitions of either polarity, allows a single line from the output port to strobe the pulse generator and also select the desired wind direction to be measured.

The trigger threshold of all receiver circuitry is determined by a single resistor, $R_t$, in a simple voltage divider string. $R_t$ sets the difference between the comparator trip point levels, $V_H$ and $V_L$. The receiver must be sensitive enough to trigger on the second, third, or fourth incoming half cycle, but it must not be so sensitive as to latch up on extraneous noise. Increasing the value shown in the schematic, for example, decreases output sensitivity. Such action may be necessary if different path lengths or transducers are used.

Transformer T should have a turns ratio of approximately 10:1. A small 120 V to 12 V filament transformer will work here as a step up device, even at 40 kHz.

A precautionary note: when testing the pulse generating circuitry, do not run it continuously with transducers connected because the 200 V peak to peak signal could result in a burn-out of these devices.

### Software

Listing 1 contains a program written for the Z-80 microprocessor which displays data from the anemometer. It sends data to one bit of an output port and accepts data from

*Text continued on page 132*

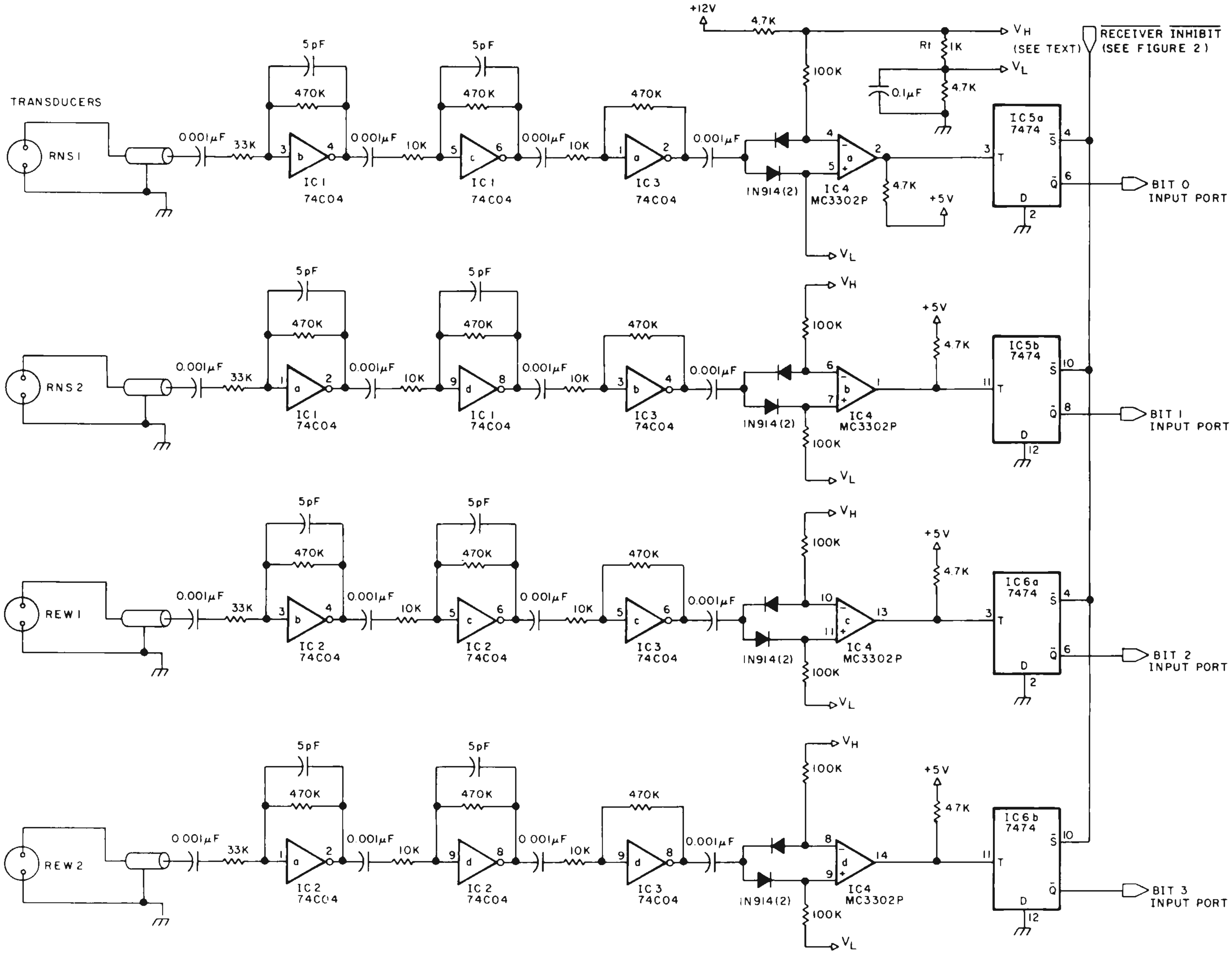



*Figure 1: Schematic diagram of receiver section of circuit. The arrival time of a pulse is perceived by the receiver as the first negative or positive half cycle to exceed a noise threshold.*

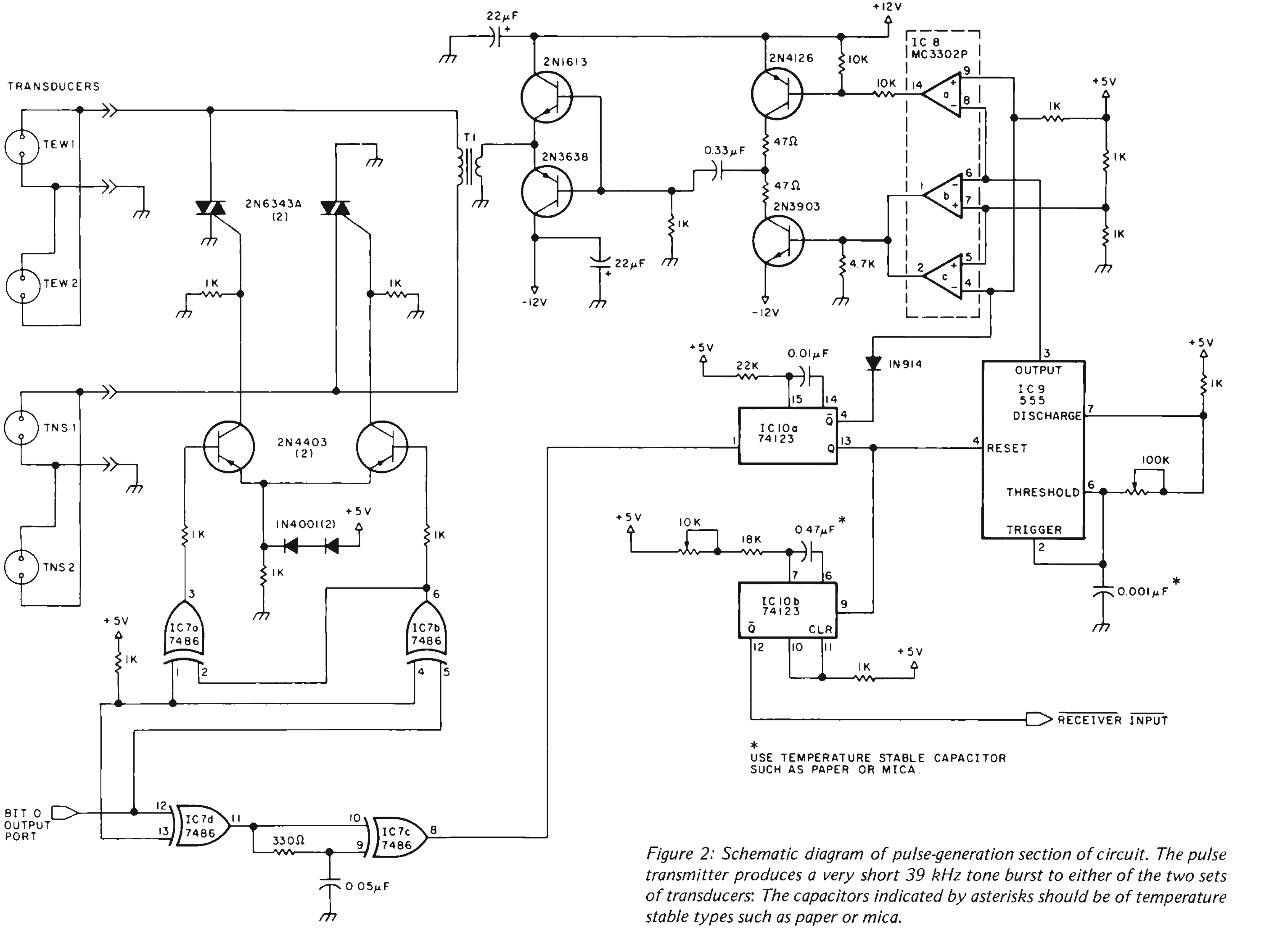
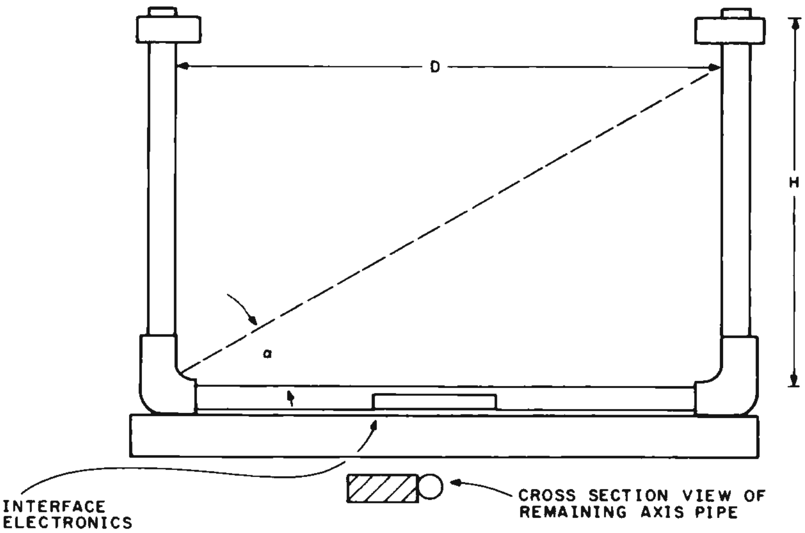



*Figure 2: Schematic diagram of pulse-generation section of circuit. The pulse transmitter produces a very short 39 kHz tone burst to either of the two sets of transducers: The capacitors indicated by asterisks should be of temperature stable types such as paper or mica.*

Table 1: *Power table for the integrated circuits used in figures 1 and 2.*

| Number | Type | +5 V | GND | −12 V | +12 V |
|---|---|---|---|---|---|
| IC1 | 74C04 | 14 | 7 | – | – |
| IC2 | 74C04 | 14 | 7 | – | – |
| IC3 | 74C04 | 14 | 7 | – | – |
| IC4 | MC3302P | – | 12 | – | 3 |
| IC5 | 7474 | 14 | 7 | – | – |
| IC6 | 7474 | 14 | 7 | – | – |
| IC7 | 7486 | 14 | 7 | – | – |
| IC8 | MC3302P | – | – | 12 | 3 |
| IC9 | 555 | 8 | 1 | – | – |
| IC10 | 74123 | 16 | 8 | – | – |

*Figure 3: Structural diagram of the sonic anemometer transducer apparatus. Shown are the two transducers for one axis; only a cross section of the other axis is seen. The electronic circuit card is seen near the center of the assembly. The measuring resolution is a function of the distance D between the transducers. The angle α should not be less than 30 degrees.*




*Listing 1: Program in Z-80 assembler code to gather data from the sonic anemometer and display wind direction on a video monitor.*

```
S 0600 063B :
0600  00             NOP  :
0601  00             NOP  :
0602  00             NOP  :
0603  CD  40  07     CALL 0740 :
0606  00             NOP  :
0607  00             NOP  :
0608  00             NOP  :
0609  00             NOP  :
060A  00             NOP  :
060B  00             NOP  :
060C  00             NOP  :
060D  00             NOP  :
060E  00             NOP  :
060F  00             NOP  :
0610  3E  00         LD   A,00 :
0612  D3  02         OUT  02 :
0614  CD  60  07     CALL 0760 :
0617  CD  34  07     CALL 0734 :
061A  CD  00  07     CALL 0700 :
061D  CD  BD  08     CALL 08BD :
0620  3E  01         LD   A,01 :
0622  D3  02         OUT  02 :
0624  CD  65  07     CALL 0765 :
0627  CD  10  07     CALL 0710 :
062A  CD  34  07     CALL 0734 :
```

0600 thru 063B triggers the pulse generator and calls all the support routines in their proper sequence.

*Listing 1 continued on page 126*

*Listing 1 continued from page 125:*

```
062D  CD 00 07     CALL 0700:
0630  CD D4 08     CALL 08D4:
0633  CD 0A 08     CALL 080A:
0636  CD 7B 08     CALL 087B:
0639  C3 00 06     JP   0600:

READY:
S 080A 0849:                          080A thru 0849 positions an ASCII asterisk in
080A  21 DF 0A     LD   HL,0ADF:      the video memory buffer.
080D  3A F3 06     LD   A,(06F3):
0810  47           LD   B,A:
0811  3A F2 06     LD   A,(06F2):
0814  98           SBC  B:
0815  38 08        JR   C,*081F*:
0817  32 F1 06     LD   (06F1),A:
081A  CD 4D 08     CALL 084D:
081D  18 08        JR   *0827*:
081F  ED 44        NEG  :
0821  32 F1 06     LD   (06F1),A:
0824  CD 51 08     CALL 0851:
0827  3A F5 06     LD   A,(06F5):
082A  47           LD   B,A:
082B  3A F4 06     LD   A,(06F4):
082E  98           SBC  B:
082F  38 0C        JR   C,*083D*:
0831  32 F6 06     LD   (06F6),A:
0834  28 05        JR   Z,*083B*:
0836  CD A0 08     CALL 08A0:
0839  00           NOP  :
083A  00           NOP  :
083B  18 0A        JR   *0847*:
083D  ED 44        NEG  :
083F  32 F6 06     LD   (06F6),A:
0842  23           INC  HL:
0843  23           INC  HL:
0844  3D           DEC  A:
0845  20 FB        JR   NZ,*0842*:
0847  36 AA        LD   (HL),AA:
0849  C9           RET  :

READY:
S 087B 0892:                          087B thru 0892 outputs the video buffer to the
087B  3E 7F        LD   A,7F:         16 by 64 character generator.
087D  CD FA 00     CALL 00FA:
0880  01 00 04     LD   BC,0400:
0883  21 00 09     LD   HL,0900:
0886  7E           LD   A,(HL):
0887  D3 00        OUT  00:
0889  AF           XOR  A:
088A  D3 00        OUT  00:
088C  23           INC  HL:
088D  0B           DEC  BC:
088E  78           LD   A,B:
088F  B1           OR   C:
0890  20 F4        JR   NZ,*0886*:
0892  C9           RET  :

READY:
S 084D 086E:                          084D thru 086E is a subroutine to raise or lower
084D  3E 42        LD   A,42:         the asterisk for routine 080A.
084F  18 02        JR   *0853*:
0851  3E 4A        LD   A,4A:
0853  32 6A 08     LD   (086A),A:
0856  3A F1 06     LD   A,(06F1):
0859  57           LD   D,A:
085A  FE 00        CP   00:
085C  C8           RET  Z:
085D  DE 07        SBC  07:
085F  38 04        JR   C,*0865*:
0861  3E 07        LD   A,07:
0863  18 01        JR   *0866*:
0865  7A           LD   A,D:
0866  01 40 00     LD   BC,0040:
0869  ED 42        SBC  HL,BC:
086B  3D           DEC  A:
086C  20 FB        JR   NZ,*0869*:
086E  C9           RET  :

READY:
S 08BD 08D2:                          08BD thru 08D2 extracts and stores vertical
08BD  3E 46        LD   A,46:         (N/S) travel times.
08BF  32 F0 08     LD   (08F0),A:
08C2  CD EB 08     CALL 08EB:
08C5  32 F2 06     LD   (06F2),A:
08C8  3E 4E        LD   A,4E:
08CA  32 F0 08     LD   (08F0),A:
08CD  CD EB 08     CALL 08EB:
08D0  32 F3 06     LD   (06F3),A:

READY:
S 08D4 08F4:                          08D4 thru 08EA extracts and stores horizontal
08D4  3E 56        LD   A,56:         (E/W) travel times.
08D6  32 F0 08     LD   (08F0),A:
```

*Listing 1 continued on page 128*


# The line your reputation should be on.

When a computer supplier or retailer sells equipment, his reputation is on the line.

And when the customer has to justify his investment decision, his reputation is on the line too.

So, naturally, everybody's a lot better off on a reliable line. Like ours.

With a name like Industrial Micro Systems, we wouldn't think of building anything short of industrial standards... even our S-100 products.

**No short cuts or jumpers.**

All of our boards — CPU, memory and controllers — are made from real fiberglass instead of a plastic substitute. And contacts are gold plated over a copper-ion barrier of nickel.

Every disk enclosure features a heavy-gauge steel chassis, and a heavy duty power supply. Built to take it.

Then come our complete systems. Aside from the disk drives, we manufacture every component that goes into them. Even the desk that goes around them. So we can vouch for rugged reliability inside and out.

**Two complete systems:**

Our new Series 5000 features a megabyte of storage on integral mini-floppies. And our Series 8000 is all business too. It can handle up to three megabytes on 8-inch drives. A large library of software, growing daily, includes CP/M,* PASCAL, and FAMOS.**

**Breaking the 64k barrier.**

Using Industrial Micro Systems 32k memory boards with Memory Management, our systems can handle up to 576k RAM internally.

**More in store.**

That's not the end of the line. It's growing all the time, along with our reputation for quality. Put your reputation on the line. Our line.

For more details just call or write. Supplier/dealer inquiries welcomed.

Industrial Micro Systems, 628 N. Eckhoff, Orange, CA 92668. (714) 633-0355.

**INDUSTRIAL MICRO SYSTEMS**

The great unknown.

Circle 173 on inquiry card.

*Trademark of Digital Research **Trademark of MV/T Systems

*Listing 1 continued from 126:*

```
08D9  CD EB 08     CALL 08EB :
08DC  32 F4 06     LD   (06F4),A :
08DF  3E 5E        LD   A,5E :
08E1  32 F0 08     LD   (08F0),A :
08E4  CD EB 08     CALL 08EB :
08E7  32 F5 06     LD   (06F5),A :
08EA  C9           RET  :
08EB  AF           XOR  A :
08EC  21 00 0D     LD   HL,0D00 :
08EF  CB 5E        BIT  3,(HL) :
08F1  C0           RET  NZ :
08F2  23           INC  HL :
08F3  3C           INC  A :
08F4  18 F9        JR   *08EF* :
```

08EB thru 08F4 counts the number of memory spaces necessary to find a bit set in the data.

```
READY :
S 0700 073A :
0700  06 00        LD   B,00 :
0702  0E 02        LD   C,02 :
0704  21 00 0D     LD   HL,0D00 :
0707  ED B2        INIR :
0709  C9           RET  :
070A  00           NOP  :
070B  00           NOP  :
070C  00           NOP  :
070D  00           NOP  :
070E  00           NOP  :
070F  00           NOP  :
0710  3E CE        LD   A,CE :
0712  32 1F 09     LD   (091F),A :
0715  3E D3        LD   A,D3 :
0717  32 DF 0C     LD   (0CDF),A :
071A  3E C5        LD   A,C5 :
071C  32 FF 0A     LD   (0AFF),A :
071F  3E D7        LD   A,D7 :
0721  32 C0 0A     LD   (0AC0),A :
0724  3E EF        LD   A,EF :
0726  32 DF 0A     LD   (0ADF),A :
0729  C9           RET  :
072A  00           NOP  :
072B  00           NOP  :
072C  00           NOP  :
072D  00           NOP  :
072E  00           NOP  :
072F  00           NOP  :
0730  06 01        LD   B,01 :
0732  18 02        JR   *0736* :
0734  06 01        LD   B,01 :
0736  0E 7F        LD   C,7F :
0738  ED BB        OTDR :
073A  C9           RET  :
```

0700 thru 0709 inputs a block of data from a port to a page in memory.

0710 thru 0729 places the N-S-E-W graticule in the buffer allocated for the video display.

0730 thru 073A is for delay only.

```
READY :
S 0740 074F :
0740  21 00 09     LD   HL,0900 :
0743  06 00        LD   B,00 :
0745  3E A0        LD   A,A0 :
0747  77           LD   (HL),A :
0748  23           INC  HL :
0749  10 FC        DJNZ *0747* :
074B  77           LD   (HL),A :
074C  23           INC  HL :
074D  10 FC        DJNZ *074B* :
074F  C9           RET  :
```

0740 thru 074F erases half the video buffer.

```
READY :
S 0760 0770 :
0760  21 00 0B     LD   HL,0B00 :
0763  18 03        JR   *0768* :
0765  21 00 0C     LD   HL,0C00 :
0768  06 00        LD   B,00 :
076A  3E A0        LD   A,A0 :
076C  77           LD   (HL),A :
076D  23           INC  HL :
076E  10 FC        DJNZ *076C* :
0770  C9           RET  :
```

0760 thru 0770 are the remaining video erase routines.

```
READY :
S 08A0 08AB :
08A0  47           LD   B,A :
08A1  DE 0F        SBC  0F :
08A3  38 02        JR   C,*08A7* :
08A5  06 0F        LD   B,0F :
08A7  2B           DEC  HL :
08A8  2B           DEC  HL :
08A9  10 FC        DJNZ *08A7* :
08AB  C9           RET  :

READY :
```

08A0 thru 08AB moves the asterisk to the left in proportion to the magnitude of the horizontal (E/W) wind vector. Limits the data to 15 decimal to prevent overwriting other memory.

# Get Ready to Make Your Own Magic

Announcement I. The first eight Personal Programs™ from Aladdin Automation are waiting for you now at your neighborhood computer retailer or direct from Aladdin.

Now you can get your full share of Aladdin magic in every one of these Personal Programs™:

**Math-Ter-Mind**™ A delightful, educational learning experience for your pre-school child. Watch the smile on your child's face as a correct answer makes the mathematician smile on the screen before you. A nursery song also serves as a reward for learning elementary addition and subtraction. With Aladdin's Math-Ter-Mind™ your child's pathway to learning will be fun-filled . . . for both of you. Math-Ter-Mind™. The first release from the Aladdin Education™ Series. (nursery song currently available only on Apple II™ program)

**Lunar Lander** In a controlled descent, you're just seconds away from your first landing on the cold, forbidding surface of the moon. As you navigate your delicate spacecraft downward to the safety of Moonbase, you must be ever watchful of the dangers rising to meet you with each passing moment: a fuel level fast approaching zero; deadly meteor showers that come from any direction, at any time; sheer-faced rock cliffs and rough terrain, choosing the correct landing pattern and rate of descent. Aladdin's Lunar Lander. Your chance to reach out and touch the stars . . . without leaving the safety and comfort of your own chair. The first release from the Aladdin Simulation™ Series.

**Craps** All eyes in the casino are on you. The dice are in your hands. Lady Luck sits at your shoulder, whispering . . . "Just one more time. Try your luck just one more time." You throw . . . and watch the dice tumbling on the screen. With Aladdin's Craps you play against the computer, so it's awfully tough to win. But when you do, it's an experience you're likely never to forget. Craps. An exciting, heart-pounding Personal Program™. The first release from the Aladdin Las Vegas™ Series.

**Mastermind** A challenging game of intrigue, centuries old, that will give you full chance to test your powers of logic, deduction and reason. And test them you will, as you try and solve the computer's puzzle, using clues as they're provided one-by-one. You control the degree of difficulty in this classic Personal Program™ that offers one simple, yet all-consuming challenge: beat the Mastermind in a direct, one-on-one battle of wits. Aladdin's Mastermind. The first release from the Aladdin Old Favorites™ Series.

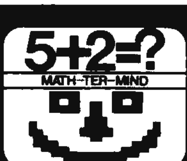
**Tic-Tac-Toe** Five different levels of difficulty allow a person of any age or skill to take part in this relaxing, enjoyable game that can act as a learning tool, as well. Level I, for example, is suitable for children and is excellent also for teaching simple mathematics. The computer plays just about perfectly at Level V. Just about, that is, so go ahead and take your best shot. See if you can beat the computer in this traditional favorite of young and old alike. Tic-Tac-Toe. Another first release from the Aladdin Old Favorites™ Series.

**Jungle Island**™ Shipwrecked in a raging storm at sea, miraculously you survive only to find yourself stranded on a seemingly deserted jungle island. Without food, water or supplies of any kind, you begin to try and find your way to safety. The computer will be your eyes and ears as you explore your jungle island and all the mysteries and dangers that lie in wait for you. Jungle Island™. A captivating first release from the Aladdin Adventure™ Series.

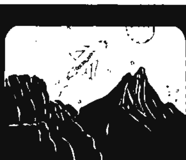
**Stix**™ Aladdin's Stix™ can be played with 2 to 5 piles of sticks and between 1 and 19 sticks in each pile. The object: to be the one to pick up the last stick. Sounds simple? Yes, but you're playing against the computer. Take heart, though, because you can control the degree of difficulty in this update of the ancient game of Nim. Stix™. Another first release from the Aladdin Old Favorites™ Series.

**Super Pro Football**™ Here's your chance to be more than just an armchair quarterback. With Aladdin's Super Pro Football™ you can replay any Super Bowl game, from the first, between Green Bay and Oakland, to last year's classic victory by Pittsburgh over Dallas. For once you can turn back the clock and go for that one big play that made the difference between victory and defeat in pro football's biggest game of all. Super Pro Football™. The first exciting release from the Aladdin Super Pro™ Series.

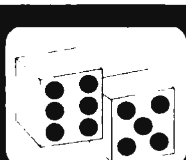
Visit your neighborhood computer retailer or contact Aladdin direct to get your full share of the magic in Announcement I, the first eight Personal Programs™ from Aladdin Automation.




Math-Ter-Mind™

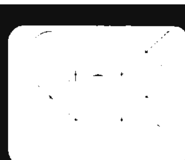


Lunar Lander



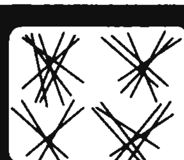
Craps

Mastermind

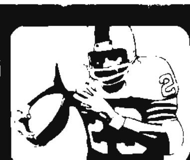


Tic-Tac-Toe

Jungle Island™



Stix™



Super Pro Football™

**Welcome to the All-New World of Aladdin. And Get Ready to Make Your Own Magic**

ALADDIN AUTOMATION, INC.
ALADDIN COMPUTER CORP.
3420 Kenyon Street, Ste. 131, San Diego, CA 9211



Design and copy by Campbell Marsh Graphic Communications

*Table 2: Summary of actions taken by program in listing 1.*

1. Output to a port to trigger the North-South transducers.
2. Erase part of the video buffer (must be a time-invariant task).
3. Input a block of data into memory from anemometer port.
4. Examine memory, extract, and save North-South travel times.
5. Output to port to trigger East-West transducers.
6. Erase remainder of video buffer.
7. Input a block of data into memory.
8. Examine memory, extract, and save East-West travel times.
9. Move cursor (equivalent to head of resultant wind vector) in video memory buffer appropriate horizontal (East-West) and vertical (North-South) distances from origin.
10. Display the contents of video buffer on the video monitor.
11. Compute and display the temperature. (optional)
12. Go to step 1.

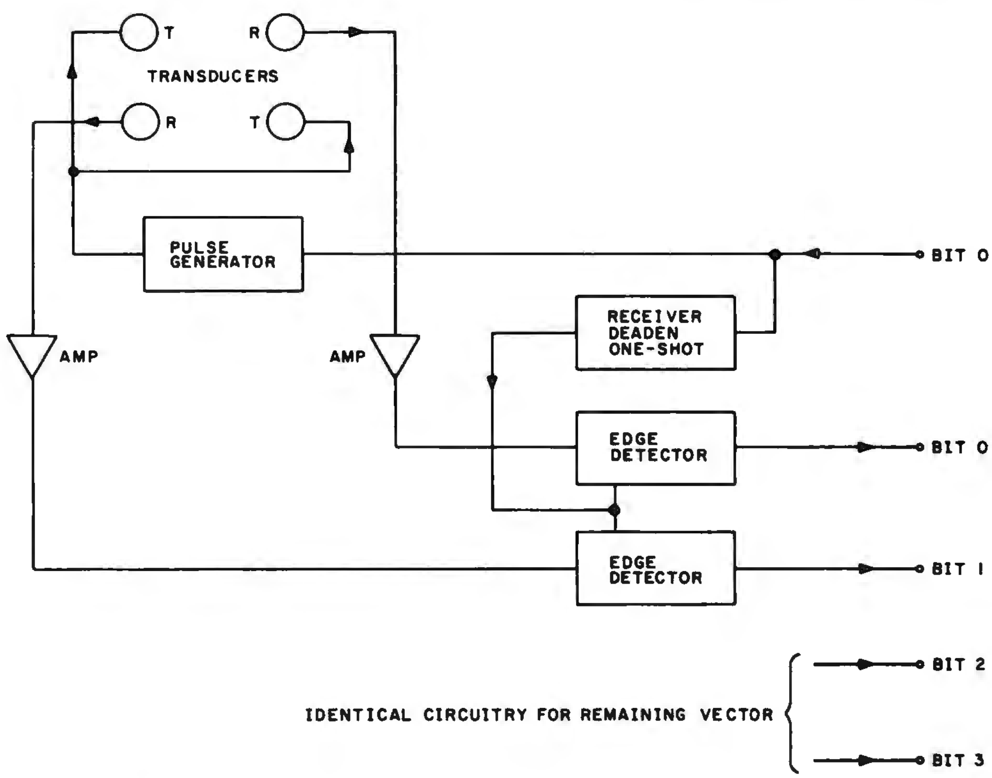



*Figure 4: Block diagram of anemometer hardware.*

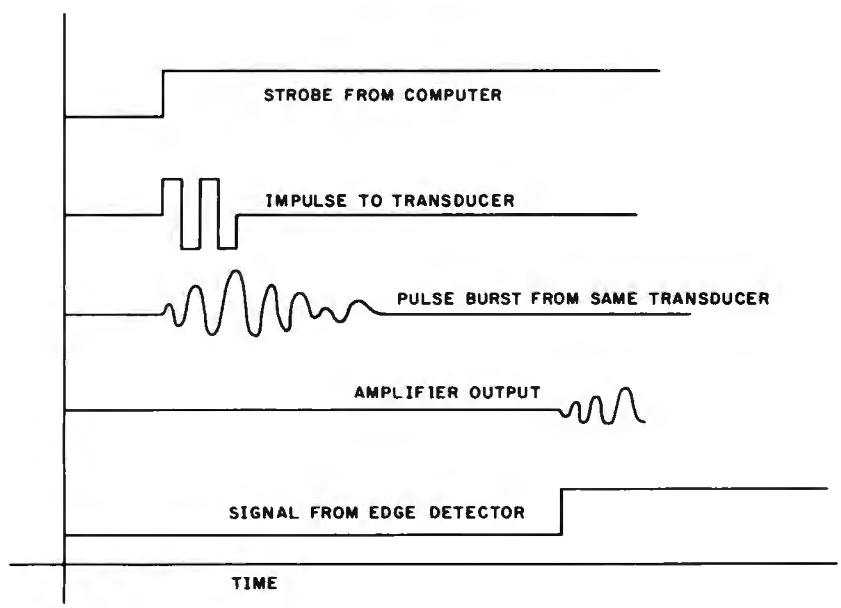
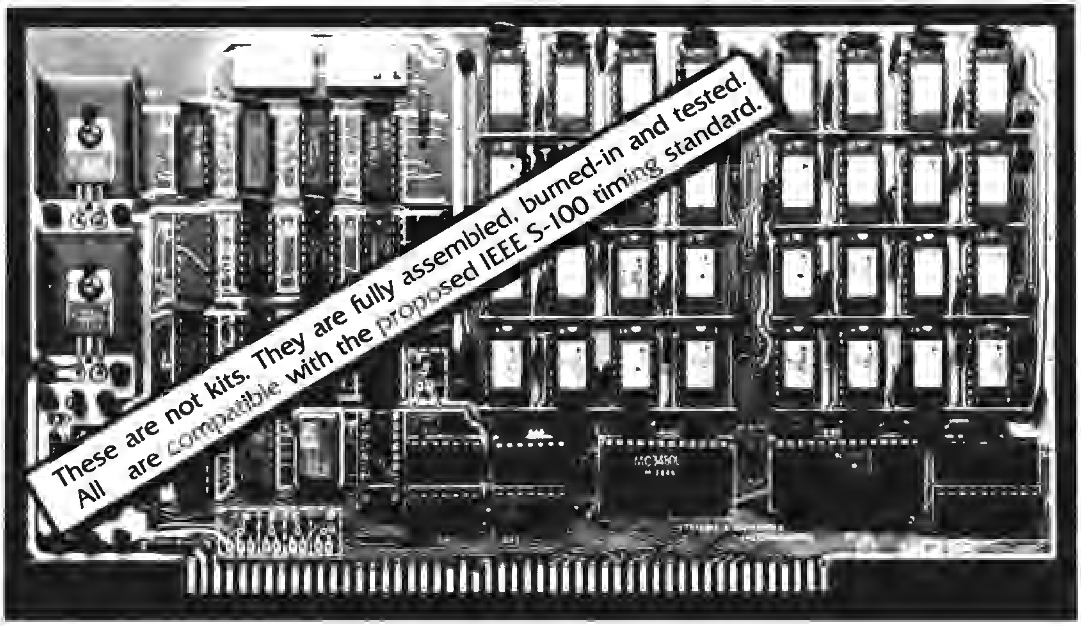



*Figure 5: Timing relationships of signals present in the sonic anemometer.*




• NEW PRODUCT •

64K bytes of fast, low power dynamic RAM. MEMORY BANK SELECT

**$895.00**

for model DMB 6400

••••••••••••

64K bytes of fast, low power dynamic RAM.

**$795.00**

for model DM 6400

••••••••••••

• NEW PRODUCT •

32K bytes of faster Static RAM which MEMORY BANK SELECT

**$695.00**

for model SMB 3200

## Both our UNCommon Dynamic and our UNCommon Static RAMS have the following features and specifications:

- They are all GUARANTEED to be compatible with the following S-100 systems:
  CROMEMCO, IMSAI, ITHACA AUDIO, MITS, NORTH STAR, PROCESSOR TECHNOLOGY, TDL, TEI, VECTOR GRAPHICS, and other S-100 systems.

## Both of Our UNCommon Dynamic RAM Series, the DMB-6400 and the DM-6400, feature:

- 64K bytes of dynamic RAM with on board transparent refresh.
- S-100 interface compatible, with crystal controlled timing INDEPENDENT of bus or processor timing
- No wait states with 8080 or Z-80 to 4MHz. Up to 5MHz with 1 wait state.
- Memory selectable or deselectable in 4K byte increments
- Low power, 8 watts maximum, in 64K byte configuration

## Our UNCommon Static RAM Series, the SMB-3200, features:

- Memory Bank select capabilities: Either two (2) 16K byte banks of memory, or one (1) 32K byte bank per board
- 32K bytes of low power static RAM
- No wait states with 8080, 8085, or Z-80 processors up to 5MHz
- Addressable in 4K byte increments at 4K boundaries. Deselectable in 4K byte increments.

## Announcing the Model DMB-6400 Series of UNCommon Dynamic RAMS. This Series incorporates the features which are standard in the DM-6400 Series and adds the following capabilities:

- MEMORY BANK SELECT which is compatible with CROMEMCO NORTH STAR, and other systems using output port bank select
- 256 ports selectable (including CROMEMCO 40H and NORTH STAR COH)
- Eight (8) 64K bytes banks of memory per output port
- Each 16K bytes of addressable memory space may be individually set to start at 0000, 4000, 8000, or C000 and can be set for any of the 8 banks on one selected output port. The bank memory size can be incremented from 16K bytes to 64K bytes in 16K increments, allowing 512K byte bank sizes.
- Total memory addressing to over 100 Megabytes by using different I/O ports for memory control.

## All of our UNCommon Dynamic and Static Series RAMS feature the following:

- Inputs are RC filtered and buffered with 1 LS TTL load. Compatible with terminated or unterminated busses.
- Outputs are all tri-state.
- Disc compatible. DMA compatible.
- Phantom memory selectable on pin 67
- DIP switch selectable addressing.
- Reliability — all boards fully tested and burned-in
- FULL DOCUMENTATION — Schematics, layout, parts list, theory of operation, timing diagrams, and option selection.
- Industrial quality design, components and construction. Glass epoxy boards. Fused, double solder masks. Silk screened legends. Gold plated connector contacts. All IC's socket mounted.
- Guaranteed performance on parts and labor for one year

NEW PRODUCTS

● UNCommon Dynamic RAMS with MEMORY BANK SELECT.

- ☐ DMB 6400 64K RAM — $895.00
- ☐ DMB 4800 48K RAM — $795.00
- ☐ DMB 3200 32K RAM — $695.00

● UNCommon Static RAMS with MEMORY BANK SELECT.

- ☐ SMB 3200, 32K RAM — $695.00

● UNCommon Dynamic RAMS

- ☐ DM 6400 64K RAM — $795.00
- ☐ DM 4800 48K RAM — $695.00
- ☐ DM 3200 32K RAM — $595.00

Delivery stock 30 d.

• PRICES SUBJECT TO CHANGE WITHOUT NOTICE •

**MEASUREMENT systems & controls incorporated**

867 North Main Street
Orange, California 92668
Telephone: (714) 633-4460

For the above products see your local dealer or order directly.

ORDERS — BA, VISA, MASTER CHARGE, money orders, or personal checks. Please allow 14 days for checks to clear bank. California residents please add 6% sales tax to your order.

All orders shipped postpaid. All orders in U.S. funds. Please add 10% on all orders outside the U.S.A. and Canada.

Circle 215 on inquiry card.

*Photo 1: Wind direction display as it appears in the 16 line by 64 character format on the video monitor. The wind speed and velocity are indicated by the position of the asterisk with respect to the origin at center.*

*Text continued from page 122:*

four bits of an input port. The composite wind vector is displayed on a video monitor in a 16 line by 64 character format. The program does not include a routine for computing the square root of the sum of squares; the plotting method used in the video formatting makes this unnecessary.

1 K bytes of memory is allocated for video display storage. A fixed cardinal direction graticule and an asterisk varying with the wind vector are placed in this buffer by subroutines. For each complete measurement by the transducers, the program erases and restores the video buffer. The position of the asterisk with respect to the center of the screen indicates wind direction and magnitude. Photo 1 shows the display. This arrangement, although cramping the display to only a ±8 unit variation in the North-South direction and allowing a ±32 unit variation in the East-West direction, still permits an interesting, virtually instantaneous display of wind activity.

Some of the subroutines in the listing have been split up so that the computer is doing useful things even during the transit times of the sound pulses (eg: erasure of the video buffer). The program measures the pulse travel times in a manner analogous to a counter. At a fixed time after pulse initiation, the input routine begins to look at a particular *pulse has arrived* flip flop (part of the hardware interface that connects to an input port) and records its logic state, 0 or 1, into an initial location in memory. This routine repeats itself 256 times, each time entering another observation into the next memory location allocated for logging. Checking for a 1 state (pulse has arrived) is saved for later in order to get the best time resolution out of the processor. A number whose value is proportional to travel time is finally obtained by a routine which starts at the beginning of the logging buffer and counts the number of successive memory bytes necessary to find a 1 in a particular bit position. Since a bit is allocated for each of the four travel times, determining the wind direction and speed is simply a matter of testing each of the four bits in an identical fashion.■

*Photo 2: The interface electronic circuits are housed inside a convenient but necessary weather barrier.*

*Photo 3: This view shows the plastic pipe construction of the sonic anemometer and the positioning of the transducers.*

# The Nature of Robots

## Part 2: Simulated Control System

**William T Powers**
**1138 Whitfield Rd**
**Northbrook IL 60062**

*Figure, table, and listing numbering continued from part 1.*

> **Note on North Star BASIC**
>
> The method of accessing strings in North Star BASIC is different from that of Microsoft and other BASICs. Translate as follows:
>
> | | | |
> |---|---|---|
> | A$(1,n) | becomes | LEFT$(A$,n) |
> | A$(n) | becomes | RIGHT$(A$,n) |
> | A$(m,n) | becomes | MID$(A$,m,n) |

In part 1, we went through a chain of reasoning that ended with the conclusion that the behavior of an organism is not what it seems. Behavior appears to be at the end of a cause and effect chain that starts with the inputs to a nervous system, but that chain is subject to disturbances that can occur *after* the output of the nervous system. Nevertheless, the behavior at the end of this chain is stable and repeatable, while events closer to the organism become less predictable as we get nearer to the neural signals at the output of the nervous system. By analyzing an example in which a car is maintained in the center of its lane, we saw that this measure of behavior belongs at both the cause and effect ends of the chain, and that if this variable is shown only once in the diagram, a closed loop results.

We are going to look in more detail at the behaving system in this closed loop, to see how it might be organized to produce the results seen. We will start using a simulator written in BASIC which allows the user to vary many parameters of the control system to see the effects on its actions. Human behavior will not be mentioned much in this installment; there are many fundamentals to cover before we can get back to the main purpose of this series. The object here is to retrain the intuition so that the closed loop way of seeing behavior becomes as natural as the old straight through cause and effect way.

### Organization of a Control System

The simulator (listing 2) is set up to demonstrate the properties of a standard sort of control system organization. We will first look at that organization, then at the simulator itself, and finally at some details of the operation of the control system. You

*Listing 2: A control system simulator written in North Star BASIC.*

```
1    PRINT "PROGRAM TWO: SIMULATION OF CONTROL SYSTEM BEHAVIOR"
2    PRINT
3    PRINT "AFTER PROMPT (COLON), YOU MAY TYPE"
4    PRINT "'PLOT XXXXXX', WHERE XXXXXX MEANS"
5    PRINT "ANY ONE OR MORE CHARACTERS FROM THE"
6    PRINT "SET P,E,R,I,O,D, IN ANY SEQUENCE."
7    PRINT
8    PRINT "YOU MAY ALSO SET PARAMETERS BY TYPING IN"
9    PRINT "THE PARAMETER SYMBOL IMMEDIATELY FOLLOWED"
10   PRINT "BY AN EQUAL SIGN AND THE VALUE (NO SPACES)."
11   PRINT
12   PRINT "PARAMETERS ARE L, K1, K2, S1, S2, O, P, R, AND D"
13   PRINT "DEFAULT VALUES 16, 1, 2, 1, 1, 0, 0, 0, AND 15"
14   PRINT
15   PRINT "TO RUN, TYPE '.' (INITIALIZE), OR '/' (DON'T INIT)."
16   PRINT
17   K1=1
18   K2=2
19   S1=1
20   S2=1
21   P0=0
22   O0=0
23   R0=0
24   D0=15
25   V(4)=1
26   V(5)=1
27   V(6)=1
28   INPUT "DISPLAY WIDTH: ",W
29   W=W-2
30   C=W/2\ REM                          CENTER OF DISPLAY
31   DIM Z$(W),M$(W),A$(20),B$(6),K(6),U(6),E$(72)
32   B$="PERIOD"
33   L1=15
34   FOR J=1 TO W
35   Z$(J,J)=" "
36   NEXT J\ REM                         CREATE BLANK FILE
37   DEF FNI(X)\ REM                     INPUT FUNCTION
38   P=P+S1*(K1*X-P)
39   RETURN P
40   FNEND
41   DEF FNO(X)\ REM                     OUTPUT FUNCTION
42   O=O+S2*(K2*E-O)
43   RETURN O
44   FNEND
45   DEF FNF(X)=0.5*X\ REM               FEEDBACK FUNCTION
46   DEF FND(X)=0.8*X\ REM               DISTURBANCE FUNCTION
```

can do much more experimenting than we will discuss here.

Figure 5 is a diagram of a typical control system. Almost every control system can be expressed in this form, although in the real system, functions that are shown here as separate are often combined into one physical entity. The symbols for functions and variables are those which appear in the BASIC simulator.

The behaving system is entirely above the boundary line. All that is not the behaving system (or systems inside the organism at a higher level, not considered here) is called the *environment* of the system. Variables inside the system will always be called *signals*, and variables in the environment will always be called *quantities.*
In the environment we have three quantities mentioned in part 1. The *input quantity* is a physical variable that the system can sense. The state of this quantity is the result of all influences acting on it (which in our limited universe means the influence from the system's own output) and one representative *disturbing quantity* that can vary independently from what the system does. The system's output is represented by the *output quantity*. The input quantity is called I, the output quantity O, and the disturbing quantity D.

The output and disturbing quantities are separated in space from the input quantity, and they influence the input quantity through properties of the intervening environment. The connection that translates the state of the output quantity into an influence on the input quantity is called the *feedback function*, symbolized in BASIC as FNF. The function that translates the state of the disturbing quantity into another influence on the input quantity is the *disturbing function*, symbolized FND. If the input quantity is associated with some physical object, then FNF and FND may both contain properties of that object (eg: its mass). There are less redundant ways to handle this in special cases.

The meaning of the previous paragraph is summed up in line 102:I = FNF(O) + FND(D). The state of the input quantity is the sum of the influences from the output quantity and the disturbing quantity. In the real world, both the output quantity and the disturbing quantity may have many effects other than those on I, but those effects are irrelevant to the operation of this system (perhaps not to the designer or user of the system, if it is artificial). We have therefore considered everything about the environment that is of interest here.

```
47   REM **
48   REM ** COMMANDS FOR SETTING PARAMETERS
49   GOTO 51
50   A$ = " " \ IF E1 > LEN(E$) THEN 51 ELSE 53
51   INPUT ":",E$ \ A$ = " " \ E1 = 1
52   IF LEN(E$) < > 0 THEN 53 \ PRINT \ GOTO 51
53   E1$ = E$(E1,E1) \ E1 = E1 + 1
54   IF E1$ = "," THEN 57 ELSE IF E1 > LEN(E$) THEN 56
55   A$ = A$ + E1$ \ GOTO 53
56   A$ = A$ + E1$
57   IF A$ = "." THEN 95
58   IF A$ = "/" THEN 99
59   IF A$ < > "?" THEN 62
60   PRINT \ PRINT%7F3,"K1 = ",K1,"  K2 = ",K2,"  S1 = ",S1,"   S2 = ", S2
61   GOTO 51
62   IF LEN(A$) < 5 THEN 72
63   IF A$(1,5) < > "PLOT" THEN 72
64   A$ = A$(6)
65   FOR J = 1 TO 6 \ REM                       TAG VARIABLES TO
66   V(J) = 0 \ REM                             BE PLOTTED.
67   FOR K = 1 TO LEN(A$)
68   IF A$(K,K) = B$(J,J) THEN V(J) = 1
69   NEXT K
70   NEXT J
71   GOTO 50
72   IF LEN(A$) < 3 THEN 91
73   IF A$(1,3) < > "K1 =" THEN 75
74   K1 = VAL(A$(4)) \ GOTO 50
75   IF A$(1,3) < > "K2 =" THEN 77
76   K2 = VAL(A$(4)) \ GOTO 50
77   IF A$(1,3) < > "S1 =" THEN 79
78   S1 = VAL(A$(4)) \ GOTO 50
79   IF A$(1,3) < > "S2 =" THEN 81
80   S2 = VAL(A$(4)) \ GOTO 50
81   IF A$(1,2) < > "O =" THEN 83
82   OO = VAL(A$(3)) \ GOTO 50
83   IF A$(1,2) < > "P =" THEN 85
84   PO = VAL(A$(3)) \ GOTO 50
85   IF A$(1,2) < > "R =" THEN 87
86   RO = VAL(A$(3)) \ GOTO 50
87   IF A$(1,2) < > "D =" THEN 89
88   DO = VAL(A$(3)) \ GOTO 50
89   IF A$(1,2) < > "L =" THEN 91
90   L1 = VAL(A$(3)) \ GOTO 50
91   PRINT "???", \ GOTO 50
92   REM **
93   REM **                                     SIMULATION AND PLOTTING LOOP
94   REM **
95   P = PO \ REM                               ENTRY WITH INITIALIZATION
96   O = OO \ D = DO \ R = RO
97   I = FNF(O) + FND(D)
98   E = R - P \ GOSUB 109 \ REM                PLOT INIT. CONDITIONS
99   D = DO \ REM                               ENTRY, NO INITIALIZATION
100  R = RO
101  FOR L = 1 TO L1 \ REM                      CONTROL LOOP SIMULATION
102  I = FNF(O) + FND(D)
103  P = FNI(I)
104  E = R - P
105  O = FNO(E)
106  GOSUB 109 \ REM                            CALL PLOTTING SUBROUTINE
107  NEXT L
108  GOTO 50
109  REM **
110  REM **                                     PLOTTING SUBROUTINE
111  REM **
112  U(1) = P + C
113  U(2) = E + C
114  U(3) = R + C
115  U(4) = I + C
116  U(5) = O + C
117  U(6) = D + C
118  PRINT
119  M$ = Z$ \ REM                              CLEAR OUTPUT BUFFER
120  M$(C + 1,C + 1) = "." \ REM                MARK SCREEN CENTER
121  FOR J = 1 TO 6 \ REM                       LOAD BUFFER
122  U = INT(U(J) + .5) + 1
123  IF U < 1 THEN U = 1
124  IF U > W THEN U = W
125  IF V(J) = 1 THEN M$(U,U) = B$(J,J)
126  NEXT J
127  PRINT M$, \ REM                            PRINT BUFFER
128  RETURN
999  END
```

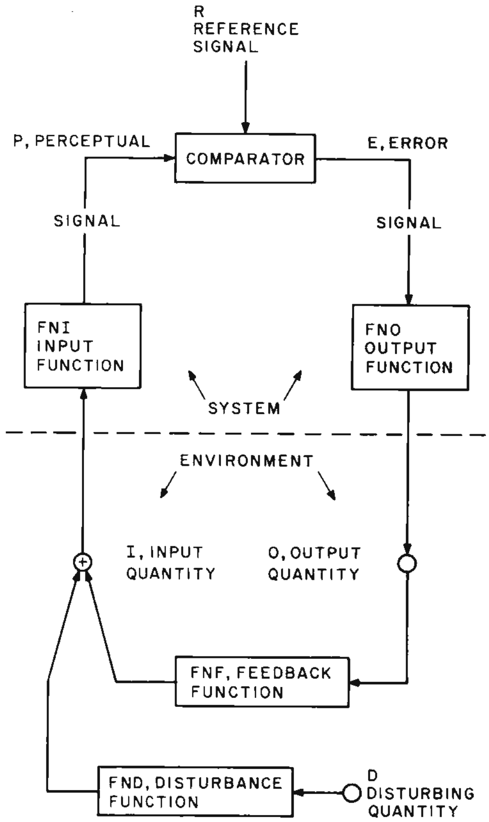



*Figure 5: The system's output quantity, O, influences the input quantity, I, via the feedback function, FNF. The disturbing quantity, D, influences the input quantity via the disturbance function FND. Both FNF and FND represent physical links in the environment. The state of the input quantity is determined by the sum of these two influences.*

*The system's input function, FNI, converts the state of the input quantity into a magnitude of the perceptual signal P. P is compared with the reference signal R in the comparator function, which emits an error signal E = R − P. The error signal is converted into a magnitude of the output quantity via the output function, FNO.*

Above the line we have the behaving system. We cross the boundary at the *input function*, FNI. This is the function which turns the state of an external quantity, I, into the magnitude of a *perceptual signal*, P. Both sensors and computing processes may be involved in a complex input function. The outcome, however, is always the magnitude of a single signal, whatever it represents. This signal can only increase or decrease; we will always work with one-dimensional control systems, treating multidimensional control phenomena by using multiple control systems. The perceptual signal is the system's internal representation of the external world – its only such representation.

Line 103 expresses the definition of the input function and the way it relates the input quantity and perceptual signal: P = FNI(I).

Inside the system is another signal, the *reference signal*, R. In living systems, this signal is generated elsewhere in the organism; it is not accessible from outside. The reference signal, along with the perceptual signal, enters a function called the *comparator*, which subtracts one signal from the other and emits an *error signal*, E, representing the signed difference of magnitudes. It does not matter which signal is subtracted from which, but for uniformity we will always treat the reference signal as the positive input and the perceptual signal as the one subtracted from it. Thus, a positive error signal always means that the reference signal is larger than the perceptual signal. This function does not have to be generalized, as nonlinearities and amplification can always be absorbed into one of the other functions.

# FREE SOFTWARE!

FINALLY....


a quality assembler for the TRS-80*

## PRACTICAL MICROCOMPUTER PROGRAMMING: THE Z80
### by W.J. Weller

Here from W.J. Weller and Northern Technology Books is the third in the **Practical Microcomputer Programming** series. It is a comprehensive text covering assembly language programming for Z80 based microcomputers. The first 16 chapters cover Z80 programming comprehensively, from binary operations to interrupt handling. Included are chapters on moving data, logical and arithmetic operations, use of the stack, communications with the terminal, floating point arithmetic and graphic output. All programming techniques are illustrated with formal tested examples. An important feature of the book is that it uses the universal standard 8080 mnemonics. This is of great help to users who are upgrading their machines and software to utilize the Z80 processor.

The last part of the book is software; an editor/assembler which will run on any 8080 or Z80 machine and a debugging monitor.

Hardcover $29.95

## FREE:

Send in the coupon supplied with the book and receive FREE the object programs of the editor/assembler and debug on either **paper tape** or on **cassette tape for the TRS-80 microcomputer.** (Loads in Level I with 16K RAM or Level II.)

Both the book **and** software for only $29.95!

- Also available is the editor/assembler object code for the Z-80 on diskette for North Star disc systems. Only $14.00
- In addition, the editor/assembler and debug source code for the Z 80 are available on diskette for North Star disc systems. Only $14.00
- Or both for $26.00!

## AN EDITOR/ASSEMBLER SYSTEM FOR 8080/8085 BASED COMPUTERS
### by W.J. Weller and W.T. Powers

This 148 page book contains complete information for initializing and using a powerful new editor/assembler and debugging monitor system, and the full SOURCE text of both. The assembler fully supports all Intel instruction mnemonics as well as the entire language used in **Practical Microcomputer Programming™:The Intel 8080**. The editor/assembler is resident in less than 8K RAM and will run on any 8080, 8085 or Z80 based computer with peripherals which transfer on a character basis or can be made to do so by buffering. The user supplies his or her own I/O drivers. The text editor is extremely simple to use and does not require irrelevant line numbers. Also included is a program to convert Processor Technology™ format tapes to a format usable by the editor/assembler.

This system is not the usual "quickie" software, riddled with errors and limitations, but a professionally created, thoroughly tested and debugged system. At $14.95 it is the best software bargain you are ever likely to see.

**AND BY THE WAY...paper tape object copies of all this software are sent FREE to book purchasers when the coupon at the back of the book is returned to Northern Technology Books.** 8½" x 11". $14.95

- 8080/8085 editor/assembler object code on diskette for North Star disc systems. Only $14.00
- 8080/8085 editor/assembler and debug source code for North Star disc systems. Only $14.00
- Or both for $26.00!

## PRACTICAL MICROCOMPUTER PROGRAMMING: THE INTEL 8080
### by W.J. Weller, A.V. Shatzel and H.Y. Nice

Here is a comprehensive source of programming information for the present or prospective user of the 8080 microcomputer, including moving data, binary arithmetic operations, multiplication and division, use of the stack pointer, subroutines, arrays and tables, conversions, decimal arithmetic, various I/O options, real time clocks and interrupt driven processes, and debugging techniques.

This 306-page hardcover book is well worth its $21.95 price and should be in every 8080 or Z80 user's library.

## PRACTICAL MICROCOMPUTER PROGRAMMING: THE M6800
### by W.J. Weller

This second volume of the **Practical Microcomputer Programming** series addresses the problems of applications programming at assembly level for the M6800. In 16 chapters and more than 100 formal examples, the fundamental techniques of assembly level programming are applied to the solution of specific problems with the 6800. Nowhere theoretical, it is a thorough and detailed methods text for the beginning and intermediate application programmer using the 6800. $21.95 hardcover.

*A trademark of Tandy Corporation

**BITS** inc T.M.

Books to erase the impossible

25 Route 101 West,
PO Box 428,
Peterborough, NH 03458
603-924-3355

Please send me the items checked below.

☐ **Practical Microcomputer Programming — The Z80** (with choice of either FREE paper tape or FREE TRS-80 cassette when I mail coupon supplied with book) $29.95
☐ **Editor/Assembler System for 8080/8085 Based Computers** (with paper tape object copies of the software sent FREE when I mail coupon supplied with book) $14.95
☐ **Practical Microcomputing Programming — The Intel 8080** $21.95
☐ **Practical Microcomputing Programming — The M6800** $21.95
☐ Editor/assembler object code for the Z 80 on diskette for North Star disc systems $14.00
☐ Editor/assembler and debug source code for the Z 80 on diskette for North Star systems $14.00
☐ **SAVE $2.00!** Both object code and source code for the Z 80 $26.00
☐ Editor/assembler object code for 8080/8085 on diskette for North Star disc systems $14.00
☐ Editor/assembler and debug source code for 8080/8085 on diskette for North Star disc systems $14.00
☐ **SAVE $2.00!** Both object code and source code for 8080/8085 $26.

Total enclosed $________(Please enclose 75¢ shipping/handling for each item ordered)

☐ Check Charge Card ☐ Master Charge ☐ Visa

Card #________________

Expires________ Signature________________

Name________________

Address________________

City__________ State________ Zip________

Dial your charge card orders toll-free: 800-258-5477 Mon.-Fri 9 - 5 PM (In New Hampshire dial 924-3355)

You may photocopy this page

Dealer inquiries invited

### Anatomy of the Simulator

*Let's run through the simulator quickly before we start using it, to see how this control organization operates.*

*Lines 1 thru 16 are user instructions. Lines 17 thru 27 initialize the system in a way that will be used to illustrate a point. Lines 28 thru 33 do more initializing, and ask for the width of your display. Lines 34 thru 36 create a blank string in case your BASIC doesn't set dimensioned strings initially to spaces.*

*Lines 37 thru 46 define the various functions of the control system. If your BASIC can't do multiline functions, you can substitute subroutines here. The idea is to make it easy to try out different kinds of functions in the control system.*

*Lines 49 thru 91 comprise the interpreter, which accepts character strings and sets initial conditions and parameters before each run. Variables are initialized and constants are set by typing a string of the form A=m or An=m (no spaces; terminated by a carriage return). To set up the plotter, the statement is PLOT XXXXXX, where XXXXXX is one or more characters from the set P,E,R,I,O, and D, in any sequence. The plotter comes up set to plot P, E, and R. If you forget the last values of the parameters K1, K2, S1, and S2, type ? and they will be printed out. We will eventually define them.*

*The control system itself is simulated from line 95 to line 108. Entering the simulator at line 95 initializes the perceptual and output variables to values given to the interpreter. Entering at line 99 runs the simulation from the conditions left at the end of the last run. This is taken care of by the two* run *commands in the interpreter: a dot (.) means run with initialization, and a slash (/) means run without initialization. All commands require a carriage return termination.*

*The plotting subroutine goes from line 112 to line 128. Its operation deserves a note, since it was arrived at after some more normal schemes were rejected for being too slow. When the interpreter is given a string of symbols to set up the plotting, a table is set up (V(j)) in which a 1 means plot and a 0 means don't plot. When the plotter is entered, it transfers all six variables to another table, U(j). The output buffer is then cleared, and a short loop scans the V table, picking up variables from the U table when V(j)=1, and putting the symbol into the output buffer in a position corresponding to the value of the variable. Then the output buffer is printed out. This eliminates sorting the variables by size or printing the line as many times as there are variables. This method nicely cures the fundamental "rheumatism" of BASIC, as it is able to plot about two lines per second on my Polymorphics VTI display.*

*When two variables fall on the same spot, the variable that actually appears is the latest one in the series PERIOD. Thus far is has always been easy to figure out where a missing variable is hidden.*

*Once we have a set of variables connecting functions together, and an overall arrangement, we can treat the system by assembling it piece by piece. Let's look at the pieces we have, represented by the four statements in listing 2 from line 102 to 105:*

```
102  I = FNF(O) + FND(D)
103  P = FNI(I)
104  E = R – P
105  O = FNO(E)
```

*Looking at figure 5, we can see that these four statements lead us clockwise around the closed loop. I is the result of combining the outputs of the feedback and disturbance functions. It becomes the* input *to the input function, producing a value of the perceptual signal P. P is one of the inputs to the comparator, which produces the error signal E.*

*Continued on page 140*

Therefore line 104 represents the comparator without using a function; it *is* the comparator function itself: E = R – P.

The error signal drives the output of the system via the *output function*, FNO. The output of the system, therefore, depends not on the input quantity or the perceptual signal alone, but on the *difference* between the perceptual signal and the reference signal. The output function translates a signal inside the system into a quantity outside it, according to whatever rule is described by FNO. If the error signal changes sign, the output quantity also changes; in other words, we assume that output functions have no constant term. Any such constant term would have the same effect as a reference signal, creating an offset in the overall system response. Not every system can handle error signals and output quantities that go through zero and thus change sign, but the principles remain the same in the region where the system works.

Line 105 expresses the operation of the output function: O = FNO(E). This closes the loop of cause and effect since the output quantity appears in line 102 where the input to the system is calculated.

If the system functions are properly designed for the properties of the system's environment, this entire closed loop will seek an equilibrium state. Our simulator will let us look at time-varying effects, but for the most part we will be concerned with steady state relationships.

Once we have seen how time variations come into the picture, we will concentrate on variations that occur slowly enough that the system and its environment never get far from a steady state relationship. This is the whole trick in grasping how control systems work. If you allow yourself to become embroiled in the interesting details of stabilization, or interested in the limits of performance in the presence of large and rapidly changing disturbances, you may learn a lot about one control system, but you will miss the organizational features that are obvious only when the system is not being subjected to unusual stresses. We will be concerned mainly with the *normal range of operation*, the range within which this system can behave very nearly like an ideal control system. Once that mode of operation is understood, there is plenty of time to explore the limits of operation. (See "Anatomy of the Simulator" text box).

### A Wrong Approach

Let us start off by assuming that we have a simple linear system. The input function is a multiplier of 1, the comparator is already

# NEW!

WHY CUT?
WHY STRIP?
WHY SLIT?

## WHY NOT...

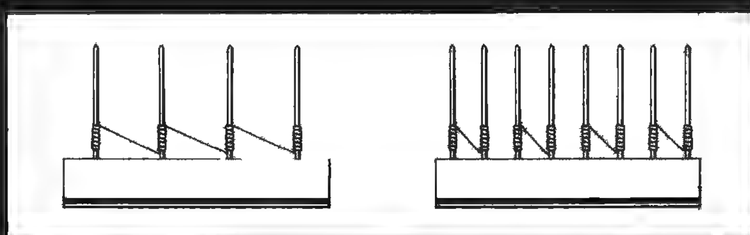
# JUST WRAP™

U.S.A
FOREIGN
PATENTS
PENDING

- AWG 30 Wire
- .025" Square Posts
- Daisy Chain or Point To Point
- No Stripping or Slitting Required
  ....JUST WRAP TM....
- Built In Cut Off
- Easy Loading of Wire
- Available Wire Colors:
  Blue, White, Red & Yellow

*$14.95

| COLOR | PART NO. | U.S. LIST PRICE |
|---|---|---|
| JUST WRAP TOOL WITH ONE 50 FT. ROLL OF WIRE | | |
| BLUE | JW-1-B | $14.95 |
| WHITE | JW-1-W | 14.95 |
| YELLOW | JW-1-Y | 14.95 |
| RED | JW-1-R | 14.95 |
| REPLACEMENT ROLL OF WIRE 50 FT. | | |
| BLUE | R-JW-B | 2.98 |
| WHITE | R-JW-W | 2.98 |
| YELLOW | R-JW-Y | 2.98 |
| RED | R-JW-R | 2.98 |



DAISY CHAIN POINT TO POINT

OK MACHINE & TOOL CORPORATION 3455 CONNER ST., BRONX, N.Y. 10475 (212) 994-6600/TELEX 125091

*MINIMUM BILLING $25.00 / AD SHIPPING CHARGE $2.00 / NEW YORK CITY / STATE RESIDENTS ADD APPLICABLE TAX.

*E is the input to the output function that produces O, the output quantity. The output quantity is the input to the feedback function, which leads us back to the start.*

*It might seem that all we have to do now is to supply some specific forms for the functions, and turn the system on to see what it will do. In a sense, this is right. If this were an analogue computation, we might even get a correct idea of how the system works. However, it is unlikely that anyone who hasn't done this before would plug in the right functions to make a digital computer give us anything more than a fairy tale. It is so important to understand this point that I have written the simulator to come up initialized in order to illustrate it.*

simple and linear, the output function is a multiplier of 2, the feedback function is a multiplier of 0.5, and the disturbance function is a multiplier of 0.8. These choices are dictated partly by the need to keep variables from falling on each other when we plot them. The simulator initializes D to 15.

Our four system equations, with these values substituted, now look like this:

$$I = 0.5xO + 0.8xD = 0.5xO + 12 \qquad (1)$$
$$P = I \qquad (2)$$
$$E = R - P \qquad (3)$$
$$O = 2xE \qquad (4)$$

This system of equations is iterated during a simulation of behavior.

The above is a pretty simple system of equations. So why can't we just solve it algebraically and skip the rest? I suggest, in fact, that you do solve it (by successive substitutions). Solve for the value of the perceptual signal in terms of R and D. You'll get P=I=(R – 0.8 x D)/2.

Ready for a shock? Your computer can't come up with that solution! Let's fire up the BASIC simulator, which is initialized according to equations 1 thru 4 above, and plot I, D, and O. Type RUN, and answer the question with a reply that tells the width of your display. After the colon prompt appears, type in the following:

```
:.
```

I trust nobody had trouble with that.

The dot says "do a plotting run after initializing the variables." A slash (/) would say "do the run from where the last run left off." The result can be found in figure 6.

The disturbance is set to a steady +15 units, and the reference signal is initialized to 0. According to the algebraic solution above, the input signal should be a steady 0.8 x 15/2, or 6 units, to the right of center (dots indicate center when nothing is there). It is clear that something else happened. The whole system is in a state of endless *oscillation*. (When variables fall on top of each other in a plot, the visible one is the latest in the sequence PERIOD.)

Nature has a way of slapping your wrist when you forget something important. Our wrist has just been slapped. Naturally we do not get the same result that algebra gives: the algebraic solution comes from treating all of those relationships *simultaneously*. Our computer program is treating them *one at a time*. The algebra says that if one variable changes, they all change. The computer, being a purely sequential machine, thinks it can change one variable without changing the others. If the physical system being modeled is of that nature — if it, too, is a sequential state machine — then the computer will produce a correct picture of behavior. But, if the system being modeled works in terms of continuous variables, *even in part*, the computer will turn it into a sequential-state machine and analyze *that* kind of system instead of the one we actually have. That is what has happened here. We forgot to tell the computer that these variables can't change as fast as the computer can compute.

```
:.

                                      O              I   D
         O                            .              I   D
                                      O                  D
         O                            .              I   D
                                      O                  D
         O                            .              I   D
                                      O                  D
         O                            .              I   D
                                      O                  D
         O                            .              I   D
                                      O                  D
         O                            .              I   D
                                      O                  D
         O                            .              I   D
                                      O                  D
         O                            .              I   D                   :
```

*Figure 6: The initial plot generated by the BASIC simulator. Disturbance is set to 15 units and the reference signal is initialized to 0. The system is in a state of oscillation.*

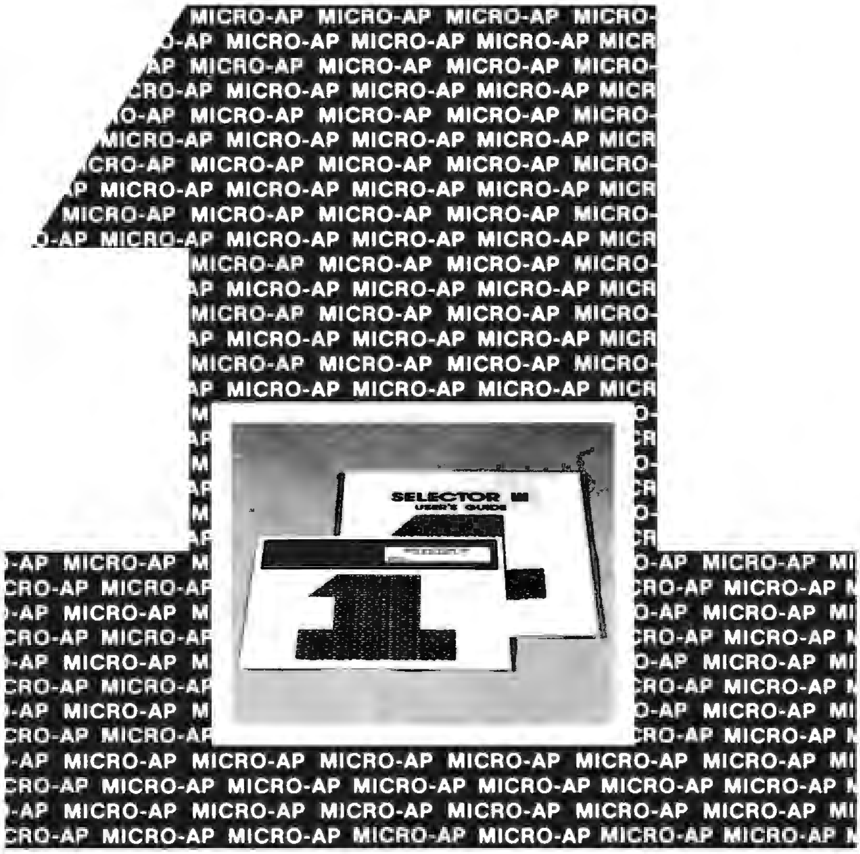
## A More Accurate Approach

In order to make this simulated system behave the way the algebra says it should, we have to slow down changes in one or more variables to take account of the fact that we are dealing with real, physical variables

# ONE PACKAGE DOES IT ALL

## Includes these Application Programs . . .

**Sales Activity, Inventory, Payables, Receivables, Check/Expense Register, Library Functions, Mailing Labels, Appointments, Client/Patient Records**



**RANDOM, MULTI-KEY RECORD RETRIEVAL under CP/M, CDOS, IMDOS, ADOS . . .**

**SELECTOR III ALLOWS INSTANT RECALL OF ANY RECORD USING ANY INFORMATION ITEM IN THE RECORD.** That statement deserves re-reading, because that ability makes SELECTOR III the most powerful Date Base Management System in microcomputers today!

With **SELECTOR III** you can...

- define a record format, assign retrieval keys, and begin entering data in minutes.
- create sorted pointers to records matching your specif or range of requirements.
- automatically generate reports with control-break summaries and unlimited variety.
- bring an application online in hours instead of months.

**SELECTOR III** comes complete with eight application programs that perform the tasks listed at top of page. And, since it's distributed in source code form, you can easily add subroutines to do specific computations or file updates.

**SELECTOR III** runs under CBASIC Vers. 1 or 2, and is priced at $295. SELECTOR III-C2 is dedicated to Vers. 2 only, runs about twice as fast, and costs $345.

Both systems are available in a variety of CP/M, diskette size and density formats including IBM 8"; North Star; Micropolis; TRS-80; Processor Tech Helios II; Altair; iCOM; Dynabyte; Imsai; and others.

Available from computer stores nationwide:

**LIFEBOAT Associates**
2248 Broadway, Suite 34,
New York, N.Y. 10024 • (212) 580-0082

Or order direct from

**MICRO-AP**
9807 Davona Drive, San Ramon, CA 94583
(415) 828-6697

Circle 213 on inquiry card.

```
:S1=0.5

:S2=0.2

:.

                                O           I   D
                              O .           I   D
                           O    .          I    D
                        O       .        I      D
                      O         .       I       D
                     O          .      I        D
                    O           .      I        D
                    O           .     I         D
                    O           .     I         D
                    O           .     I         D
                    O           .     I         D
                    O           .     I         D
                    O           .     I         D
                    O           .     I         D
                    O           .     I         D
                    O           .     I         D               :
```

*Figure 7: The slowing factors have been changed. S1 equals 0.5 and S2 is 0.2. We now have a much smoother curve.*

and not abstract numbers. The simulator does this in the input and output functions, lines 37 thru 40 (input) and 41 thru 44 (output). We will be basically dealing with a linear system in which both the input and output functions are constants of proportionality. As you can see from listing 2, however, there's a little more to it than that.

Consider line 42: O = O + S2* (K2*E – O). The O on the left side is the new value of that quantity after this program step has been executed. On the right side, O indicates the last value of the output quantity. We recognize K2*E as a calculation of the output quantity as if it were simply proportional to the error signal, E. The expression in parentheses, therefore, is the *difference* between this calculated new value and the old value of O. This is how much the output quantity would change if it could change instantly.

This calculated amount of change is multiplied by S2, a *slowing factor*, and the result is added to the old value of O. We calculate the amount of change that an instantly reacting system would produce, but allow only a fraction S2 of it to occur on any one iteration. S2 is a positive number between zero and one. We've put a low-pass filter into the output function, without affecting the *steady state* proportionality constant.

The same thing is done for the input function. A slowing factor S1, between zero and one, acts to slow P down. We need only one slowing factor to make this simulator behave realistically, but there is provision for two, so that you can explore the effect of having two if you wish. In all the plots to follow, we'll use a modest slowing factor of S1=0.5 in the input function, and essentially all of the required slowing in the output function. Once you get the hang of this you can put slowing factors into *any* of the functions.

The simulator is initialized with S1 and S2 set to 1, which reduces O + S2x (K2xE – O) to O + K2xE O or just K2xE (no slowing at all). The same is done for the input function. Let's set them to other values and see what happens. The values of S1 and S2 can be set by typing S1=n or S2=n and a carriage return:

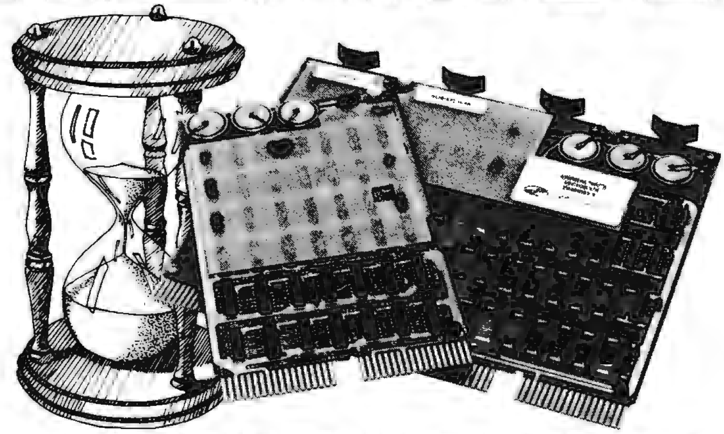
```
:S1=0.5
:S2=0.2
: .          (run with initialization)
```

Suddenly we see nice, smooth relationships (figure 7). If you measure, you'll see that the input signal, I, ends up just six units to the right; the same solution given by the algebraic approach.

Does this mean we can just use algebra to analyze a control system? Not at all. We won't delve into this, but the algebraic solutions are valid *only* if the differential equations which really describe the system have steady state solutions. Then the algebraic solutions *are* the steady state solutions. In our simulator, we see all the time variations that lead toward the steady state, and the algebra says nothing about these. By putting the slowing factors into our calculations we have caused this system to seek a steady state. Therefore, it is the stability of the system that tells us we can use algebra, not the other way around. Predicting stability can become a messy process. We fiddle around with slowing factors until we *get* stability, which is more or less how Nature does it anyway.

We have now established the fact that using natural logic and following causes and effects around the closed loop as a sequence of events will lead to a wrong prediction of control system behavior. This immediately eliminates three-quarters of what biologists, psychologists, neurologists, and even cyberneticians have published about control theory and behavior. We are just beginning to see that one must view all the variables in a control system as changing together, not one at a time. This is what I mean by retraining the in-

# TIME after TIME



# BATTERY SUPPORTED CALENDAR CLOCKS

## PDP-11*

TCU-100 • $495

- Provides month, day, hour, minute and second.
- Can interrupt on date/time, or periodic intervals.

TCU-150 • $430

- Provides year, month, day, hour, minute and second.
- Automatic leap year.
- Patches for RSX-11M, RT-11 FB/SJ VO2, VO3 and UNIX.

## LSI-11/2*

TCU-50D • $295

- Provides month, day, hour, minute and second.
- Dual size board.
- Patches for RT-11 SJ/FB VO2, VO3B.

## Lockheed SUE

TCU-200 • $550

- Provides year, month, day, hour, minute, second and milli-second.
- Interval interrupts between 1/1024 seconds and 64 seconds.

## Computer Automation (Naked Mini)

TCU-310 • $385

- Provides year, month, day, hour, minute and second.

## Multi-Bus**

TCU-410 • $325

- Provides year, month, day, hour, minute and second.
- SBC/BLC compatible.

## HP 2100

TCU-2100 • $395

- Correct time restored after power failure.
- Compatible with the HP TBG card.

## Serial Clock (RS 232 or 20 mA)

SLC-1 • $575

- Connects between any terminal and host computer.
- Provides date, time and more!

All Digital Pathways TCUs have on board NICAD batteries to maintain time and date during power down. Timing is provided by a crystal controlled oscillator. Prices are U.S. domestic single piece. Quantity discounts available.

*For more information on these products, contact:*
*Digital Pathways Inc.*
*4151 Middlefield Road*
*Palo Alto, CA 94306*
*Phone: (415) 493-5544*

**Trademark of Digital Equipment Corporation*

***Trademark of Intel Corporation*

# DIGITAL PATHWAYS

Circle 86 on inquiry card.

tuition. Cartesian concepts of cause and effect, and Newtonian physics, have trained us to think along directed lines. What we need to do to understand control systems is to learn how to think in circles.

### Using the Simulator

*The simulator is run from the keyboard, using commands that tell it which variables to plot and what values of variables and parameters to start with. The instructions can be given one at a time, terminated by carriage returns, or they can be given in a continuous string with commands separated by commas. The latter is useful for altering parameters in the middle of a plot in order to see their effects.*

*The only time a space is permitted in a command or string of commands is when it is separating the word PLOT from the string of variable symbols to be plotted.*

*In order to tell the simulator what variables to plot, type:*

*PLOT XXXXXX*

*where XXXXXX means a string of 1 to 6 symbols from the set PERIOD. The order of the symbols makes no difference. When two or more symbols land on the same plot, the one that you see is the latest in the series PERIOD, regardless of the order in which they were given.*

*To start a plotting run, type a period followed by a carriage return or comma if initialization is to occur first, and type a slash (/) if the run is to start from the conditions at the end of the previous run. Initializing creates one extra line of plot showing the initial conditions.*

*The parameters and variables that can be set are as follows:*

| | |
|---|---|
| *L* | *Number of lines to be plotted in any plotting run.* |
| *K1* | *Steady state proportionality factor of the input function.* |
| *S1* | *Slowing factor for the input function; positive and between 0 and 1.* |
| *K2* | *Steady state proportionality factor of the output function.* |
| *S2* | *Slowing factor for the output function; positive and between 0 and 1.* |
| *O* | *Initial value of output quantity.* |
| *P* | *Initial value of perceptual signal.* |
| *R* | *Setting of reference signal.* |
| *D* | *Magnitude of disturbing quantity.* |

**Examples:** *(colon is prompt from computer. Always terminate a string with a carriage return).*

| | |
|---|---|
| *Set L to 16* | *:L=16* |
| *Set D to 0, run without initializing* | *:D=0,/ or*<br>*:D=0*<br>*:/* |
| *Set D to 0, plot 2 points after initializing, set D to 10, plot 13 points from previous conditions. Plot P,E, and R* | *:PLOT PER,D=0,L=2,.,D=10,L=13,/* |

*The program is written so that after a plot is completely done (a complete string has been interpreted), the prompt character appears to the right without a carriage return. That allows a 16 point plot to be shown on a 16 line video display screen without the final carriage return bumping the first line off the screen. If you want your next string to start at the left, just hit a carriage return.*

*To find out the values of K1, K2, S1, and S2 when you forget them, type "?" followed by carriage return and they will be printed.*

### Properties of a Control System

Figure 8 shows the control system and its environment as we will be dealing with it from now on. Let's start with some definitions:

> *Loop Gain* means the product of all the steady state factors encountered in one trip around the closed loop, counting the comparator as a factor of −1. In the initial setup, K1 was 1, K2 was 2, and the feedback function FNF was a multiplier of +0.5, so the loop gain was −1. The sign of the loop gain is the sign of the feedback; we have (and will continue to have) negative feedback.
>
> *Error Sensitivity* is the factor K2, the steady state proportionality factor in the output function FNO. This number expresses how much output will be generated by a given amount of error signal.
>
> *Input Sensitivity* is the factor K1, the steady state proportionality factor in the input function FNI. This number expresses how much perceptual signal will be generated by a given amount of input quantity.

We are going to perform a series of experiments with this control system in order to arrive at some useful rules of thumb for thinking about how control systems work. These rules are approximations, but by doing the experiments and seeing how good the approximations are, you will learn to think precisely about control phenomena, even when using approximate language.

We will set the system parameters to give a loop gain of −10. As a way of summarizing where we are (refer to figure 8), the commands are given one at a time with annotations:

| | |
|---|---|
| :K1=1 | Input sensitivity = 1. |
| :K2=20 | Error sensitivity = 20. |
| :S1=0.5 | Input slowing factor = 0.5. |
| :S2=0.07 | Output slowing factor = 0.07. |
| :R=0 | Reference signal = 0. |
| :O=0 | Output initialization = 0. |
| :P=0 | Perception initialization = 0. |
| :D=0 | Disturbance = 0. |

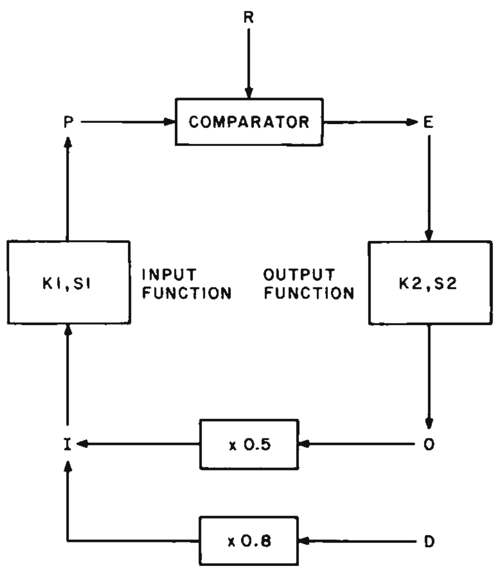



*Figure 8: Adjustable parameters are K1 (input sensitivity), S1 (input slowing factor), K2 (output sensitivity), and S2 (output slowing factor). P and O can be initialized to any starting value (normally zero). R and D can be set, and remain the same during a run. The value of the feedback function is set at 0.5, the value of the disturbance function at 0.8.*

Type those commands, and the system is now set up in a "home base" condition. Remembering that the comparator is equivalent to the factor of −1 and the feedback function is permanently set to be a factor of +0.5, this combination of parameters gives a loop gain of 1 x (−1) x 20 x 0.5 = −10.

There are two fundamental rules of thumb: a control system keeps its perceptual signal matching its reference signal, and the output of a control system cancels the effects of disturbances on the input quantity. We will take these up in order.

*Rule 1: P = R*

We're looking at the system with no disturbance acting (D=0). If you want to be sure that everything stays at zero, type PLOT PERIOD. followed by a carriage return. You will see a row of Ds, D being the last symbol in the sequence PERIOD and hence the only one visible when all variables are at zero.

Now we will plot just the reference signal and the perceptual signal. The first two points will be done with the initial conditions set up above. The reference signal will then be set to +25 units, and the plot will be continued for 13 more points. Since this plot will commence *with* initialization (the dot command), an extra line showing the initial conditions will be plotted first. This makes a total of 16 lines, which will fit on most video displays. Of course, if you're doing this on paper you don't have to worry about the number of points plotted. Here is the command string:

# TRS-80 SOLUTIONS!

L I = Level I — L II = Level II — D = Disk — ALL THIS AND MORE!!!

## BUSINESS

**Appointment Log** by M. Kelleher Perfect for the professional. Accepts name and address, meeting start and endings, subject matter, derives elapsed time. For Level II, 16K $9.95

**Payroll** by Stephen Hebbler Comprehensive 24 pg. manual with step-by-step instructions included in the package. Supports W2 and 941 information. D, $59.95

**Mail List I** by Michael Kelleher is the economy model of disk-based mailing list programs. Uses a single drive and handles up to 1400 names per disk, plus provisions for sorting options. 16K, D $19.95

**BIZ-80**
The Business Software People®
**Just about everything you need ... within 1 year, participants receive programming for Inventory, Accounts Receivable, Accounts Payable and General Ledger systems, plus Sales and Payroll. Complete documentation and software on diskette, $200.00**

**Mail List II** by BIZ-80 Complete mail list system for dual disk. Enter, update, merge, sort, and print mailing labels. D, 32K $99.95

**Small Business Bookkeeping** by Roger W. Robitaille, is based on the Dome Bookkeeping Journal, sold for years in stationery and discount outlets. Level II, 4K with ($22.00) or without ($15.00) Dome journal.

**Inventory System II** by BIZ-80 Proper inventory management is the backbone of a profitable business, yet it's very difficult to keep current on price increases, shrinkage, low-on-stock items, profitable items versus losers, without an efficient and prompt method of surveying your inventory levels at any given time. This program can help you to achieve optimal management — it can handle up to 1,000 items on one disk; each additional disk can handle another 1,000 items. With Documentation, $150.00

**Inventory S** by Roger W. Robitaille, Sr. 240 stock items can be contained using the full 6 data areas and 2 pieces of alpha information. Level I or II,16K $25.00

**Inventory II.2** Disk based program allows for creation, maintenance and review of over 2,000 items per clean diskette. Operates under Disk BASIC,DOS 2.1 with minimum memory allocation. D, $59.95

**ST 80 - SMART TERMINAL**
Lance Micklus
**Turns your TRS-80 into a computer terminal. Features include CONTROL key, REPEAT key, ESC key, RUN key and a functioning BREAK key. Lets you list incoming data on line printer. Reprogram RS-232-C switches from keyboard, making baud rate changes simple. Level II, 16K $49.94**

**Text-80** by Frank Rowlett Fully-documented text processing system for disk. Create, edit, move, delete, insert, change, print words or lines. D, 32K $59.95

**KVP Extender** by Lance Micklus Corrects keyboard bounce, upper case lock, permits use as a terminal, screen printing. On tape ($24.95) or disk ($29.95)

**8080-Z80 Conversion** by M. Kelleher Permits you to enter 8080 codings and returns the Z80 equivalent. L II, 16K $15.00

**Basic Statistics** by Steve Reisser Pearson product-movement correlation coefficient, chi-square, Fisher T-test, sample analysis of variance, Z-scores and standard scores, with a random number generator built in to simulate data. L II, 16K $20.00

**Renumber** by Lance Micklus Complete user control over which lines are renumbered, and how, including all GOSUB's and GOTO's. Specity 4, 16, 32, or 48 K version when ordering. **Operates in Disk mode.** L II, 4 through 48K, $15.00 **Source Listing**, $20.00 **All 4 versions on disk**, $25.00

**NEWDOS**
Apparat
**DISK ERROR SOLVED! Stop blaming your drive, fix your DOS with NEWDOS: an enhanced disk-operating system capable of correcting over 70 errors in TRSDOS 2.1 to improve reliability, end key bounce, enable DOS commands to be called from BASIC and much more! Available NOW for 16K systems with a minimum of 1 disk drive. $49.95**

## ACTION GAMES

**Slalom** by Denslo Hamlin Choose between Slalom, Giant Slalom and Downhill. Level II, 16K $7.95

**X-Wing Fighter** by Rev. George Blank Put yourself into the cockpit of this fighter. Extensive use of INKEY function puts all ship controls at your fingertips without hitting ENTER key. Long range sensors warn of approaching aircraft prior to visual contact. Level II, 16K $7.95

**Air Raid** by Small System Software High speed machine language program with large and small aircraft flying at different altitudes. Ground-based missile launcher aimed and fired from keyboard. Planes explode when hit, cause damage to nearby aircraft. Score tallied for hits or misses. Level I or II, 4K $14.95

**Batter Up** by David Bohlke Level II, 16K $5.95

**Ten Pin** by Frank Rowlette A game of coordination, the scoring is true to the rules of the sport. Level II, 16K $7.95

**ADVENTURES**
Scott Adams
**Feel as if you're manipulating HAL from 2001 when you play these games. Hardly any rules, finding out is part of the fun. Two adventures on 32K disk, $24.95 Tape, one adventure on each tape - pirate or land - Level II, 16K $14.95**

**DOG STAR ADVENTURE**
Lance Micklus
**You're trapped aboard an enemy battlestar ... can you find the gold, rescue the princess, discover the plans and safely escape? Level II, 16K $9.95**

**Amazin' Mazes** by Robert Wallace Ever-changing maze situation Level II, 16K $7.95

**Sink 'UM** by Rev. George Blank L II, 4K $4.95

**Breakaway** by Lance Micklus Level I or II, 4K $4.95

**Treasure Hunt** by Lance Micklus Explore caves in search of twenty hidden treasures. L I or II, 16K $7.95

**Kamikaze** by Russell Starkey Command your ship against attacking suicide planes. Machine language graphics make this fast and fun! L II, 16K $7.95

## MISCELLANEOUS

**Diskettes** Dysan 104/1 Box of five, $24.95 + $1.00 shipping Verbatim, box of ten, $34.95 + $1.00 shipping/handling

**Z80 Instruction Handbook** by Scelbi Publ. $4.95

**The BASIC Handbook** by Dr. David A. Lien $14.95 + $1.00 shipping/handling

## SIMULATIONS

**3-D Tic Tac Toe** by Scott Adams Three skill levels — author warns you to practice before tackling computer's third skill level. L I or II, 16K $7.95

**Star Trek III.3** by Lance Micklus One of the most advanced Star Trek games ever written. Level II, 16K $14.95

**End Zone** by Roger W. Robitaille, Sr. Authentic football simulation, right down to the 2-minute warning. Level I or II, 16K $7.95

**Cribbage** by Roger W. Robitaille, Sr. You versus the computer cribbage played by standard rules. Level I or II, 16K $7.95

**Bridge Challenger** by George Duisman You and the dummy play 4-person contract bridge against the computer. Level II, 16K $14.95

**'Round the Horn** by Rev. George Blank You're the captain of a clipper ship racing from New York to San Francisco. Level II, 16K $9.95

**Concentration** by Lance Micklus One of the most popular television games L I or II, 16K $7.95

**Safari** by David Bohlke You're in the running for a film contract at a major Hollywood studio. To qualify, you must photograph the most wild animals in their natural habitat. Level II, 16K $7.95

**Pork Barrel** by Rev. George Blank Places you in the shoes of an aspiring Congressman. L II, 16K $9.95

**Backgammon** by Scott Adams Level II, 16K $7.95

**Chess Companion** by M. Kelleher Combines chess clock features with ability to record your moves while action is fast and furious. Level II, 16K $7.95

**Sargon Chess** by Dan & Kathe Spracklen Winner of the 1978 San Jose Microcomputer Chess Tournament Level II, 16K $19.95

**Mastermind II.2** by Lance Micklus Lets you and the computer take turns making and breaking codes. Level II, 16K $7.95

## PERSONAL

**RPN Calculator** by Russell Starkey A self-documenting calculator program. Uses Reverse Polish Notation with 4-level stack, 100 memories, scientific functions. Level II, 16K $9.95

**Home Financial Management** by M. Kelleher Turns your computer into a personal financial advisor. Level II, 16K $9.95

**Tarot** by Frank B. Rowlett, Jr. Probably the best future-gazing type program ever written.Try it — you'll like it! Level I or II, 16K $9.95

**Ham Radio** by M. Kelleher Amateur frequency Allocations, ID Timer, Q-signal File, Amateur Log Routine, Propagation forecasting. L II, 16K $9.95 **Special Disk-enhanced version, 32K $24.95**

**Educator Assistant** by Steve Reisser Five programs of value to educators. Compute percentage, individual student averages, class averages, standard test scores, final grades. L II, 16K $9.95 D, $14.95

**Electronic Assistant** by John Adamson A group of 8 subprograms designed to solve problems such as tuned circuits and active and passive filters. L II, 16K $9.95

**Personal Finance** by Lance Micklus 33 different budgets can be easily adapted by user to fit his individual needs. A 2-part program, entry and search. Level II, 16K $9.95

**Advance Personal Finance** by Lance Micklus Same as above with advanced analysis routine. **Supports Disk Files** D, 32K $19.95

**SOFTSIDE Your BASIC software magazine**
Regular 1 year subscription - 12 issues - $15.00

**PROG/80 For the serious programmer, from beginner to professional** 1 year regular subscription - 4 issues - $10.00

# T$E 1·603·673·5144

## TRS-80 Software Exchange

master charge — VISA

17 Briar Cliff Drive Milford, New Hampshire 03055

Circle 374 on inquiry card.

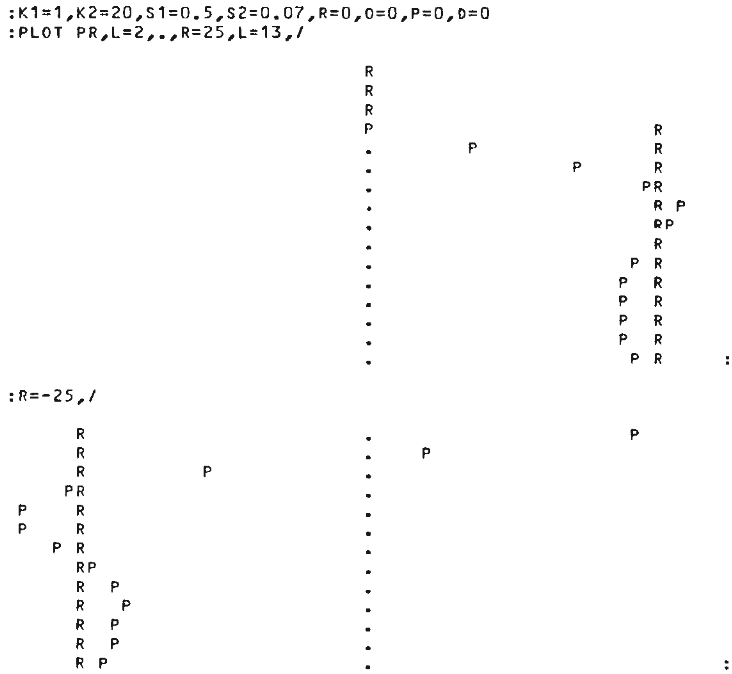



*Figure 9: The values of variables are listed in this plot. The disturbance value is changed from +25 to −25.*

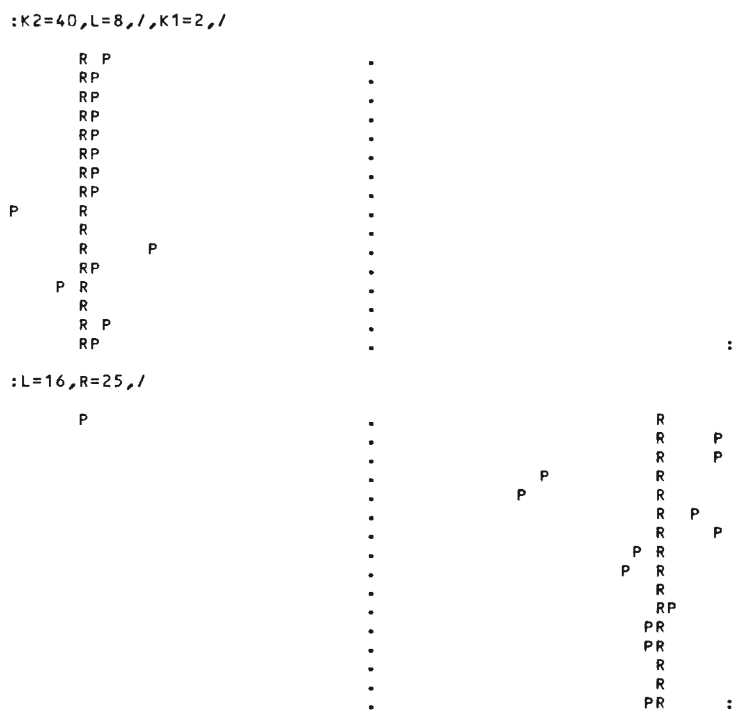



*Figure 10: Change of gain during plot. After 8th line, gain goes from 20 to 40. Reference signal is changed to check operation.*

:PLOT PR,L=2,.,R=25,L=13,/

Before discussing this, let's do another run of 13 points (figure 9), setting the reference signal to −25 units and continuing without initialization (the slash command,/):

:R=−25,/

It is clear that the perceptual signal comes to a steady state quite close to the magnitude of the reference signal, whatever the reference signal may be. The question is, how critically does this tracking effect depend on the input sensitivity and error sensitivity?

Let's leave the reference signal at −25 and do a run in which the error sensitivity is doubled at the start, and the input sensitivity is doubled halfway through the run. We will start from the previous conditions. The loop gain will now be −40 instead of −10.

:K2=40,L=8,/,K1=2,/

To insure that everything is working correctly, let's flip the reference signal to +25 units (figure 10):

:L=16,R=25,/

While there is an effect on the *way* the tracking takes place, the only effect of these rather drastic changes in input and error sensitivity is to make the tracking a little better. What about a *decrease* in these parameters?

:L=16,K1=0.5,K2=10,/,R=25,/
(Loop gain now 2.5)

Figure 11 shows that the approximation P=R isn't very accurate any more. For loop gains smaller in magnitude than about 10 (negative), the approximation begins to lose accuracy.

You will notice that doubling the error sensitivity, which doubles the amount of output generated by a given error, does *not* double the amount of output that actually occurs. Far from it. When, for any reason, the loop gain goes up, the steady state error simply gets smaller, assuming that the system remains stable. This fact does violence to the popular idea that the brain commands muscles to produce behavior. If that were the case, doubling the sensitivity of a muscle to the nerve signals reaching it ought to produce twice as much muscle tension. Nothing of the sort happens, unless you've lopped off the rest of the nervous system, particularly the feedback paths.

As long as the loop gain is sufficiently large and negative (−10 or more negative will do for a number), a stable control system will match its perceptual signal nearly to its reference signal, regardless of the reference setting. We are ignoring, of course, transient effects.

All of this was done with the disturbance set to zero. Now let us set the reference signal to zero, and check the second fundamental rule of thumb.

*Rule 2: (delta 0) = −(delta D)*

This rule requires some interpretation. It says, for the sake of brevity, that (with the reference signal constant) a change in the output quantity is equal and opposite to (the minus sign) a change in the disturbing quantity. Generally, the input and disturbing quantities will affect the input quantity through different physical paths. In our model, the output quantity acts through a multiplier of 0.5, and the disturbance through a multiplier of 0.8. The rule has to be interpreted to mean that the *effects* of the changes *on the input quantity* are equal and opposite. We will see this demonstrated.

We will now plot the output quantity, O, the disturbing quantity, D, and the input quantity, I (to make the above clear). The reference signal could be left where it is, but to avoid confusion let's set it to zero for this set of plots. The loop gain is set to −10.

```
:PLOT OID, R=0,K1=1,K2=20,L=1,D=0,
.,L=15,D=15,/
```

Let this plot run out, then:

```
:D=−15,/
```

There is some lurching back and forth in figure 12, but in the steady state the behavior of the input quantity shows that the effect of the disturbance is essentially cancelled by the final effect of the output quantity.

If you did some measuring on the plot, you would find that the final value of the output quantity is very close to 8/5 of the value of the disturbing quantity. This follows from three facts: the input quantity ends up nearly at zero; one unit of output has 0.5 unit of effect on the input quantity; one unit of disturbance has 0.8 unit of effect on the input quantity. This is the kind of reasoning that helps in understanding how a control system works.

The primary observation about a control system is always the existence of an input quantity which is stabilized against disturbances by variations in the output quantity. If the input quantity is held essentially

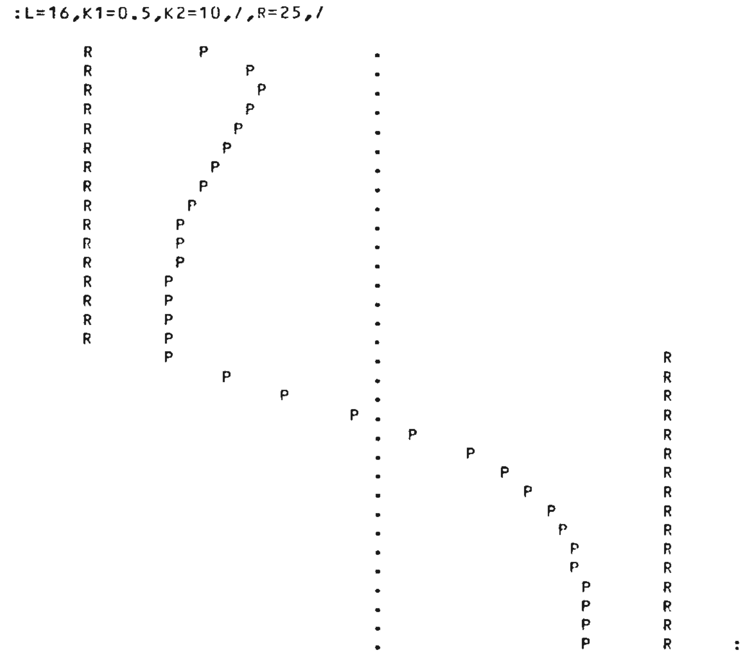



*Figure 11: The simulation parameters have been changed to produce a gain of 2.5. Notice that the approximation P=R is now inaccurate.*

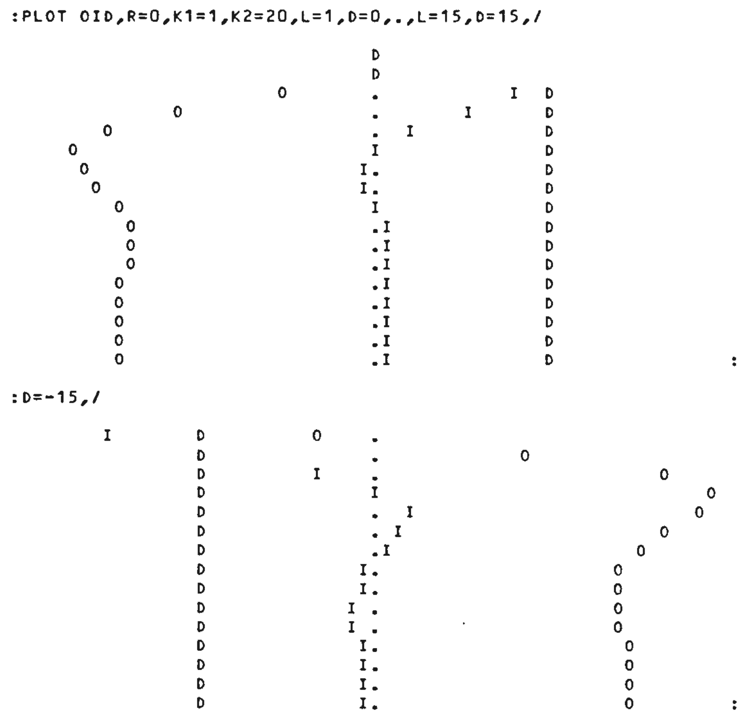



*Figure 12: The reference signal has been set to zero. This plot shows us the input quantity, the output quantity and the disturbance signal for D=+15 and then D=−15.*

```
:PLOT RIOD,D=0,R=0,L=1,.,R=12,L=15,/,D=15,/
                                D
                                D
                                D            R    O
                                D        I   R              O
                                D            RI               O
                                D            R  I            O
                                D            R I          O
                                D            I         O
                                D           IR        O
                                D          I R        O
                                D          I R        O
                                D          I R        O
                                D           IR         O
                                D           IR         O
                                D           IR         O
                                D           IR         O
                                D           IR         O
                                .            RO D       I
                                .   O        R  D   I
                              O .            R ID
                            O   .           IR  D
                             O  .          I R  D
                              O .          I R  D
                                O           IR  D
                                O            I  D
                                .O           I  D
                                O            I  D
                                O            I  D
                                O            I  D
                                O            I  D
                                O            I  D
                                O            I  D               :
```

*Figure 13: A look at different reference signal effects. As explained in the text, the disturbance signal has made the perceptual signal match the reference signal.*

constant (in the steady state), then one can *deduce* the relationship between disturbances and the system's output quantity simply from observing the properties of the system's environment. On inspection, an external observer can see both the feedback function and the disturbance function, here multipliers of 0.5 and 0.8 respectively. For any given disturbance, the effect on the input quantity for a constant output quantity can be calculated on purely physical grounds. Since the input quantity remains undisturbed in the steady state, one can then look at the connection between the output quantity and the input quantity, and deduce how the output quantity must change to account for the fact that the input quantity *doesn't* change.

Thus, in order to predict how this system will react to any external disturbance, it is necessary only to know that the system is a control system and to look closely at its environment. The kind and amount of reaction follow from the nature of the feedback and disturbance functions which are properties of the visible environment.

Most important, as far as the life sciences are concerned, the form and amount of reaction do *not* depend on any property of the control system; not enough to make any difference. Therefore, when you apply a stimulus and see a response, you are using the organism as a complicated analogue computer in order to study the physics of the local environment. This is not what the life sciences have thought they were doing.

All that remains to wrap up this section is to see the effects of disturbances when the reference signal is set to different values. This will lead to the definition of a useful technical term: the *reference level of the input quantity* (see figure 13):

```
:PLOT RIOD,D=0,R=0,L=1,.,R=12,
      L=15,/,D=15,/
```

If you have a 16 line video display this will scroll past you, losing the early parts, but no matter. The first event is that the reference signal is set to 12, and the input quantity moves essentially to +12. The output quantity goes to +24 in order to accomplish this. Then the disturbing quantity goes to +15, which has the exact effect on the input quantity that +24 units of output have. As a result, the output quantity drops to zero – *exactly* zero, if you look at the numbers.

In effect, the disturbance, by itself, has enough effect to make the perceptual signal match the reference signal. Looking at figure 8, you can see that this would mean a zero error signal and no drive to the output function. So, whenever the output drops to zero, we know that the perceptual signal is matching the reference signal, even if we can't see it.

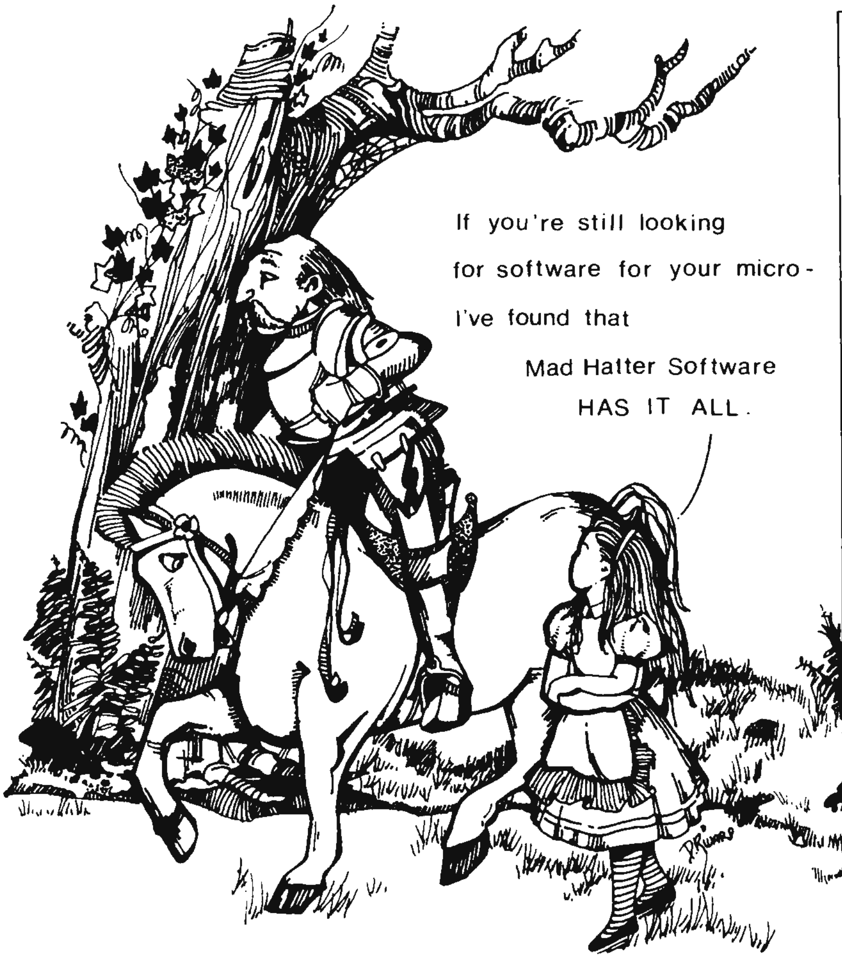
In our model right now, the input sensitivity is 1, so the perceptual signal is numerically equal to the input quantity. That's a coincidence, since the units are different: physical units outside, impulses per second inside. Even if K1 wasn't 1, the output would still drop to zero when P = R. Thus, we can give a special name to the particular value of input quantity (however created) that brings the error signal, and hence the output quantity, to zero: the *reference level* of the input quantity. The reference signal clearly determines what this reference level will be, but so does the form of the input function.

### Main Points Reviewed

All of this is supposed to have established two principal ideas. The first is that control systems control what they sense, not what they do. The second is that control systems act on the outside world only in order to protect a controlled perception against disturbance.

As we have demonstrated these principles, we have established some other odd facts. We have found that the main effect

# MAD HATTER SOFTWARE




## Business & Application Software Available

- Client Billing
- A/R • A/P
- Gen. Ledger
- Inventory
- Payroll
- Mailing List
- File Handling

for

TRS-80 APPLE & OTHERS

Call or Write for Details

| PROGRAM | DESCRIPTION | TRS-80 L1/2 | TRS-80 L2 | PET | APPLE | |
|---|---|---|---|---|---|---|
| GALACTIC BLOCKADE RUNNER | • AN EXCITING SPACE WAR GAME WITH GRAPHICS | ● | | ● | ● | $ 9.95 |
| SCI-FI GAME SAMPLER | • 3 GAMES—LUNAR LANDER—STAR MONSTER—SPACE BATTLE | ● | | ● | ● | $ 7.95 |
| R/T LUNAR LANDER | • A REAL TIME LUNAR LANDER WITH GRAPHICS | | ● | | | $ 7.95 |
| MICRO-TEXT EDITOR | • FORMAT TEXT—SAVE & LOAD TO TAPE—OUTPUT TO PRINTER | | ● | | | $ 9.95 |
| OTHELLO III | • A STRATEGY BOARD GAME—PLAY AGAINST COMPUTER OR OTHERS | ● | | ● | ● | $ 7.95 |
| AIR RAID | • A REAL TIME, ARCADE TYPE SHOOTING GAME IN MACH. LANG. | ● | | | | $ 14.95 |
| MICRO-CHESS | • PLAY CHESS WITH YOUR COMPUTER—VARIOUS LEVELS OF DIFF. | ● | | ● | ● | $ 19.95 |
| BRIDGE CHALLENGER | • DON'T WAIT FOR OTHERS TO PLAY—YOUR COMPUTER'S READY | ● | | ● | ● | $ 14.95 |
| APPLE 21 | • BLACKJACK WITH HIRES GRAPHICS | | | | ● | $ 9.95 |
| STAR WARS/SPACE MAZE | • SCI-FI GAMES FOR THE APPLE | | | | ● | $ 12.95 |
| RENUMBER | • RENUMBER YOUR BASIC PROGRAMS—RENUMBERS EVERYTHING | ● | | | ● | $ 14.95 |
| DISK RENUMBER | • SAME AS ABOVE, BUT ON DISK | ● | | | | $ 19.95 |
| PILOT 2.0 | • THE EDUCATIONAL LANGUAGE, IN MACH. LANG.—INC. EDITOR | | ● | | | $ 14.95 |
| PILOT 3.0 | • THE DISK VERSION OF THE ABOVE | | ● | | | $ 24.95 |
| APPLE TALKER | • YOUR APPLE SPEAKS! NO NEW HARDWARE REQUIRED | | | | ● | $ 15.95 |
| APPLE LIS'NER | • SPEECH RECOGNITION THE EASY WAY—GREAT WITH THE TALKER | | | | ● | $ 19.95 |
| TIC-TAC-TALKER | • TIC-TAC-TOE USING SPEECH SYNTHESIS AND RECOGNITION | | | | ● | $ 19.95 |
| SYSCOP | • MAKE BACKUP TRS-80 SYSTEM TAPES THE EASY WAY | | ● | | | $ 9.95 |
| ANDROID NIM-2 | • GAME OF NIM WITH ANIMATED ROBOTS AND SOUND | | ● | | | $ 14.95 |
| SNAKE EGG | • A BETTING GAME WITH ANIMATED SNAKES AND SOUND | | ● | | | $ 14.95 |
| LIFE 2 | • 100 GEN. PER MIN. LIFE & BATTLE OF LIFE W/ANIMATION & SOUND | | ● | | | $ 14.95 |
| DCV-1 | • PUT SYSTEM TAPES ON DISK EVEN IF IN SAME MEM AS DOS | | ● | | | $ 9.95 |
| MUSIC MASTER | • ENTER SHEET MUSIC—THE TRS-80 THEN COMPILES & PLAYS IT | | ● | | | $ 14.95 |
| DISK MUSIC MASTER | • SAME AS ABOVE BUT ON DISK W/MANY SELECTIONS | | ● | | | $ 24.95 |
| TRS-80 CP/M | • OPENS UP THE WHOLE WORLD OF CP/M SOFTWARE TO THE TRS-80 | | ● | | | $150.00 |

**10% OFF IF YOU ORDER 3 SOFTWARE PACKAGES OR MORE SEND FOR FREE CATALOG—GIVE TYPE OF COMPUTER**

TO ORDER BY PHONE OR FOR DEALER INFO—CALL—(617) 682-8131
ADD 75c SHIPPING & HANDLING • MASS. RESIDENTS ADD 5% SALES TAX
**MAD HATTER SOFTWARE** • 900b SALEM RD • DRACUT, MA 01826

master charge VISA

**AVAILABLE FROM THESE FINE MICRO COMPUTER DEALERS**

CAPITOL COMPUTER SYSTEMS
3396 EL CAMINO AVE
SACRAMENTO CA 95821

TRS-80 SOFTWARE EXCHANGE
17 BRIARCLIFF DR
MILFORD NH 03055

OP AMP TECH BOOKS
1033 N. SYCAMORE AVE
LOS ANGLES CA 90038

COMPUTER CABLEVISION, INC
2617 42ND ST NW #2
WASHINGTON DC 20007

KENNEDY SYSTEMS
74 BROAD ST
LYNDONVILLE VT 05851

ADVANCED COMPUTER PRODUCTS
1310B E. EDINGER
SANTA ANA CA 92705

HOBBY WORLD ELECTRONICS
19355 BUSINESS CENTER DR #6
NORTHRIDGE CA 91324

AUGUST AUTOMATION
28 MILK ST
WESTBORO MA 01581

THE CPU SHOP
39 PLEASANT ST
CHARLESTOWN MA 02129

J&J ELECTRONICS LTD
28 COLLINGTON AVE
BEXHILL-ON-SEA, E. SUSSEX, ENG.

COMPUTER VILLAGE
931 SW 87TH AVE.
MIAMI FL 33174

L. C. SALES
100 HINCHEY AVE #705
OTTAWA, ONT., CAN. K1Y4L9

Circle 202 on inquiry card.

of negative feedback in a control loop is to diminish the effects which disturbances would otherwise have on the system's input quantity. While we have had only one disturbance at our disposal, it should be clear that the number or the causes of disturbances make no difference. If ten different disturbances were acting at once, they could only end up increasing or decreasing the value of the controlled input quantity. Since the system maintains control by acting directly on the input quantity, and not by acting to oppose the cause of the disturbance, the system does not have to take account of the number of causes acting, or the phenomena that are involved. It acts to oppose the *net effect* of any disturbances on the input quantity.

From the point of view of the behaving system itself, reality consists of the magnitude of one perceptual signal, because that is the only internal representation of the outside world. If the system can be said to have a purpose or intention, it must be to maintain the perceptual signal matching the reference signal. The reference signal specifies to the system what it is to sense, but not what it is to do.The output that matches perceptual and reference signals is determined by the nature of the feedback function and by the strength and direction of any disturbances that may be acting. Whatever sets the reference signal, thus effectively controlling the perceptions of this system, does not have to know anything about *how* the control system comes up with a matching perception.

What is perhaps most amazing to a person who has not previously worked with negative feedback systems is the capability that this system has to maintain quite precise control over its own perceptual signal, even if its own properties change. If its output apparatus becomes stronger or weaker, or its perceptual apparatus becomes more or less sensitive, there is scarcely any effect on the perceptual signal. As long as some *minimum* loop gain is maintained and the system does not become unstable and begin oscillating, it does not really matter how much loop gain there is, or whether most of it is in the output or the input function.

A servomechanism engineer might find this approach somewhat odd. Why all this fuss about the system's internal perceptual signal? When you build a control system for a practical use, you worry more about the external variables than internal variables, because the customer is interested in the external variables.

This is exactly the point. Living control systems are *not* interested in the external variables. They can't be. They don't know about them, except indirectly. All they know is what happens to themselves. The point of behavior is not to accomplish something for a user in the external world, but to affect the system itself. Everything that a living system knows about the outside world has to first exist in the form of perceptual signals, or some other internal effect of external events (not all organisms have nervous systems).

In part 3 we will start looking at living systems more directly, and this will become much clearer. We now know that control systems control, above all, their own internal perceptual signals. Next time we will see *why* they do that.

In the meantime you might enjoy using this simulator to do further explorations. We have looked into only a few of the questions that might be raised about control systems. The simulator can reveal far more than we have seen. For example, it is instructive to look at the effects of the disturbance strictly from the external point of view (plotting I, O, and D), and then to look at exactly the same effects from inside (plotting P, E, and R). We haven't even raised the question of what a control system looks like when it becomes unstable, how the slowing factors interact with loop gain to determine stability, or what happens when the input function, the output function, or both are nonlinear. Speaking of nonlinearity, you might try rewriting the definition of the feedback function as follows:

```
45 DEF FNF(X)=X*X*X/2048+X/2
```

and then performing some of the experiments again. Try to make the input function *logarithmic* (adding a constant to make sure you don't make the perceptual signal negatively infinite), and see how the input quantity and perceptual signal behave as the reference signal or disturbance is changed.

The main objective before the next article in this series appears is to understand how a control system controls its perceptual signal, and why an external observer, who doesn't know about the controlled input quantity, might think the disturbance acts on the system to make it respond, like a doorbell. The simulator is there to help you grasp this closed loop phenomenon. I hope it *does* help.■

# CBASIC BUSINESS SOFTWARE for MICROCOMPUTERS

Osborne & Associates is publishing its business systems in book form. These systems represent five years of development and testing by O&A programmers. Each book includes source listings, program and system documentation and user's manual.

What systems are we selling?

* PAYROLL WITH COST ACCOUNTING
* ACCOUNTS PAYABLE AND ACCOUNTS RECEIVABLE
* GENERAL LEDGER

The books sell for $15.00 each. All three are available right now with listings in Wang BASIC. New editions of each book, with listings in CBASIC-2, will be published during the summer of 1979. These new editions will run on many floppy disk-based microcomputer systems.

Don't know programming? Don't want to key in thousands of lines of source code? We have over 130 dealers who will sell you the CBASIC-2 programs on ready-to-run floppy disks. Contact us for the name of your nearest dealer.

CBASIC or Wang BASIC won't work? Take a look at the list of consultants who have converted O&A programs to run on many popular systems, and are waiting to hear from you.

CONSULTANTS···
COMPUTER STORES···
SOFTWARE HOUSES·······GOOD NEWS···

## * * *CP/M CBASIC* * *

You can be a dealer for the magnetic surface of O&A's CBASIC-2 software, which runs under CP/M on many floppy disk-based computer systems. We will sell the CP/M magnetic surface only to consultants, computer stores and software houses. Osborne & Associates prefers to write and sell books, not customize the programs or answer the end user's questions. The disk for each book sells for $250.00. Once you buy the floppy disk you can copy it, resell it, change it or use it. We place no restriction on the magnetic surface; we copyright only the printed word in our books. CBASIC-2 Payroll is available now. The magnetic surfaces for all three systems are scheduled to be available in the first half of 1979; call or write for the exact availability of each system.

**ALPHA MICRO:**
Peter Burke
THE BASIC BUSINESS SOFTWARE CO., INC.
P.O. Box 2032
Salt Lake City, Utah 74110
(801) 363-1199

**APPLE II:**
Jimmy Walter
A.C.E. COMPUTER SYSTEMS
2449 North Westshore
Tampa, Florida 33607
(813) 872-2798

**COMMODORE PET:**
Chuck Stuart
CMS SOFTWARE
5115 Menefee Drive
Dallas, Texas 75227
(214) 381-0690

**CROMEMCO:**
Richard French
THE NETWORK COMPUTER SYSTEMS
495 Third Avenue #8
San Francisco, California 94118
(415) 668-7777

**DATAPOINT 1500:**
Paul Marchetti
OCCIDENTAL COMPUTER SYSTEMS, INC
6666 Valjean Avenue
Van Nuys, California 91406
(213) 782-3005

**MICROPOLIS, NORTHSTAR (REQUIRES CP/M)**
James Rountree
R&A COMPUTER SERVICES
7700 Edgewater Drive, Suite 723
Oakland, California 94621
(415) 562-3133

**MICROSOFT disk BASIC:**
Dan Kindred
GNAT COMPUTERS
7895 Convoy Court
San Diego, California 92111
(714) 560-0433

**NORTHSTAR BASIC:**
Isaac Bornstein
COMPUTER SERVICES OF ENCINITAS
341 Willowspring Drive
Encinitas, California 92024
(714) 436-2486

**POLYMORPHIC 8813:**
Bruno Caprez
KEYSTONE SYSTEMS
P.O Box 767
Spokane, Washington 99210
(509) 747-5623

**TRS-80:**
Don French
APPLIED DATA CORPORATION
P.O. Box 16020
Fort Worth, Texas 76133
(817) 738-0251

Irwin Taranto
TARANTO & ASSOCIATES
P.O. Box 6073
San Rafael, California 94903
(415) 472-1415

**WANG BASIC ON FLOPPY DISK:**
Richard Armour
ATLANTIC COMPUTING & CONSULTING
P.O. Box 7294
Hampton, Virginia 23666
(804) 340-9350

**OSBORNE & ASSOCIATES, INC.**
P.O. Box 2036, Dept. 122
Berkeley, CA 94702 USA
(415) 548-2805 TWX 910-366-7277

| Title | Price | Quantity | Amount |
|---|---|---|---|
| 09-8 Payroll with Cost Accounting in Wang BASIC | $15.00 | | |
| 13-6 Accounts Payable/Receivable in Wang BASIC | $15.00 | | |
| 20-9 General Ledger in Wang BASIC | $15.00 | | |

Calif. residents add 6% sales tax
S.F. BART residents add 6-1/2 % sales tax

California residents tax ______
Shipping ______
**Total Amount Enclosed** ______

**Shipping** (Shipping for large orders to be arranged)
- ☐ All foreign orders $4.00 per book for airmail
- ☐ $0.45 per book 4th class in the U.S (allow 3-4 weeks)
- ☐ $0.75 per book UPS in the U.S (allow 10 days)
- ☐ $1.50 per book special rush shipment by air in the U.S

**Name:** ______
**Address:** ______
**City:** ______
**State:** ______ **Zip:** ______
**Phone:** ______

☐ I am an end user interested in running CBASIC programs on disk. Please notify me of the closest dealer for

☐ I am a dealer interested in purchasing CBASIC programs on disk for resale. Please send description and order information for

- ☐ CBASIC Payroll with Cost Accounting
- ☐ CBASIC Accounts Payable/Receivable
- ☐ CBASIC General Ledger

Please send the following information
- ☐ Becoming an O&A dealer
- ☐ School discounts
- ☐ List of foreign distributors
- ☐ Other O&A publications
- ☐ I am interested in the CBASIC business books. Please send me description and order information.

Payment in advance must be enclosed for purchases of up to $70.00. Invoicing U.S. purchases over $70.00 available upon approval of your account. All foreign orders must be prepaid in U.S. dollars drawn on a U.S. bank.

S1034

Circle 292 on inquiry card.

# Clubs and Newsletters

**Virginia Club for Radio and Computer Amateurs**

The Amateur Radio Research and Development Corp (AMRAD) is a Virginia based club of over 200 radio and computer amateurs. The purpose of the club is to develop skills and knowledge in radio and electronic technology; advocate design of experimental equipment and techniques; promote basic and applied research; organize forums and technical symposiums; collect and disseminate technical information; and provide experimental repeaters.

Meetings are on the first Monday of each month at 8 PM at the Patrick Henry Branch Library, 101 Maple Av E, Vienna VA. The *Amrad Newsletter* is a monthly publication which is mailed to all AMRAD members. Contact Amateur Radio Research and Development Corp, 1524 Springvale Av, McLean VA 22101.

**Sacramento Microcomputer Users Group**

The Sacramento Microcomputer Users Group newsletter, *Push & Pop*, comes our way on a monthly basis. The most recent contained articles on CP/M Sleuth and CP/M to Pencil and Pencil to CP/M in Z-80 Mnemonics. The general club meetings are the fourth Tuesday of every month at 7:30 PM. The club also sponsors a PET workshop, SOL workshop, and hardware study group. For meeting location and more information about this group contact SMUG, POB 161513, Sacramento CA 95816.

**Attention: Fox Valley Area Illinois Computer Enthusiasts**

A fairly new computer club is getting started in the Fox Valley area of Illinois. Called the Microprocessor User Group, they are primarily an Apple user group but wish to include all processors in their discussions. With a current membership of 50, it is the club's intention to publish a newsletter and a calendar. Meetings are held in the Fermi National Accelerator Laboratory Hi-Rise on the main floor in the SW meeting facility on the third Monday of every month at 8 PM. Contact Mike Urso, 641 Woodlawn, Aurora IL 60506.

**Association of Computer Experimenters**

Besides holding monthly meetings, this active Canadian group publishes a most impressive newsletter called *Ipso Facto*. The most recent issue contains 72 pages and is filled with articles and program listings. Membership in the club is $10 per year, and they are constantly seeking new members with interesting ideas. Contact Bernie Murphy, 102 McCrany St, Oakville, Ontario, CANADA L6H 1H6.

**Pacifica TRS-80 Club**

The Pacifica TRS-80 Users Group is meeting in the Radio Shack store in the Eureka Square Shopping Center which is located about 10 miles from San Francisco. They meet the second and fourth Thursday of the month to exchange programs and ideas regarding the TRS-80. All individuals are cordially invited to attend one of their meetings. Contact John F Strazzarino, 637 Brussels St, San Francisco CA 94134.

**Evansville Indiana Computer Club**

Evansville IN *does* have a computer club according to a recent letter from Robert Heerdink. He is concerned with increasing the club's membership and encourages any interested individuals in that area to attend one of their monthly meetings. The group is interested in several types of microcomputers. They usually meet at the Blind Association, Second Av and Virginia, at 7:30 PM on the second Wednesday of the month, although he recommends checking with him first. Robert can be reached at the Evansville Computer Club, c/o National Sharedata Corp, POB 3895, Evansville IN 47737, (812) 426-2725.

**Theater Computer Users Group**

The Theater Computer Users Group is sponsored by Theater Sources Inc, a corporation created to gather and distribute information about theater in the United States. Their newsletter, called *TCUG Notes*, is $4 per year, and back issues are available. Contact *TCUG Notes*, Mike Firth, Editor, 104 N St Mary, Dallas TX 75214.

**Alamo Computer Enthusiasts**

Located in San Antonio, the Alamo Computer Enthusiasts meet on the third Friday of each month at the Norris Technical Center, Room 208, St Philip's College, San Antonio TX. Their monthly newsletter is available for a one year subscription fee of $2. Contact Dave Fashenpour, 5411 Cerro Vista, San Antonio TX 78233.

**LISP Users Newsletter**

The first issue of the *Lisp Users* Newsletter is an announcement of this new organization which is designed to spread information about applications, implementation and general information on LISP-like languages. It will pay particular attention to those LISPish things which are within the realm of currently available microprocessor systems. As an initialization process, a mailing list of prospective interested individuals is being compiled. If you are interested, send your name and address, background, interest in and knowledge of LISP, what system (if any) you own, and memory capacity (internal and mass storage) to John R Allen, 18215 Bayview Dr, Los Gatos CA 95030. Please include any suggestions you may have about organization or topics you would like to see discussed. John requests that you submit a dollar to cover postage and duplication of the newsletter.

**Free TRS-80 Newsletter**

The free *TRS-80 Bulletin*, which prints club and product news about the TRS-80, is sporting a new format and is now monthly. The *Bulletin* is an offshoot of *TRS-80 Computing* (see BYTE Clubs and Newsletters December 1978). For a free copy of the *Bulletin* write to the Computer Information Exchange Inc, POB 158, San Luis Rey CA 92068.

**Ventura County TRS-80 Club**

This group is comprised of about fifty computer enthusiasts living in the Ventura County area. Their main purpose is to share information relating to the practical applications as well as the entertainment possibilities for Radio Shack's TRS-80 microcomputer. Membership dues are $10 a year which includes a newsletter. Meetings are held the first Tuesday of each month at the Camarillo Public Library, 3100 Ponderosa Dr, Camarillo CA at 7 PM. Visitors are welcome and prospective members are encouraged to attend a meeting. Contact Lee Steinmetz, Secretary, 567 W Loop Dr, Camarillo CA 93030, (805) 484-1724.

## BYTE's Bits

**A Public TV Series**

In a recent phone conversation with Robert J Whitney, Producer/Director at WMVS/WMVT television, channels 10 and 36 in Milwaukee WI, we learned of a proposed television series for public television concerning people who build, buy and use small computers. The series idea is for 8 to 15 half hour shows about small computer hobbyists, experimenters and enthusiasts. Robert is looking for people in the Southeastern Wisconsin area who could lend their expertise to the creation, organization and realization of this idea. Interested persons should contact him by mail, in care of the station or by phone at (414) 271-1036.

**Chess Tournament**

The second London microprocessor chess tournament will be held in the West Centre Hotel, Lilee Road, Fulham, London England, from November 1 thru 3 1979. Any individual or company wishing further details should write to David Levy, c/o Personal Computer World, 62a Westbourne Grove, London, W2.

This year's event will be the first European Open Microprocessor championship. The highest placed participants will automatically qualify for places in the final of the first World Micro Championship which is scheduled to be held in London in 1980.

**IECI '80 Issues Call for Papers**

IECI '80, the Sixth Annual Conference and Exhibit on Industrial and Control Applications of Microprocessors, will be held at the Sheraton Hotel in Philadelphia next March 17 thru 19, 1980. IECI '80 will offer an exhibition and technical program dealing with the current and new work of industrial microprocessor applications. Papers dealing in such areas as automotive diagnosis and operation, intelligent instrumentation, transducers and sensors, automated manufacturing, numerical control and robotics, and biomedical control and monitoring are invited. Ten copies of the paper in extended summary form, 500 to 600 words long and an abstract of no more than 40 words describing work not generally published or previously presented, should be mailed by September 14 1979 to H T Nagle Jr, Electrical Engineering, Auburn University, Auburn AL 36830. The extended summary will be used for paper selection and session assignment and should clearly define the salient concepts and novel features of the work described. Notification of acceptance and format required for publication in the IECI '80 proceedings will be sent by October 19 1979.

**Forth Inc Announces 13 Software Seminars**

Forth Inc, producer of Forth based software systems for mini and microcomputers, has announced a schedule of 13 software seminars to be held throughout the US this summer. The seminars, which are half day events, cover design, application and demonstration of miniForth, microForth, and new polyForth software systems, are open to programmers, project managers, data processing professionals, electronic engineers, and OEMs. Following the seminar, there will be a meeting to discuss various plans for OEMs of Forth software. For a complete list of seminar locations and dates, write to Forth Inc, 815 Manhattan Ave, Manhattan Beach CA 90266.

**Call for Papers for Computer Networking Symposium**

The 1979 Computer Networking Symposium to be held December 12 1979 in Gaithersburg MD has issued a call for papers. The symposium will feature papers of a tutorial nature which describe practical experiences with computer networks and network applications, and papers that present new research results. Accepted papers will be published in the symposium proceedings. Four copies of 1,000 word abstracts should be submittd by July 15 1979 to Dr Vijay Ahuja, E96/B629, IBM Corp, Research Triangle Park NC 27709. Authors will be notified of acceptance by August 1 1979. The Computer Networking Symposium is sponsored by the National Bureau of Standards, and the IEEE Computer Society.

**Byte News Update**

Byte News in the March 1979 issue began with a paragraph listing computer manufacturers that support Pascal. For a year Texas Instruments has supported Pascal which can be implemented on the Model 990 minicomputer in the form of TI Pascal (TIP). There are also two executive operating systems available which are appropriate for use with Pascal.

The Texas Instruments Microprocessor Executive Library (TIPMX) is a collection of operating system components available to users of the TMS 9900 family of microprocessors. Minimum developments tools required for use with TIPMX are a text editor, an assembler and a link editor, all of which are available with the purchase of a Texas Instruments software license. TIPMX may be used on either the floppy disk based FS990 development system, or any of the several hard disk systems such as the DS 990. TIP and its extensive run time support library are available on the hard disk based system and are highly recommended for use with TIPMX.

The latest addition to TI's software support is called Texas Instrument's Modular-Based Executive in Read only memory (TIMBER). TIMBER is targeted for users of the TM 990 series of microcomputer modules. In this way it is possible to do Pascal software development on a low cost system incorporating the TM 990/302 software development system. The fundamental benefit of TIMBER is that it is pre-packaged in erasable read only memory, thereby freeing the user from the concern of inadvertently destroying the operating system.

In the second paragraph of Byte News, Intel was mistakenly credited with being the first to market a 32 K byte programmable read only memory. Texas Instruments introduced the world's first 32 K programmable read only memory in April of 1978. The TMS 2532 is organized as 4 K by 8 bits and operates from a single +5 V power supply with standby power dissipation of 50 mW typical. A +25 V supply is only needed for programming.

**Dale R Gibble**
**Project Engineer Texas Instruments**
**POB 1443 MS 6750**
**Houston TX 77001** ■



## BYTE's Bugs

**A Reorganized Wine Cellar**

As one who enjoys both wine and computers, I must comment on several inaccuracies in the article entitled "Computerized Wine Cellars" (February 1979 BYTE, page 128). The second wine in listing 1 is Rauenthaler Rothenberg Riesling Spatlese 1976 — von Simmern. The varietal name is not, as listed, Rhine, which is a geographical area, but Riesling, which is the name of the best grape variety produced in Germany. The Rheingau, which is the Rhine sub-area from which this wine comes, produces Riesling almost exclusively, but other Rhine areas produce Sylvaner, Müller-Thurgau, and other varietals as well.

More strangely, the producer is listed as Rauenthaler. In top quality German wines, the first name refers to the village and the second to a particular vineyard. Here, we have the village of Rauenthal and the vineyard Rothenberg. The producer is the Graf (Count) von Simmern.

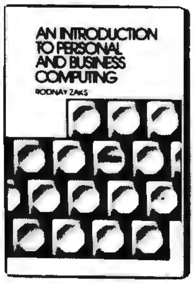
In any case, I have recently enjoyed tasting this wine and I can verify the accuracy of the tasting notes in the listing — it is a super wine, and in wine drinking, that's what counts!

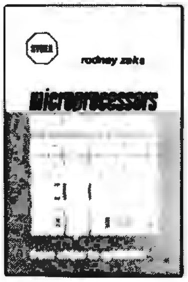
Larry Rosenblum
12008 Trailridge Dr
Potomac MD 20854■

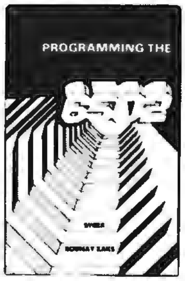
SYBEX THE LEADER IN MICROCOMPUTER EDUCATION PRESENTS

# BEST-SELLING BOOKS

***Used by Universities and Industry worldwide (ten languages)***








**C200 - AN INTRODUCTION TO PERSONAL AND BUSINESS COMPUTING**
Rodnay Zaks, 250 pp
A comprehensive introduction to small computers, their peripherals, and what to select. **$6.95**

**C201 - MICROPROCESSORS: FROM CHIPS TO SYSTEMS**
Rodnay Zaks, 416 pp
The basic text on all aspects of microprocessors and the assembly of a system. **$9.95**

**C202 - PROGRAMMING THE 6502**
Rodnay Zaks, 320pp
A complete introductory programming text for the 6502. **$10.95**

**D302 - 6502 APPLICATIONS BOOK**
Rodnay Zaks, 288 pp
Connecting a 6502 board to the outside world: from home alarm to music and industrial control. **$12.95**

**C280 - PROGRAMMING THE Z80**
Same as C202 for the Z80. **$10.95**

**C207 - MICROPROCESSOR INTERFACING TECHNIQUES**
A. Lesea & R. Zaks, 416 pp
How to interface a microprocessor to external devices: from keyboard to ADC to floppy disk, including standard busses. **$11.95**

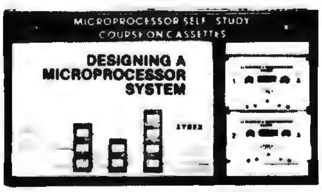
**X1 - MICROPROCESSOR LEXICON,** 125pp
All the definitions of the microprocessor world in a convenient pocketbook format. **$2.95**

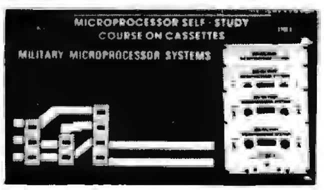
AVAILABLE AT BOOKSTORES, COMPUTER STORES, AND ELECTRONIC SHOPS EVERYWHERE!

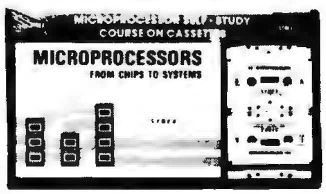
# SELF-STUDY COURSES ON CASSETTES

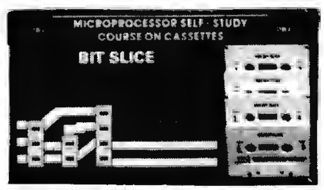
**"The time-efficient way to learn"™**

Self-study courses include two to eight audio-cassettes and a special book. They offer the fastest way to learn the topic covered (from ½ to two days). Highly effective.










**1 - INTRODUCTORY - SHORT (2.5 hrs ea)**
S1-INTRODUCTION TO MICROPROCESSORS $29.95
S2-PROGRAMMING MICROPROCESSORS $29.95
S3-DESIGNING A MICROPROCESSOR SYSTEM $29.95
S10B-INTRODUCTION TO PERSONAL AND BUSINESS COMPUTING $21.90

**2 - INTRODUCTORY - COMPREHENSIVE (10 to 12 hrs ea)**
SB1-MICROPROCESSORS $59.95
SB2-MICROCOMPUTER PROGRAMMING $59.95

**3 - SPECIALIZED (4.5 to 6 hrs ea)**
SB3-MILITARY MICROPROCESSOR SYSTEMS $49.95
SB5-BIT-SLICE $49.95
SB6-INDUSTRIAL MICROPROCESSOR SYSTEMS $49.95
SB7-MICROPROCESSOR INTERFACING $49.95

## TO ORDER

**By Phone:** 415 848-8233, Visa, M.C., Amer Express.
**By mail:** Include payment.
**Shipping:** add $1.50 per book (UPS) or 75¢ (4th class - allow 4 weeks). Double for cassettes and overseas.
**Tax:** in California add tax.

FREE DETAILED CATALOGUE

SYBEX
2020 Milvia Street
Berkeley, CA 94704
Tel 415 848-8233 Telex 336 311

NAME ______ POSITION ______
COMPANY ______
ADDRESS ______
CITY ______ STATE/ZIP ______
☐C200 ☐C201 ☐C202 ☐C207 ☐D302 ☐C280 ☐S1 ☐S2
☐S3 ☐S10B ☐SB1 ☐SB2 ☐SB3 ☐SB5 ☐SB6 ☐SB7
☐charge my ☐Visa ☐M.C. ☐Amer Express
Number ______ Exp. date ______
Signature ______ ☐Send Catalogue B7

Circle 356 on inquiry card.

# Creativity in Computer Music

Hubert S Howe, Jr
Dept of Music
Queens College CUNY
Flushing NY 11367

The most important result of computer applications in music is that they have encouraged a variety of new ideas to be formulated, tested, and reformulated in a short period of time. Concepts that previously would have taken much longer to develop and verify have already been brought forth. Musical thought is consequently on a higher level today. This applies to all aspects of work with computers – as much to research as to creative work, but perhaps less to music than to some other fields. The increasing availability of microcomputers can only further this trend.

The distinction between research and creative work is artificial, for much research is creative – certainly when it pursues new ideas. Computer work in music has for many years been carried out in the creative areas of musical composition and sound synthesis, and the research areas of musical analysis and music theory. In this discussion I will not consider such areas as music bibliography, music printing, CAI (computer-assisted instruction), and related disciplines. While much important work has been done in these areas, they are not generally concerned with creative or conceptual problems. Music bibliography essentially involves an information storage and retrieval system. Music printing is a problem of automation. CAI certainly encompasses creative work, but thus far only the most rudimentary instructional tasks have been delegated to computers, and then only with limited success.

Instead, I will discuss music theory, analysis, sound generation, and composition and their relation to the computer. While we are discussing the question of musical tasks in general, it is relevant to point out that *any* musical task is in principle subject to delegation to a computer. Musical tasks include any activities carried out by human beings that involve interaction with music.

Music is composed by composers in ways that, traditionally at least, involve only their own internal ideas, realized sometimes with the assistance of an instrument like the piano. Music is performed by performers who usually play from printed notation. Music is heard by listeners who may then engage in reflection, evaluation, or criticism. Musical documents such as recordings or scores are kept in libraries where they are filed according to careful systems of classification. Computer composition, performance, printing, recording, documentation, bibliography – even evaluation or criticism – all are feasible.

The problems involve simulating the sensory processes with which people interact with music – listening, writing, visual communication – and the mental processes with which they make their conclusions about music. The sensory processes are simulated by the I/O (input/output) devices on computers, and since computers cannot do all of these tasks in the same way that people can, various languages have been devised to enable translation from the human form into a form computers can read. The mental processes are simulated by the programs.

Now that we have described these general features of computer work in music, let us consider some specific problems concerning the four disciplines mentioned above and how using a computer has encouraged the formulation of new concepts in creative ways.

## Music Theory

There has always been a close relationship between the discipline of music theory and other fields, particularly analysis and composition. In the twentieth century, composers have produced some highly original theories that they have attempted to apply to their compositions, either before or after they have written them. The term "speculative music theory" describes theories conceived before the music illustrating them is written.

After the music has been composed, it is usually a simple matter to verify whether

a theory applies to it, assuming that the theory is adequately formulated. For example, Schoenberg's conception of atonal music and later his "method of composition with 12 tones" originated as speculative ideas which were described, albeit confusingly, in his books *Harmonielehre* and *Style and Idea.* More recent composer/theorists like Milton Babbitt have extended Schoenberg's original ideas to even greater levels of abstraction.

The availability of computers has spurred great efforts to test these theories and other ideas to a greater degree than people had been willing to attempt before. (Though most theoretical work has been done on large general-purpose computers, much can now be done on microcomputers, as is shown in our example of pitch structures.) Concomitantly, the results obtained by these researchers have brought forth new ideas and exposed the inadequacies in old ones.

Part of the reason for such success has been the fact that it is by no means obvious or easy to explain how theoretical ideas apply to a piece of music in detail, and people have had to clarify their methods in ways that go beyond what had been accepted before in noncomputer work. Computers do not presently have the ability to *listen* (in the sense that a person "listens") to a piece of music to decide whether or not a given idea applies to it. Researchers have instead formulated methods that involve detailed inspection of the *notation* for a piece. Sometimes these produce results that seem mystifying or incorrect when compared with results reported by listeners. This conflict demonstrates the difficulty of accepting unintuitive ideas about music and the question of whether or how listeners ought to revise their listening habits in order to perceive musical structures that are verified to exist in the music.

Theories, in the sense employed thus far, are ideas that explain the structure of entire compositions or passages of music. Few musical theories, even that of triadic tonality, are in fact expressed in any such detail. Usually there is a large gap between the theoretical constructions and their applications to the music that is supposed to be filled by the reader. This is obviously true of such modern concepts as pantonality (see reference 6), or of attempts to perceive Schoenberg's 12 tone compositions in a certain key (see page 407 of reference 7). Allen Forte's theory of set complexes (reference 8) presents a collection of ideas that would still not be a theory in the sense described above, but he nevertheless has much more success

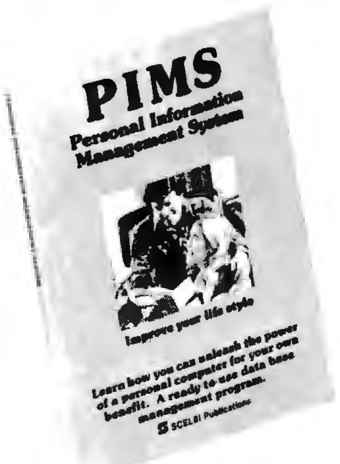
# BITS inc

Books to erase the impossible

POB 428, 25 Route 101, Peterborough, NH 03458

# LET'S GET PERSONAL

## How to use use your personal microcomputer to its fullest potential




### SMALL COMPUTER SYSTEMS HANDBOOK
**By Sol Libes**

☐ This is an introductory text for those with a minimal knowledge of electronics who are interested in microcomputers as a hobby. The author explains microcomputer concepts from the number systems and logic circuits to programming languages and applications with the emphasis on practical knowledge.

197 pp. $8.45

### YOUR HOME COMPUTER
**By James White**

☐ A clearly written, nontechnical description of personal computers that requires no prior knowledge of computers or electronics. The emphasis is on understanding; over 100 illustrations are included. Topics include: computing and you; communication inside a computer; computer thought processes; fixed memory; inputs and outputs; peripherals; systems components; how to choose a microcomputer; and so on. **Your Home Computer** is the ideal book for readers who thought they could never understand how computers work. The best part is that this is both easy and fun to read.

235 pp. $9.95

### PIMS — PERSONAL INFORMATION MANAGEMENT SYSTEM
**By Madan Gupta**

☐ This book describes a data base management system designed for personal use on small computer systems using BASIC. Along with complete source listings of the program and comprehensive operating instructions, the book discusses the personal microcomputer and its potential for personal use. The tasks you could perform with this information management tool include: maintaining a mailing list, balancing small business and personal checking accounts, keeping track of inventories, maintaining log books, and countless others. Powerful commands allow you to create, add to, change, list, delete, sort, and search your data files.

This book can really unleash the full power of your personal computer!

88 pp. $9.95.

## ... and for your personal programming needs

## The Bits Programmer Pad™

☐ Improve your style with BITS Programmer's Pads.™ One side is an ingenious memory/register allocation form, custom-tailored to the microprocessor you're programming: 8080A, 6800, 6502, Z-80, or 1802. The other side is for your program, with pre-numbered lines and columns for addresses, object code, labels, instructions, and those essential comments. Write your source code first and later assemble by filling in the object codes and assigning memory addresses. (If you're hand assembling your programs, these pads are a must!)

50 page pads on durable paper. Prepunched for a 3-ring notebook, too. **ONLY $2.50 each.**

Postage: $.75 for one Pad, $1.25 for two or more Pads.

## ✱ NEW ✱

### Z80 ASSEMBLY LANGUAGE PROGRAMMING
**By Lance A. Leventhal**

☐ Hot off the press, here's the Z80 assembly language and its use in Lance Leventhal's graphic style. A parallel to his earlier book "6800 Assembly Language Programming", it covers the Z80 instruction set in depth and programming techniques for its use, including many sample products.

*FROM OSBORNE & ASSOC.* $9.50

### INTRODUCTION TO TRS-80 GRAPHICS
**By Don Inman**

☐ TRS-80 users, here in Level-I BASIC are the techniques you need to make full use of your video display. This book is ideal for the beginner, with complete explanations for the statements and commands used. The plot and print methods are detailed and used in creating lines, curves, and more complex geometric shapes. Moving patterns are explored. The basic techniques will stir your imagination to make full use of the TRS-80's graphics capabilities.

133 pp. $8.95

### 32 BASIC PROGRAMS FOR THE PET COMPUTER
**By Tom Rugg and Phil Feldman**

☐ Here are 32 fully documented programs, ready to run on your Commodore PET 2001 computer, including applications programs, educational programs, games, graphical display programs, math programs, and more. The program instructions are detailed enough to allow you to modify any program and sample runs are included. PET users, don't leave your PET on the shelf, put it to work for you!

267 pp. $15.95

### PEANUT BUTTER AND JELLY GUIDE TO COMPUTERS
**By Jerry Willis**

☐ This book is much like a peanut butter and jelly sandwich: simple but nutritious. It is an introduction to the hardware of personal computers that is free of technical jargon; a combination consumer's guide and tutorial. The fundamental parts of a computer, computer languages, and personal applications are covered. Included are case studies of people, their computers and applications. It's a humorous and painless introduction to the world of the personal computer.

207 pp. $7.95

**DIAL YOUR CHARGE CARD ORDERS TOLL FREE**

**800-258-5477**

VISA

master charge

(in NH call 924-3355) Mon-Fri 9-5

For your convenience in ordering, please use this page plus the order form on page 137

in describing how his ideas relate in specific ways to actual musical content and the difficulties encountered in this process.

Forte's theory is a striking example of a series of ideas that grew out of a continuing interaction with computers. In his case, as well as that of this author, computers figured prominently in the development of the concepts employed in formulating the theory. Composers have sometimes referred to these concepts as precompositional ideas.

In my own work (described in "Some Combinational Properties of Pitch Structures," *Perspectives of New Music,* volume 4, number 1, fall/winter 1965, pages 45 thru 61) the basic concepts were pitch, pitch class, pitch class collection, and pitch structure. A computer was used to generate the basic sets of all of these elements, and then to test various operations such as inversion and cycle of fifths equivalence. The fact that a computer was used in this work meant that my list of pitch structures generated by the computer was one of the first accurate lists ever produced. Howard Hanson's count of distinct pitch sets in *The Harmonic Materials of Modern Music: Resources of the Tempered Scale,* Irvington Pub, NY, 1961, is incorrect. Forte's list published in "A Theory of Set Complexes for Music," *Journal of Music Therory,* volume 8, number 2, winter 1962, is correct according to his definition, but his definition does not reflect current practice. I have continued to use computers in my work, which now includes computer composition as well as theory development.

It is not difficult to see that the continuing use of computers in music theoretical work will have beneficial results, if only because it will force the theorists to formulate their ideas in more specific terms. Such continuing investigations could reveal and clarify many inadequacies in the theory of harmony or tonality as presently stated in harmony textbooks.

In fact, the *computerization* of the principles stated in any of these books ought to point up all kinds of basic problems that are never acknowledged – for example, just exactly what the theory explains about which music and what that has to do with any activity the reader would be likely to engage in with respect to either the theory or the music. *Harmony,* as we understand it from harmony texts, is simply the translation of a piece of music into a different notation or vocabulary – a vocabulary that is imprecise and overlooks many of the most important characteristics of the music itself. The reasons why such a translation is made and what advantages it possesses are unclear. Because they have to clarify their purposes to impersonal machines, computer researchers are not able to ignore these questions as easily as traditional ones.

### Musical Analysis

Computer work in musical analysis highlights the question: what do we really want to know about a piece of music? There are

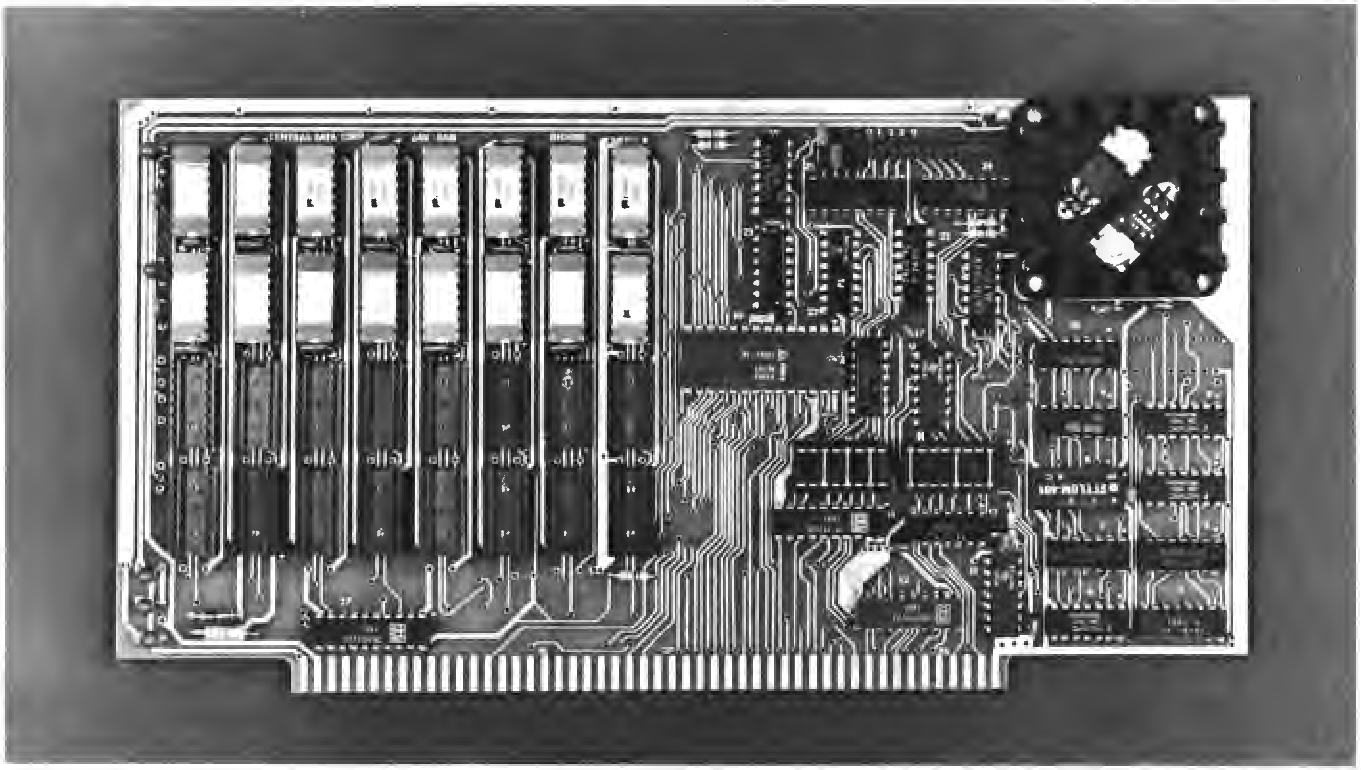
### Pitch Structures

Pitch structures are sets of notes related by transposition (see Glossary). Within 12 tone equal temperament, there are exactly 351 of them available. The BASIC program shown in listing 1 generates all pitch structures in normal form. Normal form is a method of notation that chooses the representation with the smallest overall intervallic span. Individual pitch classes are indicated according to their intervallic distance from the lowest note. For example, the major triad (C E G) has three orderings: 0 4 7 (C E G), 0 3 8 (E G C), and 0 5 9 (G C E). The normal form representation would be 0 4 7. When two representations have the same overall span, the choice is made according to the smallest second, third, etc, interval. Some sets, like the augmented triad 0 4 8 (C E G#) have the same intervallic pattern for each ordering; these sets all have two or more transpositions that produce the same set of pitch classes.

This program is written in TRS-80 Level 1 BASIC, which allows multiple statements to be given in a single line using the colon (:) as a delimiter. LET is optional in assignment statements, and has been omitted in the program.

*Listing 1: Program written in TRS-80 Level 1 BASIC to compute pitch structures in normal form.*

```
  5 REM PROGRAM TO COMPUTE PITCH STRUCTURES IN
  6 REM NORMAL FORM
 10 FOR I=1 TO 24: A(I)=0: NEXT I
 20 S=2: K=1: L=S-1: M=2: N=1: Q=0: A(25)=0
 30 FOR I=1 TO L: A(I)=A(10*N+14+I): NEXT I:A(S)=K: V=A(2)
 40 GOSUB 210: IF T <> 0 THEN 80
 50 FOR I=1 TO S: A(10*M+14+I)=A(I): NEXT I
 60 Q=Q+1: PRINT S, Q,: FOR I=1 TO S: PRINT A(I);: NEXT I: PRINT
 70 K=K+1: M=M+1: GOTO 30
 80 IF V <= A(10*(N+1)+16) THEN 100
 90 L=S: S=S+1: Q=0: IF S >10 THEN 110
100 N=N+1: K=A(10*N+14+L)+1: GOTO 30
110 GO TO 999
200 REM SUBROUTINE NRMLFM
210 P=S-1: X=0: T=0
220 IF A(1)=0 THEN 250
230 X=A(1): T=X
240 FOR I=1 TO S: A(I)=A(I)-X: NEXT I
250 FOR I=1 TO S: A(I+12)=A(I): NEXT I: A(S+13)=12
260 U=A(14): X=X+U: IF X >= 12 THEN RETURN
270 FOR I=2 TO S: A(12+I)=A(13+I)-U: NEXT I
280 IF A(S+12) < A(S) THEN 330
290 IF A(S+12) > A(S) THEN 260
300 FOR F=2 TO S: IF A(F+12) < A(F) THEN 330
310 IF A(F+12) > A(F) THEN 260
320 NEXT F: RETURN
330 FOR I=1 TO S: A(I)=A(I+12): NEXT I: T=X: GOTO 260
999 END
```



32K Board Pictured Above

# New RAM Prices.

## From <u>The</u> Dynamic Memory Company.

| | |
|---|---|
| **16K—$249** | **32K—$375** |
| **48K—$500** | **64K—$625** |

**Ever since we started making these memory boards over a year ago we have continued to lower our prices to stay competitive. Due to your confidence in us, we are again able to lower our prices!** Our reliability has been proven by months of superior performance in thousands of installations. Our low-power boards are being used by quality-minded systems manufacturers across the country and overseas.

**4MHz boards now available.**

After receiving hundreds of requests, our engineering staff has come up with a new version of our board which runs on 4MHz Z-80 systems. It wasn't easy to come up with a high speed board which would operate as reliably as our 450ns version, but after months of careful design and testing, we did it. The price of the 250ns board is $10 per 16K additional.

**All of our features remain.**

Our boards didn't become great sellers only because of the price. We still offer you our deselect feature which allows our RAM to overlap with any fixed memory areas in your system. Also, the RAM area of our board is fully socketed so that you can expand the board yourself.

Other standard features include: plug selectable addressing on 16K boundaries (shorting plugs are placed over wire-wrap pins to address the board — located on the top of the board for easy changes), S-100 and Z-80 compatability and totally invisible refresh — no wait states.

**Fully assembled, tested, and guaranteed.**

All of our boards go through a rigorous testing procedure. They are then placed on burn-in running a series of memory tests to detect any other possible faults. After you receive the board, you are backed by us with a one year warrantee.

**Low power consumption keeps your computer from "losing its cool."**

The total power consumption of our 16K board is typically less than 4 watts (+8V @ 300ma, +16V @ 150ma and −16V @ 20ma). Boards with additional memory typically increase power consumption only 1 watt per 16K!

**Standard S-100 Interface.**

Our board is designed to interface with any standard S-100 CPU. All of the timing of the board is independent of the processor chip, and the board is set up for different processors by changing two plugs on the board.

**Contact your local dealer.**

To find out more about our RAM boards, contact your local dealer. If he is unable to help you, call or write us for a fast response. Central Data Corporation, 1207 North Hagan Street, Champaign, IL 61820. (217) 359-8010

**Central Data**

Circle 45 on inquiry card.

two ways in which computer programs attempt to answer this question. One is to pose a traditional question to which we supposedly know the answer, in the hope that the task of carrying this out on the computer will improve our understanding; in other words, it helps us understand musical intuition. The second way is to ask the computer to solve problems we have not yet understood when traditional methods have been used. Serious research will always be concerned with questions of this sort, but unless those of the first sort can also be handled we will be in doubt about the success of the second.

There have been many research projects in musical analysis, most of which are ongoing and have thus far produced only tentative or incomplete results. In my opinion, the most important conflict in this area concerns both purpose and procedure: should computers be used to examine the characteristics of a single work, or rather to analyze the common properties of whole bodies of works?

Some of the most significant results in musical analysis have been obtained by theorists who are more interested in the theoretical problems than the musical works, and who have analyzed works more in order to prove their theories than to discover new aspects of the music. I have already mentioned the work of Allen Forte, who has produced excellent analysis of diverse atonal works with the assistance of a computer. Another example is the well-known "Josquin Project" conducted by Professors Arthur Mendel and Lewis Lockwood at Princeton University. While this project is remarkable for the sheer quantity of music that has been encoded for study, all analytical work has been carried out from the standpoint of music analysis according to the theoretical principles formulated by contemporaries of Josquin. (See reference 9. Josquin des Prés, c1445-1521, was a prominent composer of early music.) Few studies have produced results as successful as these. (Some are absurd. See my review of "Two Parameters of Melodic Line as Stylistic Discriminants" by David Sheldon Lewis and "Some Techniques for Computer-Aided Analysis of Musical Scores" by Donald Margedo Pederson in *Perspectives of New Music*, volume 9, number 2 and volume 10, number 1, 1971.)

One of the most time-consuming aspects of computer analysis of music has been the development of encoding languages in which musical notation can be transcribed for computer input. Analytical studies have become studies of musical notation, not of sound or auditory experience, and are thus open to the objections raised above. Many languages have been developed for different purposes, and many of them choose to represent the notational characteristics in very different ways. There has been much unfruitful debate over these languages prompted by some researchers trying to get others to adopt their language as universal. One researcher, Eric Regener, has instead taken the intelligent line of writing programs to transcribe other notations into his.

An important quality of these languages is that each was developed with a different purpose in mind, and thus each emphasizes different aspects of the music. Some languages developed for the purpose of music printing (like DARMS) are necessarily quite complex in their methods of representing spacing between notes and other features of page layout that are irrelevant for languages concerned with analytical work. Other languages developed for use with a specific body of music (like IML) cannot be used for other types of music because they do not provide methods for encoding features that do not occur in the works for which the languages were developed.

The point I would like to make about these languages is that any notation expresses a conceptual structure of the music

**Encoding Languages**

The following example shows a line of music from the Hosanna of the *Missa Ave Maris Stella* by Josquin des Prés encoded for computer input in the IML encoding language:

```
'= CLEF G' '= KEY F'
R11/R1 *B1,HO,/B2* C2 D1,SAN,/G11,NA,/
R1 *C1/C2*B4 A4 *A1//
```

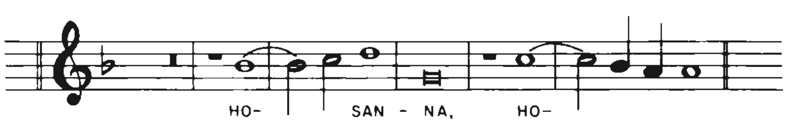
Keywords indicating the clef and key signature are enclosed in quotation marks with an equal sign at the beginning. Rests are indicated by the letter R. Notes are indicated by the letter name (C, D, E, F, G, A, B) as read on the staff (people who encode this data must be able to read music). Following the letters A through G or R is a number indicating the duration: 11 is a breve, 1 is a whole note, 2 is a half note, and 4 is a quarter note. Text syllables are enclosed in commas following the note, and the slash (/) indicates a bar line; the double slash indicates the end of the line. Asterisks indicate the beginnings or ends of ties.


*Figure 1: Musical excerpt from the Hosanna of the* Missa Ave Maris Stella *by Josquin des Prés (see "Encoding Languages" text box).*

# Basic Time INTRODUCES THE $15,995* 10MB HARD DISK COMPUTER

*Quantity 1 Price

## Complete System Includes:

- ★ 10 MILLION CHARACTER HARD DISK STORAGE
- ★ DISK OPERATING SYSTEM & EXTENDED BASIC
- ★ POWERFUL 16-BIT CENTRAL PROCESSING UNIT
- ★ VIDEO DISPLAY TERMINAL
- ★ 150 CPS LINE PRINTER AND PRINTER STAND
- ★ DESK ENCLOSURE

basic time

1215 E. El Segundo Boulevard
El Segundo, California 90245
213/322-4435

Circle 23 on inquiry card.

it represents. Printing languages are very specific about graphical features because those are important aspects of the intended result. Other languages developed for analysis represent elements like pitches and rhythms as the basic aspects to be examined, and provide many ways for representing nuances in these domains but not others. For another example, guitar notation providing only a melodic line and a basic chord pattern is a remarkably efficient means of indicating a diverse quantity of sounds in a few symbols, and the symbols convey a sense about the irrelevance of certain nuances in the chord structure that would be unthinkable in other music. No language currently in existence is universally satisfactory *and* efficient to represent all music that could be transcribed for any purpose. Arguments about encoding languages show only a lack of understanding of the diverse and even cross-disciplinary purposes for which certain research is carried out.

### Sound Generation

In the area of sound generation, computers have already had a significant impact on the music of our time, and the future is even more promising. (An excellent description of the facilities needed to produce music on a microcomputer is contained in Tom O'Haver's article "Audio Processing with a Microprocessor" in June 1978 BYTE, page 166.) To explain the reasons for this, it is necessary to review some of the history of both computer generated music and electronic music, of which computer music is a part.

Electronic music has been shaped by several disparate influences. Historically, the first was the use of nonconventional sounds that were previously considered to be nonmusical. Early works were based on the sounds of railroads, water dripping, and other noises of everyday experience and of nature. Another significant influence was the availability of analog electronic sound generating and modifying equipment, which is now packaged into devices called synthesizers. (A third influence, less important than the first two, was the use of speech and language in noncommunicative ways.) Music based on sounds of nature tended to be extremely complex in the foreground, whereas music produced by electronic equipment often lacked dynamic variations in tonal characteristics.

Today, these early tendencies have been mollified as a result of the experiences of many people working with these ideas, often in conjunction with computers. People have begun to analyze the characteristics of nonconventional sounds, often by computer, to discover and generalize the properties of interest in them. Thus, early exploratory use of these sounds has now given way to conceptual thinking about them. Such thinking is a natural result of processes which people must employ to generate sounds by computers, because users must present information to the computer in concrete ways. A computer cannot generate a sound from a person's abstract recollection of what it sounds like.

Sounds produced by electronic music synthesizers have evolved from the lifeless organ-like sounds that contain no dynamic variation in tonal characteristics to sounds that mimic live musical instruments. A synthesizer is, indeed, a musical instrument, and it is natural that people would begin to develop performance techniques once they are able to work with them for a while. Such expert performers as Walter Carlos and Isao Tomita can routinely produce any quality such as vibrato, tremolo, or dynamic spectral variations. They have also developed excellent methods of imitating specific instruments, particularly woodwinds, brass, and percussion (solo string sounds are easier to imitate than the lush sounds of an entire string section).

The trouble with this procedure is that even expert synthesizer performers now at least partly judge their work not by the originality of the sounds they produce, but by their resemblance to the familiar sounds of musical instruments. This is in complete contrast to computer music composers.

To cause a computer to generate music, it is necessary for the composer (who is actually the performer in this case) to provide a detailed description of the sound desired. The description can be anything that is mathematically sufficient to cause the desired properties to be produced by the computer. *Any* describable sound can be produced; the limitation is not in the capability of the computer but rather in the ability of composers to provide adequate descriptions of what they want. A detailed explanation of how computers generate musical sound is contained in *Electronic Music Synthesis* (see reference 3).

This factor has been one of the primary reasons for the recent interest in the analysis of sounds of all types. Computer music synthesis is thus not limited by the comparison of the results produced to any pre-existing standard. Indeed, composers are encouraged to be creative with the qualities of the sounds they produce, by the very procedure by which they must work.

Of course, people who work with syn-

# 5 reasons why you should <u>not</u> buy
# the electric pencil II™



Check the appropriate box(es):

- ☐ **You love typing the same copy 20 thousand times a day.**
- ☐ **Your secretary can type 250 words per minute.**
- ☐ **You're dying to spend $15,000 on a word processing system, just for the tax investment credit.**
- ☐ **All your capital assets are tied up in a 10-year supply of correction fluid.**
- ☐ **You never commit a single thought to paper.**

If you have checked one or more boxes, you do not need The Electric Pencil. On the other hand, you may want to join the thousands of people who haven't checked a single box.

The Electric Pencil II is a *Character Oriented Word Processing System.* This means that text is entered as a string of continuous characters and is manipulated as such. This allows the user enormous freedom and ease in the movement and handling of text. Since line endings are never delineated, any number of characters, words, lines or paragraphs may be inserted or deleted anywhere in the text. The entirety of the text shifts and opens up or closes as needed in full view of the user. The typing of carriage returns or word hyphenations is not required since lines of text are formatted automatically.

As text is typed and the end of a line is reached, a partially completed word is shifted to the beginning of the following line. Whenever text is inserted or deleted, existing text is pushed down or pulled up in a wrap around fashion. Everything appears on the video display as it occurs, which eliminates guesswork. Text may be reviewed at will by variable speed scrolling both in the forward and reverse directions. By using the search or search and replace functions, any string of characters may be located and/or replaced with any other string of characters as desired.

Numerous combinations of line length, page length, line spacing and page spacing permit automatic formatting of any form. Character spacing, bold face, multicolumn and bidirectional printing are included in the Diablo versions. Multiple columns with right and left justified margins may be printed in a single pass.

## Wide screen video

Versions are available for Imsai VIO video users with the huge 80x24 character screen. These versions put almost twice as many characters on the screen!!!

## CP/M versions

Digital Research's CP/M, as well as its derivatives, including IMDOS and CDOS, and Helios PTDOS versions are also available. There are several NEC Spinwriter print packages. A utility program that converts The Electric Pencil to CP/M to Pencil files, called **CONVERT**, is only $35.

## Features

- CP/M, IMDOS *and* HELIOS compatible
- Supports four disk drives
- Dynamic print formatting
- DIABLO and NEC printer packages
- Multi-column formatting in one pass
- Print value chaining
- Page-at-a-time scrolling
- Bidirectional multispeed scrolling controls
- Subsystem with print value scoreboard
- Automatic word and record number tally
- Cassette backup for additional storage
- Full margin control
- End-of-page control
- Non-printing text commenting
- Line and paragraph indentation
- Centering
- Underlining
- Bold face

## Upgrading policy

Any version of The Electric Pencil may be upgraded at any time by simply returning the *original* disk or cassette and the price difference between versions, plus $15 to Michael Shrayer Software. Only the originally purchased cassette or diskette will be accepted for upgrading under this policy.

MSS
MICHAEL SHRAYER SOFTWARE, INC.
1253 Vista Superba Drive
Glendale, CA. 91205
(213) 956-1593

## Have we got a version for you?

The Electric Pencil II operates with any 8080/Z80 based microcomputer that supports a CP/M disk system and uses an Imsai VIO, Processor Tech. VDM-1, Polymorphic VTI, Solid State Music VB-1B or Vector Graphic video interface. REX versions also available. Specify when using CP/M that has been modified for Micropolis or North Star disk systems as follows: for North star add suffix A to version number; for Micropolis add suffix B, e.g., SS-IIA, DV-IIB.

| Vers. | Video | Printer | Price |
|---|---|---|---|
| SS-II | SOL | TTY or similar | $225. |
| SP-II | VTI | TTY or similar | 225. |
| SV-II | VDM | TTY or similar | 225. |
| SR-II | REX | TTY or similar | 250. |
| SI-II | VIO | TTY or similar | 250. |
| DS-II | SOL | Diablo 1610/20 | 275. |
| DP-II | VTI | Diablo 1610/20 | 275. |
| DV-II | VDM | Diablo 1610/20 | 275. |
| DR-II | REX | Diablo 1610/20 | 300. |
| DI-II | VIO | Diablo 1610/20 | 300. |
| NS-II | SOL | NEC Spinwriter | 275. |
| NP-II | VTI | NEC Spinwriter | 275. |
| NV-II | VDM | NEC Spinwriter | 275. |
| NR-II | REX | NEC Spinwriter | 300. |
| NI-II | VIO | NEC Spinwriter | 300. |
| SSH | SOL | Helios/TTY | 250. |
| DSH | SOL | Helios/Diablo | 300. |

## Attention: TRS-80 Users!

The Electric Pencil has been designed to work with both Level I (16K system) and Level II models of the TRS-80, and with virtually any printer you choose. Two versions, one for use with cassette, and one for use with disk, are available on cassette. The TRS-80 disk version is easily transferred to disk and is fully interactive with the READ, WRITE, DIR, and KILL routines of TRSDOS 2.1.

| Version | Storage | Price |
|---|---|---|
| TRC | Cassette | $100. |
| TRD | Disk | $150. |

# <u>Demand</u> a demo from your dealer!

Circle 319 on inquiry card.

thesizers are not limited either; nevertheless it seems that there have been far more successful electronic performances of conventional music recently than there has been original music. Part of the reason for this difference is actually one of the limitations of computer music as it is produced in most situations today: the lack of immediacy between performance time (at which the input is fed to the computer) and audition time (when the sound is produced). Very few installations today have any immediate playback facilities except for more than a few simple sounds, and real time computer music synthesis of entire compositions is practically impossible (see reference 3, chapter 9).

The inherent difficulty in producing sounds by computer, together with the delay in hearing the results, forces computer music composers to evaluate their work reflectively and to think carefully before trying things out. While immediacy is important to the act of performance, it is not necessary for the act of composition itself, which is a conceptual task.

An excellent example of the kind of creativity on the sound level made possible in computer music is described by J K Randall in his article "Operations on Waveforms" (*Perspectives of New Music*, volume 5, number 2, spring/summer 1967, page 124). These ideas are exploited compositionally in Randall's *Lyric Variations for Violin and Computer*, written in 1968 and recorded on Vanguard C-10057. Randall describes several original ideas he used to synthesize new timbres from completely new theoretical principles. He produces sounds in which the individual constituent partials (harmonics of the fundamental) of a tone are operated upon just as the pitch, rhythm, and dynamics are in conventional music. The result is music of unusual interest in which all variations of the sounds relate in novel ways to the structure of the music.

The same kinds of processes in which partials are treated as pitches were employed in my compositions entitled *Studies in Timbre*, of which there are presently four. The first study employs sounds that dissolve into others by glissandos that move from one partial to another. These are contrasted with other sounds of fixed pitch but changing timbre. The second study uses instrument-like sounds with transient elements in the attack and variations in the steady state portions of the tones. The third is based on varying timbral patterns associated with specific musical events. The typical instrument produces 12 partials that fade in and out in different ways over the course of each tone, correlated to amplitude and location changes. Contrasts between partials of some tones and the fundamental frequencies of others are emphasized in conjunction with rhythmic, dynamic, and harmonic properties that develop concomitantly. The fourth study uses harmonic series that progress to nonharmonic ones, but which nevertheless preserve some abstract ratio between the elements, thus producing another kind of dissolution of a tone into a somewhat clangorous sound.

The main point of these considerations is to show how a computer music composer is encouraged to experiment with original ideas that often lead to results unobtainable by any other method of music production. But since computers are theoretically capable of producing any sounds, this is not the only beneficial or distinguishing result. The fact is that the *same* sounds can be produced by *different* descriptions that are based on generalizations about different aspects of the sounds. Thus, the really important point is that it is the *relationship between the input and the output* which is clarified by the process of computer music synthesis. Whatever this may be in relation to a specific musical work is in itself a concept about the structure of that work.

**Musical Composition**

Although I have emphasized the creativity involved in the other disciplines discussed, it is obvious that composition is, by comparison, the ultimate creative act to be delegated to a computer. Many people express disbelief or doubt that this can really be done successfully. I believe that this doubt is rooted more in misunderstanding of what musical composition itself involves rather than of what the computer does. Composers may employ mysterious methods, but they are even more prone to making mysterious explanations of straightforward methods in order to preserve their compositional mystique.

Any detailed consideration of musical analysis or theory suggests numerous avenues of approach to the compositional method employed. Whether a piece actually has been composed according to the principles that can be abstracted from its structure is not necessarily relevant, and often can never be known. In recent years, the work of Heinrich Schenker and post-Schenkerian analysts has attempted to discover large scale structural properties in tonal music which almost certainly were not consciously considered by the composers who wrote the music, and which raise questions not previously posed in the history of theory.

Computer composition can be employed as a method of verifying theories and analyses of specific works, especially if they consist of multilayered abstractions like Schenkerian analysis. These abstractions suggest a generative approach starting from a background and building through successive layers of elaboration to the foreground. One method of verifying such an analysis is to prove that it can actually regenerate the work, at least in its essential structural aspects, by a particular sequence of operations. Another method is to change some of the background or middleground structure or the particular sequence of operations to produce new works that can be compared with the original. Although they might not be intended to be taken seriously as original compositions (since their structure is derivative of another work), such compositions would unquestionably be computer composed music.

Instead of using this structural approach, early work in computer composition has tended to emphasize the sensational aspects of "machine music" and has been based on random and aleatoric methods (see reference 2). Very few conditions of the intended music have been specified, with the result that certain characteristics are quite clearly present (eg: consonances) or absent (eg: dissonances) but the music is otherwise aimless and without structure. These facts are quite obvious when the music produced resembles some early style like a Bach invention, but are less clear when some avant-garde style is imitated. Nevertheless, such experiments (often designated as experiments rather than as music) are not typical of musical compositional methods in general, nor of computer work in this field.

In my own music I have employed computer composition in extensive ways ("Composing by Computer," *Computers and the Humanities,* volume 9, 1975, page 281). Nevertheless, the computer does not make any decisions that I would not make myself, nor indeed that I have not already made when it executes my instructions; it merely carries out many time-consuming calculations that would otherwise have to be done by hand. Basically, the program works out aspects of the foreground syntax which assure various rhythmic, harmonic and structural properties. All aspects of this syntax are specified by instructions provided together with the data (pitch classes) on which they are carried out.

Using a program like this allows composers to work more from a background perspective than from the foreground. They are able to concentrate on the large scale structural properties without being encumbered with the foreground details that may arise from these characteristics; these are handled quickly and automatically by the computer. Several different possibilities for working out a given passage can be tried before any commitment is made to them.

Even more important, the process of writing a program to carry out such compositional details forces composers to be absolutely explicit about their procedures and intentions. Any errors or incompleteness in the specifications will be exposed when a program is executed by the computer. Ad hoc methods that composers may use to fix certain passages when they don't work out as expected are not necessary, because it is easy enough to revise the program once these problems are exposed. The composer's attention is thus always directed to the most important conceptual aspects of the music, and his or her ability to solve problems in these areas is facilitated.

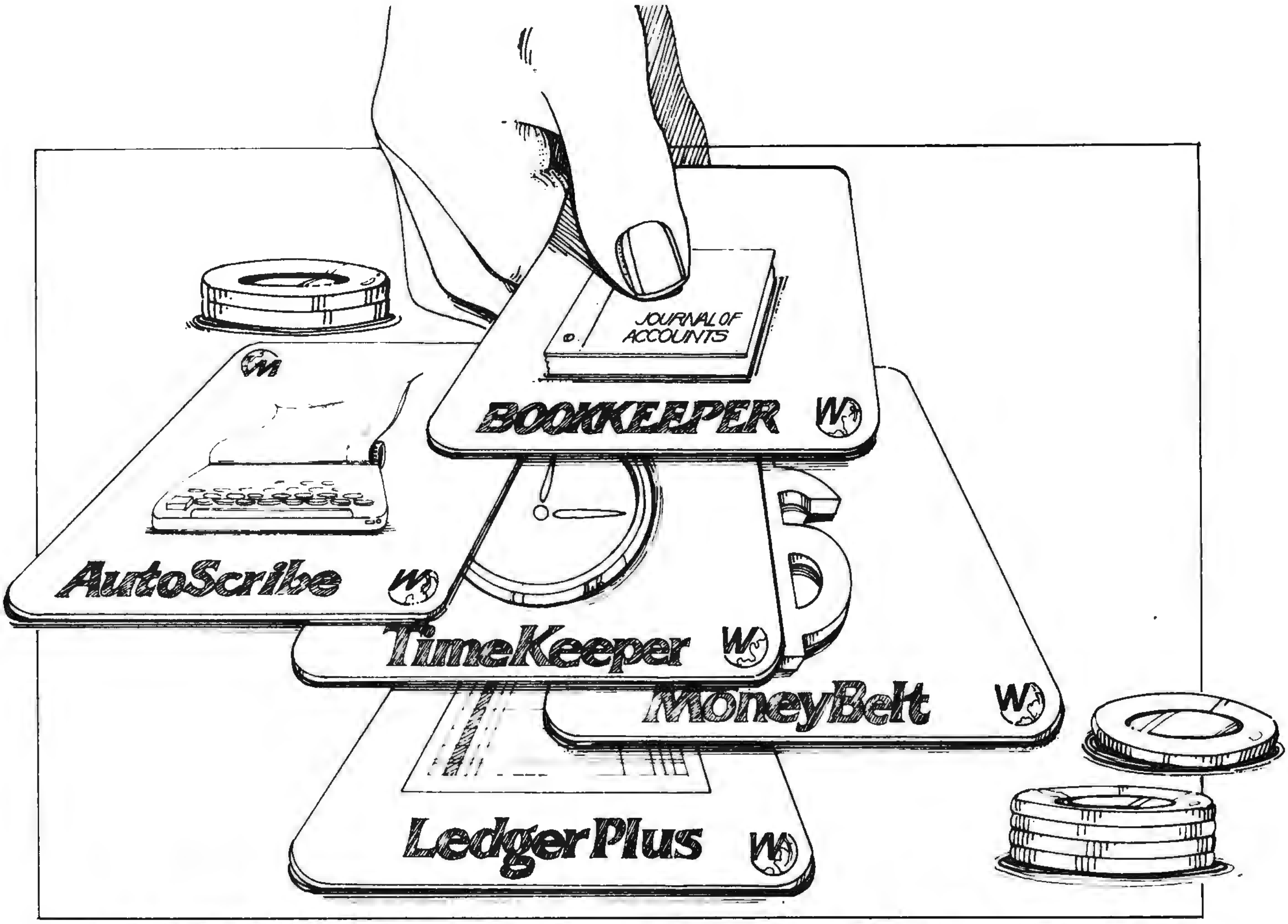
## Conclusion

By examining aspects of the four musical disciplines discussed above, we have seen many instances where the use of computers





Circle 78 on inquiry card.


JOURNAL OF ACCOUNTS
BOOKKEEPER
AutoScribe
TimeKeeper
MoneyBelt
LedgerPlus

necessitates a conceptual approach to the questions at issue. While many projects continue to pose problems because of the difficulty of computer programming, and of preparation of data, progress in these fields is nevertheless being made in significant ways. There still remain, and there always will be, gaps between those subjects that computer researchers tend to deal with and those that traditional, noncomputer people will choose, partly in the belief (from both sides) that some subjects are not suitable or possible for computer analysis. While important questions are always difficult to answer and sometimes require generations of work in order to achieve accurate answers, they are never impossible. We will know that we have truly reached a high state of conceptual thinking when the conclusions of computer research tend to agree with those of noncomputer research – both will then be operating on the same level.■

## GLOSSARY

**Aleatoric music:** music, usually composed by computer, in which all or many of the most important characteristics are chosen randomly.

**Analysis:** description of music according to certain fundamental properties that are judged to be relevant to a given piece, showing how different elements in the work may be related by these properties.

**Atonality:** music which is not tonal, and where a specific attempt is made to avoid reference to a key, or when the concept of key is not relevant. The term has two basic uses, specific and general: specifically, it describes a body of early twentieth century music by Schoenberg, Berg, Webern, and others, preceding 12 tone music (qv), in which tonality (qv) was consciously avoided. Generally, the term is used to describe any twentieth century music avoiding tonality that cannot be described simply by other methods.

**Background:** musical substructure describing large scale properties and relations that may not be evident from an inspection of the immediate note to note properties, or from a superficial auditioning of the music; see **foreground.** In certain theories, the background is considered divisible into several levels.

**Computer music:** music employing computers at any stage of its composition or realization as sound.

**Cycle of fifths equivalence:** an operation on a group of notes in which each element is replaced by its equivalent on the circle of fifths; analogous to inversion (qv). The circle of fifths generates the total chromatic (qv) by starting on any note, adding the note a perfect fifth (seven half steps) higher, and continuing this process.

**Electronic music:** music in which the sounds are produced by electronic means. In a somewhat simplified form, electronic music as currently practiced can be broken down into four general areas: (1) **musique concrète,** in which use is made of natural sounds that are recorded on tape and modified by recording processes or other means; (2) **tape music,** in which sounds are modified according to procedures that may be applied to magnetic tape or tape recorders; (3) **synthesized music,** in which the sounds are produced by electronic music synthesizers (qv); and (4) **computer music** (qv), in which the sounds are produced or controlled by computers.

**Encoding language:** a method in which musical notation may be represented in code suitable for computer input. The most widely used and documented languages to date are IML (Intermediary Musical Language), DARMS (Descriptive Alphanumeric Representation of Musical Symbols), the "Plaine and Easie Code System for Musicke," LMT (Linear Music Transcription), and ALMA (Alphanumeric Language for Music Analysis).

**Foreground:** the "surface" of a piece of music, including sounds that are simultaneous or that appear in direct succession; distinguished from the background (qv).

**Glissando:** a continuous sliding from one pitch to another.

**Half step:** the smallest interval (qv) used in music based on equal temperament (the tuning system in widespread use in Western cultures since the eighteenth century). Music employing smaller intervals is said to be **microtonal.**

**Harmony:** a theory describing properties of simultaneous sounds (chords) in tonal music (see **tonality**). Chords are expected to move in certain progressions, and dissonances resolve into consonances according to various rules.

**Interval:** the distance, measured in half steps, between two pitches or pitch classes. Tonal music also employs another definition of interval, based not on the sound but on the notation for the two notes involved.

**Inversion:** an operation on a group of notes in which each element is replaced by its equivalent on the descending chromatic scale, or ascending circle of major sevenths (11 half steps). (In this formulation, an ascending chromatic scale would be the identity operation.) Identity and inversion, along with cycle of fifths equivalence and its inversion (cycle of fourths equivalence) are the only single interval cycles that generate the total chromatic (qv).

**Octave:** a musical interval of 12 half steps, corresponding to the frequency ratio of 2 to 1. Pitches related by octaves possess a strong similarity, which has been called **octave equivalence.**

**Pitch:** a single tone in a musical composition. Most pitches used in music are taken from the range of the 88 produced by the piano, but some extend beyond this range.

**Pitch class:** a group of pitches separated by any number of octaves. In musical theories, pitches in different octaves employ the same letter names (C, C # or D♭, D, D # or E♭, etc) reflecting the fact that theories are based on pitch classes rather than pitches. Pitch classes are also often called notes. While there are many pitches, there are only 12 distinct pitch classes in Western music.

**Pitch structure:** a set of pitch class collections that all possess the same intervallic structure, so that they are related by transposition (qv). Pitch structure is the basic way that collections of tones, such as chords, are compared: as major triads, minor triads, etc.

**Spectrum:** the overtone structure of a sound, represented as a series showing the amplitude of each overtone present; see **timbre.**

**Structure:** any abstract method in which the properties of a piece of music can be encompassed. Generally, a piece is divided into several sections, each of which has a different structure. Sometimes structure is described in terms of a function or purpose at work in an entire section, such as introduction, development, or statement (of a theme or idea). Sometimes structure is described numerically, proportionally, or in other abstract ways.

**Synthesizer:** a machine that generates and processes sounds automatically, used in the production of electronic music. Most synthesizers consist of a number of discrete components that perform different functions (eg: oscillators that generate tones, filters that modify their spectrum, etc). Some recently designed synthesizers include microprocessors, which are used as controlling devices.

**Theory:** a set of concepts describing properties and relations that can be shown to exist in a body of musical literature. The most commonly known theories today include tonality (qv) and 12 tone music (qv), but there are many others that have been described and used by various authors. Most theories have originated after the music they purport to describe has been written. *Speculative* theory originates as speculation, before such music has been written, so that its success or failure has not yet been demonstrated.

**Timbre:** musical tone quality, descriptive of the way in which different tones may possess a similarity not on the basis of pitch, amplitude, or rhythm. In electronic music, this term is normally used synonymously with spectrum (qv), but in contexts where it is described qualitatively rather than numerically.

**Tonality:** a theory describing music which is in a key, or a series of keys, usually pertaining to music written during the eighteenth and nineteenth centuries. Basic concepts include the major and minor scales, triads, and specific rules according to which dissonances resolve into consonances. Much of the music of the twentieth century is based on an extended notion of tonality, in which some, but not all, of the basic concepts are employed.

**Total chromatic:** any series of notes including all 12 pitch classes.

**Transposition:** the addition or subtraction of a constant interval to each tone in a collection, moving the set up or down by a uniform amount.

**Triad:** a chord consisting of three notes (or pitch classes) with a root note, and other notes a third and fifth above the root.

**12 tone music:** music in which all note successions, and sometimes other properties are based on sets of orderings of the total chromatic.

## REFERENCES


1. von Forester, H, and Beauchamp, J W, (editors), *Music by Computers,* John Wiley and Sons Inc, 1969.
2. Hiller, Jr, L A, and Isaacson, L M, *Experimental Music,* McGraw-Hill Book Company Inc, 1959.
3. Howe, Jr, H S, *Electronic Music Synthesis,* W W Norton and Co Inc, 1975.
4. Lincoln, H B, (editor), *The Computer and Music,* Cornell University Press, 1970.
5. Morgan, C P, (editor), *The BYTE Book of Computer Music,* BYTE Publications, 1979.
6. Reti, R, *Tonality in Modern Music,* Crowell-Collier, 1962.
7. Sessions, R, *Harmonic Practice,* Harcourt, Brace, Jovanovich Inc, 1951.
8. Forte, A, *The Structure of Atonal Music,* Yale University Press, 1973.
9. Mall, T, et al, *IML-MIR User's Manual,* Princeton University Music Dept, 1972.

### Journals

*Computers and the Humanities,* Pergammon Press Inc, Maxwell House, Fairview Park, Elmsford, NY 10523.

*Computer Music Journal,* People's Computer Company, Box E, 1263 El Camino Real, Menlo Park CA 94025.

*Journal of Music Theory,* Yale School of Music, New Haven CT, 06520.

*Perspectives of New Music,* Perspectives of New Music Inc, Annandale on Hudson, NY 12504.

Nybbles

## "Tiny" Pascal in 8080 Assembly Language


Dr G Louis
OB/GYN Department
St Michael's Hospital
30 Bond St
Toronto CANADA M5B 1W8


The p-code interpreter, Pascal to p-code compiler, and p-code to 8080-code translator described in "A 'Tiny' Pascal Compiler" by Kin-Man Chung and Herbert Yuen (September thru November 1978 BYTE) have been rewritten in 8080 assembly language. In addition to providing approximately two orders of magnitude increase in speed, the object versions run in far less memory.

To use the assembly language version of tiny Pascal you need: an 8080 (or Z-80) microcomputer system with not less than 12 K bytes of memory (the package would run in 8 K bytes but you need file space); a copy of the articles by Chung and Yuen, which have been reprinted by BYTE magazine and are available for $3 (BYTE document number 100); an elementary acquaintance with 8080 assembly language; the package of assembly language modules.

Additionally, it is desirable to have a text editor, an assembler that stores lines in the form: length byte, 4 digit line number, space or tab, text, carriage return, and a relocator program that is useful for tailoring the package to your system.

*The Nybbles Library is an inexpensive means for BYTE readers to share some interesting but specialized forms of software. These programs are written by readers with small computers and printer facilities, and are therefore designed for particular systems. The algorithms and programming techniques in these programs can be directly used by readers with similar equipment, or can serve as an inspiration for improvisation on computers of different characteristics.*

*Potential authors of such programs should send us a self-addressed stamped envelope, with a request for a copy of our "Guidelines for Nybbles*

Authors." Payment for Nybbles items is based on sales and length of the item. Rates are set at the time of acceptance.

Nybbles Library programs are sent in listing form, printed on 8.5 by 11 inch paper on both sides. The Nybbles Library programs are 3 hole punched for collection in loose leaf binders, and come in an attractive folder which serves as a cover.

This month "'Tiny' Pascal in 8080 Assembly Language" (document #111) has been added to the Nybbles Library. Use the coupon below to order your personal copy, at $10.00 (US and Canada) or $12.00 (foreign airmail) postpaid.

**Please send ________ copies of BYTE Nybble #__________ at $__________ postpaid.**

**Please remit in US funds only.**

__________ *Check Enclosed*

*Bill my VISA #* ______________________ *Exp Date* ______________

*Bill my MC #* ______________________ *Exp Date* ______________

Name ______________________________________________

Street ______________________________________________

City ____________________ State ____________ Zip Code ____________

**BYTE Nybbles Library, 70 Main St, Peterborough NH 03458**

You may photocopy this page if you wish to keep your BYTE intact.

# Quest

**Roger Chaffee**
**POB 7495**
**Menlo Park CA 94025**

*You are at one end of a vast hall stretching forward out of sight to the west. There are openings to either side. Nearby, a wide stone staircase leads downward. The hall is filled with wisps of white mist swaying to and fro almost as if alive. A cold wind blows up the staircase. There is a passage at the top of the dome behind you. Rough stone steps lead up the dome.*

> GO DOWN

*You are in the hall of the Mountain King, with passages off in all directions. A huge green fierce snake bars the way!*

> ATTACK SNAKE

*Attacking the snake both doesn't work and is very dangerous.*

> FEED SNAKE

*There's nothing here it wants to eat (except perhaps you).*

> YELL

*Are you trying to somehow deal with the snake?*

> YES

*I am prepared to give you a hint, but it will cost you 2 points. Do you want the hint?*

The preceding paragraphs are taken from a session I played on a computer running a program called Adventure, which has used many, many cycles of machine time on many, many computers in the past year or two. The original version, I believe, was written in FORTRAN, by Willie Crowther (now at Xerox Palo Alto Research Center), and was subsequently modified and rewritten by Don Woods at Stanford's Artificial Intelligence Laboratory.

The version I ran has been bootlegged through several generations, and I'm not sure of its exact ancestry. It requires close to 200,000 bytes of memory on an IBM 370/168. Adventure has been translated into PL/I, APL, and BASIC. It is the successor to Hunt the Wumpus and the many *Star Trek* games.

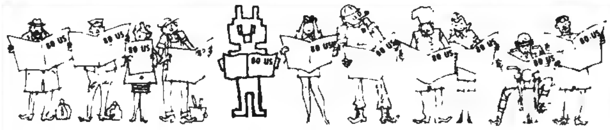
I hope it is the precursor of more elaborate games which combine computers with fantasy to produce an "electric novel," which the user and the computer write or experience together. Already, a few computers around the country are offering a child of Adventure called Zork or Dungeon, which has a more sophisticated understanding of English, and a whole new set of problems to solve and monsters to defeat. Space War, which used to belong to the "freaks" and the "hackers" (in the hours after the managers went home), is now available in your neighborhood tavern for 25¢ per enemy starship. How much longer will it be until we can each rule our own kingdom and rescue our own fair maidens?

```
                         QUEST

YOU WERE WALKING THROUGH THE
WOODS, AND YOU CAME ACROSS THE ENTRANCE
OF A CAVE, COVERED WITH BRUSH.

PEOPLE SAY THAT MANY YEARS AGO A
PIRATE HID HIS TREASURE IN THESE
WOODS, BUT NO ONE HAS EVER FOUND IT.
IT MAY STILL BE HERE, FOR ALL I KNOW.

WHEN YOU ANSWER A QUESTION, I LOOK AT
ONLY THE FIRST LETTER, ALTHOUGH YOU CAN
TYPE THE WHOLE WORD IF YOU WANT.

TYPE N, S, E, W, U, OR D FOR NORTH, SOUTH,
EAST, WEST, UP OR DOWN. TYPE P FOR SCORE

YOU'RE OUTSIDE THE CAVE.
GO SOUTH TO ENTER.

                                   WHICH WAY? SOUTH

YOU'RE IN A LOW CHAMBER. A TIGHT TUNNEL
GOES EAST, AND YOU CAN WALK TO THE
SOUTH OR WEST. THERE IS LIGHT
TO THE NORTH.

                                   WHICH WAY? WEST

YOU'RE AT A DEAD END!

                                   WHICH WAY? EAST

YOU'RE IN A LOW CHAMBER. A TIGHT TUNNEL
GOES EAST, AND YOU CAN WALK TO THE
SOUTH OR WEST. THERE IS LIGHT
TO THE NORTH.

                                   WHICH WAY? SOUTH

YOU'RE AT THE HOME OF THE GNOME-KING.
FORTUNATELY, HE'S GONE FOR THE DAY

                                   WHICH WAY? SOUTH

YOU CAN'T GO IN THAT DIRECTION
```

*Listing 1: Sample dialogue from the Quest game. This shows only the first few moves.*

## Quest

There aren't many personal computers yet with 200 K bytes of memory available, and not all of us have free or inexpensive access to the machines on which Adventure can run. A smaller computer would require a floppy disk file for keeping the cave descriptions, and most users have no hardware for



# YOU'VE READ THEM ALL-NOW GET THE BEST!

## The 80-US Journal

A PUBLICATION FOR TRS-80 USERS - COVERS **ALL** TRS-80 CAPABILITY!

*BI-MONTHLY, SEND $16.00 CHECK OR MONEY ORDER FOR A 1 YEAR SUBSCRIPTION TO:*

**80 US**
**PO Box 7112**
**TACOMA, WA. 98407**

MASTER CHARGE, VISA OK - (206) 759-9642
*WANT TO CHECK IT OUT FIRST? SEND $3.00 for current sample copy.*

A SELECT CHOICE OF SOFTWARE IS AVAILABLE



# T.D.Q.
# TAPE DATA QUERY

**PET-8 K SOL-IIA TRS-80-LEVEL II**

***FILE MANAGEMENT SYSTEM**
—UTILIZES DUAL AUDIO CASSETTE RECORDERS
***INTERACTIVE QUERY LANGUAGE**
—ENGLISH-LIKE COMMANDS
—POWERFUL INFO RETRIEVAL CAPABILITY
***COMPUTERIZED BUSINESS & PERSONAL RECORDS**
—CUSTOMIZE YOUR OWN FILE STRUCTURES
—CREATE & MAINTAIN DATA FILES
—NO PROGRAMMING EXPERIENCE REQUIRED
***IMPLEMENTED IN BASIC**
**T.D.Q. CASSETTE WITH MANUAL & REF. CARD $50.00**

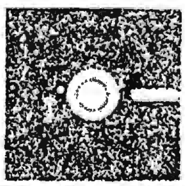
**THE FOLLOWING PRE-DEFINED T.D.Q. FILE STRUCTURES ARE AVAILABLE TO SOLVE YOUR DATA PROCESSING NEEDS:**

| | |
|---|---|
| INVENTORY CONTROL | $35.00 |
| ACCOUNTS RECEIVABLE | $35.00 |
| ACCOUNTS PAYABLE | $35.00 |
| ORDER PROCESSING | $35.00 |
| CUSTOMER DIRECTORY | $25.00 |
| APPOINTMENT SCHEDULING | $25.00 |

**EACH WITH CASSETTE AND MANUAL**

*SEND SELF-ADDRESSED STAMPED ENVELOPE FOR COMPLETE SOFTWARE CATALOGUE.*
*SEND CHECK OR MONEY-ORDER TO:*

**H. GELLER COMPUTER SYSTEMS**
DEPT. B, P.O. BOX 350
NEW YORK, NY 10040
*(NEW YORK RESIDENTS ADD APPLICABLE SALES TAX)*



# CP/M®
## LOW-COST MICROCOMPUTER SOFTWARE

**CP/M® OPERATING SYSTEM:**
- Includes Editor, Assembler, Debugger and Utilities.
- Standard version for 8080, Z80, or Intel MDS (other versions available.)
- For IBM-compatible floppy discs.
- **$100**-Diskette and Documentation.
- **$25**-Documentation (Set of 6 manuals) only.

**MAC™ MACRO ASSEMBLER:**
- Compatible with new Intel macro standard.
- Complete guide to macro applications.
- **$90**-Diskette and Manual.

**SID™ SYMBOLIC DEBUGGER:**
- Symbolic memory reference.
- Built-in assembler/disassembler.
- **$75**-Diskette and Manual.

**TEX™ TEXT FORMATTER:**
- Powerful text formatting capabilities.
- Text prepared using CP/M Editor.
- **$75**-Diskette and Manual.

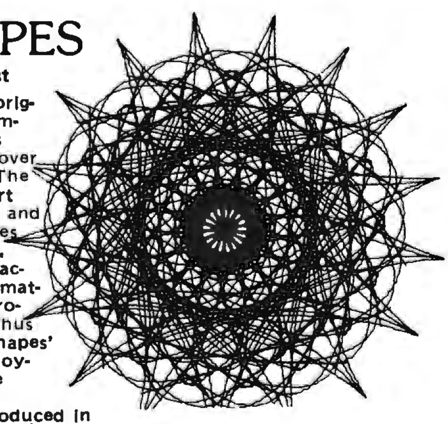
**DESPOOL™:**
- Background print utility.
- Use with CP/M (version 1.4)
- **$50.** Diskette and Manual.

**DIGITAL RESEARCH®**
**P.O. Box 579 • Pacific Grove, California 93950**
**(408) 649-3896**



# NEW SHAPES

by **Julius Guest**



is a collection of 110 original and fascinating computer-generated designs created by the author over an eight-year period. The designs are satisfying art creations in themselves and may evoke the responses that their titles suggest.

As each art piece is accompanied by a mathematical formulation and program, the reader may thus create his own 'New Shapes' and derive as much enjoyment from them as the author obviously did.

The designs are reproduced in color (some in exciting solid reverse) on high quality art paper.

**Details:** 174 pages, 28cm x 27cm (11"x10½"). (At $12.00 U.S, that's only 11¢ per diagram!) Add $1.25 postage. ISBN 0-9596384

## INTRODUCTION TO PL/1 & STRUCTURED PROGRAMMING

**R.A. Vowels**, Senior Lecturer, Royal Melbourne Inst. of Technology. 1978; 311 pages, 60 diagrams, ISBN 0-9596384-0-7.

This book covers an easy-to-learn subset of **ANSI PL/1.** Fundamentals are emphasized and structured programming is encouraged. The book contains numerical and non-numerical worked examples, and 200 exercises and problems. The versatile subset is defined in only 3 pages of syntax diagrams. Suited to a one- or two-semester course. Also appropriate for self-study. An excellent reference document–only **$5.00 U.S.** (Add 80¢ postage.)

**Contents:** ten chapters and six appendixes. Data representation, fixed and floating-point, character strings, LIST and EDIT-directed I/O, DO, WHILE, declarations, arrays (fixed and variable bounds), procedures and functions, built-ins, computational error detection and correction, ON-units.

**Order** direct from publisher R.A. Vowels, 93 Park Drive, Parkville 3052, Australia.
Personal checks accepted (in your currency), or Bank Draft Check in Aust. currency. **Discount:** 5% for 2 or more books.

rapid random access outside the computer memory. Adventure on everybody's computer is still in the future, although it is coming soon.

I was playing Adventure at about the time Peninsula School, Menlo Park CA, received two Commodore PET 2001 machines for the children to work with; and the incredible idea of Adventure on a PET was too exciting to ignore. Adventure on a PET, with only 7123 bytes available for the BASIC program, is impossible, but couldn't I do something just a bit less ambitious? In a couple of intense work weeks, I wrote a program I called Quest, which ran on the PET.

### Comparing Quest and Adventure

Compared to Adventure, Quest is a toy, in the same measure that the PET is a toy compared to larger computer systems. But it is an enjoyable and even exciting toy, in the same ways that the PET can be enjoyed by someone who can also play with the big computers. Each system has its own problems and pleasures. Adventure, as you can see from the opening paragraphs, has some novel problems for the adventurer to solve, and will proceed according to the adventurer's 2 word commands. On the other hand, the problem you must solve in Quest is basically to find your way around the cave. The only commands Quest normally understands are six directions: NORTH, SOUTH, EAST, WEST, UP, and DOWN, and even there, only the first letter is examined.

No huge green snake will confront you, and even the pirate, who swoops down to protect his treasure at some point, is beyond your control. He steals back the treasure after you have found it, and the problem continues: find the treasure again, and find your way out of the cave. To make it more interesting, various passages open and close according to your progress through the game.

One limitation which Quest and Adventure share is that they never change. Once you know how to get past the snake, you always know, and once you can find the treasure, you always know where it is. A friend of mine has suggested having earthquakes, which open and close passages at random. It seems to me, however, that that simply makes a bigger problem of the same kind, and I would rather have different problems. In that respect, both Adventure and Quest are very limited.

In a closer approach to the electric novel, there would be no guarantee that the problems can really be solved. In a Star Trek game, for instance, the fate of the Enterprise depends on the random number generator, which can set the starship down in the same quadrant with four Klingon commanders and two super commanders, or cause all the starbases to be overrun by a plague of tribbles. In Adventure and Quest, as in crossword puzzles, the solution is part of the problem definition, and you know it exists.

It is possible in Adventure to do something which will ruin your chance for success. For instance, if you feed the bird to the snake, you will never get past the problem to which the bird is the solution. However, this doesn't change the basic limitation.

An important part of both games is the descriptions of the locations. These are of course not created by the computer, but were elaborated over a long period by the programmer and his friends. The topology of the Quest cave could be more complex if the location descriptions were something like "YOU'RE AT LOCATION 28. NOW WHAT?", but that would spoil a good part of the game. The descriptions in Quest have been worked out very carefully. Some of them are just for fun, and some of them have hints about the neighborhood.

## Scoring

The original version of Quest had no scoring at all, to minimize the competitive situations I thought might develop. The children who used it soon said that they wanted points. My own cynical analysis suggests that they don't know whether or not they're having fun unless they keep score, but perhaps there is simply a need in all of us to track our progress in a quantitative way. In any case, Quest now awards you one point for each location that you visit, plus up to 25 for the various checkpoints you pass along the way. In this version you can get up to 66 points.

## Playing Time

Many people who play Adventure find that it takes them about two weeks of serious study to get through. The corresponding time for Quest is about two hours, although it has been done in an hour, and some people haven't finished after a day of frustrating search!

## The Program

Quest was written for the PET 2001, and used some features of the PET to reduce the size of the program. The version given here has been rewritten in a simpler dialect of BASIC, which I believe will run with minor

*Text continued on page 186*

*Listing 2: Game of Quest in BASIC.*

```
1  REM QUEST    BY ROGER CHAFFEE
2  REM INSPIRED BY WILL CROWTHER'S "ADVENTURE"
3  REM COPYRIGHT(C) 1978
4  REM    PENINSULA SCHOOL, MENLO PARK, CA
5  REM    PERMISSION TO USE, NOT TO SELL
6  REM THE ORIGINAL VERSION OF THIS PROGRAM WAS WRITTEN ON A
7  REM COMMODORE PET 2001.  THIS VERSION HAS BEEN CONSIDERABLY
8  REM RECODED, AND IS IN "PLAIN VANILLA" BASIC, WITH THE
9  REM EXCEPTION OF THE RANDOM NUMBER GENERATOR IN LINES
10 REM 6600-6800, THE STRING MANIPULATION, AND THE IF ... THEN
11 REM STATEMENTS WHICH GIVE A STATEMENT TO PERFORM INSTEAD OF
12 REM A STATEMENT NUMBER TO GO TO.
13 REM VARIABLES USED
14 REM    N NODE (CAVE) NUMBER
15 REM    MO   MOVE COUNTER
16 REM    T    CURRENT LOCATION OF TREASURE (-1 FOR CARRYING)
17 REM    T1,T2 FIRST AND SECOND HIDING PLACES (NODE NUMBERS)
18 REM    M6   SAVES THE MOVE NUMBER WHEN HE SAID NO, HE DIDNT
19 REM          WANT TO TAKE THE TREASURE WITH HIM.
20 REM    M   MAP OF INTERCONNECTIONS.
21 REM          M(I,J) IS NEXT NODE FROM NODE J, WHEN YOU GO
22 REM             N,E,U,D,W,S FOR I=1,2,3,4,5,6
23 REM
24 REM                      BOUNCES
25 REM        TO GO TO NODE -2 MEANS BOUNCE BACK TO THE NODE YOU
26 REM           CAME FROM.
27 REM
28 REM                      FORCED AND/OR RANDOM MOVES
29 REM        M(1,N)=-2 MEANS A FORCED MOVE AS SOON AS YOU REACH
30 REM        NODE N.  IN THAT CASE, M(2,N) OF THE TIME YOU GO
31 REM        TO NODE M(3,N).  IF YOU DON'T GO THERE,   M(4,N)
32 REM        OF THE TIME YOU GO TO NODE (5,N), AND THE REST OF
33 REM        THE TIME YOU GO TO NODE (6,N).
34 REM
```

*Listing 2 continued on page 180*

*Listing 2 continued from page 179:*

```
 35  REM                   NODES > 100:
 36  REM        NODE N+100 MEANS NODE N IF YOU DON'T HAVE THE
 37  REM        TREASURE, AND NODE N+1 IF YOU DO.
 38  REM        NODE N+200 MEANS NODE N+1 IF YOU HAVE THE TREASURE
 39  REM        THE SECOND TIME, AND NODE N OTHERWISE.
 40  REM        NODE N+500 MEANS RUN THROUGH A DELAY LOOP AND THEN
 41  REM        GO TO NODE N. THIS IS USED FOR FALLING DOWN THE
 42  REM        EXIT CHUTE,  AND YOU MAY WANT TO ADJUST THE DELAY
 43  REM        TIME (LINE 6250).
 44  REM
 45  REM    Q$  INPUT STRING
 46  REM    A$  CHARACTERS TO MATCH IN THE INPUT ROUTINE
 47  REM    A2  NUMBER OF CHARACTERS IN A$
 48  REM    A1   OUTPUT FROM THE INPUT ROUTINE
 49  REM    P    PIRATE FLAG  1 IF PIRATE HAS GOT YOU, 0 OTHERWISE
 50  REM    P1   COUNTER FOR PIRATE ROUTINE
 51  REM    N9   SAVES OLD NODE IN MOVE ROUTINE, FOR BOUNCE
 52  REM    N8   SAVES NODE WE BOUNCED FROM IN MOVE ROUTINE,
 53  REM            FOR PRINT FLAG
 54  REM    N0   SAVES OLD NODE IN MOVE ROUTINE, FOR DEAD END
 55  REM    A0   SAVES OLD DIRECTION IN MOVE ROUTINE
 56  REM    D    DEBUG FLAG (NON-ZERO TO PRINT)
 57  REM    I,J MISC. COUNTERS
 58  REM    W   TRAVEL FLAG, USED IN SCORING. W(I)=1 IF HE'S
 59  REM        BEEN TO NODE I, 0 OTHERWISE
 60  REM    S   SCORE
 61  REM    M9  MAXIMUM NUMBER OF NODES
 80  D=0
100  REM -------------------------------------------------------
110  REM GIVE 'EM SOMETHING TO READ WHILE I GET THE DATABASE SET UP
120  PRINT "               QUEST"
130  PRINT
140  PRINT "YOU WERE WALKING THROUGH THE"
150  PRINT "WOODS, AND YOU CAME ACROSS THE ENTRANCE"
160  PRINT "OF A CAVE, COVERED WITH BRUSH."
170  PRINT
180  PRINT "PEOPLE SAY THAT MANY YEARS AGO A"
190  PRINT "PIRATE HID HIS TREASURE IN THESE"
200  PRINT "WOODS, BUT NO ONE HAS EVER FOUND IT."
210  PRINT "IT MAY STILL BE HERE, FOR ALL I KNOW."
400  READ M9,T1,T2
490  REM DIMENSION OF W, M IS M9, IF YOU HAVE DYNAMIC ASSIGNMENT
500  DIM W(42),M(6,42)
510  REM READ MAP INTO M ARRAY
520     FOR I=1 TO M9
530        READ N
540        IF I=N THEN 570
550           PRINT "DATABASE PROBLEM"I,N
560           STOP
570        FOR J=1 TO 6
580           READ M(J,I)
590        NEXT J
600     NEXT I
900  PRINT
905  PRINT "WHEN YOU  ANSWER A QUESTION, I LOOK AT"
906  PRINT "ONLY THE  FIRST LETTER, ALTHOUGH YOU CAN"
907  PRINT "TYPE THE WHOLE WORD IF YOU WANT."
920  GOSUB 7500
1000 REM -------------------------------------------------------
1010 N=5
1020 M0=0
1030 M6=0
1040 T=T1
1050 P=0
1060 P1=0
1070 FOR J=1 TO M9
1080    W(J)=0
1090 NEXT J
1100 PRINT
1110 REM DESCRIBE
1120    GOSUB 8000
1400 REM -------------------------------------------------------
1405 REM               MAIN LOOP STARTS HERE
1410 REM COUNT MOVES
1420    M0=M0+1
1430 REM MOVE
1440    GOSUB 6000
1450 REM CHECK FOR FINDING THE TREASURE
1460    GOSUB 2000
```

```
1470 REM TRY THE PIRATE
1480    GOSUB 4000
1490 REM LOOP UNLESS FINISHED
1500    IF T>0 THEN 1400
1510    IF N<>5 THEN 1400
1700 REM CALCULATE SCORE
1710    GOSUB 3000
1720    PRINT
1730    PRINT "CONGRATULATIONS! YOU GOT THE TREASURE"
1740    PRINT "OUT IN";M0;
1750    PRINT "MOVES AND YOU GOT"S+10"POINTS!"
1760    PRINT "WANT TO HUNT AGAIN? ";
1770    A$="YN"
1771    A2=2
1780    GOSUB 5000
1790    ON A1 GO TO 1000,9999,1760
2000 REM -----------------------------------------------------
2010 REM FOUND?
2100 IF T<>N THEN RETURN
2110 IF T<0 THEN RETURN
2120 IF M6+5>M0 THEN RETURN
2200 PRINT "DO YOU WANT TO TAKE IT WITH YOU? ";
2210 A$="YN"
2220 A2=2
2230 GOSUB 5000
2240 ON A1 GO TO 2300,2400
2250 PRINT "WELL?...."
2260 GO TO 2210
2300 T=-1
2310 PRINT
2320 PRINT "OK, LETS GET OUT OF HERE!"
2330 RETURN
2400 PRINT
2410 PRINT "WE'LL LEAVE IT HERE AND YOU CAN EXPLORE"
2420 PRINT "SOME MORE."
2430 M6=M0
2440 RETURN
3000 REM -----------------------------------------------------
3010 REM                   SCORE
3020 S=0
3030 IF T=-1 THEN S=S+5
3040 IF P=1 THEN S=S+10
3050 FOR J=2 TO N9
3060    S=S+W(J)
3070 NEXT J
3080 RETURN
4000 REM -----------------------------------------------------
4010 REM                   PIRATE
4020 IF N=T2 THEN RETURN
4030 IF P=1 THEN RETURN
4040 IF T1=T2 THEN RETURN
4050 IF T<>-1 THEN RETURN
4060    REM HES AT THE EXIT WITH THE TREASURE. ZAP HIM.
4065    REM (ARRGH. HOW DID HE GET HERE, ANYWAY?)
4070       IF N=16 THEN P=160
4080    REM COUNT MOVES SINCE HITTING TIGHT TUNNEL WITH TREASURE
4090       IF P1>0 THEN P1=P1+1
4100       IF N=3 THEN P1=P1+1
4110    REM GIVE HIM A FEW MORE MOVES, THEN ZAP HIM
4120       IF P1<15 THEN RETURN
4130          PRINT
4140          PRINT"SUDDENLY THE PIRATE LEAPS OUT OF THE"
4150          PRINT"GLOOM AND GRABS THE TREASURE FROM YOU!"
4160          PRINT"'HAH!', HE SHOUTS, 'YOU FOUND MY"
4170          PRINT"TREASURE, DID YOU?!  WELL, I'LL HIDE"
4180          PRINT"IT BETTER THIS TIME!'"
4190          PRINT"AND HE DISAPPEARS INTO THE DARKNESS"
4200          PRINT"WITH THE TREASURE."
4210          P=1
4220          T=T2
4230 RETURN
5000 REM -----------------------------------------------------
5010 REM                   INPUT
5020 REM FIRST CHARACTER OF A1 INPUT STRING IS COMPARED WITH
5030 REM THE LETTERS OF A$, AND IF THERE IS A MATCH, THE INDEX
5040 REM IN A$ IS RETURNED IN A1. IF NO MATCH, SIZE(A$)+1 IS
5050 REM RETURNED.
5060 REM GET INPUT STRING
5070 INPUT Q$
```

*Listing 2 continued on page 182*

*Listing 2 continued from page 181:*

```
5080  REM USE ONLY FIRST CHARACTER
5090    Q$ = LST(Q$,1)
5100  REM SEARCH FOR THE CHARACTER Q$ IN THE STRING A$. IN THIS VERSION
5110  REM OF BASIC, NDX IS THE INDEX FUNCTION, WHICH DOES EXACTLY THAT.
5120     A1=NDX(A$,Q$)
5130  REM BUT CHECK FOR THE CASE WHERE THE CHARACTER WAS NOT FOUND
5140     IF A1=0 THEN A1=A2+1
5145  RETURN
5150  REM IF YOUR VERSION OF BASIC DOESN-T HAVE THE NDX FUNCTION, BUT
5160  REM DOES, FOR INSTANCE, HAVE A FUNCTION WHICH WILL PICK A
5170  REM PARTICULAR CHARACTER FROM A STRING, SUCH AS MID(A$,A2,1)
5180  REM PICKING THE A2-TH CHARACTER FROM A$, YOU MIGHT USE THE
5190  REM FOLLOWING CODE.
5200  REM     FOR A1=1 TO A2
5210  REM        IF Q$=MID(A$,A1,1) THEN RETURN
5220  REM     NEXT A1
5230  REM     A1=A2+1
5240  REM     RETURN
6000  REM -------------------------------------------------------------
6010  REM               MOVE
6020  REM REMEMBER WHERE WE ARE, FOR BOUNCE.
6030     N9=N
6040  REM SET N8 TO ANYTHING BUT YOU CANT GO THAT WAY
6050     N8=0
6060  REM ASK WHICH WAY
6070     GOSUB 7000
6080  REM REMEMBER WHERE WE ARE, UNLESS A DEAD END
6090     IF N=1 THEN 6120
6100        N0=N
6110        A0=A1
6120  PRINT
6130  I=M(A1,N)
6200  IF I=-2 THEN I=N9
6210  IF D<>0 THEN PRINT '               DEBUG';N;'TO';I
6220  IF I<500 THEN 6300
6230     REM DELAY LOOP TO WASTE SOME TIME
6240        I=I-500
6250        FOR J=0 TO 100
6260        NEXT J
6270     GO TO 6200
6300  ON I/100 GO TO 6340,6370
6310     REM NORMAL ROUTE--LESS THAN 100
6320        N=I
6330        GO TO 6400
6340     REM N+100. ADD ONE IF CARRYING THE TREASURE
6350        N=I-100
6355        IF T=-1 THEN N=N+1
6360        GO TO 6400
6370     REM N+200. ADD 1 IF CARRYING TREASURE THE SECOND TIME
6380        N=I-200
6390        IF T=-1 THEN N=N+P
6400  IF N<>1 THEN 6500
6410     REM DEAD END.  TURN IT SO YOU GET OUT THE OTHER WAY
6420        FOR J=1 TO 6
6430           M(J,N)=2
6440        NEXT J
6450        M(7-A0,N)=N0
6500  REM PRINT OUT THE NODE DESCRIPTION
6510     IF N8<>2 THEN GOSUB 8000
6520  REM REMEMBER WEVE BEEN HERE
6530     W(N)=1
6540     N8=N
6600  IF M(1,N)<>-2 THEN 6800
6610     REM FORCED MOVE, WITH RANDOM DESTINATIONS
6620     REM ON THIS VERSION OF BASIC, J=-1 FOLLOWED BY RND(J)
6630     REM GETS YOU A NUMBER BETWEEN ZERO AND ONE.
6640     REM YOUR VERSION WILL DIFFER, AND THE NEXT FIVE
6650     REM LINES WILL HAVE TO BE CHANGED.
6660        I=M(6,N)
6670        J=-1
6680        IF M(4,N) > 100*RND(J) THEN I=M(5,N)
6690        J=-1
6700        IF M(2,N) > 100*RND(J) THEN I=M(3,N)
6710     IF D<>0 THEN PRINT '               DEBUG BOUNCE TO';I
6720     REM NOW HAVE A NEW DESTINATION.  GO BACK AND REDO IT
6730        GO TO 6200
6800  RETURN
7000  REM -------------------------------------------------------------
7010  REM               WHICH WAY?
```

*Listing 2 continued on page 184*

Circle 53 on inquiry card.

# The Best MICRO-SOFTWARE

**GENERAL LEDGER • ACCTS. PAYABLE**
**ACCTS. RECEIVABLE • INVENTORY**
**PAYROLL**

for TRS • 80 Apple II Micropolis
Vector MZ Exidy Sorcerer Cromemco
CBasic-CP/M Microsoft-CP/M

Buy simple, effective programs designed specifically for your machine. Call us once for same week delivery on a product you can use the day you receive it, and two years from now when your company is twice as big. **And ask about MICROMAX** - our computer output videotape system.

**TRS • 80: $100/each. CBASIC-CP/M: $200/each. Apple II, Micropolis, Vector, Exidy, Cromemco, and Microsoft: $140/each.**

Order today by U.P.S. COD. We'll pay postage and handling on Visa, Mastercharge, or prepaid orders.

**505 Hamilton Ave., Palo Alto, CA 94301** *(415) 321-2881*

COMPUMAX ASSOCIATES, INC.

*Announcing. . .*

## THE GUIDE TO DESKTOP MICROCOMPUTER SELECTION

A guide to the selection of a desktop computer that is *right* for your needs. Over 100 pages are crammed with features & specifications on models offered by the leading suppliers in the industry today. Manufacturers such as Radio Shack, Apple, Wang, Tektronix, IBM & Hewlett-Packard are included.

*Other features include:*

- a matrix of BASIC statements with over 250 entries comparing the capabilities of 8 different BASIC Interpreters.
- a matrix of hardware characteristics comparing desktop systems and their associated options.
- capabilities to consider when determining your needs & options.
- space within each matrix to include your choice of system not listed.
- key elements for development of software which can convert from one computer to another.

**TO OBTAIN YOUR COPY, RUSH $13.65 PLUS $1.35 POSTAGE AND HANDLING TO:**

AAC ATLANTIC ANALYSIS CORPORATION

5 KOGER EXECUTIVE CENTER • SUITE 219 • NORFOLK, VIRGINIA 23502
(804) 461-1980
VA Residents, add 4% sales tax

## TRS-80 means business! ! !
## ...with CP/M, CBASIC2, & applications software.

CP/M ("the software bus") & CBASIC2 (the standard for business software) bring new power and versatility to the TRS-80 for practical use.

CP/M Operating System (w/Editor, Assembler, Debugger, Utilities & 6 manual set) . . . . . . . $150
CBASIC2 Compiler (w/manual) . . . . . . . . . . . . $ 95
DESPOOL Print Spooler (w/manual). . . . . . . . . $ 75

*Osborne & Assoc. **PAYROLL W/COST ACCTNG** . . . . . . . . . . . . . . . . . . . . . . . $250
*Osborne & Assoc. **ACCTS. RECEIVABLE & ACCTS. PAYABLE** . . . . . . . . . . . . . . . . . . $250
*Osborne & Assoc. **GENERAL LEDGER**. . . . . . $250

*=CBASIC2 source programs; add $15 (each) for applicable O & A book.

**APH** - Self-administered Automated Patient History. . . . . . . . . . . . . . . . . . . . . . . . . . $175

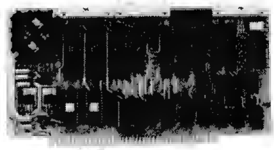
Write/call for additional software available & FREE "CP/M Primer".

CA residents add 6% tax. Visa, M/C,M/O,Check OK.

CYBERNETICS INC (714) 848-1922

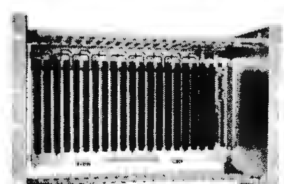
8041 NEWMAN AVENUE • SUITE 208 • HUNTINGTON BEACH, CALIFORNIA 92647

BUILDING BLOCKS FOR MICROCOMPUTER SYSTEMS, CONTROL & TEST EQUIPMENT



**R² I/O**
2K ROM 2 K RAM
3 SERIAL PORTS 1 PARALLEL PORT

**TT-10**
TABLE TOP MAINFRAMES



**ECT-100-F**
RACKMOUNT CARD CAGES

***POWER SUPPLIES, CPU's, MEMORY, OEM VARIATIONS***

SPECIALIZING IN QUALITY MICRO-COMPUTER HARDWARE

INDUSTRIAL
EDUCATIONAL
SMALL BUSINESS
PERSONAL

**ELECTRONIC CONTROL TECHNOLOGY**

763 RAMSEY AVE.
HILLSIDE, N.J. 07205
**(201) 686-8080**

Circle 85 on inquiry card.

Circle 120 on inquiry card.

*Listing 2 continued from page 182:*

```
7100  PRINT
7110  PRINT "                          WHICH WAY? ";
7120  A$="NEUDWSP"
7130  A2=7
7140  GOSUB 5000
7150  IF A1<8  THEN 7300
7160     PRINT "WHICH WAY DO YOU WANT TO GO?"
7170     REM GIVE INSTRUCTIONS
7180        GOSUB 7500
7190     REM DESCRIBE THE LOCATION AGAIN
7200        GOSUB 8000
7210     GO TO 7100
7300  IF A1<7 THEN 7400
7310     REM CALCULATE AND PRINT SCORE
7320        GOSUB 3000
7330        PRINT "YOU HAVE"S"POINTS!"
7340     REM START AGAIN
7350        GO TO 7100
7400  RETURN
7500  REM ---------------------------------------------------------
7510  REM     SUBROUTINE TO GIVE INSTRUCTIONS
7520     PRINT
7530     PRINT "TYPE N,S,E,W,U, OR D FOR NORTH, SOUTH,"
7550     PRINT "EAST,WEST, UP OR DOWN.  TYPE P FOR SCORE"
7560     PRINT
7570     RETURN
8000  REM ---------------------------------------------------------
8010  REM DESCRIBE THE CURRENT LOCATION
8050     I=INT(N/5)
8060     J=N-5*I+1
8070     REM THERE ARE ENOUGH STATEMENT NUMBERS HERE TO HANDLE NODES
8080     REM ZERO THROUGH 49.  YOU WILL HAVE TO ADD MORE IF YOU ADD
8090     REM NODES 50 AND BEYOND.
8100     ON I+1 GO TO 8200,8210,8220,8230,8240,8250,8260,8270,8280,8290
8200        ON J GO TO 9000,9010,9020,9030,9040
8210        ON J GO TO 9050,9060,9070,9080,9090
8220        ON J GO TO 9100,9110,9120,9130,9140
8230        ON J GO TO 9150,9160,9170,9180,9190
8240        ON J GO TO 9200,9210,9220,9230,9240
8250        ON J GO TO 9250,9260,9270,9280,9290
8260        ON J GO TO 9300,9310,9320,9330,9340
8270        ON J GO TO 9350,9360,9370,9380,9390
8280        ON J GO TO 9400,9410,9420,9430,9440
8290        ON J GO TO 9450,9460,9470,9480,9490
8400     IF T<>N THEN 8500
8410        PRINT
8420        PRINT "THE TREASURE IS HERE!"
8500     IF T<>T2 THEN 8600
8510     IF T1=T2 THEN 8600
8520     IF T1<>N THEN 8600
8530        PRINT
8540        PRINT "A NOTE ON THE WALL SAYS"
8550        PRINT "  'PIRATES NEVER LEAVE THEIR TREASURE"
8560        PRINT "   TWICE IN THE SAME PLACE!"
8600     RETURN
9000  REM ---------------------------------------------------------
9001  REM FIRST DATA STATEMENT IS NUMBER OF NODES, AND THE 2
9002  REM HIDING PLACES FOR THE TREASURE.
9003  DATA 42,23,12
9010  DATA 1,0,0,0,0,0,0
9011     PRINT "YOU'RE AT A DEAD END!"
9012     GO TO 8400
9020  DATA 2,-2,101,-2,0,0,0
9021     PRINT "YOU CAN'T GO IN THAT DIRECTION"
9022     PRINT
9023     GO TO 8400
9030  DATA 3,33,2,1,10,106,4
9031     PRINT "A TUNNEL GOES NORTH-SOUTH."
9032     PRINT "THERE IS AN OPENING TO THE WEST."
9037     GO TO 8400
9040  DATA 4,3,30,2,11,2,1
9041     PRINT "YOU'RE ON THE BRINK OF A PIT."
9047     GO TO 8400
9050  DATA 5,8,8,15,10,8,16
9051     PRINT "YOU'RE OUTSIDE THE CAVE."
9052     PRINT "GO SOUTH TO ENTER."
9053     GO TO 8400
9060  DATA 6,16,3,2,10,2,2
9061     PRINT "YOU'RE AT THE HOME OF THE GNOME-KING."
9062     PRINT "FORTUNATELY, HE'S GONE FOR THE DAY"
9067     GO TO 8400
9070  DATA 7,-2,101,-2,0,0,0
```

```
9071      PRINT "THE GNOME KING IS HERE!"
9072      PRINT "YOU'D BETTER GET OUT!"
9073      PRINT
9077      GO TO 8400
9080   DATA 8,18,18,15,10,18,9
9081      PRINT "YOU'RE LOST IN THE WOODS."
9087      GO TO 8400
9090   DATA 9,-2,33,5,1,0,-2
9097      GO TO 8400
9100   DATA 10,-2,101,-2,0,0,0
9101      PRINT "YOU'RE NOT GOING TO GET FAR, DIGGING"
9102      PRINT "THROUGH ROCK."
9103      PRINT
9107      GO TO 8400
9110   DATA 11,1,13,4,2,1,2
9111      PRINT "YOU'RE AT THE BOTTOM OF A PIT. A LITTLE"
9112      PRINT "STREAM FLOWS OVER THE ROCKS HERE."
9117      GO TO 8400
9120   DATA 12,36,2,1,2,1,2
9121      PRINT "YOU'RE AT A DEAD END!"
9127      GO TO 8400
9130   DATA 13,2,37,2,1,11,14
9131      PRINT "YOU'RE AT A WIDE SPOT.  THERE IS A"
9132      PRINT "SOOTY PATCH WHERE SOMEBODY HAS RESTED"
9133      PRINT "A TORCH AGAINST THE WALL.  THERE ARE"
9134      PRINT "JAGGED ROCKS ABOVE YOU."
9137      GO TO 8400
9140   DATA 14,13,1,19,2,31,31
9141      PRINT "YOU'RE IN A CANYON. HIGH ON THE WALL"
9142      PRINT "ABOVE YOU IS SCRATCHED THE MESSAGE"
9143      PRINT "     'BILBO WAS HERE'"
9147      GO TO 8400
9150   DATA 15,-2,101,-2,0,0,0
9151      PRINT "YOU'RE NOT A BIRD. YOU CAN'T FLY!"
9152      PRINT
9157      GO TO 8400
9160   DATA 16,5,33,2,10,1,106
9161      PRINT "YOU'RE IN A LOW CHAMBER. A TIGHT TUNNEL"
9162      PRINT "GOES EAST, AND YOU CAN WALK TO THE"
9163      PRINT "SOUTH OR WEST. THERE IS LIGHT"
9164      PRINT "TO THE NORTH."
9167      GO TO 8400
9170   DATA 17,-2,101,-2,0,0,0
9171      PRINT "IT'S A TIGHT SQUEEZE. YOU CAN'T"
9172      PRINT "GET PAST WITH THE TREASURE."
9173      PRINT
9177      GO TO 8400
9180   DATA 18,-2,101,8,0,0,0
9181      PRINT "I DON'T THINK YOU CAN FIND THE CAVE."
9187      GO TO 8400
9190   DATA 19,224,2,2,14,1,42
9191      PRINT "YOU'RE AT THE  TOP OF A CLIMB."
9192      PRINT "BELOW YOU A MESSAGE SAYS"
9193      PRINT "     'BILBO WAS HERE'"
9197      GO TO 8400
9200   DATA 20,226,1,2,2,25,2
9201      PRINT "YOU'RE AT THE NORTH SIDE OF A CHASM,"
9202      PRINT "TOO WIDE TO JUMP.  RINGING ECHOES FROM"
9203      PRINT "BELOW ARE THE ONLY INDICATION OF DEPTH."
9207      GO TO 8400
9210   DATA 21,1,226,2,2,38,25
9211      PRINT "YOU'RE IN XANADU. BELOW YOU"
9212      PRINT "ALPH, THE SACRED RIVER RUNS"
9213      PRINT "THROUGH CAVERNS MEASURELESS TO MAN,"
9214      PRINT "DOWN TO A SUNLESS SEA."
9217      GO TO 8400
9220   DATA 22,-2,33,13,50,29,30
9227      GO TO 8400
9230   DATA 23,2,1,2,31,2,2
9231      PRINT "YOU'RE ON THE LEDGE ABOVE THE GUILLOTINE ROOM."
9237      GO TO 8400
9240   DATA 24,-2,101,19,0,0,0
9241      PRINT "I HEAR THE GIANT THERE!!!"
9242      PRINT "YOU'D BETTER GO BACK!"
9243      PRINT
9247      GO TO 8400
9250   DATA 25,21,20,2,2,1,19
9251      PRINT "YOU'RE IN THE GIANT'S CAVERN.  BETTER"
9252      PRINT "NOT BE HERE WHEN THE GIANT COMES!"
9257      GO TO 8400
9260   DATA 26,-2,65,-2,50,11,14
9261      PRINT "YOU'RE IN THE QUEST RESEARCH AND"
```

*Listing 2 continued on page 186*

*Listing 2 continued from page 185:*

```
9262     PRINT "DEVELOPMENT AREA."
9263     PRINT
9264     PRINT "I'M SORRY, BUT VISITORS ARE NOT"
9265     PRINT "ALLOWED.  YOU'LL HAVE TO LEAVE."
9266     PRINT
9267     GO TO 8400
9270  DATA 27,2,40,2,2,21,20
9271     PRINT "YOU'RE IN THE CRYSTAL PALACE. THE"
9272     PRINT "WALLS RESONATE WITH"
9273     PRINT "AWESOME MUSIC."
9280  DATA 28,-2,60,221,50,14,19
9287     GO TO 8400
9290  DATA 29,2,42,2,13,1,1
9291     PRINT "YOU'RE AT THE  TOP OF A GIANT STALACTITE."
9292     PRINT "YOU COULD SLIDE DOWN, BUT YOU COULDN'T"
9293     PRINT "CLIMB BACK UP."
9297     GO TO 8400
9300  DATA 30,34,34,2,1,4,2
9301     PRINT "YOU'RE IN A LITTLE GROTTO. THERE IS A"
9302     PRINT "BOOK HERE CALLED JANE'S FIGHTING SHIPS,"
9303     PRINT "DATED 1763."
9307     GO TO 8400
9310  DATA 31,14,14,23,2,1,2
9311     PRINT "YOU'RE IN THE GUILLOTINE ROOM. A SHARP"
9312     PRINT "ROCK BALANCES PRECARIOUSLY ON THE"
9313     PRINT "LEDGE ABOVE YOU."
9317     GO TO 8400
9320  DATA 32,-2,101,516,0,0,0
9321     PRINT "YOU'RE IN A CHUTE, SCRAMBLING DOWN THE"
9322     PRINT "ROCKS! NO WAY TO STOP! HANG ON!"
9323     PRINT
9327     GO TO 8400
9330  DATA 33,2,1,2,1,116,3
9331     PRINT "THE TIGHT TUNNEL TURNS A CORNER."
9332     GO TO 8400
9340  DATA 34,1,35,2,1,30,30
9341     PRINT "YOU'RE IN A LITTLE TWISTY MAZE"
9347     GO TO 8400
9350  DATA 35,2,1,2,37,34,36
9351     PRINT "YOU'RE IN A LITTLE TWISTING MAZE"
9357     GO TO 8400
9360  DATA 36,35,2,1,37,34,12
9361     PRINT "YOU'RE IN A TWISTING LITTLE MAZE"
9367     GO TO 8400
9370  DATA 37,2,1,35,2,13,2
9371     PRINT "YOU'RE IN A TWISTY LITTLE MAZE"
9377     GO TO 8400
9380  DATA 38,2,21,2,116,1,2
9381     PRINT "YOU'RE IN A PREHISTORIC DWELLING. ON"
9382     PRINT "THE WALL ARE DRAWINGS OF BISON DONE IN"
9383     PRINT "RED CLAY. THE FLOOR IS STREWN WITH"
9384     PRINT "BONES, THE REMAINS OF ANCIENT RITUALS."
9385     PRINT "A SMALL TUNNEL GOES THROUGH THE FLOOR."
9387     GO TO 8400
9390  DATA 39,2,40,2,32,21,26
9391     PRINT "YOU'RE IN A BLACK HOLE.  THE"
9392     PRINT "FORCE OF GRAVITY IS OVERWHELMING."
9397     GO TO 8400
9400  DATA 40,40,40,2,2,40,41
9401     PRINT "YOU'RE IN THE LABYRINTHE"
9407     GO TO 8400
9410  DATA 41,40,40,40,2,40,39
9411     PRINT "YOU'RE IN THE LABYRINTHE"
9412     PRINT "IT'S VERY DARK IN HERE."
9417     GO TO 8400
9420  DATA 42,28,28,28,28,28,28
9421     PRINT "YOU'RE IN THE ASHRAM.  INCENSE IS HEAVY"
9422     PRINT "IN THE AIR, AND ALL DIRECTIONS"
9423     PRINT "SEEM THE SAME."
9427     GO TO 8400
9430  REM NO NODES SET UP FOR THESE VALUES.  IF YOU GET HERE,
9440  REM YOU HAVE A MISTAKE IN A DATA STATEMENT.
9450  REM
9460  REM
9470  REM
9480  REM
9490  REM
9500  REM
9510  REM
9520  REM
9530  REM
9999  END
```

*Text continued from page 179:*

modifications on most machines which have enough memory to hold it. The number of locations has been increased slightly, and a few surprises have been added to make it more interesting. The original Quest was made for seventh-graders at Peninsula School, who were doing a map making unit. For this reason, most of the connections between locations work as if they were in physical three-dimensional space, although there is no requirement for this in the program.

The program and the description of the cave are well documented by the comments in the code, so I won't go into great detail here. Besides its description, each location has a set of six numbers which give the next location to move to, in case of a move NORTH, EAST, UP, DOWN, WEST, or SOUTH. Special events, such as the pirate and the treasure, are done in the program rather than in the descriptions. There is a provision in the connection codes for an immediate return to the location you came from, which is used, for instance, at the location called "YOU CAN'T GO IN THAT DIRECTION." There is also provision for different connections chosen according to a random number, and for different connections depending on whether or not you are carrying the treasure, and whether or not the pirate has found you.

## A Final Statement

It is possible to get through the cave by reading the program and decoding the data which defines the connections. If you do that, you will deprive yourself of the pleasure of finally finding your way through. It is also possible to "help" a friend by telling him how to get through. I don't think the easy pleasure of knowing how to get through can equal the joy of discovering the way, or the satisfaction of having discovered it, or the excitement of being on the way to discovering it. I also don't think that anyone who merely plays Quest can have as much fun as I have had in writing it, and watching other people use it.

## Acknowledgments

My thanks go to Larry Tesler and Phyllis Cole, of the Peninsula School Computer Project, for their encouragement and technical help, and to Mary Artibee for her help with this article. A tape of the PET version of Quest is available for $9.95 from the Peninsula School Computing Project, Menlo Park CA 94025.■

# Book Reviews

### Techniques of Program Structure and Design

*by Edward Yourdon*
*Prentice-Hall, 1975*
*364 pages hardbound*
*$19.50*

As you might surmise from the title, this textbook promotes structured programming, design, and testing concepts. Although familiarity with current hardware, systems, or software doctrines is not a prerequisite for reader comprehension, the greatest benefits will accrue to those individuals active in software development or maintenance efforts.

The book's tone and structure are established in the first chapter with a discussion of the seven most desirable qualities of a good program. The program should work well and according to specification; the simplicity of design should reduce development, testing, and maintenance costs; and the program's inherent flexibility should allow change through expansion, modification, or upgrade.

Of perhaps more importance to the average programmer is the analysis of ten practices which commonly exacerbate the debugging problem. Also included are suggestions for easing the difficulties of maintaining or modifying existing programs for the maintenance programmer.

The next chapter deals with top-down design, coding, and testing. Top-down design is the process by which the programmer identifies the major functions of a programming problem, and organizes the solution in such a manner that it is recognizable to both the computer and the maintenance programmer.

Mr Yourdon provides five suggestions for successfully applying this concept. The benefits and disadvantages of flowcharts figure prominently in the arguments for and against the concept of writing code concurrently with designing the top-down structure. An offshoot of this discussion of top-down coding concerns how best to display modular code: the horizontal and vertical approaches have unique merits which should be considered prior to presenting the code.

Following the presentation of the nature and advantages of top-down testing, the author deals with practical variations of the *pure approach* to top-down design, coding, and testing. These modifications are frequently the result of organizational problems and management compromises. Two of the most successful variations discussed are the IBM developed *structured walkthrough* technique and IBM's highly innovative *chief programmer team* concept.

In Chapter Three, the reader is introduced to the concept of modular programming (precursor to the now popular structured programming technique). After a discussion of the characteristics of a programming module, the pros and cons of modularity are detailed. Techniques for achieving modular programs are of particular interest, especially those sections dealing with decision tables, separate I/O (input/output) functions, and use of symbolic parameters. The chapter concludes with the closely related subject of general purpose subroutines.

After a review of the history and background of the structured programming movement, both the theoretical and practical aspects of the concept are detailed in Chapter Four. While analyzing structured programming's objectives and motivations (in terms of testing, productivity, clarity, and efficiency), Mr Yourdon highlights significant differences between structured and modular approaches to program design. Although structured programming is becoming the standard method of achieving modular programs, the concept involves more than the simple conjunction of "GOTO-less" programming with the process, loop, and binary-decision constructs. As with most evolving techniques, there are tradeoffs in terms of efficiency, convenience, hardware and software requirements which concern many of those involved with computer science. The structured technique is not a panacea for programming ills.

Mr Yourdon has found that imposing structure upon poorly designed, unstructured programs usually results in poorly designed, GOTO-less, structured programs that are almost as difficult to understand and debug as the originals. As a result, he recommends that experienced programmers rework their previously unstructured code *only* as a means of bringing their thought processes into compliance with current top-down design concepts. There are three common techniques for restructuring this code, each with its own strengths and weaknesses: the *duplication-of-coding* technique which is recommended for programs with network or lattice structures; the *state-variable* technique which is helpful to the maintenance programmer and can be applied to very complex processing programs; and the *Boolean flag* technique which is applicable to loop-oriented programs.

The chapter concludes with a discussion of the applicability of structured programming to the currently available high level programming, systems implementation, and assembly languages. ALGOL, PL/I, COBOL, FORTRAN and assembly languages are covered quite extensively; PL/S, BLISS, PL/360, Burroughs B5500 ESPOL, Project MAC EPL, and GE-645 MULTICS languages are discussed in less detail. I noted with interest the author's claim that FORTRAN and assembly languages do not provide the necessary facilities for structured programming.

In Chapter Five, programming method, system software, and system hardware figure prominently in the discussion of the elements of programming style which dramatically affect the coding and debugging process. Of particular note are the questions concerning how to best reduce program complexity so as to limit the introduction of errors, and how to construct programs so that someone other than the original programmer can read and comprehend the logic flow.

The focal point of the next chapter is *defensive* programming — the practice of writing programs in such a way that the inevitable bugs are readily noticeable to both programmer and user. Effective arguments are presented against the most common objections to this antibugging technique. Following the presentation of a comprehensive list of aspects of a computer program which require checking, the author summarizes nine useful error-checking techniques. Although many of the antibugging techniques belong more properly in the realm of system, rather than program design, the discussion is valid at both the programming and coding levels.

Testing is reviewed in Chapter Seven. The magnitude of this effort can reach staggering proportions on projects which utilize many programmers, large numbers of manhours, and vast segments of code. Current module, system, and acceptance testing techniques expose most common types of error. These errors are divided into eight categories: logic, documentation, overload, timing, throughput and capacity, fallback and recovery, system hardware/software, and standards errors.

It is Mr Yourdon's contention that a program is economical to test and debug only if the team programming concept is used in combination with an antibugging programming style and an unambiguous, structured language. Also, because programmers use slow, tedious methods and are psychologically inclined to justify their output as correct, he advocates the removal of the human element from the testing process. Ideally, the testing process would be fully automated: the automatic test harness would collect the test input from the automatic test data generator, pass it to the program being tested, and use an automatic output checker to list the discrepancies. Upon program execution, the automatic testing monitor would print a report showing the portions of the program exercised by the test data. The program would be subjected to a thorough retesting if it is further modified. The chapter concludes with a review of some of the experimental techniques currently being evaluated for their abilities to increase program reliability.

Once the existence of a bug has been established, it is time to employ the technique of finding and correcting the error, known as debugging. The procedure is more art than science, and in Chapter Eight the reader is provided with both an explanation for 26 common bugs and 11 detailed suggestions to assist in the formulation of a workable debugging strategy. A terminal-oriented debugging system greatly reduces the frustration of using memory dumps and program traces. The system is called DDT or Dynamic Debugging Technique. After describing the general features of the process, the author expands his explanation on the currently available stand alone and sophisticated timesharing and realtime packages. The chapter concludes with the implementation of a simple version of DDT.

The four programming exercises in the appendices illustrate the principles of program design and structured programming, and allow the reader to apply the techniques presented in the book to real problems. The two management system problems are quite complex and can best be attacked by a team of three to four programmers. The master file update and tic-tac-toe problems are more simple and are suitable for individual programmers.

This highly informative book is packed with information for both the commercial and personal aspects of programming. The book's heavy concentration on the general philosophy and techniques of good programming deserves the highest accolades. *Techniques of Program Structure and Design* should be a part of every dedicated computer user's library.

**Lee C Matthews**
**9659 B Jefferson**
**Ellsworth AFB, SD 57706■**

# Book Reviews

**Top-down Structured Programming Techniques**
*by Clement L McGowan and John R Kelly*
*Mason/Charter Publishers*
*Brown University,*
*New York 1975*
*288 pages hardback*
*$15.95*

This book provides a good introduction to the top-down structured programming concept. This concept requires, as a first step, that a problem's functional specifications be determined. The problem is then refined until that programming code directly expresses the resulting subfunctions, or modules. The authors believe that programmers who use the top-down structured methodology are more productive, and consistently produce a reliable product which works despite continuous testing or future modification. The book also details recent programming advances in the large system, multiprocessing environment.

Chapter 1 presents an overview of structured programming with an emphasis on the development of a language-independent methodology whose principal goal is program reliability. Of special interest is the section on the importance of developmental and modification software (in terms of time, cost, and reliability). Following a preview of the remaining chapters, the authors discuss their reason for illustrating programming concepts with code fragments in the PL/I language.

Chapter 2 emphasizes program reliability. The data processing industry spends a great amount of time and money to insure that programs are developed with as few flaws as possible. The current doctrine of testing programs for reliability is gradually giving way to a more efficient method of checking for correctness. Finally, the DO-WHILE construct is introduced by emphasizing the importance of properly initializing conditional loops, and explicit guidelines for programming conditional iterations are given.

In the next chapter, the SEQUENCE and IF-THEN-ELSE programming tools are discussed. Correctness questions, which the programmer should pose while using these constructs, are used as an aid in determining the functional specifications of the program. Program specifications must be determined prior to commencing the actual coding process because the authors consider inaccurate specifications to be the most important source of software errors. The ITERATIVE-DO, SELECT-CASE, REPEAT-UNTIL, and LOOP-EXITIF-ENDLOOP constructs are formed by combining the basic figures.

Unfortunately, the authors elect to leave the related correctness questions as an exercise for the reader, feeling that ". . .the reader is now sufficiently aware of the correctness considerations to properly examine his use of the nonbasic figure." Although the authors admit to writing for the computer professional, rather than the mathematician or newcomer to the computer field, I feel they did their readers a disservice. These correctness questions are vital (without them, the book suffers from a lack of comprehensiveness which limits its usefulness to both the salaried professional and computer hobbyist). Chapter 3 concludes with an interesting section on how to impose structure on programs written in FORTRAN and COBOL. The examples are illuminating.

The advantages of the top-down structured approach to program design, coding, and integration are discussed in Chapter 4. Top-down segmented implementation is compared (somewhat one-sidedly) with the more standard bottom-up strategy. Among the advantages discussed is the following: the effectiveness of top-down structured programming in eliminating construct errors (missing paths, inappropriate path selection, and inappropriate action under a given condition). The top-down method concentrates on the more time-consuming and error prone aspects of programming during the initial, rather than the final stages of development. A structured program has the advantage of becoming its own principal documentation. And the final advantage, especially associated with large systems programs, is that integration and debugging are easier to implement under the top-down strategy. The programming technique of recursion is introduced in the concluding sections of the chapter.

The remaining two chapters and two appendices are of less interest to the computer hobbyist. Chapter 5 examines IBM's innovative "chief programmer team" organization and its managerial approach to the operation of software projects. Chapter 6 provides extended examples of the top-down structured approach to program design, with the primary focus on parallel processes (whereby several computations proceed either simultaneously on separate processors, or by multiprogramming on a single processor) and multiprocessing resource management. The chapter concludes with the specification and top-down design in pseudocode, of a hypothetical multiprogrammed, multiprocessing operating system. The REPEAT-UNTIL, SELECT-CASE, and LOOP-EXITIF-ENDLOOP structured modules are implemented in PL/I in Appendix A. In Appendix B, six of the seven structured programming constructs are specified in 360/370 assembly language macros.

As a result of the omission mentioned earlier, I have qualms about recommending this book as a mandatory reference resource. If you are interested in using structured techniques to increase your programming productivity and reliability, I would recommend that you get a copy of the book through a nearby college or computer club library and spend a few hours reading the material. I found the authors' style lucid, although on occasion I had difficulty following some of their examples due to my unfamiliarity with the PL/I language.

Overall, the book successfully presents the authors' contention that structured coding, top-down design, formal and informal proofs of correctness, chief programmer team organizations, and code reading result in reliable, flexible software. I fully concur with their belief that "...top-down structured programming is one way to make programming the enjoyable activity it should be."

**Lee C Matthews**
**9659B Jefferson**
**Ellsworth AFB SD 57706**■

# Programming Quickies

## BASIC Bit Twiddling

**Ralph Owens**
**POB 202**
**Enterprise KS 67441**

The following are several functions which allow the user to manipulate individual bits and nybbles (groups of four bits) using BASIC. I like to use these functions with the PEEK and POKE commands.

| | |
|---|---|
| Nybble functions | A contains the value of the byte.<br>N contains the value of the nybble. |
| FNU(N) = INT(A/16) | Reads upper nybble. |
| FNL(N) = A − 16*FNU(N) | Reads lower nybble. |
| FNA(N) = 16*N + FNL(N) | Sets upper nybble to value of N. |
| FNB(N) = FNU(N) + N | Sets lower nybble to value of N. |
| Bit functions | A contains the value of the byte.<br>N (0 thru 7) is number of desired bit. |
| FNQ(N) = INT(A/2↑N) − 2*INT(A/2↑(N + 1)) | Reads Nth bit. |
| FNR(N) = A − FNQ(N)*2↑N | Resets Nth bit to zero. |
| FNS(N) = A + (1 − FNQ(N))*2↑N | Sets Nth bit to one. |
| FNT(N) = FNR(N) + FNS(N) − A | Toggles the Nth bit. |

Here are two examples of the use of these functions:

```
1000  REM FIND LOWER NYBBLE VALUE
1010  A=PEEK(16422)
1020  V= FNL(A)
1030  PRINT "VALUE OF LOWER NYBBLE
      IS"; V
9999  END

2010  A=PEEK(16422)
2020  REM SET EIGHTH BIT OF DECIMAL
2030  REM LOCATION 16422 TO ONE
2050  A= FNS(8)
2060  POKE 16422, A
9999  END
```
■

# World Power Systems: A Report

This spring, World Power Systems Inc of Tucson AZ began "business" with a promotion campaign, ordering goods from suppliers, and soliciting orders and money from customers. It is now known that only a few initial orders were filled (apparently to establish credibility), and that money accompanying later orders was simply stashed away.

Because of this, our industry has received a black eye, and many of us, BYTE included, are apparently out of substantial amounts of money. Acting on a tip received April 25, the Pima County Attorney's Office promptly investigated and, within a few days, had sufficient evidence of illegal activities to obtain warrants for the arrest of those responsible for the operation. An arrest was made, goods were seized and impounded, and others, who were out of Tucson at the time, are being sought by the authorities.

The question that all of us must be asking is "How can we prevent this from happening again?"

It would be nice if there were a simple answer. The main things consumers can do are (1) don't send money unless you have reason for confidence in the company and (2) promptly report (eg: to postal authorities) anything suspicious.

What neither consumers nor the media can afford to do, however, is to establish a presumption of guilt. Just because a company is new does not mean that the media can refuse advertising or that consumers should avoid the company's products. Such actions would only lead to stagnation in the industry and possible claims of unfair competition and/or antitrust violations. Indeed, since many of the biggest companies in the industry began in founders' homes, such a policy might have retarded the development of the personal computer industry by several years.

BYTE has had, and continues to have, stringent standards on the advertising it accepts. Advertisements have been rejected for such reasons as being tasteless, misleading, or having originated from unreliable companies.

As to reliability, we have to depend in part on readers' input. Financial reports on advertisers are, unfortunately, of limited value. Some financially unstable companies are totally honest, while others, with a superficially impressive report, may turn out to be fraudulent.

At BYTE, any reader's complaints about advertisers are sent to the advertiser along with our request that BYTE be notified of what action is being taken. Since there are always two sides to every argument, having a single disgruntled customer would be inadequate reason for sanctions against the advertiser. A series of complaints without adequate response by the company, however, is cause for BYTE refusing to run further advertisements by that company.

In summary, there is no absolute protection in any industry against well-planned and well-executed frauds. Prompt communication between (1) consumers, (2) reputable companies, (3) the media, and (4) the authorities can nip poorly planned frauds in the bud and limit the damage that is done.

Chris Morgan
BYTE Executive Editor





Circle 169 on inquiry card.

Circle 172 on inquiry card.

# Photo Essay: Physical Hardware of a New Computer Backplane

by Carl Helmers

As noted in a recent editorial, I am in the process of designing and building a new computer system based on the 6809 processor. It is my intention, as this design evolves, to provide a fairly complete set of plans in the form of an irregular series of articles in BYTE. The building and developing of the software systems of homebrew computers is, after all, the basis from which personal computing has developed. This series of articles will document the development of an up-to-date design that utilizes contemporary components which were unavailable to me when I first started building crude and imperfect homebrew computers in 1974.

The regularity of this documentation will be dependent upon the time that is available to engineer, build and test the component parts. I will try to provide an update on the progress of the project with each future issue of BYTE. From time to time there will be gaps in the series since, like all people, I have only 24 hours each day in which to work. Having issued this caveat concerning the irregularity of the information, let us turn to the starting point of my documentation; a physical basis for a bus oriented homebrew computer.

The following is documentation of a key part of any homebrew computer; its backplane. At this stage, the computer system is depicted in photographs 1 through 5, which I took while assembling the backplane on a recent evening. For the photographers among our readers, all these pictures were taken with a 35mm single lens reflex (SLR) camera, highly stopped down (f/32, f/22, depending on lens) to emphasize depth of field, and using lots of light (1200 W).

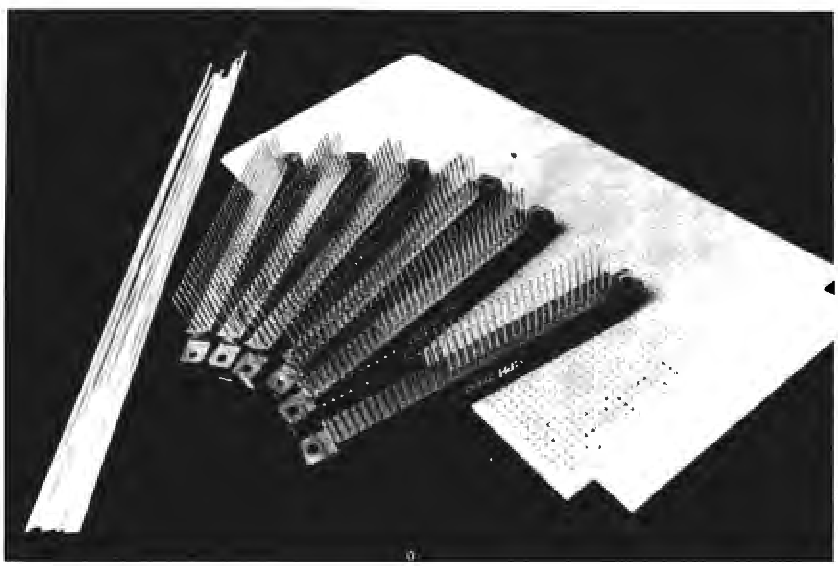


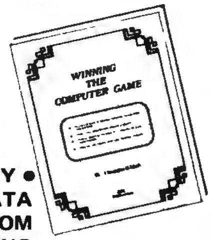
*Photo 1: This pile of parts represents the beginning of the project's physical hardware assembly. At left is a set of eight copper rods made from #12 gauge household electrical wire. After stripping the insulation, one end of the wire is clamped in my bench vise, and the other end is clamped in "Vise-Grip" pliers. Five uniform, careful rotations of the wire while holding tension suffice to make the rods as straight as those shown in the photo. This torsion straightening process gives amazing results. These rods will be the bus wires of the power and ground distribution.*

*The matrix for assembly of the backplane is one Vector Electronic Co #3719—4 "P" pattern prototyping board. This board was chosen as standard for the new computer because it has a 0.1 inch (.254 mm) square grid consistent with integrated circuit sockets, and an identical 0.1 inch spacing for the edge connector sockets. Thus, the same board style that will be used for the assembly of the computer modules can also be used for assembly of this motherboard.*

*The final part that is going into the physical assembly of the backplane is a set of 6 edge connector sockets for the circuit modules. The sockets have 72 circuit connections in two rows of 36 pins.*

Circle 89 on inquiry card.

# HOW TO BUY A BUSINESS COMPUTER




WHEN TO BUY • WHEN NOT TO BUY • HOW TO CHOOSE CONSULTANTS • DATA GATHERING • PACKAGED VS CUSTOM SOFTWARE • CONTRACTS • FINANCING • WARRANTEES AND MAINTENANCE • INSTALLATION MANAGEMENT AND MUCH MORE .....

**"SUPER BOOK!" DP Consultant, Mercer Island, Washington.**

**"I wish I'd had this book before I bought my computer!" TV Station General Manager, California.**

**"It's the best book I've seen on the subject. Send me 5 copies to send to my clients." CPA, Orange County, California.**

**"I really learned something from this book and I'm in the business. I'm recommending it to my clients." DP Consultant, Alhambra, California.**

**Here's a straight forward approach to business computer purchase and installation. It will save you time and money.**

**ORDER NOW! If not completely satisfied, return within 30 days for a full and immediate refund.**

**8½ x 11 Softbound ▪ 178 pp. ▪ $12.95**
**Credit Card Orders 1-805-964-7448**

DDC PUBLICATIONS b
**5386 Hollister Ave., Santa Barbara, CA 93111**

**Rush___ copies of "Winning The Computer Game" at $12.95 per copy. (CA residents add 6% sa'es tax)**

NAME __________
ADDRESS __________
CITY/STATE/ZIP __________

☐Check ☐MasterCharge ☐Visa

Card No. __________ Exp. Date ______

Circle 32 on inquiry card.

# SHOULD it be a Heathkit® ?

Whether you are considering the purchase of an additional peripheral or your first computer you should know more about Heath® computer products. Heathkit® has a continuing commitment to selling well-documented computer kits and software. Are they for you? How can you find out?

Read Buss: The Independent Newsletter of Heath Co. Computers, where information on new products is printed as it leaks out of Benton Harbor, not held back to suit a marketing plan. Buss is not a company-controlled publication, so it can deal with weaknesses of Heathkit® products as well as their strengths. It features news of compatible hardware and software from other vendors. Every Buss issue has candid accounts of experiences of Heathkit® owners. Results of their discoveries, which often include hardware modifications, save subscribers headaches--and money. That's proven by two years' experience. So Buss can guarantee a full refund any time you're not satisfied.

Buss is mailed first class (by airmail outside North America). The 24-issue subscription gives you the choice of starting with the latest issue or with available back issues (about 8 are still in stock). Send $ 8.00 for 12 issues or $ 15.25 for 24 (overseas, $10.00/12; $19.25/24 US funds) to: Buss
325-B Pennsylvania Ave., S.E.
Washington, DC 20003

***The Independent Newsletter of Heath Co. Computers***

# FREE Catalog

## New 4-way relief from problems with minicomputer supplies and accessories.

**1. One-stop shopping.** Inmac (formerly known as Minicomputer Accessories Corporation) has a catalog of over 800 products. Everything from racks and line-printer paper to connectors and cables. Each designed to help keep your minicomputer or word processing system up and running.


**2. Hassle-free ordering.** Inmac lets you order by mail or phone. So keep our Summer '79 catalog close. It makes those once-tough tasks like ordering magnetic media easy, fast and foolproof.

**3. Fast shipment of just the quantity you need.** Inmac ships your order within 36 hours from centers in California and New Jersey. In a bind? Call us for many special services that insure you get your order to your installation within 24 hours. Call now and give us a chance.

**4. Field-proven quality means precision performance.** Inmac guarantees every product in these 70 pages for at least 45 days. And even some for up to ten years.

Inmac — International Minicomputer Accessories Corporation

**Solve problems, send for your FREE Inmac catalog.**

130 S. Wolfe Rd. P.O. Box 9004 Sunnyvale, CA 94086 (408) 737-8700
© 1979 International Minicomputer Accessories Corporation

Circle 223 on inquiry card.

# 'TINY' PASCAL
## for TRS-80®
## & NORTH STAR®

Now you too can have Pascal! The Chung/Yuen 'Tiny' Pascal has been specially designed for TRS-80 & North Star owners. The full power & elegance of 'Tiny' Pascal is at your command. Programs written in 'Tiny' Pascal run at least 4 times faster than the same program in BASIC! 'Tiny' Pascal is also a great way to learn Pascal Programming, & fun too.

The minimum system requirements are: Level II, 16K for TRS-80, & 24K for North Star (specify single or double density).

**SOURCE TOO!**

But most important, you also get source to 'Tiny' Pascal written in Pascal with each purchase! You can even compile the compiler! (Requires 36K for North Star systems, & 32K, Level II for TRS-80). You can customize your own version, or just use it the way it is.

'Tiny' Pascal is a subset of Standard Pascal & includes:
RECURSIVE PROCEDURE/FUNCTION, IF-THEN-ELSE, REPEAT/UNTIL, 'PEEK & POKE', WHILE, CASE, & MORE!

(Plus full graphics for TRS-80 as well)

Also you can save & load programs.

You get all this & more, plus a user's manual for $40.00.

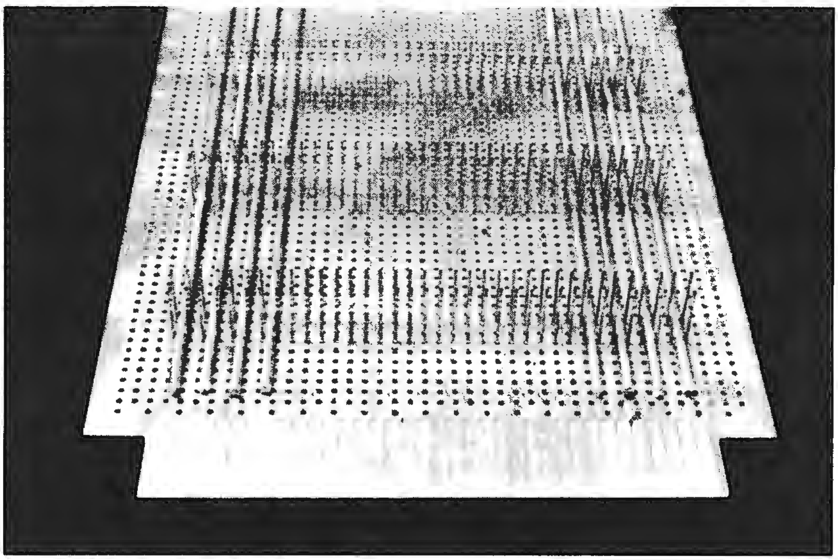
*available from:*

**SUPERSOFT**
**P.O. Box 1628**
**Champaign, IL 61820**
**(217) 344-7596**

*All orders pre-paid, Illinois residents add 5% sales tax*

Circle 357 on inquiry card.

*Photo 2: The physical assembly of the backplane begins by noting the fact that a single #12 guage wire will fit quite nicely between a set of four pins on a 0.1 inch grid. Thus, if we insert all the sockets in the backplane board, we can drop one copper bus connection across four corresponding pins of all six sockets. The 12 gauge wire size for the buses is just slightly larger than the 0.06 inch spacing between the wire wrap socket pins of a 0.1 inch grid. This fact causes the pins to spread just a bit, locking the sockets in place prior to soldering. In this photo we show eight bus wires occupying 32 pins of the backplane socket.*

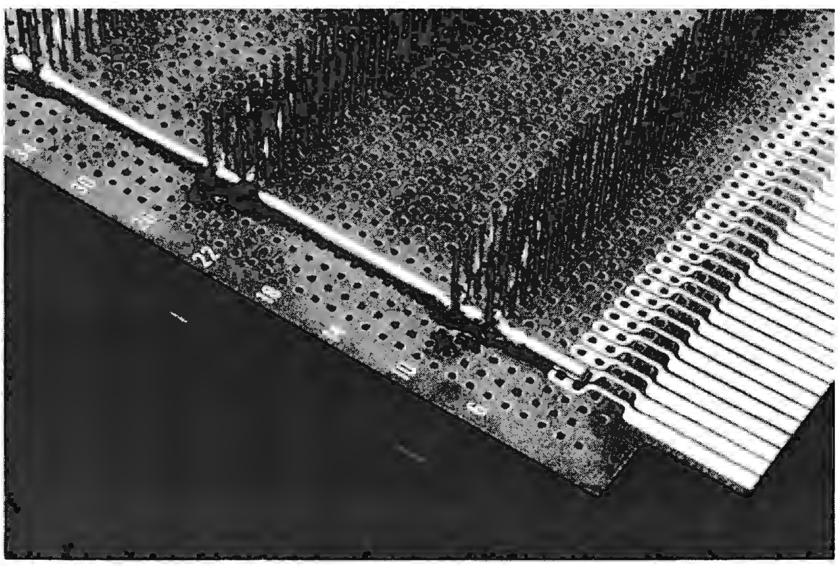


*Looking ahead to the design of my computer and basing my conclusion on previous experience, I will need a total of four power supply buses. A standard +5 V is the main power supply, used by most of the digital integrated circuits. Ground is the common point for all power supplies. Two symmetrical supplies of −12 V and +12 V will be used by analog circuits.*

*Noting that a symmetrical arrangement of the backplane prevents power supply destruction through inadvertent reversal of boards, two sets of four buses are used. As we will see later (photo 4), the innermost buses will become the basis for the ground distribution grid.*

*With four power voltages occupying a total of 32 pins, the 72 pin sockets of the backplane have 40 uncommitted pins available for communications between boards. This is more than adequate for a good general purpose computer system based on an 8 bit microprocessor, such as the 6809 I will be using with this design.*

*Photo 3: Assembly begins with the outermost bus, laying it down in its niche in the pin forest of all six sockets. It is then soldered to each set of four pins, as depicted in this photo. I used a 120 W light-duty soldering gun for this operation, since the extremely high heat carrying capacity of the copper wires made my 25 W soldering iron impossible to use.*



*After each bus wire is soldered in place, the four pins at each socket are clipped off just above the copper wire and solder bead. This process is repeated for each of the eight bus wires assigned to the power voltages of the new computer.*

*Care must be taken while soldering to avoid forming a bridge between adjacent buses. The last step in soldering a bus wire is incomplete in this photo: the wire is just resting on the four holes in the tab from the backplane's edge connector. When soldering this part, be extremely careful about forming bridges from one bus wire to the next.*

*Photo 4: The completed backplane assembly includes a set of jumpers bridging the symmetrically arranged power supply voltages. Six bus wire jumpers connect the inner ground buses across the socket area. Three heavy insulated stranded wire jumpers create an aesthetically pleasing, but electrically useless arch form at one end of the board.*

*The bus wires in this photo of the assembly process have been soldered to the edge connector pins corresponding to power distribution voltages. Not shown at this stage is a set of bypass capacitors installed between the three voltages and ground. On each voltage, six 0.1 μF ceramic capacitors and one 3.3 μF tantalum electrolytic were installed for local bypass of the power supply voltages.*

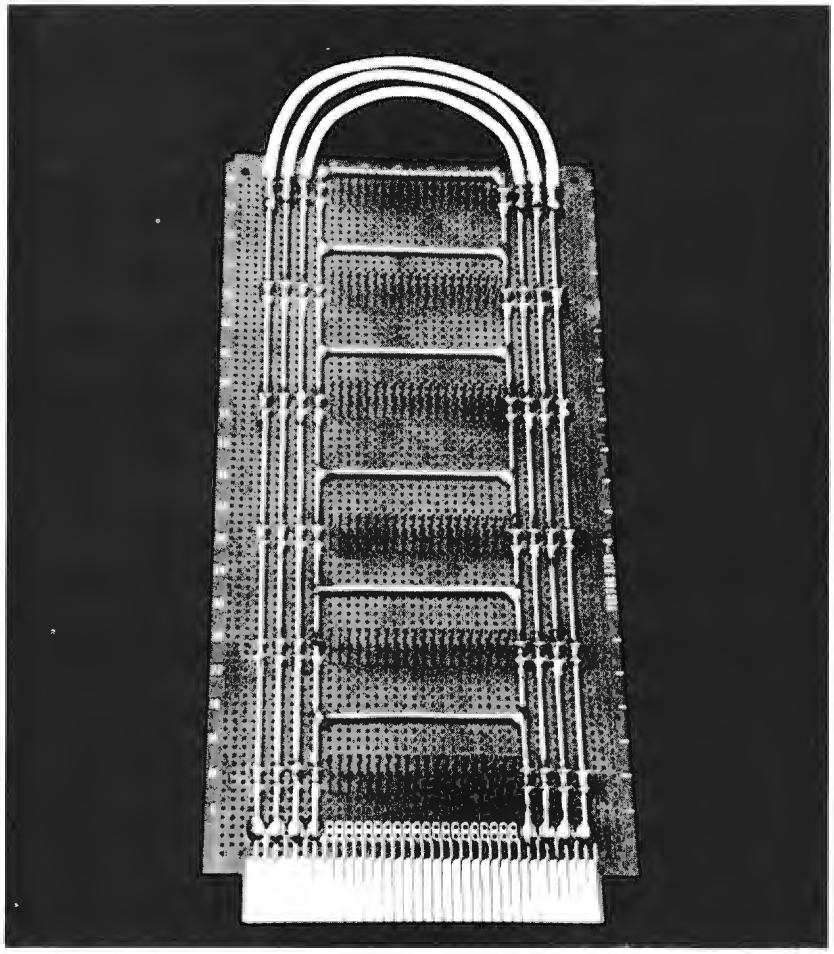


*Photo 5: No backplane idea is complete without a discussion of the physical support of the boards in the final assembly. Here we show the newly completed backplane together with six boards and a set of 1.375 inch (1⅜ inch) spacers.*

*Readers wishing to duplicate this board should learn from an imperfection I introduced. If the boards in this picture appear to be a little crooked, rest assured that this is true, and not caused by the wide angle lens. The actual center to center spacings I finally used (count the holes in photo 4 if you wish) were 1.4, 1.5, 1.7, 1.4, and 1.5 inches! The spacing between backplane sockets should be 1.4 inches, center to center. In my final assembly, I will have to use extra washers as spacers due to this flagrant indiscretion during initial assembly.*

Physically, this completes the motherboard and its power supply electrical connections. In part II of this description of the physical assembly of my new computer system we will complete the wiring of the backplane (by using a homebrew adaptation of a Vector "Slit-N-Wrap" tool to my electric eraser), wooden cabinetry which forms a base for the computer while hiding its power supply modules, and the final assembly of the computer system's basic hardware. ∎

# Mouse

## A Language for Microcomputers

**Peter Grogono**
**73 Roxton Crescent**
**Montreal West**
**Quebec CANADA**
**H4X 1C7**

This article describes Mouse, a computer programming language which can be implemented on most microcomputers. The word Mouse is not an acronym, merely an appropriate description for something small and active.

There are many available languages for microcomputers already, so the introduction of a new language warrants some explanation. The justification for Mouse is that it incorporates many features of high level languages, yet it can be implemented without the resources needed by most high level languages. More specifically, Mouse programs demonstrate the use and implementation of arrays, functions, procedures, nested control structures, local variables, recursion, and several methods of passing parameters from one procedure to another. Mouse also embodies some of the principles of structured programming, in that it uses nested, single entry control structures, and does not allow unrestricted jumps. Nonetheless, Mouse can be implemented on a minimal system consisting of a microprocessor, 4 K bytes of memory, and a terminal.

All of these features cannot be incorporated into a simple language without making some sacrifices. Identifiers in Mouse consist of a single letter, so that a symbol table is not required. Expressions are written in postfix notation, which is more easily interpreted by a computer than the conventional infix notation. Parameters are passed to subroutines as strings, eliminating the need for complex parameter transmission mechanisms. Mouse programs are easier to write than to read, and it is possible to write Mouse programs that are very obscure. Although readability is a highly desirable feature of a programming language, a language cannot be condemned solely on the grounds that obscure programs can result: witness the popularity of APL. The extraordinary thing about Mouse is that so much can be achieved with such a small amount of implementation effort.

Mouse is descended from an older programming language called Musys. In 1970 I was working for Electronic Music Studios (London) Limited, the company which now manufactures the SYNTHI series of electronic music equipment. At that time the company, under the direction of Peter Zinovieff, was exploring the possibilities of using minicomputers to control electronic music instruments. The studio had two DEC PDP-8 computers, but very little software. My job was to write software which would relieve composers of the tedious chore of entering musical compositions in the form of strings of octal numbers. Since the older PDP-8 was already connected to a variety of digitally controlled oscillators, filters, envelope shapers, and other musical equipment, we decided to use the newer and faster PDP-8 to do the language processing. Space was very limited: the PDP-8 had only 4 K 12 bit words of memory, and a very restricted instruction set by comparison with today's microprocessors.

The system that I designed for the studio enables composers to write their compositions in a high level language called Musys. The heart of Musys is a simple yet powerful macroprocessor. A Musys composition consists of a hierarchy of macroinstructions, in which the higher level macroinstructions determine the overall form of the composition, and the low level macroinstructions specify details such as the pitch and duration of the individual sounds. The Musys interpreter contained about 700 instructions. About 600 additional instructions were required for supporting software, including disk control. The system is described in reference 1. The idea of using a macroprocessor as the basis for a minicomputer

**Note:** The examples of the Mouse language which appear in the *text* of this article have been italicized for the sake of clarity. Examples of the language appearing on *lines by themselves* have not been italicized, however.

language occurred to me after I had read a paper by the late Christopher Strachey (reference 2).

The remainder of this article consists of two main sections: the first describes the language Mouse; the second describes the implementation of a Mouse interpreter.

**Mouse User's Guide**

A Mouse program is a string of characters. Blanks may be inserted anywhere in the program, but they are ignored by the interpreter, except in a few contexts which are defined below. The last two characters of a program are always *$$*. The interpreter starts executing at the first character of the program and processes one character at a time until it encounters the character *$*, at which point it stops. This processing sequence is broken only by specific control strings: conditions, loops, macro calls, and formal parameters. These are described below.

When the interpreter encounters the quote character *"*, it prints or displays characters up to, but not including, the next quote character. For example, the program:

```
"JACQUELINE" $$
```

will print the message:

```
JACQUELINE
```

The quoted string may contain blanks, which are printed, and the exclamation point *!* that prints a carriage return/line feed. Thus the program:

```
"FIRST LINE!SECOND LINE" $$
```

will print:

```
FIRST LINE
SECOND LINE
```

Mouse performs calculations using a stack. An operand pushes a value onto the stack, and an operator removes one or two values from the stack and may replace them with other values. The question mark *?* is an operand which tells the interpreter to read a number from the keyboard or input file and push it onto the stack. The operator *!* removes the top value from the stack and prints it. (Do not confuse the use of *!* as an operator with its use within quotes.) The following program reads one number from the keyboard and prints it:

```
? ! $$
```

The stack is a *last in, first out* data structure. The program:

```
? ? ? ! ! ! $$
```

reads three numbers and prints them in reverse order. For example, if the input file contained:

```
45 46 47
```

then the program would print:

```
474645
```

If we wanted them to be printed in the order in which they were read, we would have to write:

```
? ! ? ! ? ! $$
```

This program, using the same data as before, would print:

```
454647
```

A decimal integer is another kind of operand. When the interpreter reads an integer, it pushes its value onto the stack. This program prints *365*:

```
365! $$
```

We can push two numbers onto the stack by writing them one after the other, with a *blank* in between. This program prints *7 5*:

```
5 7 ! " " ! $$
```

This is the only context in Mouse where a blank character is required; without it, *5 7* would be read as the single number *57*. However, we often insert blanks into Mouse programs to improve their readability. Furthermore, Mouse does not print leading or trailing blanks when it prints a number, so we must include a blank string *" "* between print operators if we want numbers separated in the output. Note also that Mouse does not process floating point numbers: all operands have integer values.

Mouse has 26 variables, the names of which are *A*, *B*, *C*,. . .*Z*. The name of a variable is an operand, and when the interpreter encounters a variable name, it pushes the address of that variable onto the stack. This program prints the addresses of A and T:

```
A! T! $$
```

The operator . (period) replaces an address on the stack by the value stored at that

*Peter Grogono's first computer experience consisted of feeding a short piece of paper into EDSAC II at Cambridge University. He joined a group conducting research into automatic pattern recognition in 1965, and has spent the ensuing 13 years writing computer programs of various kinds in England, Australia, and Canada. The programs include a package for highway design, an operating system for an electronic music studio, and an accounting system for travel agencies. Peter is currently working for the Computer Centre of Concordia University, and his interests are programming language design and text processing.*

address. If we want to print the value of A we write _A.!_ .

The assignment operator is =. An assignment is performed in three steps:

- An address is pushed onto the stack;
- A value is pushed onto the stack;
- The assignment operator is applied.

Thus in order to give the value 17 to X we write:

X17=

This statement has the effect that X := 17 would have in a Pascal program. Similarly, we write _XY.=_ to assign the value of Y to X in Mouse. The most common programming error in Mouse is to forget the period; no error will be reported, but the calculation will use the address rather than the value of a variable, and the result will probably be wrong.

The arithmetic operators in Mouse are +, –, *, and /, denoting addition, subtraction, multiplication, and division, respectively. Because operands in Mouse have integer values, / means divide and truncate. A Mouse operator is always written after its operands. The resulting notation is called _postfix_ notation or reversed Polish notation, in contrast to the conventional algebraic notation which is called _infix_ notation. Postfix notation may be confusing at first, but it does have some advantages. The infix expression A+B is written in Mouse as _A.B.+_; the periods signify that we are adding values, not addresses. One of the advantages of postfix notation is that parentheses or brackets are not required. The infix expression A+B*C, in which the multiplication is performed before the addition, is written _A.B.C.*+_ in Mouse. The infix expression (A+B)*C, in which the parentheses indicate that the addition is to be performed first, is written _A.B.+C.*_.

It is not hard to translate expressions into postfix notation if you remember these two rules:

- Operands appear in the same order in both expressions;
- Operators are written as soon as both operands have been written in full.

As an example, consider the conversion to postfix form of the infix expression (A+B)/(C–D). First write down the operands in sequence:

A. B. C. D.

The addition is performed after *B.*, and the subtraction after *D.*. The division cannot be performed until both its operands have been computed, so the complete expression in postfix notation is:

A.B.+ C.D.– /

The arithmetic operators always have two operands in Mouse. The infix expression –X means 0–X, and it must be written in the form *0X.–*.

The top value in the stack may be used as an *anonymous variable.* In fact, we have already used the stack in this way, in the program *? ! $$*. Here is a more subtle use of this feature:

A A5= . A6= ! $$

These five steps have the following effect:

- *A* puts the address of A on the stack;
- *A5=* assigns the value 5 to A (this uses the stack, but leaves it unchanged);
- . converts the address of A to the value of A, which is 5;
- *A6=* changes the value of A to 6;
- *!* prints the value, 5, from the stack.

Care must be taken, of course. If you are writing programs in this fashion, you must at all times know what is supposed to be in the stack. The following example uses the stack to interchange the values of two variables, X and Y. In most languages this interchange can only be done with a temporary variable. For example, in Pascal we would write:

T := X; X := Y; Y := T

In Mouse we can write:

X Y. Y X. = =

The addresses of A, B, C,. . .Z are 1, 2, 3, . . .26. This means that B, for instance, can be regarded as either a variable in its own right, or as the second element of the array A. The address of B can be written as either *B* or *A1+*, and its value as *B.* or *A1+.* . A general element of the array A, written A[I] in Pascal, can be written as *AI.+.* in Mouse, for $0 \leqslant I \leqslant 25$. Any letter can be used in this way: thus *K5+* is equivalent to *P*, and *Z3–.* is equivalent to *W.* . You must be careful not to use the same address for two different purposes. If you decide to use A as an array with ten components, you cannot use the variables B, C,. . .J in the same program. The following illustrates these points:

A7= BA.= A.! B.! A1+.! $$

*A7=* assigns the value 7 to A, and *BA.=* copies the same value to B. This value is then printed three times: *A.!* prints the value of A; *B.!* prints the value of B; and *A1+.!* prints the value of A[1].

Before we pass on to a discussion of the control structures of Mouse, we need one more concept: *clause*. A clause is a string which contains quoted strings and expressions, which have been defined above; and complete control structures, which are defined below.

### Control Structures

A *condition* in Mouse is written *E*[*C*]; E is an expression, and C is a clause. The condition itself is a clause. E is evaluated, and if its value is greater than zero, C is executed. If E is zero or negative, C is skipped. The Pascal statement:

if $X > 0$ then Y := X

becomes in Mouse:

X.[YX.=]

Since Mouse distinguishes only $E > 0$ and $E \leqslant 0$, more complicated expressions must be devised for different conditions. For example, $X \geqslant 0$ is equivalent to $X+1 > 0$, and so the Pascal statement:

if $X \geqslant 0$ then C

becomes in Mouse:

X.1+[C]

Similarly, $X = 0$ is equivalent to $(1+X > 0)$ and $(1-X > 0)$, and so Pascal:

if $X = 0$ then C

becomes in Mouse:

1X.+[1X.– [C]]

Now, reconsider the definition of clause. A complete condition *E*[*C*] is both a clause itself, and contains a clause, *C*. Suppose C is itself a condition, *F*[*D*]. If we replace C by *F*[*D*] in the clause *E*[*C*], we obtain the clause *E*[*F*[*D*]]. This demonstrates that the nested condition *1X.+*[*1X.–*[*C*]] of the last example is a legitimate Mouse construction.

A loop in Mouse is written:

$$(C_1 \; E\uparrow C_2)$$

where E is an expression and $C_1$ and $C_2$ are clauses. This loop is a clause; it may be paraphrased as follows:

```
START:  C1
        if E <= 0 goto EXIT
        C2
        goto START
EXIT:
```

In other words, the loop is executed for as long as $E > 0$, and the exit test may appear anywhere within the loop. Either $C_1$ or $C_2$ may be omitted. *(E↑ $C_2$)* is equivalent to Pascal:

**while** $E > 0$ **do** $C_2$

and *($C_1$ E↑)* is equivalent to Pascal:

**repeat** $C_1$ **until** $E \leqslant 0$

There may be more than one exit test in a loop. For example, *($C_1$ E↑ $C_2$ F↑ $C_3$)*, in which E and F are expressions, is allowed.

We now consider some simple Mouse programs which use these control structures. The following program reads a number N, and prints its factorial, 1*2*3*. . .*N:

```
N?= F1= ( N.↑ FF.N.*= NN.1—= ) F.! $$
```

We can shorten the factorial program by using the top of the stack instead of the variable F. The program then becomes:

```
N?= 1 ( N.↑ N.* NN.1—= ) ! $$
```

In this version, *1* puts the value one on the stack; *N.** multiplies the value on top of the stack by N; and *!* removes the value from the top of the stack and prints it. We can expand this program so that it continues to ask for data until it reads a number which is less than or equal to zero. Note the nested loop:

```
( "ENTER A NUMBER" M?= M.↑
   NM.= 1 ( N.↑ N.* NN.1—= )
"FACTORIAL ( " M.! ") = " ! "!" )
$$
```

Here is a dialogue produced by this program:

```
ENTER A NUMBER 3
FACTORIAL (3) = 6
ENTER A NUMBER 7
FACTORIAL (7) = 5040
ENTER A NUMBER 0
```

The program which follows reads numbers into an array and then prints them in reverse order. A is used as an array index, and B is the array. The input is terminated by a negative number.

```
A0=
( BA.+.?= BA.+.1+↑ AA.1+= )
( A.↑ AA.1−= BA.+.! )
$$
```

We can improve this program so that it will read no more than 25 numbers, the maximum capacity of the array B. Note the use of two exit conditions in the first loop:

```
A0=
( BA.+.?= BA.+.1+↑ AA.1+=
A.25− [ "ARRAY FULL!" 0↑ ] )
( A.↑ AA.1−= BA.+.! )
$$
```

### Macroinstructions

A complex algorithm always has a hierarchical structure, and a programming language must provide the means of defining and calling procedures and functions so that programs which have an analogous hierarchical structure can be written. In Mouse, procedures and functions are implemented by macroinstructions. A macroinstruction call is written like this:

```
#M;
```

This is a call to macroinstruction M. There must not be a blank between # and the macro name. In this example there are no actual parameters. When there are actual parameters, they follow the macro name and are preceded by commas. The following is a call to macroinstruction M with parameters X and Y:

```
#M,X,Y;
```

A macro name is a single letter, so a program may use up to 26 different macroinstructions.

Macro definitions come between the main program and the terminating *$$*. A macro definition starts with the character *$* and the name of the macroinstruction, and is terminated by the character @. There must not be any blanks between *$* and the name of the macroinstruction being defined. The definition must be a clause, and definitions cannot be nested. The following program uses a macroinstruction, M, to print a message:

```
#M;
$M "A MESSAGE" @
$$
```

Note that the main program code is terminated by *$*, which introduces the first macro definition. As usual, the entire program is terminated by *$$*. When the interpreter encounters *#M;* in the program, it substitutes the definition of the macroinstruction, excluding *$M* or @. Thus when the program above is interpreted, it prints:

```
A MESSAGE
```

Each macroinstruction has its own complete set of local variables, A, B, C,. . .Z. Assignments to these variables do not affect the values of main program variables. The program which follows will print the number 3, despite the assignment to N in macroinstruction X:

```
N3= #X; N.!
$X N99= @
$$
```

A macro definition may have up to 26 formal parameters, written %A, %B, %C, . . .%Z, with no blank between % and the letter. When a formal parameter is encountered in a definition, the actual parameter is substituted as a string. Macroinstruction P in the following program prints the value of its parameter:

```
#P,3; A5= #P,A.; #P,A;
$P %A! @
$$
```

The call *#P,3;* is equivalent to *3!*, and prints *3*. The call *#P,A.;* is equivalent to *A.!* and prints the value of A, which is 5. These are known as *calls by value,* because the macroinstruction can obtain only the value of the parameter. The final call, *#P,A;*, is equivalent to *A!*, and prints the address of A. It is known as a *call by reference,* because the parameter is an address. Calls by reference can be used to return values to the calling program. Macroinstruction A in the following program adds the values of its first two parameters, and returns the result as the third parameter:

```
#A,2,3,X; #A,3,4,Y; #A,X.,Y.,Z;
$A %C %A %B + = @
$$
```

This program adds 2 and 3, giving X, then adds 3 and 4, giving Y, then adds X and Y, giving Z. Note that the actual parameters in the last call are the values *X.* and *Y.* and the address *Z*.

You must know, when writing a macro call, whether the macroinstruction expects a value or an address. The device mentioned before for exchanging two values can be incorporated into a macroinstruction:

```
$E %A %B. %B %A. = = @
```

In this case, both actual parameters must be addresses. To use this macroinstruction to interchange the values of X and Y write:

```
#E,X,Y;
```

An array can be passed to a macroinstruction by the address of its first component. The following macroinstruction sums %B components of the array %A, and returns the result as %C:

```
$S %C0= N0=
( %BN.–↑ %C%AN.+.%C.+= NN.1+= )
@
```

The call *#S, A,3,Z;* would store the value of A+B+C in Z.

A macroinstruction which leaves a value on the stack acts as a function. We can rewrite macroinstruction A, which was defined above, as a function:

```
$A %A%B+ @
```

The call *#A,2,3;* is now an operand which leaves the value 5 on the stack. This macroinstruction evaluates the factorial function:

```
$F N%A= 1 ( N.↑ N.* NN.1– ) @
```

The clause *#F,5; #F,6; + !* prints 840 (=5!+6!).

A macro definition may include macro calls. Moreover, a macroinstruction may call itself, either directly or indirectly, so that it is possible to define recursive macroinstructions. The factorial function can be defined recursively in this way:

```
$F %A [ %A #F,%A1–; * ]
1%A– [ 1 ]
@
```

The character @ acts like a RETURN statement in BASIC or FORTRAN, and we can use it more than once in a macro definition. Here is a slightly shorter version of the recursive factorial macro that exploits this fact:

```
$F %A [ %A #F,%A1–; * @ ] 1 @
```

This definition may be paraphrased in ALGOL:

```
F(N) :=
if N > 0
then N * F(N–1)
else 1
```

If %A > 0, the conditional clause is executed, and the macroinstruction is terminated by the @ between the brackets. If %A ⩽ 0, the value *1* is placed on the stack and the macroinstruction is terminated by the final @.

If there is no actual parameter corresponding to a formal parameter, the formal parameter is processed as a null string. Actual parameters for which there are no formal parameters are ignored. A call to an undefined macroinstruction is also processed as a null string. These points are illustrated by the following program:

```
#T; #T, "A"; #T, "A", "B";
#T, "A", "B", "C"; #U;
$T "<" %A%B ">!"@
$$
```

This program prints:

```
<>
<A>
<AB>
<AB>
```

In the first call, *#T;*, there are no actual parameters. In the second call, *#T, "A";*, there is one actual parameter, *"A"*, which is printed. In the third call, both parameters are printed. In the fourth call, there are three actual parameters, of which the third, *"C"*, is ignored because there is no formal parameter *%C*. The last call is to an undefined macroinstruction U, and has no effect.

Now that all of the control structures of Mouse have been described, we can define *clause* more precisely. A clause may be: an expression, a literal string, a condition, a loop, a macro call, or a clause followed by another clause. Actual parameters and macro definitions must be clauses. These rules are not precise enough to formally define the syntax of Mouse, but they serve as a guide for the Mouse programmer. Their principal purpose is to ensure that all of the components of a control structure are on the same level. They forbid, for example, the use of *(* or [ without the matching *)* or ] at the same level in a condition, loop, macro definition or actual parameter.

### Example Programs

We conclude this section of the article with some sample Mouse programs. The first program illustrates mutual recursion. Two macroinstructions are said to be *mutually recursive* if each one calls the other. The square numbers $S_n$ (1, 4, 9, 16,. . .) and the triangular numbers $T_n$ (1, 3, 6, 10,. . .) can be defined in terms of each other in the following way:

$$S_1 = 1$$
$$S_n = T_n + T_{n-1} \text{ for } n > 1$$
$$T_1 = 1$$
$$T_n = (S_{n-1} + 3n - 1)/2 \text{ for } n > 1$$

The program, which appears in listing 1, uses mutually recursive macroinstructions S and T to compute $S_n$ and $T_n$.

The second program, which appears in listing 2, prints prime numbers. It loops indefinitely, calling macroinstruction P for each integer N in turn. Macroinstruction P prints N, if N is prime. The expression *%AF./F.*%A–1+* is equivalent to 1 – %A **mod** F.; it is positive only if F divides %A exactly. The program will fail when incrementing N causes overflow, but the algorithm is so inefficient that there is little danger of this happening.

The third program, shown in listing 3, uses recursive macroinstruction V to print the words of a song. The British and American versions of this song are different, and I am not sure that the words printed by the program are a correct statement of either version. The limit of nine verses is a restriction imposed by the interpreter described in this article, not an inherent limitation of Mouse.

```
"SQUARE AND TRIANGULAR NUMBERS"
( "ENTER A NUMBER" N? = N . ↑
      "S(" N . ! ") = " #S,N . ; !
   "   T(" N . ! ") = " #T,N . ; ! "!" )

$S %A1 – [ #T, %A; #T, %A1 –; + @ ] 1 @
$T %A1 – [ #S, %A1 –; 3%A*1 – + 2/ @ ] 1 @

$$
```

*Listing 1: This Mouse program reads an integer and prints the corresponding square number $S_n$ and triangular number $T_n$. It uses two mutually recursive macroinstructions S and T.*

```
"PRIME NUMBERS!" N1 = ( NN.1 + = #P,N.; )

$P F1 = N1 =
    ( FF. 1 + = %AF. –↑ %AF./F. * %A – 1 + [ N0 = 0↑ ] )
    N. [ %A! "!" ] @

$$
```

*Listing 2: This program prints a list of prime numbers. Macroinstruction P determines whether its parameter N is a prime number and, if it is, prints it.*

The program which appears in listing 4 prints the moves required to solve the Towers of Hanoi problem for a given number of disks. It uses recursion, and demonstrates the use of strings as actual parameters.

```
( "HOW MANY VERSES?"
   S? = S.↑ 10S. - [ #V,S.1 +; ] )

$V %A1 - [ #N,%A; #B; #W; ",!"
           #N,%A; #B; ";!" #F; ",!"
           #N,%A1 -; #B; #W; ".!!"
           #V,%A1 -;                    ] @

$B " GREEN BOTTLES" @
$W "STANDING ON THE WALL" @
$F "IF ONE OF THOSE BOTTLES SHOULD HAPPEN TO FALL" @

$N %A9 - [ "NINE"    @ ]
   %A8 - [ "EIGHT"   @ ]
   %A7 - [ "SEVEN"   @ ]
   %A6 - [ "SIX"     @ ]
   %A5 - [ "FIVE"    @ ]
   %A4 - [ "FOUR"    @ ]
   %A3 - [ "THREE"   @ ]
   %A2 - [ "TWO"     @ ]
   %A1 - [ "ONE"     @ ]
   %A    [ "NO"      @ ] @

$$
```

*Listing 3: Macroinstruction V calls macroinstructions B, W, F, and N to print one verse of a song, and then calls itself recursively to print the next verse. The number of verses is limited to nine, but the interpreter is quite easily modified to enable the program to print more verses.*

```
"TOWERS OF HANOI!HOW MANY DISKS?" D? =
D. [ #H,D.,"LEFT","RIGHT","CENTER"; ]

$H %A [ #H,%A1 -,%B,%D,%C;
        "MOVE " %B " TO " %C "!"
        #H,%A1 -,%D,%C,%B;        ] @

$$
```

*Listing 4: The famous Towers of Hanoi problem. The program prints a list of the moves required to solve the problem for a specified number of rings.*

```
"BUBBLE SORT!" #P,A;

$P "HOW MANY NUMBERS?" N? = N.26 - [ "TOO MANY!" @ ]
   "ENTER " N.! " NUMBERS!" NN.1 - =
   #F,M,0,N.,%AM. +? =;
   "INPUT ARRAY!" #F,M,0,N.,%AM. + .! " "; "!"
   #B,%A,0,N.;
   "SORTED ARRAY!" #F,M,0,N.,%AM. + .! " "; "!" @

$B #F,I,%B1 +,%C,
     #F,J,%C,I.,
        %AJ.1 - +. %AJ. +. -
              [ #E,%AJ. +,%AJ.1 - +; ] ; ; @

$F %A%B = %C%A. -1 +
   [ ( %C%A. -1 +↑ %D %A%A.1 + = ) @ ]
    ( %A. %C -1 +↑ %D %A%A.1 - = ) @

$E %A %B. %B %A. = = @

$$
```

*Listing 5: This is the most elaborate Mouse program presented in this article. The main program contains one macro call (#P,A;) only, and so all 26 variables at the lowest level can be used to store an array. The program sorts the array using the bubble sort algorithm, and then prints it.*

The last program, shown in listing 5, is the most elaborate. It reads an array, prints it, sorts it into ascending sequence, and prints the sorted array. The main program prints the title and then calls macroinstruction P to do everything else. The address of A is given to P, and since no variables are used in the main program, all 26 components of array A can be used to store the input data. Thus, the program can sort up to 26 numbers. The program makes extensive use of macroinstruction F, which simulates a **for** statement. The effect of a call to F is:

**for** %A := %B **to** %C **do** %D

If %B > %C then the index variable %A will be decremented rather than incremented. Macroinstruction E is the exchange macroinstruction introduced earlier. Macroinstruction B does the actual sorting in two nested loops, using the bubble sort algorithm.

### Implementation

The Mouse interpreter is presented in listing 6 in the form of a Pascal program. It is not intended, however, that Mouse should be implemented by that program. Listing 6 is intended to be a machine independent guide to the implementation of Mouse in assembly or machine language. Accordingly, the explanation which follows contains hints as to how the Pascal statements can be translated into machine language. Numbers in parentheses refer to line numbers in listing 6. Some explanation of the meaning of the Pascal statements is given here. (If you need more, consult either Jensen and Wirth (reference 3) or my book (reference 4)).

The program starts with a heading (1). This line merely states that the program is called Mouse and that it uses two files, INPUT and OUTPUT. Lines 12 thru 18 declare the global variables of the program. Global variables can be used anywhere in a Pascal program, even in subroutines. The first five declarations (12 thru 16) define arrays. PROG is an array of 500 characters used to store the text of the Mouse program. The components of PROG are written PROG[1], PROG[2]. . .PROG[500]. DEFINITIONS is an array of 26 integers. Each nonzero component of DEFINITIONS forms an index for the start of a macro definition in the array PROG. The array CALSTACK is the stack used for calculations. The array STACK is the main stack, used to store the status of the program during the expansion of a macro call, a formal parameter, or a loop. Each component of STACK is a variable of type

FRAMETYPE. This type is defined in lines 5 thru 9. A FRAMETYPE variable has three components: TAG, POS, and OFF. A TAG has one of three values: MACRO, PARAM, or LOOP; these values can be coded as 0, 1, 2, or −1, 0, +1. POS and OFF are both integers. The most convenient way to represent STACK is as a block of 20 units, each unit having three words. Within a unit, the word required is addressed by its offset: 0 for TAG, 1 for POS, and 2 for OFF. For example, the address STACK [N].OFF (the address of component OFF of the Nth unit) is STACK+3*(N−1)+2. Finally, the array DATA is used to store the values of both local and global variables of the Mouse program. The sizes of the arrays are adequate for simple Mouse programs, including all programs in this article. The choice of array sizes is discussed below in greater detail.

Lines 17 and 18 declare global scalar variables. CAL is a pointer to CALSTACK; it is an index for the next free stack word. LEVEL performs the same function for STACK, the main stack. CHPOS is a pointer to the array PROG; it indexes the character currently being processed. The current character itself is stored in CH. Assuming that a single byte is used to store a character, and two bytes are used to store an integer, global data for the interpreter occupies 1248 bytes of memory.

Lines 20 thru 84 define the subroutines used by the interpreter. The function NUM (20 thru 23) maps letters into integers. The Pascal function ORD means simply *ordinal value of*. The ordinal value of a character is its ASCII (or other) code. For example, if the code for A is 65, then ORD('A') = 65. Thus, NUM('A') = 1, NUM('B') =2, and so on. The interpreter assumes that the letters have consecutive codes, and hence that NUM('Z') = 26. The function VAL performs a similar task for digits. VAL('0') = 0, VAL ('1') = 1, and so on. In machine language, these functions can be implemented by a single instruction which subtracts the appropriate constant from the character value. The procedure GETCHAR (30 thru 33) increments the character pointer CHPOS and sets CH to the next character in the array PROG. The Mouse program is accessed by means of this procedure only.

PUSHCAL (35 thru 38) and POPCAL (40 thru 43) are used to store and remove values from the calculation stack CALSTACK. The parameter of PUSHCAL is the value which is to be pushed onto the stack. The value returned by POPCAL is the value removed from the stack. In a machine language implementation, these values can be passed in a register. PUSH (45 thru 51) and POP (53

*Listing 6: The Mouse interpreter expressed as a Pascal program.*

```
 1  program MOUSE (input,output);
 2
 3  type
 4     TAGTYPE = (MACRO,PARAM,LOOP);
 5     FRAMETYPE =
 6        record
 7           TAG : TAGTYPE;
 8           POS,OFF : integer
 9        end;
10
11  var
12     PROG          : array [ 1..500 ] of char;
13     DEFINITIONS   : array [ 1..26  ] of integer;
14     CALSTACK      : array [ 1..20  ] of integer;
15     STACK         : array [ 1..20  ] of FRAMETYPE;
16     DATA          : array [ 1..260 ] of integer;
17     CAL, CHPOS, LEVEL, OFFSET, PARNUM, PARBAL, TEMP : integer;
18     CH : char;
19
20  function NUM (CH : char) : integer;
21     begin
22        NUM := ord(CH) - ord('A') + 1
23     end;
24
25  function VAL (CH : char) : integer;
26     begin
27        VAL := ord(CH) - ord('0')
28     end;
29
30  procedure GETCHAR;
31     begin
32        CHPOS := CHPOS + 1; CH := PROG[CHPOS]
33     end;
34
35  procedure PUSHCAL (DATUM : integer);
36     begin
37        CAL := CAL + 1; CALSTACK[CAL] := DATUM
38     end;
39
40  function POPCAL : integer;
41     begin
42        POPCAL := CALSTACK[CAL]; CAL := CAL = 1
43     end;
44
45  procedure PUSH (TAGVAL : TAGTYPE);
46     begin
47        LEVEL := LEVEL + 1;
48        STACK[LEVEL].TAG  :=  TAGVAL;
49        STACK[LEVEL].POS  :=  CHPOS;
50        STACK[LEVEL].OFF  :=  OFFSET
51     end;
52
53  procedure POP;
54     begin
55        CHPOS := STACK[LEVEL].POS;
56        OFFSET := STACK[LEVEL].OFF;
57        LEVEL := LEVEL - 1
58     end;
59
60  procedure SKIP (LCH, RCH : char);
61     var CNT : integer;
62     begin
63        CNT := 1;
64        repeat
65           GETCHAR;
66           if CH = LCH
67              then CNT := CNT + 1
68           else if CH = RCH
69              then CNT := CNT - 1
70        until CNT = 0
71     end;
72
73  procedure LOAD;
74     var THIS, LAST : char; CHARNUM : integer;
75     begin
76        for CHARNUM := 1 to 26 do DEFINITIONS[CHARNUM] := 0;
77        CHARNUM := 0; THIS := ' ';
78        repeat
79           LAST := THIS; read(THIS);
80           CHARNUM := CHARNUM + 1; PROG[CHARNUM] := THIS;
81           if (THIS in [ 'A'..'Z' ] ) and (LAST = '$')
82              then DEFINITIONS[NUM(THIS)] := CHARNUM
```

*Listing 6 continued on page 210*

*Listing 6 continued from page 209:*

```
 83           until (THIS = '$') and (LAST = '$')
 84        end;
 85
 86     begin
 87        LOAD;
 88        CHPOS := 0; LEVEL := 0; OFFSET := 0; CAL := 0;
 89        repeat
 90           GETCHAR;
 91           case CH of
 92
 93              ' ', ']', '$' : ;
 94
 95              '0','1','2','3','4','5','6','7','8','9' :
 96                     begin
 97                        TEMP := 0;
 98                        while CH in [ '0' . . '9' ] do
 99                           begin
100                              TEMP := 10 * TEMP + VAL(CH); GETCHAR
101                           end;
102                        PUSHCAL(TEMP); CHPOS := CHPOS - 1
103                     end;
104
105              'A','B','C','D','E','F','G','H','I','J','K','L','M',
106              'N','O','P','Q','R','S','T','U','V','W','X','Y','Z' :
107                     PUSHCAL(NUM(CH) + OFFSET);
108
109              '?'   : begin
110                        read(TEMP); PUSHCAL(TEMP)
111                     end;
112
113              '!'   : write(POPCAL : 1);
114
115              '+'   : PUSHCAL(POPCAL + POPCAL);
116
117              '−'   : PUSHCAL(−POPCAL + POPCAL);
118
119              '*'   : PUSHCAL(POPCAL * POPCAL);
120
121              '/'   : begin
122                        TEMP := POPCAL;
123                        PUSHCAL(POPCAL div TEMP)
124                     end;
125
126              '.'   : PUSHCAL(DATA[POPCAL]);
127
128              '='   : begin
129                        TEMP := POPCAL;
130                        DATA[POPCAL] := TEMP
131                     end;
132
133              '"'   : repeat
134                        GETCHAR;
135                        if CH = '!'
136                           then writeln
137                        else if CH ≠ '"'
138                           then write(CH)
139                     until CH = '"';
140
141              '['   : if POPCAL ≤ 0 then SKIP('[',']');
142
143              '('   : PUSH(LOOP);
144
145              '↑'   : if POPCAL ≤ 0
146                        then
147                           begin
148                              POP; SKIP('(',')')
149                           end;
150
151              ')'   : CHPOS := STACK[LEVEL].POS;
152
153              '#'   : begin
154                        GETCHAR;
155                        if DEFINITIONS[NUM(CH)] > 0
156                           then
157                              begin
158                                 PUSH(MACRO);
159                                 CHPOS := DEFINITIONS[NUM(CH)];
160                                 OFFSET := OFFSET + 26
161                              end;
162                           else SKIP('#',';')
163                     end;
```

*Listing 6 continued on page 212*

thru 58) perform similar functions for the main stack, but they are more complicated because each stack entry has three words. The data placed on the stack by PUSH consists of the parameter, TAGVAL, and the values of the two global variables, CHPOS and OFFSET. The procedure POP restores the values of CHPOS and OFFSET but ignores the stacked value of TAG.

The procedure SKIP (60 thru 71) is used to skip over nested pairs of characters. For example, SKIP('[',']') is used to skip over conditional clauses. SKIP uses a local variable CNT to count occurrences of left and right brackets so that it can correctly skip over sequences such as:

[ . . . [ . . . ] . . . ]

On entry, SKIP assumes that the first character, [, in the example, has been read, so CNT is initially set to 1. Subsequently, [ increments CNT and ] decrements CNT. The procedure terminates when CNT = 0. This procedure is used for skipping over [. . .], (. . .), and #. . .;.

The procedure LOAD (73 thru 84) has two functions: it reads the program from the input file into the array PROG, one character at a time; and it stores pointers to macro definitions in the array DEFINITIONS. It uses local variables THIS, LAST, and CHARNUM (74), and it initializes all components of the array DEFINITIONS to zero (76), so that the interpreter can later recognize a call to an undefined macro. LOAD recognizes a macro definition by the sequence *$<letter>* and the end of the program by the sequence *$$*.

In this version of the interpreter, both the Mouse program and its input data are read from the system input file. If you have auxiliary storage, such as disks or cassette tapes, it is probably better to store the Mouse program there. If you use disks or tapes, READ(THIS) on line 79 will call a procedure that reads one character from the chosen device. It is much easier to develop a Mouse program if it is stored on an external medium where it can be attacked with a text editor, rather than keying it directly into memory. Alternately, if you have room in memory, you can elaborate the procedure LOAD into a Mouse editor, rather than using it simply as a loader.

The main program begins at line 86. Initialization consists of loading the program (87) and setting various global variables to zero (88). The rest of the interpreter consists of a single **repeat** statement (89 thru 197). The body of the **repeat** statement contains two statements: GETCHAR (90),

Circle 331 on inquiry card.







Circle 159 on inquiry card.

*Listing 6 continued from page 210:*

```
164
165            '@' : begin
166                     POP; SKIP('#',';')
167                  end;
168
169            '%' : begin
170                     GETCHAR; PARNUM := NUM(CH); PUSH(PARAM);
171                     PARBAL := 1; TEMP := LEVEL;
172                     repeat
173                        TEMP := TEMP - 1;
174                        case STACK[TEMP].TAG of
175                           MACRO : PARBAL := PARBAL - 1;
176                           PARAM : PARBAL := PARBAL - 1;
177                           LOOP :
178                        end
179                     until PARBAL = 0;
180                     CHPOS := STACK[TEMP].POS;
181                     OFFSET := STACK[TEMP].OFF;
182                     repeat
183                        GETCHAR;
184                        if CH = '#'
185                           then
186                              begin
187                                 SKIP('#',';'); GETCHAR
188                              end;
189                        if CH = ',' then PARNUM := PARNUM - 1
190                     until (PARNUM = 0) or (CH = ';');
191                     if CH = ';' then POP
192                  end;
193
194            ',',';' : POP
195
196         end
197      until CH = '$'
198   end.
```



and a case statement (91 thru 196) which selects an action according to the character obtained by GETCHAR. The **repeat** statement, and hence the interpreter, terminates when CH = '\$'. The operation of the interpreter can therefore be described solely in terms of the action taken for each character returned by GETCHAR. The case statement can be implemented by comparing CH to each legal character in turn. A jump table addressed by the ordinal value of CH is more efficient, but it may use more memory.

The characters (blank), ], and \$ require no action by the interpreter (93). If the character is \$, the **repeat** statement terminates, otherwise control returns to GETCHAR (90) which gets the next character.

If the character is a digit (95), this digit and succeeding digits are read, and the value of the corresponding number is accumulated in TEMP (97 thru 101). The value of TEMP is then pushed onto the stack (102). The interpreter has now read one character past the last digit, which it has to do in order to recognize the end of the number, so the character pointer is backspaced (102). If the character is a letter (105 and 106), its address NUM(CH)+OFFSET is pushed onto the stack. If OFFSET = 0, the address of A is 1, the address of B is 2, and so on. (The use of OFFSET will be explained later.) The remaining operand is *?* (109 thru 111), which reads a number from the input file and pushes its value onto the stack. The Pascal statement READ(TEMP) reads a signed number from the input file, having skipped over leading blanks.

Lines 113 thru 131 define the actions taken when an operator has been read. Operators use the calculation stack CALSTACK. The character ! (113) pops the top value off the stack and prints it. The Pascal statement WRITE(POPCAL : 1) prints the value of POPCAL without leading or trailing blanks. If blanks are required to separate numbers, they must be explicitly coded in the Mouse program.

The arithmetic operators (115 thru 124) remove two operands from the stack, apply the appropriate operation, and push the result onto the stack. Note that the second operand is on top of the stack, but must be used after the first operand in the noncommutative subtraction and division operations. The Pascal operator **div** means "divide and truncate" (123). The operator . (period) replaces the address on the stack by the corresponding component of the array DATA (126). The assignment operator (=) uses the address and value on the stack to update a component of the array DATA (128 thru 131).

All of the operations on CALSTACK are

coded using the subroutines PUSHCAL and POPCAL. They can be coded more efficiently without these subroutines. For example, the action required for division can be written:

```
CAL := CAL - 1;
CALSTACK[CAL] := CALSTACK[CAL]
div CALSTACK[CAL+1]
```

The advantage of using the subroutines PUSHCAL and POPCAL is that they can check for stack underflow (CAL $\leqslant$ 0 in POPCAL) and stack overflow (CAL $\geqslant$ 20 in PUSHCAL), although these checks are not shown in this listing.

When the double quote character " is encountered, the interpreter prints successive characters up to, but not including, the next quote character (133 thru 139). The Pascal procedure WRITELN writes a carriage return/line feed to the output file, and WRITE(CH) writes the single character CH.

The left bracket [ introduces a conditional clause. The value on top of the stack is removed and examined. If it is positive and nonzero, no action is taken, and the interpreter proceeds to execute the bracketed clause. If the value on the stack is zero or negative, the interpreter skips to the matching right bracket ]. The use of the procedure SKIP enables the interpreter to process nested conditions correctly.

Loops are implemented by lines 143 thru 151. The effect of the left parenthesis *(* is simply to push a stack frame of type LOOP onto the main stack. This stores the current value of CHPOS on the stack. (It also stores the current value of OFFSET, but OFFSET is not used for loops.) When the interpreter encounters the up arrow symbol (↑), it removes and examines the value on top of the calculation stack. If this value is positive and nonzero, there is nothing to do, but if it is zero or negative, the interpreter must exit from the loop. It does this in two steps (148). First, the main stack is popped. This restores the value of CHPOS, which now points to the left parenthesis *(* at the beginning of the loop. Then the procedure SKIP is used to skip over the body of the loop and leave CHPOS pointing to the closing right parenthesis *)*. This is a slightly inefficient method of terminating the loop, because the entire body of the loop is skipped, rather than just the section from ↑ to *)*. When the right parenthesis is encountered during the execution of the loop, CHPOS is set to the stacked value POS, which causes the interpreter to jump back to the opening parenthesis. The stack is used for loops to allow loops to be nested.

The rest of the **case** statement, lines 153 thru 194, handles macro expansion and parameter substitution. When the interpreter encounters the character # (153), it reads the character which follows. This character should be a letter (154). If there is a definition for the macroinstruction (DEFINITIONS[NUM(CH)] > 0), the interpreter pushes a MACRO entry onto the main stack (158) and assigns new values to CHPOS (159) and OFFSET (160). CHPOS now points to the first character of the macro definition. The effect of adding 26 to OFFSET is to allocate 26 local variables in the array DATA for the use of the macro. The address of the local variable A is NUM('A')+OFFSET (see line 107); in the main program this is 1, in a macroinstruction called from the main program it is 27, and so on. If there is no definition for the macroinstruction, the interpreter skips to the semicolon which terminates the call (162). The procedure SKIP must be used to find the semicolon because the actual parameters of the macro call may include macro calls. The interpreter continues to process the macro definition until it encounters an @, at which point it pops the main stack (166). Popping the stack resets OFFSET

Circle 93 on inquiry card.

## 25 START-AT-HOME COMPUTER BUSINESSES

**In "Low Capital, Startup Computer Businesses"**

CONSULTING • PROGRAMMING • MICRO COMPUTER OPPORTUNITIES • SOFTWARE PACKAGES • FREELANCE WRITING • SEMINARS • TAPE/DISC CLEANING • FIELD SERVICE • SYSTEMS HOUSES • LEASING • SUPPLIES • PUBLISHING • HARDWARE DISTRIBUTORS • SALES AGENCIES • USED COMPUTERS • FINDER'S FEES • SCRAP COMPONENTS • *AND MORE . . .*


Plus — ideas on moonlighting, going full-time, image building, revenue building, bidding, contracts, marketing, professionalism, and more. No career tool like it. Order now — if not completely satisfied, return within 30 days for full immediate refund.

• 8½ x 11 ringbound • 156 pp. • $20.00

Phone Orders **901-761-9090**

**DATASEARCH**
incorporated

4954 William Arnold Road, Dept. B, Memphis, TN 38117

Rush my copy of "Low Capital Startup Computer Businesses" at $20.

NAME/COMPANY ______________________
ADDRESS ______________________
CITY/STATE/ZIP ______________________

☐ Check Enclosed ☐ VISA ☐ Master Charge

# ______________ Exp. Date ______________

Circle 372 on inquiry card.

## Trans-Data Corporation

**ANNOUNCES MICROCOMPUTERS SOFTWARE FOR OHIO SCIENTIFIC, APPLE II, AND NORTH STAR.**

### OHIO SCIENTIFIC (C1-P, C2-4P, C2-8P)

| | |
|---|---|
| *Interest on Loans | $ 6.00 |
| *Checking Account | $ 8.00 |
| *Luna Lauder | $ 6.00 |
| *Star Treck 8K | $ 6.00 |
| *Hangman 8K | $ 8.00 |
| *Basic Tutor Series | $29.00 |

### APPLE II

| | | |
|---|---|---|
| *File Management System | 32K | $60.00 |
| *Word Processor | 32K | $60.00 |
| *Step By Step | 16K | $39.95 |

### NORTH STAR

| | |
|---|---|
| *Inventory | $25.00 |
| *Medical Billing | $25.00 |
| *Payroll | $25.00 |

*WE HAVE THE MOST SOPHISTICATED LIBRARY OF SOFTWARE FOR THESE MICROCOMPUTERS IN STOCK. SEND $1.00 FOR SOFTWARE CATALOG OR CALL OUR TOLL FREE NUMBER (800) 327-8455.*

PLEASE SPECIFY DISK OR TAPE VERSION AND COMPUTER MODEL.

**Trans-Data Corp.**
**161 Almeria Avenue**
**Coral Gables, Florida 33134**

## EASYSOFTWARE

*for your* **APPLE**
*Let your* **APPLE** *teach you*
**FRENCH, GERMAN, ITALIAN,** ***and*** **SPANISH**

Each package contains more than 1000 French, German, Italian or Spanish words and their English equivalents. You may answer in French, German, etc. or English either by multiple choice or by fill-in. The programs keep track of correct and incorrect responses and sort words according to their difficulty.

Then enjoy **COMPUTER ART** new beautiful symmetrical high resolution color designs. An almost infinite number! Endless hours of enjoyment. Don't get high and start watching these!

| | | |
|---|---|---|
| ( ) FRENCH | $25 | Name ______________ |
| ( ) GERMAN | $25 | Address ______________ |
| ( ) ITALIAN | $25 | City ______________ |
| ( ) SPANISH | $25 | State ______________ |
| ( ) HI ART | $10 | Zip ______________ |

*Send check or money order to:*

**EASYSOFTWARE**
PO Box 84305
Veterans Adm Br
Los Angeles CA 90073

Circle 103 on inquiry card.

## MICROPOLIS SOFTWARE (MOD II)

### WHY

Do you need the CCA DATA MANAGEMENT SYSTEM?

### BECAUSE

With this powerful system,

*You're in control*

- YOU decide on the files you need (Name & Address A/R, A/P, Payroll, Inventory, Customer Lists, etc.)
- YOU choose the names for the files and all of the data fields in the files.
- YOU let the DMS assist you in maintaining your files using YOUR field names. Records can be added, updated, deleted, inspected, or scanned for in any file.
- YOU let the DMS sort your files as needed.
- YOU select the report or mailing label format, selecting title, fields, totaling, sub-totaling, editing and more.

*Stay in control!*

Order the CCA Data Management System today!
Only $150.00 - Comprehensive Documentation

*Dealer Inquiries Invited*

Creative Computer Applications
2218 Glen Canyon Road
Altadena, CA 91001
(213) 798-4529

VISA master charge

Circle 84 on inquiry card.

correctly, but it leaves CHPOS pointing at the macroinstruction name. The call to SKIP (166) moves CHPOS past the macro call so that processing can continue.

The most difficult task for the interpreter is processing a formal parameter (169 thru 192). First, the interpreter reads the parameter name and sets PARNUM to the corresponding numeric value (170). For example, if it reads *%A*, PARNUM is set to 1. A new entry is created in the main stack. The next problem is to find, in the stack, the stack

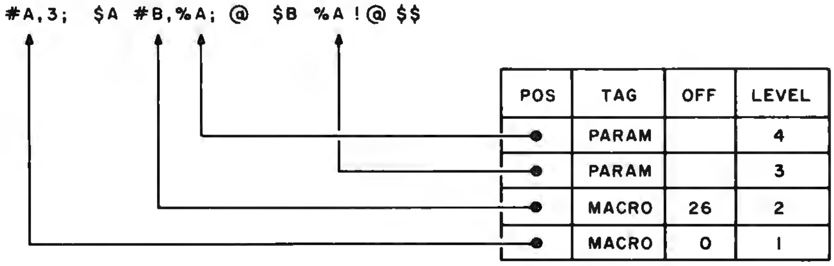



*Figure 1: The interpreter has read the formal parameter %A in the call #B,A; and is looking for the actual parameter 3. Level 1 of the stack, shown at the right, contains a pointer to the corresponding macro call #A,3; . The interpreter finds this pointer by searching the stack for a MACRO frame matching the PARAM frame at stack level 4.*

frame created by the corresponding macro call. This would be easy if the call frame was at the top of the stack, but this is not necessarily the case.

Consider the situation shown in figure 1, which shows a simple Mouse program and the main stack during its execution. The interpreter reads *#A*, creates the stack frame at level 1, and moves CHPOS to *$A*. It then reads *#B*, creates the stack frame at level 2, and moves CHPOS to *$B*. Now it encounters *%A*, creates a PARAM stack frame at level 3, and sets CHPOS to the actual parameter. The actual parameter is *%A* (in the call *#B,%A;*) and so it creates a new stack frame at level 4. Now the interpreter has to find the actual parameter corresponding to this *%A*.

In order to do this it must locate the stack frame at level 1, which contains a pointer to *#A*. It can find the correct frame by using the fact that MACRO and PARAM frames in the stack are nested. (The situation is slightly more complicated when an actual parameter contains a macro call, but the same strategy works.) The search is implemented by lines 171 thru 179 of the interpreter. LOOP frames on the stack are ignored (177).

When the interpreter has found the correct stack frame, it sets the values of CHPOS and OFFSET from it (180 and 181). The stacked value of OFFSET must be used because variables in an actual parameter belong to the level of the macro call, not to the level of its definition. CHPOS now points to the name of the macro in the macro call. The interpreter finds the correct parameter by counting PARNUM commas (182 thru 190).

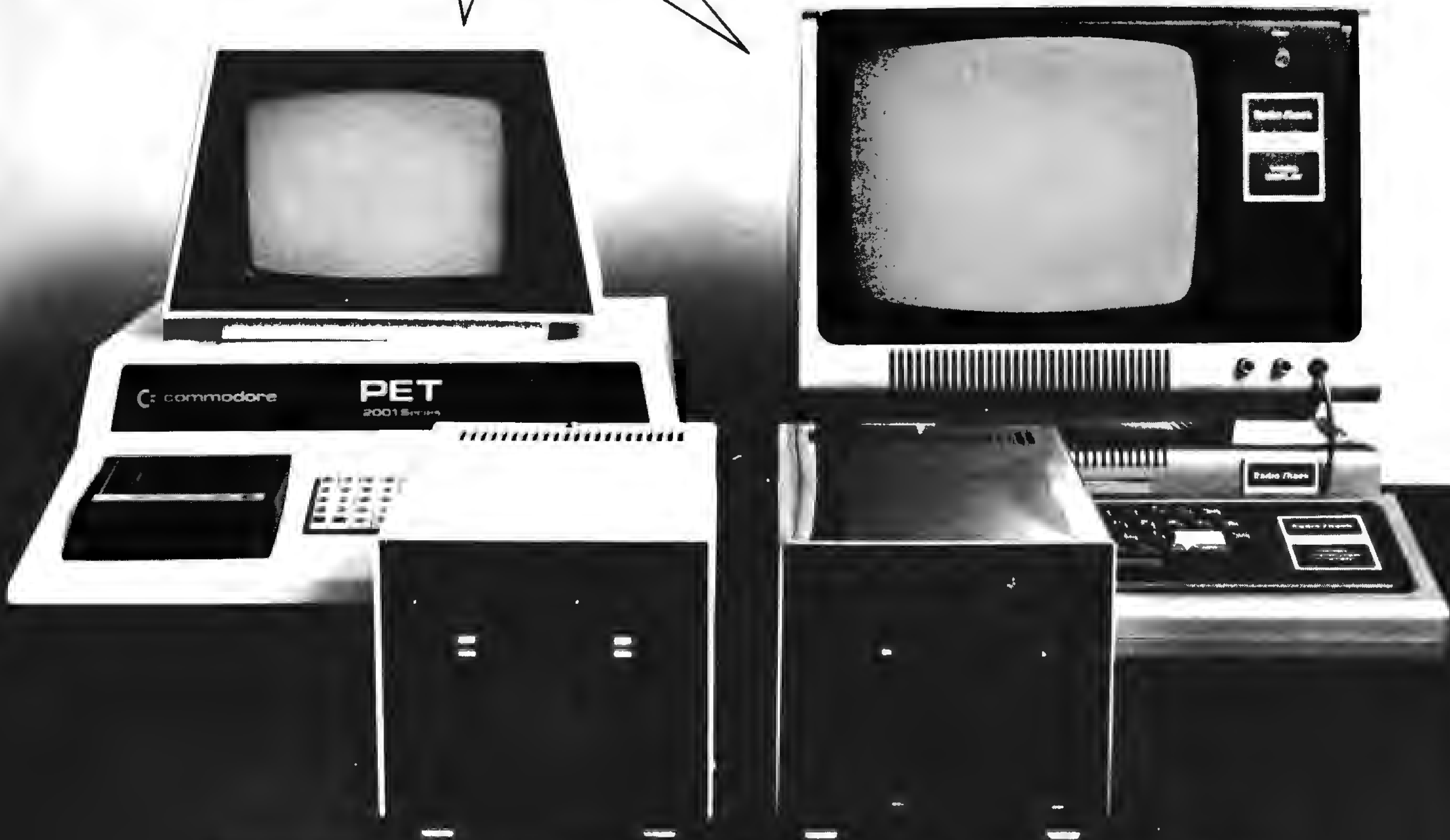
The counting process is complicated by two factors. One is that the actual parameter may contain macro calls; this contingency is handled by SKIP (184 thru 188). The other complication is that there may be no actual parameter corresponding to the formal parameter. In this case, the interpreter will encounter a semicolon (190), and must pop the stack frame that it just created (191).

An actual parameter is terminated by either a comma or a semicolon. The action of the interpreter is simply to pop the main stack (194).

### Improving the Implementation

The Mouse interpreter that is presented here has been pared to the bare essentials. It is complete and accurate, and was used to test the example programs of listings 1 thru 5. Mouse is easier to use, however, if the interpreter does some error checking.

# THANKS FOR THE MEMORIES.




Thanks to these inexpensive Pet and TRS-80 memories and disk drives you can instantly turn your personal computer into a speedy and efficient professional computer.

## Expand your Pet.

1. Disk Drive. Double density 400k on-line disk drive and DOS operating system including 17 additional BASIC commands. Random or indexed sequential access. Exceptionally fast — loads 20k program in 3 seconds. Only $1295.
2. Expandamem™ Completely internal memory that adds up to 32k. 32k unit allows for 8k of assembly language subroutines accessed via the USR command. 16k only $425. 24k only $525. 32k only $615.
3. Software. Complete library available. Write for details.

## Expand your TRS-80.

1. Disk Drive. Compact dual disk drive is completely compatible with TRS-80 DOS and uses own controller. Only $895.

## IMMEDIATE DELIVERY! ORDER TODAY!

**COMPU/THINK™**
3260 Alpine Road, Menlo Park, CA 94025
(415) 854-2577

Please send me the following:
☐ Pet Disk Drive
☐ Pet Expandamem™ ☐ 16k ☐ 24k ☐ 32k
☐ Pet Software (Info only) ☐ TRS-80 Disk Drive

Total Enclosed: $
Check or Money Order only. Calif. Res. add 6% tax. Freight pre-paid in continental U.S.

Name ____________________

Address ____________

City ____________ State ______ Zip ______

Dealer Inquires Invited.

Some of the errors which it can easily detect are listed here:

- Stack underflow and overflow (both stacks). Underflow is always the result of an error in the Mouse program; overflow may be due to an error in the Mouse program, but it is more likely that the stack is not large enough for the program. Checks for underflow should be incorporated into POP and POPCAL, and checks for overflow should be incorporated into PUSH and PUSHCAL.
- Illegal characters. The interpreter should check the program for illegal characters; this can be done during loading.
- Division by zero. TEMP = 0 at line 122.
- Illegal address. The value of the stack at line 126 should satisfy 1 ⩽ POPCAL ⩽ OFFSET+26; the same test can be made at line 129. A stricter address check would be preferable, but is not easy to devise, since a macroinstruction can access the variables both at its own level and at lower levels by means of parameters.
- Context errors. The characters # and % must always be followed by a letter.
- Undefined macroinstruction or missing actual parameter. These would be regarded as errors in a stricter implementation of Mouse.

A tracing option is a powerful aid to debugging Mouse programs. The easiest way to trace the execution of a Mouse program is to make the interpreter display the value of each character it processes. This can be done by inserting WRITE(CH) after GETCHAR at line 90 of the interpreter (listing 6). It is also useful to trace the results of assignments. This can be done by printing the value of TEMP at line 129.

The size of the arrays (12 thru 16) can be adjusted to suit your requirements. Most expressions can be evaluated with a small stack of two or three entries, and you may find it surprising that CALSTACK has space for so many entries. The reason is that some recursive macroinstructions (such as S and T in listing 1) create an entry in the calculation stack at each level of recursion, and CALSTACK must be large enough to hold these. The array DATA makes the poorest use of space; 26 words are allocated at each level of macrocall. This implementation allows ten calling levels, which is not as generous as it sounds if you are using recursive macroinstructions. You can reduce the space requirement to ten variables (A,B. . .J) at each level, by changing line 160 to OFFSET := OFFSET + 10. Note that a macroinstruction with no local variables needs space in the array STACK but not in the array DATA, so recursive macros such as S and T in listing 1, V in listing 3, and H in listing 4, are limited only by the size of the arrays STACK and CALSTACK.

### Improving the Language

It is easy to add features to Mouse. A random number generator is useful, particularly for programming games. Probably the simplest method is to use a unary operator which multiplies the number on the top of the calculation stack by a real random number R such that $0 < R < 1$, truncates the result, and increments it. If the character *:* (colon) is used to denote the operator, then *6:* would leave a simulated die throw on the stack.

The most severe restriction of this particular version of Mouse is that it cannot process character data. A more powerful version of Mouse can be obtained by redefining *?* and *!* so that they read and write a single character. The disadvantage is that macroinstructions are then required to read and print numbers – not a large price to pay for the greater generality achieved.

It is quite easy to add a case construction to the language. The following syntax is suitable:

$$E<C_1,C_2 \ldots C_n>$$

Each $C_i$ is a clause. When the interpreter reads $<$, it performs the following actions:

- skip to the matching $>$ and push a CASE frame onto the main stack;
- return to $<$;
- if $E \geqslant 1$, then scan the clause list until the (n–1)th comma is encountered, otherwise pop the stack. If the character $>$ is encountered during this scan, then $E > n$, and the case clause is null or illegal.

The action for comma, which is already defined to be POP, is correct. This construction will select and execute one of the clauses $C_1,C_2 \ldots C_n$, according to the value of E. It is very easy to write a random sentence generating program in a version of Mouse to which a random number generator and a case construction have been added.

The facilities for annotating Mouse programs are very limited. Strings in quotes may be used in the main program outside loops. These serve as comments to the program



Circle 329 on inquiry card.

text and also as a guide to progress when the program is running, since they are printed by the interpreter. The interpreter ignores text between macro definitions, so macroinstructions can be titled. However, these methods use up valuable space in the array PROG. A better solution is to use a special symbol such as ' (apostrophe), or a pair of symbols such as { and } , and to modify the loader so that it does not store comments in memory.

### Conclusion

Mouse is simple enough to be implemented on a small computer system in a few days, yet it is rich enough to give insight into the mechanisms used by much higher level languages. The Mouse interpreter can be used by itself or as part of a larger system. The General Purpose Macrogenerator is considerably more powerful than Mouse, but nonetheless I think that Strachey's appraisal of the GPM provides an apt conclusion:

> It has been our experience that the GPM, while a very powerful tool in the hands of a ruthless programmer, is something of a trap for the unsophisticated one. It contains in itself all the undesirable features of every possible machine code — in the sense of inviting endless tricks and time-wasting though fascinating exercises in ingenuity — without any of the irritating *ad hoc* features of real machines. It can also be almost impenetrably opaque, and even very experienced programmers indeed tend to spend hours simulating its action when one of their macro definitions goes wrong. Furthermore, it is remarkably good at using up machine time — fortunately the programs written for it are usually rather short. ■

#### REFERENCES


1. Grogono, Peter, "MUSYS: Software for an Electronic Music Studio," *Software: Practice and Experience,* volume 3, 1973, pages 369 thru 383.
2. Strachey, C, "A General Purpose Macrogenerator," *Computer Journal,* volume 8, 1965, pages 225 thru 241.
3. Jensen, K and N Wirth, *PASCAL User Manual and Report,* Springer-Verlag, 1976.
4. Grogono, Peter, *Programming in PASCAL,* Addison-Wesley, 1978.

# Intel 8080 Microprocessor Instruction Set

**Technical Forum**

In my work with the 8080 microprocessor, I have found the accompanying instruction set summary very useful. The instructions are grouped in five tables according to function, with a single line summary being given for each instruction. In addition, there is an explanatory table of nomenclature and symbols. The order of the instructions is the same as given in the *Intel 8080 Microcomputer Systems User's Manual*. The method of tabulation makes it very clear which registers and flags are affected by the execution of each instruction.


R S Clist PhD, Scientist,
Electronics Group, Dept of Scientific and
Industrial Research, POB 2225, Auckland
NEW ZEALAND.■


| Symbol | Meaning |
|---|---|
| A | 8 bit accumulator |
| B, C, D, E, H, L | 8 bit general purpose registers |
| F | Condition code flags Z, S, P, CY, AC (Zero, Sign, Parity, Carry, and Auxiliary Carry) |
| PSW | 16 bit Processor Status Word (comprising A and F) |
| rp | One of the register pairs: B representing BC; D representing DE; H representing HL; SP representing 16 bit stack pointer |
| PC | 16 bit program counter register |
| M | Memory location whose address is in HL |
| r, r1, r2 | One of the registers A, B, C, D, E, H, L, or M |
| $r_m$ | bit m of the register r   7 6 5 4 3 2 1 0 |
| data | 8 bit data quantity |
| addr, data16 | 16 bit address or data quantity (note: low byte is stored *first*) |
| pp | 8 bit port number |
| n | Restart number 0 thru 7 |
| cc | One of the condition code tests: NZ not zero Z = 0; Z zero Z = 1; NC no carry CY = 0; C carry CY = 1; PO parity odd P = 0; PE parity even P = 1; P plus S = 0; M minus S = 1 |
| ( ) | indirect reference |
| –(SP) | stack push operation |
| (SP)+ | stack pop operation |
| ← | is replaced by |
| ↔ | is exchanged with |
| — | boolean NOT (bar above symbol) |
| ∧ | boolean AND |
| ∨ | boolean OR |
| ⊻ | boolean Exclusive – OR |
| + | addition |
| — | two's complement subtraction |

### Data Transfer Group

| Flags affected Z | S | P | CY | AC | Registers affected | Bytes | Op code | Operands | Meaning | Action | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | r2 | 1 | MOV | r2, r1 | Move register r1 to register r2 | r2 ← r1 | |
| | | | | | r | 2 | MVI | r, data | Move immediate data | r ← data | |
| | | | | | rp | 3 | LXI | rp, data16 | Load register pair immediate | rp ← data16 | |
| | | | | | A | 3 | LDA | addr | Load accumulator direct | A ← (addr) | |
| | | | | | | 3 | STA | addr | Store accumulator direct | (addr) ← A | |
| | | | | | HL | 3 | LHLD | addr | Load HL direct | HL ← (addr) | |
| | | | | | | 3 | SHLD | addr | Store HL direct | (addr) ← HL | |
| | | | | | A | 1 | LDAX | rp* | Load A indirect | A ← (rp) | |
| | | | | | | 1 | STAX | rp* | Store A indirect | (rp) ← A | |
| | | | | | HL, DE | 1 | XCHG | | Exchange HL with DE | HL ↔ DE | *B or D only |

### Arithmetic Group

| Flags affected Z | S | P | CY | AC | Registers affected | Bytes | Op code | Operands | Meaning | Action | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| √ | √ | √ | √ | √ | A, F | 1 | ADD | r | Add register | A ← A + r | |
| √ | √ | √ | √ | √ | A, F | 2 | ADI | data | Add immediate | A ← A + data | |
| √ | √ | √ | √ | √ | A, F | 1 | ADC | r | Add register with carry | A ← A + r + CY | |
| √ | √ | √ | √ | √ | A, F | 2 | ACI | data | Add immediate with carry | A ← A + data + CY | |
| √ | √ | √ | √ | √ | A, F | 1 | SUB | r | Subtract register | A ← A − r | |
| √ | √ | √ | √ | √ | A, F | 2 | SUI | data | Subtract immediate | A ← A − data | |
| √ | √ | √ | √ | √ | A, F | 1 | SBB | r | Subtract register with borrow | A ← A − r − CY | |
| √ | √ | √ | √ | √ | A, F | 2 | SBI | data | Subtract immediate with borrow | A ← A − data − CY | |
| √ | √ | √ | | √ | r, F | 1 | INR | r | Increment register | r ← r + 1 | |
| √ | √ | √ | | √ | r, F | 1 | DCR | r | Decrement register | r ← r − 1 * | |
| | | | | | rp | 1 | INX | rp | Increment register pair | rp ← rp + 1 | *binary subtraction |
| | | | | | rp | 1 | DCX | rp | Decrement register pair | rp ← rp − 1 * | |
| | | | √ | | HL, CY | 1 | DAD | rp | Add register pair to HL | HL ← HL + rp | |
| √ | √ | √ | √ | √ | A, F | 1 | DAA | | Decimal adjust accumulator | The 8 bit number in A is adjusted to form two 4 bit BCD digits. | |

### Logical Group

| Z | S | P | CY | AC | Registers affected | Bytes | Op code | Operands | Meaning | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| √ | √ | √ | 0 | √ | A, F | 1 | ANA | r | AND register | $A \leftarrow A \wedge r$ |
| √ | √ | √ | 0 | √ | A, F | 2 | ANI | data | AND immediate | $A \leftarrow A \wedge data$ |
| √ | √ | √ | 0 | 0 | A, F | 1 | XRA | r | Exclusive –OR register | $A \leftarrow A \veebar r$ |
| √ | √ | √ | 0 | 0 | A, F | 2 | XRI | data | Exclusive –OR immediate | $A \leftarrow A \veebar data$ |
| √ | √ | √ | 0 | 0 | A, F | 1 | ORA | r | OR register | $A \leftarrow A \vee r$ |
| √ | √ | √ | 0 | 0 | A, F | 2 | ORI | data | OR immediate | $A \leftarrow A \vee data$ |
| √ | √ | √ | √ | √ | F | 1 | CMP | r | Compare register | $A - r$ (Z = 1 if A = r) (CY = 1 if A < r) |
| √ | √ | √ | √ | √ | F | 2 | CPI | data | Compare immediate | $A - data$ (Z = 1 if A = data) (CY = 1 if A < data) |
| | | | √ | | A, CY | 1 | RLC | | Rotate left | $A_{n+1} \leftarrow A_n, A_0 \leftarrow A_7, CY \leftarrow A_7$ |
| | | | √ | | A, CY | 1 | RRC | | Rotate right | $A_n \leftarrow A_{n+1}, A_7 \leftarrow A_0, CY \leftarrow A_0$ |
| | | | √ | | A, CY | 1 | RAL | | Rotate left through carry | $A_{n+1} \leftarrow A_n, CY \leftarrow A_7, A_0 \leftarrow CY$ |
| | | | √ | | A, CY | 1 | RAR | | Rotate right through carry | $A_n \leftarrow A_{n+1}, CY \leftarrow A_0, A_7 \leftarrow CY$ |
| | | | | | A | 1 | CMA | | Complement accumulator | $A \leftarrow \overline{A}$ |
| | | | √ | | CY | 1 | CMC | | Complement carry | $CY \leftarrow \overline{CY}$ |
| | | | 1 | | CY | 1 | STC | | Set carry | $CY \leftarrow 1$ |

### Branch Group

| Z | S | P | CY | AC | Registers affected | Bytes | Op code | Operands | Meaning | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | PC | 3 | JMP | addr | Jump | PC ← addr |
| | | | | | PC | 3 | Jcc | addr | Conditional jump | PC ← addr (if cc true) |
| | | | | | PC, SP | 3 | CALL | addr | Call | –(SP) ← PC, PC ← addr |
| | | | | | PC, SP | 3 | Ccc | addr | Conditional call | –(SP) ← PC, PC ← addr (if cc true) |
| | | | | | PC, SP | 1 | RET | | Return | PC ← (SP)+ |
| | | | | | PC, SP | 1 | Rcc | | Conditional return | PC ← (SP)+ (if cc true) |
| | | | | | PC, SP | 1 | RST | n | Restart | –(SP) ← PC, PC ← 8n |
| | | | | | PC | 1 | PCHL | | Jump HL indirect | PC ← HL |

### Stack, I/O, and Machine Control Group

| Z | S | P | CY | AC | Registers affected | Bytes | Op code | Operands | Meaning | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | SP | 1 | PUSH | rp* | Push register pair | –(SP) ← rp |
| | | | | | SP | 1 | PUSH | PSW | Push processor status word | –(SP) ← A, F |
| | | | | | SP, rp | 1 | POP | rp* | Pop register pair | rp ← (SP)+ |
| √ | √ | √ | √ | √ | SP, A, F | 1 | POP | PSW | Pop processor status word | A, F ← (SP)+ |
| | | | | | HL | 1 | XTHL | | Exchange stack top with HL | HL ⟷ (SP) |
| | | | | | SP | 1 | SPHL | | Move HL to SP | SP ← HL |
| | | | | | A | 2 | IN | pp | Input | A ← (pp) |
| | | | | | | 2 | OUT | pp | Output | (pp) ← A |
| | | | | | | 1 | EI | | Enable interrupts | Enable interrupts after execution of *next* instruction. |
| | | | | | | 1 | DI | | Disable interrupts | Disable interrupts after execution of *this* instruction. |
| | | | | | | 1 | HLT | | Halt | Stop the processor (may be started again only by interrupt or hardware restart). |
| | | | | | | 1 | NOP | | No Operation | No operation is performed. |

* B, D, or H only

## Technical Forum

# Headwind Progress Made

In "Computer Assisted Flight Planning" (March 1979 BYTE, page 206), the author did, indeed, identify a problem suited for the computer. One aspect of the article is unclear and I would like to suggest a solution to the headwind/tailwind limitation described by the author.

The unclear aspect of the article relates to the *two pass* system. Such a system is not really necessary as the author has already solved the problem. The *drift* equation is the exact solution of the velocity vector triangle for the crab angle (angle at which the plane must be turned so that the resultant forces of the wind and airplane produce travel in the desired direction). In other words, the *drift* plus the *true course heading* yields the *true heading* to be taken. The given equation is instructive in that all elements are included for students to identify. However, for programmable calculators with very limited program steps, this equation can be shortened by using the identities:

$\sin(-x) = -\sin(x)$
and
$\sin(x \pm 180) = -\sin(x)$.

This yields:

$$\text{drift} = \sin^{-1}\left[\frac{\text{wind}}{\text{true}} \times \sin(\text{wind} - \text{true})\right]$$

where *wind direction* is defined as the direction from which the wind is coming.

The author, in solving the velocity vector triangle for the ground speed, uses the Law of Sines. This introduces the limitation that the equation cannot be used for direct tail wind or head wind situations, since a nonphysical answer results from division by zero ($\sin(0) = 0$). The use of the Law of Cosines avoids this problem and yields:

$$GS = \sqrt{(TAS)^2 + W^2 - 2(W)(TAS)\cos(WD - TC - CA)}$$

where:

| | | |
|---|---|---|
| GS | = | ground speed, |
| TAS | = | true air speed, |
| W | = | wind speed, |
| WD | = | wind direction (the direction *from* which the wind is coming), |
| TC | = | true course heading, |
| CA | = | crab angle ■ |

**Bruce Ronald Fiene**
**Commercial Pilot, ASMEL Instrument**
**408 Big Stone Dr**
**Xenia OH 45385**

# Subroutine Parameters

W D Maurer
University Library Bldg
Room 634
George Washington University
Washington DC 20052

If you've written computer programs in any language, you must be aware by now what a subroutine is, although you might not have written any. The basic concept of a subroutine is present in all computer languages, although every language implements it a bit differently from the others. In systems based on the 8080, the 8085, or the Z-80, you write CALL SUB to call the subroutine called SUB. On the 6800 and the 6502, it's JSR SUB, while in BASIC it's GOSUB $\alpha$ where the first statement of the subroutine SUB is on line number $\alpha$. But regardless of the language, the concept is the same: you have something in your program that you want to do more than once. It may be looking up an element in a table; it may be printing out a list; it may be making an access to a data structure; but whatever it is, you need it at various times in your program. You don't want to have to write out the same instructions over again every time you need that particular job to be done, because this is wasteful of memory space. So, therefore, you group together the instructions that do this job into a *subroutine*, and then, at any point that you want the job to be done, you put in an instruction to *call* the subroutine. When the subroutine is finished, it *returns* to the point immediately following the place where it was called; and this is also done differently in different programming languages — one writes RETURN in BASIC, RET for the 8080 and Z-80, and RTS (return from subroutine) for the 6800 and 6502.

All this is fine if the job you want to do repeatedly is exactly the same every time you want to do it. But, in practice, this is usually not the case. For example, if you are looking up an element in a table, you are probably looking up a different element each time. If you are multiplying two 16 bit quantities — a very common subject for a small system subroutine — the quantities you are multiplying are probably not the same from one multiplication to the next, and the result is also probably not the same variable. This is true even though the logic of multiplication does stay the same. It is this that has led to the idea of subroutine parameters, the subject of this article.

## Parameters

In applied mathematics, there is a concept of parameter which will be familiar to those small system users who have backgrounds in engineering or physical science. Consider, for example, the graph of a function. You are usually expressing $y$ in terms of $x$, but if you are constructing the graph of a circle, it is sometimes more useful to introduce another variable $\theta$ to represent the angle, and then to express both $x$ and $y$ in terms of $\theta$. The variable $\theta$, in this context, is called a parameter. In computer programming, however, whether on large systems or small ones, the word "parameter" has a more general meaning, and one which does not require any knowledge of applied mathematics; it is simply any variable which is used by a subroutine, and which is supplied to that subroutine by the program that calls it.

Parameters of subroutines are related to *arguments* (sometimes also called parameters or formal parameters) of functions. If you have a function $f(t)$ or $g(a, b)$ or $h(x, y, z)$, then $t$, $a$, $b$, $x$, $y$, and $z$ are the arguments. On a computer, the value of a function is computed by a subroutine, and this must be considered as one special kind of subroutine. Some languages allow you to use functional notation for functions; thus $h(x, y, z)$ might be FNH(X, Y, Z) in BASIC, for example (provided that the definition of $h$ was simple enough). In assembly language, however, one generally uses the same instructions (CALL, JSR, or whatever), whether one is calling a subroutine to calculate the value of a function, or a more general subroutine.

Those who work with big computers have laid out a considerable amount of terminology dealing with parameters and how they are supplied, or *passed*, to a subroutine by the program that calls it (and sometimes vice versa). One of the purposes of this article is to lay out this terminology for the small system user so that he or she will not have to reinvent the wheel. It should be emphasized that, for a long time, mathematicians believed that there ought to be a single concept of parameter that would work well in all situations. Gradually we have come to realize that there are at least four, and probably a good deal more, reasonable implementations of parameter passing. These will be detailed in what follows.

### Two Examples

To illustrate why the concept of parameter differs from one situation to another, let us consider two simple subroutines: an output subroutine and a multiplication subroutine. The output subroutine will be called OUTPUT(X), and its job will be to output the character X. The multiplication subroutine will be called MULT16(I, J, N), and its job will be to multiply the two 16 bit quantities I and J, producing the result N. The problem we are to solve is how to call OUTPUT(Z), OUTPUT(Q), and so on, for various characters we wish to output, and similarly MULT16(A, B, C), MULT16(U, V, W), and so on, for various multiplications we wish to perform.

Consider first the case of the output subroutine. Suppose that in this subroutine there is a variable called X. In order to output Z, for example, we move Z to X just before calling OUTPUT. The same sort of thing will work for Q, or any other character we wish to output. This method of passing parameters is known as call by value. It may be defined more formally as follows. Suppose we have a subroutine such as OUTPUT(X), where X stands for *any* parameter, such as Z or Q, that might actually be supplied. Here Z and Q are called the *actual* parameters, and X is called the *formal* parameter. Then call by value consists of:

1. Moving the value of the actual parameter to the formal parameter. (If there is more than one formal parameter — as in the case of a function $h(x, y, z)$ — then they must all be moved.)
2. Calling the subroutine.

In assembly language it is very common for X, in a situation such as the above, to be a register. Then all we have to do is to load the register before we call the subroutine; the subroutine assumes that Z, or Q, or whatever stands for X, is in that register. (On the 8080, the Z-80, the 6800, and the 6502, the most common register used for this purpose is the A register, although ISIS, the operating system for the Intellec, which is an 8080 based system, uses the C register.)

If we now look at MULT16, however, we can see without too much trouble that call by value doesn't work. Let us see why not by laying out a specific example. Suppose we are calling MULT16(U, V, W), where MULT16 has been defined as a subroutine with parameters I, J, and N. That is, I, J, and N are the formal parameters, while U, V, and W are the actual parameters. To use call by value, we would first have to move the values of U, V, and W into I, J, and N. That is, U would be moved to I; V would be moved to J; and W would be moved to N. Now we would call the subroutine; and the subroutine, we are assuming, multiplies the 16 bit quantities I and J and sets N equal to the result.

What is wrong with this? Since we were calling MULT16(U, V, W), what we presumably wanted was to multiply the two 16 bit numbers U and V, and set W equal to the result. It is not too hard to see that we did, actually, multiply U by V, because we set I equal to U, and J equal to V, and then we multiplied I by J. But what happens to W? We set N equal to the result of multiplying U by V; but we didn't set W equal to anything. (We also, earlier, set N equal to W — an unneeded and useless operation.) The general situation here is that whenever we have a *formal* parameter that is *set* to some new value by a subroutine, call by value will not work; the formal parameter will not be set to the new value (or to any new value).

Because of this, people who work with big computers came up with three alternative methods of passing parameters. The first of these is known as call by value and result (or sometimes, informally, as "copy-restore"). The second is known as call by reference (or sometimes "call by address" or "call by location"). The third is known as call by name. We shall take up each of these in turn.

### Call by Value and Result

Call by value and result is a rather straightforward way of fixing the bug in call by value that should be evident from the preceding discussion. In fact, what we wanted to do in our MULT16 subroutine was as follows:

1. Set I equal to U and J equal to V.
2. Call the subroutine (which multiplies I by J, giving N).
3. Set W equal to N.

In other words, there are two parameter-passing operations — one just before the subroutine starts, the second one after it ends — and one is the reverse of the other. In the first operation, we move *actual* parameters to *formal* parameters. In the second operation, we move *formal* parameters to *actual* parameters. The parameters we move the first time are the ones that are *used* by the subroutine; the parameters we move the second time are the ones that are *set* by the subroutine.

But how can we tell which parameters are used and which ones are set? It won't always be the case that the first two are used and the last one is set (if there are three altogether). They might all be used, or two of them might be set, or any number of possible combinations. Again, there is more than one reasonable solution to this problem.

The solution chosen by the designers of a number of computer languages in widespread use by the American military establishment (NELIAC, JOVIAL, CMS-2) was to build the distinction between used and returned parameters into the syntax of the language. In other words, when you call a subroutine in any one of these languages, you would have to specify, in some way, which of these you intended to be used and which you intended to be returned. (JOVIAL, for example, uses a semicolon; we would speak of MULT16(U, V; W), for example, where the semicolon separates the used parameters U and V from the returned parameter W.) This certainly solves the problem, although only if you are going to use call by value and result, at the cost of making life a trifle more complicated for those who don't want to have to worry about how parameters are passed.

The other solution, chosen by IBM, is to regard *all* parameters as *both* used and returned at all times. This may seem a bit wasteful, but in fact, compared to call by reference (to be described below), it is more efficient, most of the time. It does, however, lead to some strange and unusual results, the most famous of which may be illustrated as follows. Suppose we have a subroutine D(X, Y), where X and Y are the formal parameters, and suppose that this sets X to zero and does not change Y. Now suppose that we call D(L,L). Of course, we would like this to set L equal to zero. But see what happens:

1. Since X and Y are treated as both used and returned, our first step is to set X equal to L and Y equal to L.
2. Now we call the subroutine, which sets X equal to zero and does not change Y.
3. Finally, we return the actual parameters. First we return X by setting L equal to X. Since X is now zero, this will set L equal to zero, which is exactly what we wanted. But now we return Y by setting L equal to Y. Since Y is still the original value of L, this will undo the previous result, and the final outcome will be that L is the same after calling D as it was beforehand!

The behavior illustrated above can be avoided simply by setting L1 equal to L and then calling D(L, L1), rather than D(L, L). In general, when using call by value and result, with all parameters used and returned, one should never use two actual parameters which are the same. The problem above actually happened to a student of this author, who wrote a big FORTRAN program that ran on the CDC 6400, a computer using call by reference – to be described below – but mysteriously failed to run on the IBM 360, a computer using call by value and result. Many hours of analysis traced the bug to a subroutine call like D(L, L) above.

### Call by Reference

Call by reference, historically, preceded call by value and result, although it was not known by that name at that time. The idea of call by reference is to give the subroutine the *addresses* of its parameters, rather than their values. Then, when the subroutine either uses or sets one of its formal parameters, it does so by making a *reference* to that address. Let us see how this would work on a small system:

1. On the 8080, you can load the HL register pair with the address of the parameter $\alpha$ with the instruction LXI H, $\alpha$ just before calling the subroutine. Then, in the subroutine, if you need to load this parameter into any register *r*, you can use MOV *r*,M; if you need to operate on it arithmetically, you can use ADD M, SUB M, ANA M, and the like; if you need to set it to a new value which is now in register *r*, you can use MOV M,*r*. If you need the HL register pair for other purposes in your routine, you can do an XCHG if you don't need the DE register pair, or you can PUSH H while you use HL and POP H afterward. If there are two parameters, you can load one into HL with LXI H, $\alpha$ as before, and load the other one into BC or DE. If there are several parameters, you can push their addresses onto the stack before calling the subroutine, and pop them back within the subroutine.
2. On the 6800, you can load the X register with the address of the parameter $\alpha$ with the instruction LDX # $\alpha$ (where the # specifies an immediate addressing instruction) just before calling the subroutine. You can now use indexed addressing instructions to manipulate the parameter by loading it (LDAA 0,X or LDAB 0,X), storing it (STAA 0,X or STAB 0,X), or performing arithmetic operations such as ADDA 0,X or ANDB 0,X.

If there is more than one parameter, you can move the addresses of all the actual parameters to fixed locations within the subroutine before calling it. The subroutine can then load each of these into the X register when needed, after which any of the indexed instructions discussed above may be used.

3. On the 6502, there is a general method involving loading the X register, just before calling the subroutine, with the address of a table of addresses of actual parameters. That is, we execute LDX #α where we have written (in page zero):

```
α     DFB     U MOD 256
      DFB     U/256
      DFB     V MOD 256
      DFB     V/256
      DFB     W MOD 256
      DFB     W/256
```

for example, defining a byte for the low order address and then for the high order address of each of the parameters U, V, and W. One can then make reference to the actual parameters by indexed indirect addressing: LDA (0,X) for U, LDA (2,X) for V, and LDA (4,X) for W. This is perfectly general, since LDA (load) can be replaced by STA (store), ADC (add with carry), CMP (compare), AND, and so on.

4. On the Z-80, you can (as always) mimic the 8080, or you can use registers IX and IY to contain the addresses of parameters.

An additional advantage of call by reference is that it allows you to have, as a parameter, the name of an array. For example, you might be writing a subroutine to compare two character strings to see if they are the same. There would be two parameters, namely the two character strings. If you used call by value, you would have to move these entire strings into new locations just before calling the subroutine. This would be wasteful of both time and space, and is, in fact, never done; even systems that use call by value or call by value and result, if they allow array names as parameters, use call by reference (or call by name, to be discussed below) for these. Thus you would only be passing, from the program to the subroutine, the two string starting addresses; that is, for each string, the address of its first byte.

One important source of confusion, when call by reference is used, has to do with how to return a parameter. A large number of programmers try, when they are writing a subroutine, to have it put its answer "somewhere" and then furnish the main program with the address of where that "somewhere" is. This never works, because the main program has no way of using that information. It is not up to the *subroutine* to tell the main program where the information is to be returned; it is up to the main program to tell the subroutine *where to return* the information, and then the subroutine *must* return the information to that point. In particular, the subroutine will never be right if it returns a parameter to a fixed location. When writing a subroutine, if call by reference is used, it should be remembered that this subroutine can be called more than once, with different actual parameters each time, and therefore, when it changes the value of one of its actual parameters, that change must be made by storing this new value in an *indexed* location – where the index is normally the HL register pair on the 8080, the X register on the 6800 and 6502, and (possibly) the IX or IY register on the Z-80.

Call by reference is, in general, more inefficient than call by value and result, particularly if we make reference to a parameter inside a loop. One technique that has been tried on big computers, and works rather well for subroutines that take large amounts of time, is address modification. This involves storing the addresses which are passed as parameters directly into the instructions that use them. Unfortunately, this technique is inappropriate in most microcomputer systems, where the instructions are in read only memory and thus cannot be modified as the program is running. It should also be mentioned that on some systems which use both call by reference and call by value and result, the second of these is implemented as a special case of the first. That is, it is always the addresses, or references, that are passed (so that there is only one kind of standard subroutine protocol rather than two), but, whenever call by value and result is to be used, the subroutine – rather than the main program -- performs the setting of formal parameters to actual parameter values and vice versa.

### Call by Name

This brings us, finally, to call by name – the easiest to define, and yet the hardest to understand, of the better known parameter passing methods. For years, call by name was a *pons asinorum* among big computer software people; that is, the way of distinguishing the bright from the dumb, or the "with-it" from the "not-with-it," was whether you understood call by name. Lately there has been a bit less interest in call by name among practical computer people, since, although it was used in ALGOL 60, one of the first big computer languages (in both senses – big [computer languages] and [big computer] languages), it has not been used in most languages developed since then. But an understanding of it, and of some of the problems that arise with it, is still essential to the amateur as well as the professional computer scientist.

Call by name is defined as follows. Suppose I have a subroutine with a formal parameter X. Suppose I call this subroutine, with actual parameter Y. Then call by name implies that the subroutine is executed as if we had gone through it and substituted Y for every occurrence of X.

There is one important proviso to the above, which may be illustrated as follows. Suppose that in the subroutine we have A = B·X. Suppose now that the actual parameter is not Y, but rather U+V. (It is quite permissible to call SUB(U+V), for example, where SUB is the name of a subroutine.) Now we would like to proceed as if A = B·X really means A = B·(U+V); but if we substitute U+V for X, as in the above definition, we obtain A = B·U+V, which is not quite the same. Therefore we need to change the definition so as to specify the insertion of parentheses. On the other hand, it should also be clear that we do not want to insert parentheses all the time. For example, the variable A could have been the formal parameter, rather than X. In this case, the actual parameter could not be U+V, because then A = B·X would be interpreted as (U+V) = B·X, which makes no sense. But suppose the actual parameter is Y, just as before; we still don't want to write (Y) = B·X (with parentheses) in BASIC or any other algebraic language. Therefore the rule is that the actual parameter is substituted for the formal parameter, *inserting parentheses wherever syntactically possible* (this is the phrase used in the definition of ALGOL 60).

So long as the actual parameters are not expressions like U+V (or like A(I), which could be either a subscripted variable or a reference to a function), call by name is almost identical to call by reference. Therefore, in studying the differences between the two, we have to look at the general rules for handling actual parameters which are expressions. These are that an actual parameter cannot be an expression (other than a single variable, either subscripted or not) when the corresponding formal parameter is *returned*, as we have illustrated above with the formal parameter A and the actual parameter U+V; and, of course, a *formal* parameter can never be an expression.

Suppose now that in our subroutine we have S = S+X, where X is a formal parameter, and the corresponding actual parameter is A(I). (This is a simplification of an actual example given with the definition of ALGOL 60.) Therefore S = S+X becomes S = S+A(I). But now suppose that we want to do this for I = 1 to 10. That would be, presumably, a way of adding the numbers A(1) through A(10), if S were originally set to zero. If we use call by reference, however, this won't work. In call by reference, the address of the actual parameter – in this case, the address of A(I) – would be given to the subroutine. When the subroutine does S = S+X, it would get X from the location which has that address. But that location is a *constant* location – the location, in fact, of A(I) where the variable I has whatever value it had before the subroutine was called. This means that we add X ten times, whatever X is, and in this case we add the *same* value of A(I) ten times, rather than adding A(1) through A(10).

How would we implement call by name? In the above case, when the subroutine does S = S+X, it has to have a way of finding out whether X will stand for a different variable each time. Therefore it loads S and then calls a subroutine to find the value of X, which it then adds and stores the result in S. This means that it is *the address of the start of this subroutine* that is passed, rather than the address of X itself as in call by reference. (This is known as Jensen's device, after a programmer at Regnecentralen, or the National Computer Center of Denmark, who used it in implementing ALGOL 60.) We should remark that there is another entirely different way of implementing call by name, which is to replace each *call* to a subroutine, separately, by the subroutine with the substitutions performed as discussed above. This won't work for ALGOL 60, because it won't work, in general, for recursive subroutines, and it also takes up quite a bit of space if the subroutines are long.

Call by name is considerably less efficient than the other methods we have discussed, which is a big reason for its general decline. Nevertheless, it has its own unexpected advantages. Let us consider a subroutine like D(X, Y), which we discussed earlier, but this time suppose that it simply *uses* X and does not use Y, and let us call D(A, F(B)), where F(B) is a reference to a function. Suppose further that the calculation of F(B) (for some reason) gets the computer into an endless loop. If we use call by name, then, since we never use Y, we have no occasion to call the subroutine that calculates Y – that is, we never call F(B). If we use call by value, however, the first thing we do is to set X equal to A and Y equal to F(B). The result is that we get into the endless loop, in this case, if we use call by value, but not if we use call by name. ■

# Languages Forum

## A "Tiny" Pascal Source Creator

Thomas W Phillips MD
RD 1-551
Chenango Lake Rd
Norwich NY 13815

I would like to thank you for publishing "A 'Tiny' Pascal Compiler" by Kin-Man Chung and Herbert Yuen (September, October and November 1978 BYTE). I now have the compiler working without problems on my 8080 system (Altair with North Star Disk). It is a fascinating way to learn about compilers and Pascal.

I am not sure how Mr Chung and Mr Yuen create their Pascal source programs; however, listing 1 shows my method of creating the Pascal source. Editing is easy with North Star BASIC.

*Listing 1*

```
100 DATA " ! CALCULATE SQUARE. ! "
110 DATA " VAR A,B:INTEGER; "
120 DATA " BEGIN "
130 DATA " READ (B#); "
140 DATA " A: =B*B; "
150 DATA " WRITE (13,10); "
160 DATA " WRITE (B#,' SQUARED IS ',
    A#,13,10); "
170 DATA " CALL (%0800) "
180 DATA " END. "
190 DATA " "
200 DATA " XX "'
210 DIM A$(100)
220 A$ = " "
230 OPEN #1,"A"
240 READ A$
250 IF A$ (1,2) ="XX" THEN 290
260 WRITE #1,A$,NOENDMARK
270 A$ = " "
280 GOTO 240
290 CLOSE #1
300 CHAIN " PASCAL "
READY
RUN


PASCAL & PASCAL 2

P-CODES STARTS AT 0000


0 $A
0 ! CALCULATE SQUARE. !
0 VAR A,B:INTEGER;
1 BEGIN
1 READ (B#);
```

*Listing 1 continued on page 232*

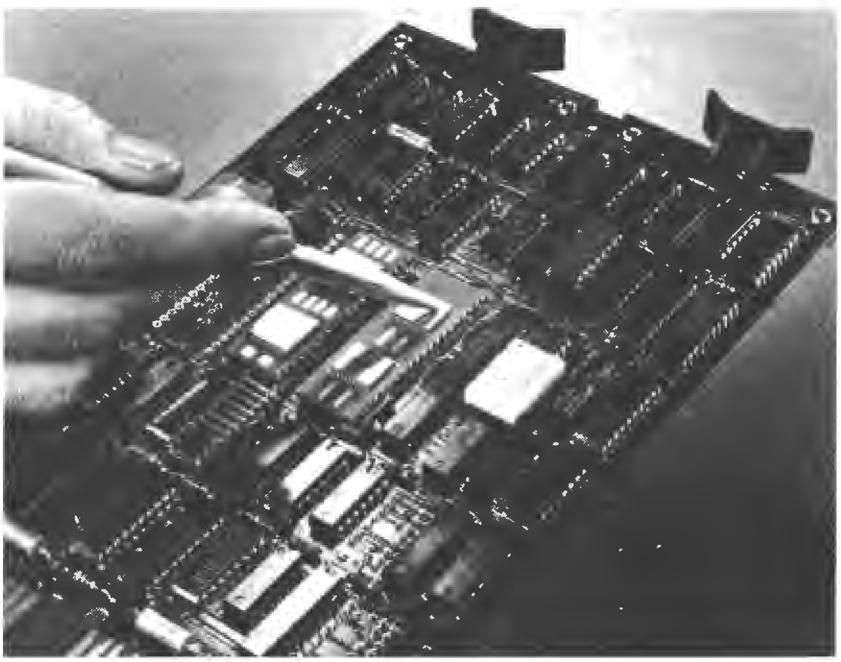
*Listing 1 continued from page 231:*

```
4 A:=B*B;
8 WRITE (13,10);
12 WRITE (B#,'SQUARED IS',A#,13,10);
34 CALL(%0800)
36 END.

PCODTR & PCODTR2
*** P-CODE TO 8080 TRANSLATION ***
PAS.LIB=1A00
P-CODE=0000
8080 OUTPUT=0800
STACK START ADDRESS=8000
STACK END ADDRESS=A000
2 REFERENCES
2 ACTUAL LABELS
0 0809 09 0F 12 18 1A 1D 20 26 2C 2F 35 38 3E 41
15 0847 4D 53 59 5F 65 6B 71 77 7D 83 89 8F 52 58
30 085B 61 64 6A 6D 73 76 79
0 FORWARD REFERENCES
P-CODE.. 37 INSTRUCTIONS
8080.. 121 BYTES
P-CODE:8080 = .81756757
* END TRANSLATION *
READY
BYE
*LF PAS.LIB 1A00
*JP0800
?1
1 SQUARED IS 1
?2
2 SQUARED IS 4 ■
```

# What's New?

## Digital Introduces LSI-11/23 and PDP-11/23 Microcomputers

This new microcomputer called the LSI-11/23 from Digital Equipment Corp, 146 Main St, Maynard MA 01754 has the funtionality and software compatibility of a midrange minicomputer, yet it consists of two 5.2 by 8.9 inch (13.42 by 22.61 cm) boards and backplane. A rack mountable, packaged version, the PDP-11/23, has also been announced. Both versions can run the RSX-11M and 11S operating systems that were previously available only on mid to high-range PDP-11 minicomputers.



The LSI-11/23 features 256 K bytes of memory capacity. It uses the full instruction set of the PDP-11/34 minicomputer, and software supported memory segmentation and protection features of the RSX-11M and 11S multitasking, multiuser operating systems. The LSI-11/23 has the same small size circuit boards as the LSI-11/2, which permit easier placement in instruments and specialized systems. The LSI-11/23 has an optional floating point processor integrated circuit.

Besides accommodating RSX-11M and 11S software, the LSI-11/23 and PDP-11/23 run all software developed for the LSI-11 family without modification. This includes the RT-11 operating system and high level languages such as BASIC, FORTRAN IV and FOCAL. The LSI-11/23 is at least twice as fast as previous LSI-11 family members.

The system is plug compatible with the entry level LSI-11/2. It is also software compatible with the LSI-11/2 and PDP-11 minicomputers.

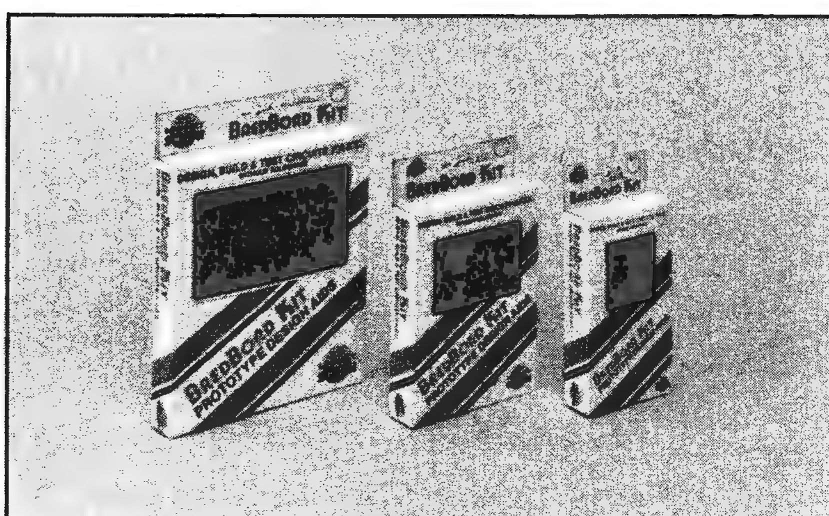
In 100 unit quantities, the LSI-11/23 and PDP-11/23 are priced at $1,758 and $4,500 respectively. The single unit price of the PDP-11/23 is $6,800. A new programmable read only memory board for $300 and programmable read only memory programmer for $1,975 have also been introduced.

Circle 623 on inquiry card

## Design, Build and Test Circuits With BredBord Kits

These BredBord kits are prototype design aids developed to simplify circuit design and building, and testing and wiring without patch cords or solder. The 8 BredBord kit models are assembled on heavy duty, glass epoxy boards, copper clad on each underside for ground plane. They feature color coded, insulated binding posts and rubber mounting feet; and a designated number of test points and capacities for multi-pin dual-in-line package integrated circuits. Each of the kits is made to interface with meters, scopes and other devices. The packages are produced in three types to accept different BredBord sizes. For further information, contact Herman H Smith Inc, 812 Snediker Av, Brooklyn NY 11207.

Circle 624 on inquiry card.



# ADVANCED COMPUTER PRODUCTS

NEW CATALOG AVAILABLE NOW!

THE FIRST TO OFFER PRIME PRODUCTS TO THE HOBBYIST AT FAIR PRICES NOW LOWERS PRICES EVEN FURTHER!

**1. Proven Quality** Factory tested products only, no re-tests or fallouts. Guaranteed money back. We stand behind our products.

## 1979 CATALOG NOW AVAILABLE.

Send $1.00 for your copy of the most complete catalog of computer products. A must for the serious computer user.

### STATIC RAM BOARDS

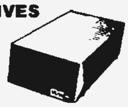
Now, just more than 1½¢ per bit!

**S-100 32K** (uses 2114)
ASSEMBLED — Kit
450ns. 599.00 — 450ns 539.95
250ns. 699.95 — 250ns 599.95
Bare Board 49.95
Bare Board w/all parts less mem. 99.95

**S-100 16K** (uses 2114) KIT (exp. to 32K)
ASSEMBLED — 450ns 279.00
450ns 325.00 — 250ns. 299.00
250ns 375.00
Bare Board 49.95

**LOGOS I 8K**
ASSEMBLED
450 ns 169.95 KIT 450ns 125.95
250ns 189.95 — 250ns 149.95
Bare PC Board w/Data $21.95
Now over 1 year successful field experience
"Special Offer" Buy (4) 8K 450ns Kits $117.00

### FLOPPY DISK DRIVES



1. VISTA V-80 MINIDISK FOR TRS-80
   ★ 23% More Storage Capacity - 40 Tracks
   ★ Faster Drive - Up to 8 Times Faster — 395.00
   2 Drive Cable Add $29.95
   4 Drive Cable Add $39.95
2. VISTA V-200 MINI-FLOPPY SYSTEM
   ★ 204K Byte Capacity ★ w/CPM, Basic "E"
   ★ One Single Sided, Double Density Drive — V-200 699.00
   ★ One Double Density Controller w/Case & P.S.
   **Add to your EXIDY, HORIZON, etc.**
3. VISTA V-1000 FLOPPY DISK SYSTEM
   ★ (2) Shugart 800-R 8" Floppy Disks
   ★ Controller Card, Cable, Case & P.S. — V-1000 1699.00
   ★ CPM & Basic "E". Instructions & Manual
4. MPI B51-5¼", 40 tracks ...... 279.00
5. Shugart SA400-5¼", 35 tracks .... 295.00
6. Siemens/GSI FDD100-8 8" ...... 375.00
7. Shugart 800/801R 8" ...... 495.00
8. PERSCI Model 277 Dual .... 1195.00

### EXPANDORAM MEMORY KITS

★ Bank Selectable ★ Uses 4115 or 4116 200 ns.
★ Write Protect ★ Power 8VDC, ±16VDC
★ Phantom ★ Lowest Cost/Bit

| Expando 32 Kit (4115) | | Expando 64 Kit (4116) | |
|---|---|---|---|
| 8K | $179.00 | 16K | $248.95 |
| 16K | $229.00 | 32K | $369.00 |
| 24K | $299.00 | 48K | $469.00 |
| 32K | $349.00 | 64K | $565.00 |

### IMS STATIC RAM BOARDS

★ Memory Mapping ★ Low Power $699.00
★ Phantom ★ Assembled & tested
Recommended by Alphamicrosystems

| | 250 ns. | 450 ns. |
|---|---|---|
| 8K Static | $209.00 | $189.00 |
| 16K Static | $449.00 | $399.00 |
| 32K Static | $799.00 | $699.00 |

### ANADEX PRINTER

Model DP-8000 compact, impact, parallel or serial. Sprocket feed, 80 cols, 84 lines/min., bi-directional. New only .. $895.00

### VERBATIM™ DISKETTES

LOW PRICE TOP QUALITY DISKETTES

★ 5¼" Minidiskettes ★ Softsector, 10 Sector, 16 Sector $4.25 Each, 10/39.95
★ 8" Standard Floppy Disks ★ Soft Sector, Hard Sector $4.50 Each, 10/41.95
*Add 4.95 for 10 Pack in Deluxe Disk Holder

### 6800

**MOTOROLA EXORCISER COMPATIBLE**

9600 MPU Module w/6802 CPU .... $495.00
9601 16 Slot Mother Board ...... 175.00
9602 Card Cage (19" Retma Rack Mount) .. 75.00
9603 8 Slot Mother Board ...... 100.00
9604 Switchmode System Power Supply .. 250.00
9610 Utility Prototyping Board ...... 36.00
9616 Quad 8K Eprom Module ...... ***
9620 16 Channel Parallel I/O Module .. 295.00
9622 Serial/Parallel I/O Combo .... ***
9626 8K Static RAM Module ...... 295.00
9627 16K Static 450ns ...... 495.00
9630 Card Extender ...... 68.00
9640 Multiple Programmable Timer (24 Timers) ...... 395.00
9650 8 Channel Duplex Serial I/O ...... 395.00
96103 32/32 I/O Module ...... 275.00
96702 32 Point Reed Relay Module ...... 350.00

**6800 BARE BOARDS**

| | | | |
|---|---|---|---|
| 9620-0 | $45.00 | 9603-0 | 27.00 |
| 9626-0 | 45.00 | 9600 | 55.00 |
| 9650-0 | 45.00 | 96103 | 55.00 |
| 9601-0 | 50.00 | 96702 | 55.00 |

Also AMI EVK System in Stock

### APPLE/EXIDY/EXPANDO TRS 80 16K-UPGRADE KIT

★ 16K with Jumpers & Instructions for either Level I or Level II .... $89.95
★ 16K for Apple II Upgrade ...... $89.95
Special: TRS80 Schematic ...... $ 4.95
Expansion Interface Schematic .... $ 4.95

### TRS 80 TO S-100 PET TO S-100 ADAPTER

Allows Pet/TRS 80 to be interfaced to popular S-100 Bus.
Pet to S-100 Kit ...... $189.95
Assembled ...... $269.95
TRS 80 to S-100 HUH 8100 Kit ... $275.00
Assembled ...... $355.00

### KEYBOARD ASCII ENCODED

One time purchase of NEW Surplus keyboards. From the Singer Corporation. The keyboard features 128 ASCII characters in a 63 key format, MOS encoder circuitry "N" key rollover, lighted shift lock, control, escape and repeat functions. Ltd Qty **63 KEY $59.95**

### UV "Eprom" Eraser

**Model UVs-11E $64.95**
Holds 4 Eprom's at a time. Backed by 45 years experience.
**Model S-52T ... $219.95**
Professional Industrial Model

### TARBELL FLOPPY INTERFACE

★ Z80/8080 S100 Compatible ★ Uses CPM
Assembled for Shugart **SALE $229.00**
Assembled Other Drives $269.95
Kit $179.95
Bare Board $36.95 (Doc Add $10.00)
Vista Double Density 5¼" Controller Assem $299.00
SD Versa Floppy Kit $159.95
SD Versa Floppy Assembled $189.95
Tarbel Cassette I/O Kit $115.00
Sale ★ 1771-01 Floppy Chip $27.95

### BYTE USER 8K EPROM BOARD

★ Power on Jump ★ Reset Jump
Assembled & Tested ...... $94.95
Byteuser Kit ...... $64.95
Bare PC Board ...... $21.95
**Special Offer: Buy 4 kits only $59.95 each**
MR-8 8K w/1K Ram ...... $99.50
MR-16 16K w/1K Ram ...... $99.50
EPM-1 4K 1702 ...... $59.95
EPM-2 2708 or 2716 Eprom ...... $69.95

### Z-80/Z-80A/8080 CPU BOARD

★ On board 2708 ★ 2708 included (450ns)
★ Power on jump ★ completely socketed
Assembled and tested ...... $185.00
Kit ...... $129.95
Bare PC Board ...... $ 34.95
★ For 4MHz Speed Add $15.00
8080A Kit ...... $ 99.95
8080A Assembled ...... $149.95

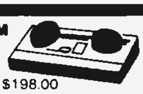
### S-100 MOTHERBOARD SPECIAL

8 slot expandable w/9 conn
reg $69.95 ...... NOW $52.95

### TARBELL FLOPPY CONTROLLER

Card assembled and tested for use with Shugart Drives $ SALE PRICE only $229.00

### ACOUSTIC MODEM



NOVATION CAT™
0-300 Baud
Bell 103
Answer, Originate $198.00

### ACOUSTIC COUPLER SPECIAL

AJ MODEL A30
SPECIAL PURCHASE OF SURPLUS UNITS
AVAILABILITY LIMITED $29.95

### DATA BOOKS • COMPUTER BOOKS

| | | | |
|---|---|---|---|
| 1978 IC Master | 49.95 | Intel MCS 80 Manual | 7.95 |
| NSC TTL Data | 3.95 | Intel MCS 40 Manual | 7.95 |
| NSC Linear | 4.95 | AMD 8080A Manual | 5.95 |
| NSC Linear App Notes II | 3.95 | AMD Schottky Databook | 4.95 |
| NSC CMOS | 3.95 | AMI MOS/LSI Data | 3.95 |
| NSC Memory | 3.95 | GI MOS/LSI Data | 4.95 |
| Intel Databook | 4.95 | Harris Analog Databook | 4.95 |
| Intel MCS 85 Manual | 7.50 | TI Linear Control Data | 3.95 |

**SALE • OSBORNE BOOKS • SALE**

| | Reg. | Sale |
|---|---|---|
| Intro to Micros Vol 0 | ~~8.50~~ | 7.75 |
| Intro to Micros Vol I | ~~8.50~~ | 7.75 |
| 8080A Programming | ~~8.50~~ | 7.75 |
| 6800 Programming | ~~8.50~~ | 7.75 |
| Z80 Programming | ~~8.50~~ | 7.75 |
| Vol II Some Real Microprocessors w/Binder | ~~30.00~~ | 27.50 |
| Vol III Some Real Support Devices w/Binder | ~~20.00~~ | 18.50 |
| Intro to Micros Vol II | ~~20.00~~ | 18.50 |

**SALE • DILITHIUM COMPUTER BOOKS • SALE**

| | | |
|---|---|---|
| Understanding Computers | ~~8.95~~ | 7.95 |
| 8080 Microcomputer Experiments | ~~13.95~~ | 11.95 |
| Beginning BASIC | ~~9.95~~ | 8.95 |
| Beginners Glossary & Guide | ~~6.95~~ | 5.95 |
| Peanut Butter & Jelly Guide to Computers | ~~7.95~~ | 6.95 |
| 8080 Machine Language Programming | ~~7.95~~ | 6.95 |
| Home Computers Vol I Hardware | ~~7.95~~ | 6.95 |
| Home Computers Vol II Software | ~~13.95~~ | 11.95 |
| Starship Simulator | ~~7.95~~ | 6.95 |

### MICROPROCESSORS

Z-80 $18.95
Z 80A 18.95
F-8 (3850) 16.95
2650 18.95
CD1802 19.95
8080A 9.45
8080A-4MHz 19.95
**SALE 8085 19.95**
8008-1 14.95
2901 16.95
2901A 24.95
TMS 9300JL 49.95
CP1600 39.95
6502 11.50
6502A 19.95
Mc100 29.95
**6800 16.95**
6802P 24.95
8035 17.50
8755 49.95
8748 69.95

### SUPPORT DEVICES

AM9511 Arith Processor [illegible]
AM 9511-1 300 ns 245.00
AM9517 DMA Controller [illegible]
AM9519 [illegible] 24.35
3881 Z80 PIO 10.45
3881 4 4MHz 14.95
3882 Z 80 CTC 10.45
3882 4 4MHz 14.95
8205 74S138 Decoder 2.95
8212 8 bit I/O 2.50
8214 Priority Int 4.95
8216 Bus Driver 2.50
8224 Clock Gen 2.95
8224 4 4MHz 9.75
8226 Bus Drive 3.95
8T26 Bus Driver 2.19
8228 Sys Contr 5.95
8238 Sys Cont 6.25
8251 Prog I/O 6.95
8253 Int Timer 19.50
8255 Prog I/O 6.50
8257 Prog DMA 19.50
8259 Prog Int 19.50
8275 CRT Controller 74.95
8279 Prog Keyboard 14.95
6810 1 128 x 8 RAM 4.75
6820 PIA 6.50
6821 PIA 6.50
6828 Priority Int 11.00
6834 1 512 x 8 Eprom 12.45
6850 ACIA [illegible]
6852 Serial Adapter 4.95
6845 HD46505 CRT Cont 39.95
6860 Modem 9.95
6862 Modulator 11.95
6871A 1.0MHz OSC 24.95
6875 8.25
6880 Bus Driver 2.19
MC68488 [illegible]
1821 SCD 1K RAM 25.00
1822 SCD 256 x 4 RAM 16.45
1824 CD 32 x 8 RAM 9.95
1852 CD 8 bit I/O 1.95
1854 Uart 10.95
1856 CD I/O 4.95
1857 CD I/O 8.95
6520 PIA 7.50
6522 Mult 9.25
6530 002 15.50
6530 003 5.50
6530 004 15.50
6530 005 15.50
3851 F8 Prog Store 14.95
3853 F8 Memory I/O 4.45

### DYNAMIC RAMS

416 4116 16K (16 Pin) 12.45
Set of 8 4116's 89.95
4115 8K (16 Pin) 6.95
4050 4K x 1 (18 Pin 4.25
4060 4K x 1 (22 Pin 4.95
4096 4K x 1 (16 Pin) 3.95
2104 4K x 1 (16 Pin) 4.75
4027 4K x 1 (16 Pin 4.95

| | | | |
|---|---|---|---|
| 5261 | 1.95 | 1103 | 1.95 |
| 5262 | 1.95 | 4008L | 4.95 |
| 5270 | 4.95 | 6605 | 7.95 |
| 5280 | 4.95 | 6604 | 4.95 |
| 5290 | 12.45 | 6002 | 1.50 |

### PROMS

2708 [illegible]
2716 [illegible]
1702A [illegible]
1702A 6 [illegible]
2716 5V 54.95
2716 5V 12V 40.00
2758 5V 29.95
5203AQ [illegible]
5204AQ [illegible]
6834 1 [illegible]
IM 5610 [illegible]
**SALE 8223 32 x 8 .......... 2.95**
82S115 512 x 8 (TS) 16.95
82S123 32 x 8 2.50
82S126 256 x 4 3.50
82S129 256 x 4 (TS) 3.50
82S130 512 x 4 (OC) 6.50
NSC DM7578 32 x 8 2.95

### CHARACTER GEN

2513-001 (5V) Upper 9.50
2513-005 (5V) Lower 10.95
2513 ADM3 (5V) Lower 14.95
MCM6571 10.75
MCM6571A 10.75
MCM6574 14.50
MCM6575 14.50

### UARTS/USRTS

TR1602B 5V 12V 3.95
AY51013 (5V 12V 4.95
AY51014A 1517 5 14V 6.95
AY51015A 1863 5V 6.95
TMS 6011 5V 12V 5.50
IM6402 7.95
IM6403 8.95
2350 USRT 9.95
1671B Astros 24.95
**SALE TR1472B .............. 9.95**

### BAUD RATE GEN

MC14411 11.95
4702 14.95
WD1941 [illegible]

### KEYBOARD ENCODERS

AY5-2376 13.75
AY5-3600 13.75
HD0165 9.95
74C922 9.95
74C923 9.95

### A/D CONVERTERS

8700 8 bit Binary 13.50
8701 10 bit Binary 22.00
8703 8 bit TS 11.50
9400 Volt to Freq Conv 7.25
8750 3 1/2 Digit BCD 13.95
1408L6 6 bit 1.95
1408L8 8 bit 5.95
DAC01 A/D 5.95

### STATIC RAM HEADQUARTERS

| | 1-24 | 25-99 | 100 |
|---|---|---|---|
| 21L02 450ns | 1.30 | 1.25 | 1.15 |
| 21L02 250ns | 1.59 | 1.55 | 1.45 |
| 2102 | 1.25 | 1.15 | 1.10 |
| 2111 | 3.75 | 3.65 | 3.55 |
| 2112-1 | 2.95 | 2.85 | 2.65 |
| 2101-1 | 2.90 | 2.70 | 2.55 |
| 2114L-250ns | 12.95 | 10.95 | 9.95 |
| 2114L-300ns | 8.95 | 7.95 | 6.45 |
| **2114L-450ns.** | 7.50 | 6.75 | **4.75** |
| 4044 4041 300ns | 9.95 | 8.75 | 7.95 |
| **4044/4041 450ns.** | 7.50 | 6.75 | **4.75** |
| EMM4200A | 9.75 | 8.75 | 7.95 |
| EMM4402 | 7.95 | 7.25 | 6.25 |
| EMM4804 | 12.50 | 11.50 | 9.95 |
| [illegible] | [illegible] | [illegible] | 7.25 |
| [illegible] | [illegible] | [illegible] | 9.25 |
| AM9124 [illegible] | [illegible] | [illegible] | [illegible] |
| AM9130 [illegible] | [illegible] | [illegible] | [illegible] |
| TSC 4K 4044 ... $18.95 Each | | | |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| 6518 1K CMOS | [illegible] | [illegible] | 7.25 |
| 6518 1K CMOS | [illegible] | [illegible] | 7.25 |
| 74S189 [illegible] | [illegible] | [illegible] | [illegible] 50 |

8155 [illegible] Ram 21.95
2147 Low Power 4K Static 14.95 ea

### CHARGE COUPLED DEVICES

16K CCD First time offered Fairchild 460 CCD 16K Memory now you can experiment with CCD technology at a reasonable price. 17 page Application notes supplied with each order. Quantity limited.
**$18.95 each (reg. 43.00)**

### CRYSTALS

**Microprocessor Timebases TV Game**

Frequency / Price / Frequency / Price [illegible]

### DISPLAYS/OPTO/LED'S

**★ 7 SEGMENT ★ CALC ★ CLOCKS ★**

DL 704 (CC) DL 707 (CA) .300 Red .99
FND 357 (CC) .357 Red .99
FND 500 507 (CC) .500 Red .99
FND 507 510 (CA) .500 Red .99
FND 800 803 (CC) .800 Red 1.75
FND 807 810 (CA) .800 Red 1.75
MAN 3062 .500 Green 1.15
HP5082-7731 (CA) .300 Red .99
9 Digit Bubble Mini Calc Display .99
9 Digit Panaplex Display .400 .99
9 Digit Fluorescent .300 .99
MA1003 12V Auto Clock Module **15.95**
Bezel for MA1003 w/Red Filter 4.95
MA1002A LED 12 hr. Clock Module 10.95

**★ HEX DISPLAYS ★ ENCODED DISPLAYS ★**

HP 5082-7340 Red Hexidecimal 15.95
HP 5082-7300 Red Nymeric 14.95
TIL 306 Numeric w/Logic 8.95
TIL 308 Number w/Logic 8.95
TIL 309 Number w/Logic 8.95
TIL 311 Hexadecimal 12.95
MAN 2A .320 Red Alpha Numer 5.95
MAN 10A .270 Red Alpha Numer 5.95

**★ LED's ★ OPTOISOLATERS ★**

LEDS Red Yel w Green .185 5/1.00
MCT 2 Photo XSTR HFE 250 30V .99
4N25 Photo XSTR HFE 250 30V 1.29
4N33 Photo Darlington 1.75
FPT 110B Photo XSTR Full Cone **SALE** 4/1.00

### ATTENTION KIM USERS

KIMSAI expansion to S-100 125.00 Kit 165.00
KIMSI to KIM Connector 5.75 pair
KIM 1 6502 Single Board Computer 179.00
KIM 1 Power Supply 59.95
KIM Memory Plus (consists of 8K Ram 8K2716 Eprom Programmer I/O etc.) 245.00

KIM SOFTWARE
- Please package (cassette) 12 games 16.95
- Help Editor package (cassette) 16.95
- Help Mailing List pkg (cassette) 16.95
- Help Info Retrival pkg (cassette) 16.95
- Microchess (cassette) 16.95
- Microaid Assembly/Disassem/Editor 27.95
- Microaid Source Listing (cassette) 27.95
- Tiny Basic for KIM (paper tape) 10.95

### COMPUTER SPECIALS

WOW!

| | LIST | SALE |
|---|---|---|
| Hi-Plot Plotter | 1085.00 | 899.00 |
| Hi-Plot Digitizer | 795.00 | 735.00 |
| Exidy w 32K | 1395.00 | 1195.00 |
| Apple w 16K | 1195.00 | 1045.00 |
| Compucolor w 16K | 1695.00 | 1595.00 |
| TEI PT208 (Naval) | 4995.00 | 3995.00 |
| Cromemco Sys III | 5990.00 | 5395.00 |
| Commodore Pet | 795.00 | 765.00 |
| Soroc IQ 120 | 995.00 | 895.00 |
| ADM3A Assem | 895.00 | 829.00 |
| Teletype 43 | 1349.00 | 1150.00 |
| Centronics P1 | 495.00 | 395.00 |
| Centronics S-1 | 595.00 | 525.00 |

### MONTHLY IC SPECIALS

LF13508 JFET Anlog Multi 8 bit 6.95
ICM7208 Seven Decade Counter 17.95
ICM7207 Oscillator Controller 6.95
ICM7045 Precision Sto Watch Timer 22.95
ICL7107 3½ Digit A D (LED) 14.95
ICL8211 Voltage Reference 1.95
LM390 Battery OP Audio Amp 3/1.00
LM1830 Fluid Detector 3/1.00
LM1850 Ground Fault IC 3/1.00
LM1800 Phase Lock Loop FM Stereo 3/1.00
LM1820 AM Radio 3/1.00
DS3625 Dual Mos Sense Amp 2.50

| | | | |
|---|---|---|---|
| 1044 5 | 4/10.95 | 10147 ECL Ram. | 9.95 |
| 1488 1489 | 2/1.99 | NE561 | 5.00 |
| 22 Pin S/T Socket | 10/1.00 | LF356HBiFet | 3/1.99 |
| 8223 Prom | 2.95 | MCM14505 | 8.95 |
| 6331 1 Prom | 1.95 | 74S89 | 3/1.99 |
| MK5014 Cal | 4/1.90 | 74107N | 6/1.99 |
| 74141N | 3/1.99 | 75452N | 8/1.99 |
| LM2917 | [illegible] | 741N-14 | 10/1.99 |
| 8T26 8T28 | [illegible] | 555CN | 5/1.99 |
| 95H90 | 9.95 | 556CN | 3/1.99 |

### ATTENTION APPLE II USERS

Apple II w 16K $1195.00
16K Upgrade Kit 89.95
Floppy Disk II w Interface 595.00
Floppy Disk II 495.00
Firmware Card 180.00

### SOCKETS

[illegible]

### TEXTOOL ZERO INSERTION FORCE SOCKETS

16 Pin $ 5.50 — 24 Pin $7.50
40 Pin $10.25

### CONNECTORS

DB25P (RS232) 3.25
DB25S Female 3.75
Hood 1.25
Set w Hood Sale $6.50
22 44 W W S/T KIM 2.45
43 86 W W S/T MOT 6.50
50 100 S 100 Connector w w 4.25
50 100 S 100 Connector [illegible]

### CTS DIPSWITCHES

CTS206-4 $1.75 — CTS206-8 $1.95
CTS206-5 $1.75 — CTS206-9 $1.95
CTS206-6 $1.75 — CTS206-10 $1.95
CTS206-7 $1.75

### LIVERMORE BASIC

ON 6SK ROM
THAT'S RIGHT ON ROM!!
OUR PRICE only $95.00

### NAKED PC BOARD SALE

Z-80 CPU (Ithaca) $34.95
8080A CPU 34.95
8K Static RAM (Logos) 21.95
16K Static RAM (2114) 29.95
32K Static RAM (2114) 49.95
Floppy I/O (Tarbel) 39.95
Cassette I/O (Tarbell) 29.95
8K Eprom (2708) 21.95
1702 Eprom Board 30.00
2708 2716 Eprom (Ithaca) 34.95
2708 2716 Eprom (WMC) 30.00
Real time Clock 34.95
ACP Proto Bd (3M Conn) 27.95
vector 8800 Proto 19.95
Vector 8800 11 slot MB 24.95
ACP Extender w/Conn 15.95
Video Interface (SSM) 27.95
Parallel Interface (SSM) 27.95
13 Slot MotherBoard (WMC) 32.95
9 Slot MotherBoard (WMC) 29.95
8 Slot Mother (expandable) 14.95

### WAVEFORM GENERATORS

8038 Function Gen 3.95
MC4024 VCO 2.45
LM566 VCO 1.75
XR2206 Function Generator 5.25

### FLOPPY DISK I/O

1771 01 5 & Mini floppy [illegible]
uPD372 Nec Floppy [illegible]
1781 Dual Floppy [illegible]
1791 Dual Floppy 44.95

### TV INTERFACES

Pixie Verter 8.50
TV 1 Video Interface 8.95
Microverter 35.00
M&R Modulator 35.00

### ATTENTION PET USERS

BETSI pet expansion to S100 105.00 Kit-160.00
PET Connector Kit Includes (4) Connectors for memory expansion IEEE 488 I/O cassette I/O and parallel user port 7.95/set
Video Buffer (converts to Std. Video) 29.95
Petunia (Music Board) 29.95
Combo (Video & Petunia) 49.95
Beeper (signals tape load) 24.95

### ATTN TRS 80 USERS

20 40 Pin Memory Exp Conn $7.95
16K Memory Add on w Inst 89.95
Vista V80 Minifloppy 395.00
40 Track DOS Patch on Disk 10.95
4 Drive Cable for V80 39.95
Centronics 779 w tractor 1179.00
Centronics 701 Bidirect 1579.00
ANADEX DP8000 995.00
Micro P same as Quick Printer 395.00
Cable to plug in Expansion Interface 39.00
Power Strip 6 outlets 17.95
Surge Suppressor Filter 24.95
Microsoft Fortran on Disk 325.00
Electric Pencil on Cassette 99.00
Electric Pencil on Disk 150.00
CP M for TRS on Disk 150.00
Library 100 (cassettes) 49.95
Micro Chess (cassette) 19.95

### TV CHIPS

AY38550 1 8 Games Color w/ [illegible]
AY38500 1 6 Games B/W [illegible]
AY38515 [illegible]
AY38603 1 Roadrace Game [illegible]
AY38F05 1 Warfare Game [illegible]
AY38606 1 Wipeout Game [illegible]
AY38607 1 Shooting Gallery [illegible]
AY38610 [illegible]
AY38615 [illegible]
AY38910 [illegible]
SN76477 [illegible]
MM57100 [illegible] 9.95
MM5369 [illegible]
LM1889 [illegible]
MM5[illegible]
MM5[illegible]
RF Mod [illegible]

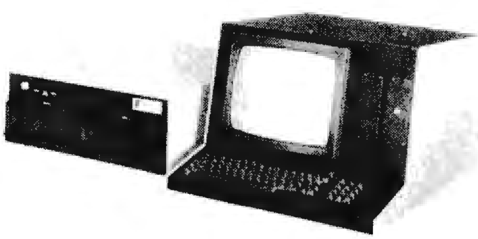
All Shipments FCM or UPS. Orders under $100.00 add 5% handling and postage. Orders over $100.00 add 2.5% handling & postage. Mastercharge Bank americard/COD accepted w/25% deposit. California Residents add 6% tax. Foreign Orders add 8% handling. All parts prime factory tested guaranteed. Same day **shipment. Add .35 cents for Data.**
Retail pricing may vary from Mail Order Pricing. All pricing subject to change without notice.

P. O. BOX 17329 Irvine, California 92713 Phone (714) 558-8813 TWX: 910-595-1565

Retail Store Open Mon. – Sat. Located at 1310 "B" E. Edinger, Santa Ana, CA 92705

Circle 4 on inquiry card.

### Key-to-Disk Software Available for Microcomputers



A new key-to-disk software (P1-KTDS), designed to run on 8080 and Z-80 microcomputers, has been announced by Phone 1, POB 1522, Rockford IL 61110. The software supports four video terminals, four floppy disk drives, line printer, and 3780 communications. The P1-KTDS package allows up to four video users to define and select as many as four screen formats per user. Each format may contain as many as 40 user specified fields. Constant data fields may also be specified.

In addition to data entry, data verification is also included as a feature of the system. Verification is done on each field specified as a verify field whenever the verify option is enabled. Each of the four users has a separate disk drive that stores the formats and data records for the particular video assigned to the drive.

The P1-KTDS software is currently shipping on Phone 1's P1-5 Data Concentrator product which utilizes the 8080 microcomputer, 8214 and 8259 interrrupt controllers, 8251 terminal and printer controllers. Mylar decals, which attach to the front edge of the video keytops, guide the user in efficiently entering data.

P1-KTDS source module on CP/M or FDOS III compatible disk, limited use license, users manual, and four sets of keyboard decals are priced at $2,500. The users manual is available separately for $35.

Circle 629 on inquiry card.

### Business Software Packages

A line of 21 fully integrated and auto-chaining business software packages is available from Univair International, 10327 Lambert International Airport, St Louis MO 63145. Some of the major programs include General Ledger, Accounts Payable, Accounts Receivable, Payroll, Inventory, Dental Management System, Medical Management System, Real Estate Multi-List, Insurance Agency, Credit Union, Data Base Management, and Word Processing. All programs are run under CP/M or IMDOS with commercial BASIC and 31 K bytes of programmable memory. A system of automatic chaining, posting, file backups, and updates is incorporated.

The cost of each program on an eight inch soft-sectored floppy disk is $395. Complete source code and operators manual are provided. Programs are also available on five inch Northstar or five inch Micropolis disks.

Circle 630 on inquiry card.

### Software for the Apple II Computer

Softape, 10756 Vanowen St, North Hollywood CA 91605 has an extensive selection of software available for the Apple II computer. One such program, Appletalker accepts voice or audio information through the cassette input port, digitizes the information, and stores it in numbered tables in the computer's memory. The stored information can then be played back using the Apple's on board speaker.

Apple-Lis'ner allows the user to communicate with the Apple II computer via spoken words. By using a cassette recorder and microphone, Apple Lis'ner will listen for the words or phrases it has learned and respond under program control.

For more information on these and other Apple II programs, write to Softape at the above address.

Circle 631 on inquiry card.

### 8080 Simulator for the 6502

Now available in a KIM-1 version, the 8080 Simulator for the 6502 processor executes the entire 8080 instruction set. All internal 8080 registers are maintained, ready for convenient examination or modification of their contents. In its minimum configuration on the KIM-1, the 8080 Simulator supports register single step, program counter single step and run modes. It also offers an input and an output port, breakpoint operation, and rejection of illegal op-codes.

The 8080 Simulator runs in less than 1 K of memory, leaving up to 224 bytes of 8080 programming space on an unexpanded KIM-1. The Simulator may be relocated in read only memory and can be adapted to other 6502 based systems.

Well suited to all but time sensitive applications, the 8080 Simulator may be used to assist in the design and testing of 8080 software, used as a training aid or used for running most 8080 application software. The package consists of a KIM-1 format cassette tape, a user manual and a complete, commented assembly level source and object listing. The price is $18 plus $1.50 for postage and handling. For further information, contact Dann McCreary, 4758 Mansfield St #2B, San Diego CA 92116.

Circle 633 on inquiry card.

### Software for the Micropolis Floppy Disk System

The Basically Speaking Co has announced the availability of software for the Micropolis five inch floppy disk systems. Statpak includes the ability to create a data file and to do multiple statistical analyses on a data base. Available statistical functions include Chi Square, analysis of variance and linear regression.

Gradebook allows school teachers to use their computer as a gradebook. Multiple classes are allowed, as well as missing assignments, excused absences, and addition and deletion of student records. A grade figuring program called Reportcard is included.

The Personal Accounts Ledger (PAL) program keeps not only a checkbook ledger, but a savings ledger, a small business ledger, and an investment ledger, and allows these ledgers and their accounts to be fully integrated. Selected account printing is also supported.

The price is $45 for a Micropolis MDOS (Model I or II) or Micropolis CP/M disk and documentation. The documentation includes full instructions for use, suggestions for modification, and complete listings. For further information contact Basically Speaking, 719 Anna Lee Ln, Bloomington IN 47101.

Circle 632 on inquiry card.

### Word Processing System for Z-80 Based Computers with North Star Disk

The Wordsmith is a word processor for Z-80 based computers with North Star floppy disk systems, an RS-232 terminal and a Diablo 1620 or equivalent printer. It provides complete cursor control, block movements, string searches and alterations, insertion and deletion of text, and other editing functions through the use of control commands.

Print formatting commands are entered along with the text and allow the format to be changed while the printing is taking place. The format commands include right justification, setting of margins, automatic paging and headers, four types of paragraphs, insertion of variable data into the text, and operator instructions. Form letters are easily produced, each one personalized for the recipient, through the use of simple text commands. Disk file creation, deletion and updating are handled automatically.

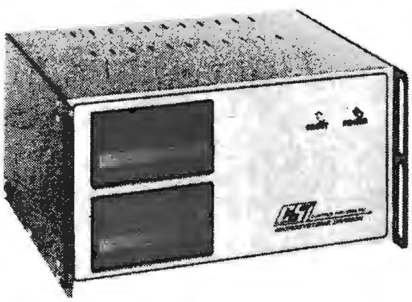
The Wordsmith is priced at $299. For further information contact Southwest Micro-Systems, POB 20088, Riverside CA 92516.

Circle 634 on inquiry card.

### Home Computer From Bally

The Bally Computer System contains the built-in "Bally Brain," a microprocessor that has a 12 K byte memory. Each optional Bally Videocade cassette adds up to 8 K additional bytes. Also built into the system are a five function, ten memory register calculator, 256 color display variations, stop-action pause control, and automatic shut-off. The unit is available in two player and four player models and has a 16,000 dot picture image. With three arcade games and a 24 key calculator built in, the two player model has a retail price of $299.95 and the four player model is $329.95. Both an optional Bally BASIC programming cassette and an audio cassette interface are priced at $49.95. For further information contact Bally Consumer Products Div, 10750 W Grand Av, Franklin Park IL 60131.

Circle 639 on inquiry card.



### Microcomputer Offers Pascal in Programmable Read Only Memory

This new microcomputer designed for use with Pascal is being marketed by Control Systems Inc, Drawer EE, Williamsburg VA 23185. The UDS 470 offers Pascal in programmable read only memory as an alternative to assembly language and BASIC for low and medium volume applications where power and fast development are important. They make available a version of UCSD Pascal specifically designed for read only memory and programmable read only memory operation for use in dedicated applications when the development cycle would be slow with assembly language. The Pascal in programmable read only memory feature makes high level programming as easy as assembly language programming. A Pascal program is compiled (instead of assembled) and compiler output (P-code) is burned into programmable read only memory and erasable read only memory.

The UDS 470 is a rack mountable system designed for industrial environments (high temperature, vibration, etc). It currently uses the 6800 microprocessor, but can be upgraded to the 6809 or 68000 when they become available. The UCSD system was designed to be machine independent. UCSD's 2.0 version is currently being supplied, but the 3.0 version will be used when UCSD releases it. The standard UDS 470 package includes a processor with 1 K bytes of programmable memory and 2 K bytes of erasable read only memory; serial I/O (input/output) port with automatic reset and VCC monitor; 48 K bytes programmable memory; 16 K bytes erasable read only memory; 5 inch double density floppy disks with interface; 5 V power supply; and a case. The approximate price of the standard UDS 470 is $4000.

Circle 640 on inquiry card.

### Where Do New Products 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 information might be of interest to the personal computing experimenters and homebrewers who read BYTE, we print it in some form. We openly solicit releases and photos 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. While we would not knowingly print untrue or inaccurate data, or data from unreliable companies, our capacity to evaluate the products and companies appearing in the "What's New?" feature is necessarily limited. We therefore cannot be responsible for product quality or company performance.*

### North Star Word Processor With Extras

IDSWORD is a comprehensive word processing package available in North Star BASIC Version 6, under North Star disk operating system release 4.0. Prompts are given as complete English sentences, and responses are accepted as words, rather than numbers. Imbedded commands are not required since IDSWORD formats the text interactively. Some of the features are insertion, deletion, and block moves of text; global searches; complete text editing on video terminal or printer; control of margin size and justification; merging of up to ten files; sorting and printing of mailing labels.

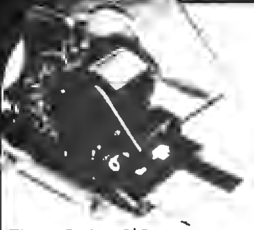
Block editing capability on the video terminal is available by a linked list of all lines of text in memory, coupled with full cursor control over the entire text display. The amount of text in memory is automatically adjusted to the available memory. About 25 pages of text may be stored on a single density disk. Longer documents may be developed and printed in segments.

IDSWORD will run with one or two disk units. It is presently configured to run with the Soroc, Lear Siegler ADM-3a, Hazeltine, and Intertube terminals, and the IBM Selectric, Qume, Diablo, and Spinwriter printers. IDSWORD is a modular system starting at $125 for the basic configuration. The complete word processor is priced at $245 for the video screen edit capability, or $220 for the editing on the printer capability. The form letter, labels, name and address file maintenance, and sort modules are $50. For further information, contact CW Applications, 1776 E Jefferson St, Rockville MD 20852.

Circle 641 on inquiry card.

# California Digital

**Post Office Box 3097 B • Torrance, California 90503**

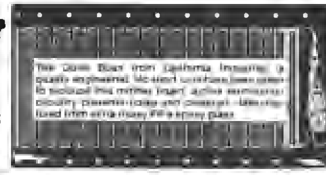


## Sankyo Magnetic Card Reader

**$59**

These Sankyo I/O units are capable of storing and retrieving over 400 characters of data in under two seconds.
The flexability of this device lends itself to numerous applications. As an input reader to a computerized security system, the computer has the ability of identifying the card holder and admitting only those individuals who are authorized to enter the premises during specified time frames. The device is also suitable for maintaining customer information files, or any other application where small amounts of information must be quickly entered into a data processing system.
Accepts 2" by 4" HP style mag-cards. (Similar to bank cards.)
Motorized reader pulls the magnetic card across one four channel read/write head. NEW surplus, original cost $300, full documentation

## S-100 Mother Board



*Quiet Buss*

$29.95
8803-18
18 slot
IMSAI

## TELETYPE MODEL 43

Even if we have to give them away, we're going to ship more 43's in 1979 than the aggregate of all our competitors.

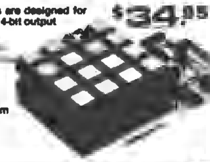
Model 43AAA TTL)

| EACH | 3 | 10 | 25 |
|---|---|---|---|
| $925. | 875. | 850. | 825. |

plus shipping

RS-232 Interface "K" Add $75.00

## HEXADECIMAL KEYBOARD



$34.95

Maxi-Switch hexadecimal keyboards are designed for microcomputer systems that require 4-bit output in standard hex code.

Reliable low friction acetal resin plungers are credited for the smooth operation and long life of this premium keyboard.

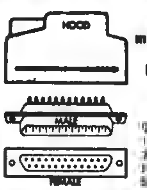
Requires single +5 volt supply.

## $29.95 DISKETTES

BOX of 10

Verbatim APPLE/TRS-80 Mini - Soft sector

## CONNECTORS



your choice
DB25P male plug & hood or DB25S female

$3.95

## Edge Connectors

GOLD 100 PIN IMSAI/ALTAIR

| | | |
|---|---|---|
| Imsai solder .125x.250 | $3.95 | 3/$ 9.00 |
| Imsai w/w .125centers | $4.95 | 3/$13.00 |
| Altair soldertail .140 row | $5.95 | 3/$15.00 |
| SPECIALS | | |
| 22/44 Kim eyelet .156" | $1.95 | 3/$5.00 |
| 25/50 solder tab .156" | $1.09 | 3/$2.00 |
| 36/72 wide post w/w.156 | $1.95 | 3/$5.00 |

$24.88

## SPERRY UNIVAC KEYBOARD

The famous Sperry Univac 1710 Hollerith keyboard assembly is now available from California Industrial for only $24.88. The ideal computer input device for accountants and mathematicians. The numeric keys are placed on the lower three rows to resemble a ten key adding machine. This format allows one handed numeric data entry. Original cost was $380. Used but guaranteed in excellent condition. Complete with documentation.

Apple Owners: TEN KEY Data Entry Pad

$79.50

Plugs directly into you Apple II. Allows you to enter numerics, punctuation and upper case alpha characters, all from the data entry pad. Sold assembled in walnut finished enclosure.

## SPECIAL

## APPLE II 16K MEMORY

COLOR • GRAPHICS • SOUND

**$1024** Mfg.Sug. Retail.... $1195

PLUS SHIPPING

TEN for $41 50+ $3.85

**Scotch** BRAND Certified Digital CASSETTES Won't drop a BIT!

DISKETTES $5.50

8 inch Soft (IBM)
8 inch 32 sector
Mini Soft sec.
Mini 10 sector
Mini 16 sector

CALIFORNIA INDUSTRIAL is an authorized Dealer of Scotch Brand Data Products

Shugart Associates

## SA800-R Floppy Disk Drive

The most cost effective way to store data processing information, when random recall is a prime factor. The SA800 is fully compatible with the IBM 3740 format. Write protect circuitry, low maintenance & Shugart quality.

**$449.50**

## CALIFORNIA DIGITAL 16 BiT 8086 S-100 CPU Board

Directly addresses one megabyte. 8 bit unidirectional & 16 bit bi-directional. 4K of static memory is supplied on board. $550.00

## DigiCast A/V-100 R.F. MODULATOR

$29.95 Broadcast both audio and vidio on your existing color television. Recommended for the Apple II.

## MEMORY

TRS-80 APPLE II $65

16k memory (8) 4116's

• As you may be aware, publishers require advertisers to submit their ad copy 60 to 90 days prior to "press" date. That much lead time in a volatile market place such as memory circuits, makes it extremely difficult to project future cost and availability.
To obtain the best pricing on memory we have made volume commitments to our suppliers, which in turn affords us the opportunity to sell these circuits at the most competitive prices. Please contact us if you if you have a demand for volume state of the art memory products.

| STATIC | 1-31 | 32-99 | 100-5C | -999 | 1K+ |
|---|---|---|---|---|---|
| 21L02 450nS. | 1.49 | 1.19 | 1.05 | .95 | .89 |
| 21L02 250nS. | 1.69 | 1.49 | 1.45 | * | * |
| 2114 1Kx4 450 | 6.95 | 6.50 | 6.25 | 6.00 | 5.75 |
| 2114 1Kx4 300 | 8.95 | 8.50 | 8.00 | * | * |
| 4044 4Kx1 450 | 5.95 | 5.50 | 5.00 | * | * |
| 4044 4Kx1 250 | 9.95 | 9.50 | 9.00 | * | * |
| 4045 1Kx4 450 | 8.95 | 8.50 | 8.00 | * | * |
| 4045 1Kx4 250 | 9.95 | 9.50 | 9.00 | * | * |
| 5257 low pow. | 7.95 | 7.50 | 7.05 | 6.75 | 6.45 |

### SPECIAL CIRCUITS

| | | | |
|---|---|---|---|
| Z80A 4 MHz. | 24.95 | AY5-1013A UART | 4.95 |
| 8080A CPU | 9.95 | Floppy Disc Controllers | |
| 8085 | 22.50 | WD 1771 single D. | 39.95 |
| 8086 Intel 16 bits | * | WD 1781 Double D | 65.00 |
| TMS9900 16 bits | 49.95 | WD 1791 D/D 3740 | * |

| EPROMS | | 1-15 | 16-63 | 64+ |
|---|---|---|---|---|
| 1702A | 2K | 4.95 | 4.50 | 4.00 |
| 2708 | 8K | 9.95 | 9.50 | 9.00 |
| 2716 5v | 16K | 49.95 | 45.00 | 42.50 |
| 2532 | 32K | * | * | * |

**$139.50**

## PORTABLE DATA ENTRY SYSTEM

These used data terminals were originally designed for chain store inventory control and order entry systems. The operator enters the inventory control number, merchandise on hand and the unit price. After all pertinent data has been entered into the recorder, the main warehouse is telephoned, the handset is placed in the acoustic coupler and all the recorded information is transmitted back to the master computer. With a little imagination and one of these portable entry systems, you should be able to exchange programs and computer information with associates across the country. All units were removed from service in working condition. Original cost $2,500.
Each system come complete with:

- Portable Cassette Drive Unit
- Removable Entry Keyboard with LED Display
- Five Gould "D" NiCads
- Acoustical Coupler
- Battery Charger
- DB25 Cable
- Shoulder starp
- Full Documentation

## MINIATURE SWITCHES

your choice

| | 10 | 50 | 100 | 1k |
|---|---|---|---|---|
| $.98 | $.88 | .81 | .73 | .66 |

SPDT Miniature Toggles

## DIP Switch

SPST

## SYSTEM X-10

It's not offen that California Digital ventures into the distribution of consumer products, but we have resently come accross a product that appears so unique that we just had to add it to our product line. This is the System X-10 manufactured by the BSR turntable company. This space age system will remotely control any light or appliance in your home or office. Command signals are transmitted from the command console over your existing wiring. From your bed or easy chair you can control up to 16 different electrical devices inside and outside your home. Use the System X-10 to control your stereo, television or any light fixture on the premises.
The basic sampler package comes complete with command console, battery operated ultrasonic controller, one each of the appliance module, lamp module and wall switch. The basic package is priced at only $99.50. Additional modules are available for $13.95 each.

## DISCOUNT Wire Wrap Center

### IC SOCKETS

| pin | wire wrap ea. | 25 | 50 | low profile ea. | 25 | 50 |
|---|---|---|---|---|---|---|
| 8 | | | | 17¢ | 16 | 15 |
| 14 | 37¢ | 36 | 35 | 18 | 17 | 16 |
| 16 | 38 | 37 | 36 | 19 | 18 | 17 |
| 24 | 99 | 93 | 85 | 36 | 35 | 34 |
| 40 | 169 | 155 | 139 | 63 | 60 | 58 |

50ft. $.98 **KYNAR** WIRE WRAP

| 500 | 1,000 | 11,000 |
|---|---|---|
| $9. | $15. | $105. |

$29.95 BW 630

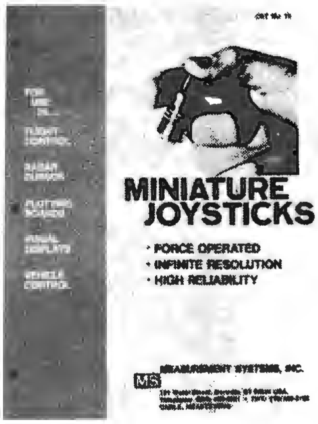
HOBBY WRAP-30 wrap & strip tool $5.45

**(213) 679-9001**

All merchandise sold by California Digital is premium grade. Sorry, no COD's. Orders are shipped the same day received. California residents please add 6% sales tax. Foreign orders add 10%. Orders over $25, when accompanied by payment, are shipped at our expense. Otherwise, please add $2.

VISA master charge

Circle 39 on inquiry card.




### Miniature Joystick Catalog

Measurement Systems Inc announces publication of their new 16 page catalog, *Miniature Joysticks.* These joysticks are used for cursor positioning of many displays including interactive terminals, computer aided drafting, and radar systems. They are also used for mechanism positioning such as microcircuit production equipment and vehicle control. These joysticks are offered for control of one, two, or three axes. They are offered in commercial, industrial and military grades. For further information contact Measurement Systems Inc, 121 Water St, Norwalk CT 06854.

Circle 642 on inquiry card.


### Buyers Guide Offered Free of Charge

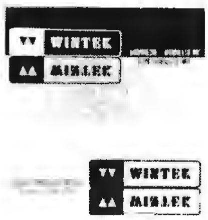
This buyers guide of microcomputer software, accessories, and supplies is available from Wallace Electronics Inc, 4921 N Sheridan Rd, Peoria IL 61614. Software and accessories for the Apple II and TRS-80, as well as a wide range of computer supplies, are listed on these sheets. The guide is updated weekly. The buyers guide is free of charge, although .50 should be included to cover postage and handling.

Circle 646 on inquiry card.

### EPROM Bulletin Shows Erase Times

An in-depth technical bulletin showing erase times and energy characteristics for various popular erasable read only memories is available at no charge from Adco Electronics, 2182 DuPont, Suite 222, Irvine CA 92715. It contains two charts which make it easier for the user to be sure of proper erasure of ultraviolet (UV) erasable programmable read only memories. The brochure, *Ultraviolet Erasing of EPROMS* (# A78286), has been prepared by the engineering staff of Spectronics Corp to take the guess work out of making sure that various programmable read only memories are empty of data. By quickly determining the specific energy level needed and required exposure time, the user can select the correct UV source.

Circle 643 on inquiry card.




### Wintek Corp Offers New Catalog

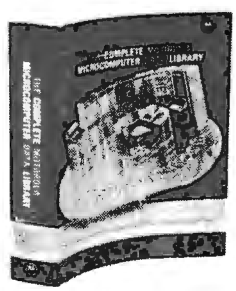
This catalog contains a 6800 based single board computer plus 15 support and interface modules on industry standard 4½ by 6½ inch cards for process control and data acquisition. Additionally, 6800 development systems, resident and cross assemblers, and compilers are listed. For further information contact Wintek Corp, 902 N 9th St, Lafayette IN 47904.

Circle 644 on inquiry card.

### The Complete Motorola Microcomputer Data Library

*The Complete Motorola Microcomputer Data Library* presents technical data for microcomputer design and implementation. It is divided into three basic segments, each further subdivided into subordinate product categories. The three segments are:

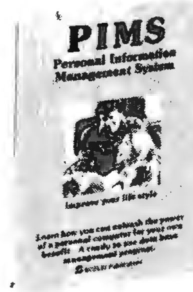
- microcomputer components—microprocessor and microcomputer unit components, together with interface and peripheral components to implement microcomputer systems,
- memory products—basic memory components and add-in and add-on memory subsystems for computer applications,
- microcomputer development systems and subsystems—support products (hardware and software) to design microcomputer systems; board-level subsystems for system implementation.



The organization within each of these basic segments is by device families and application groupings rather than in alphanumeric sequence. Therefore, a comprehensive table of contents provides the reader with a sequential listing of the chapter by chapter content of each segment.

The book is priced at $6. For further information write to Motorola Semiconductor Products Inc, POB 29024, Phoenix AZ 85036.

Circle 645 on inquiry card.




### Personal Information Management System

*Personal Information Management System* describes a data base management program designed for personal use on such small computer systems as the Radio Shack TRS-80 Level II, or other systems using a Microsoft compatible BASIC language. Along with complete source listings of the program and comprehensive operating instructions, this 88 page book discusses the microcomputer and its potential for personal use. Additionally, 15 comprehensive applications of the program are illustrated in detail. The publication is designed for the computer novice although the program may be beneficially used by anyone. The book is priced at $9.95. For further information contact Scelbi Publications, POB 133 PP STN, Milford CT 06460.

Circle 647 on inquiry card.

# Smart & Foxy

# Venus 2001 Video Board

**Assembled and Tested $259.95 • Complete Unit with 4K of Memory and Video Driver on Eprom assembled and tested $339.95**

## kit $199.95

*OPTIONAL:* • Sockets $10.00 • 2K Memory $30.00 • 4K Memory $60.00 • Video Driver Eprom $20.00 • Text Editor Eprom (Includes Video Driver $75.00)

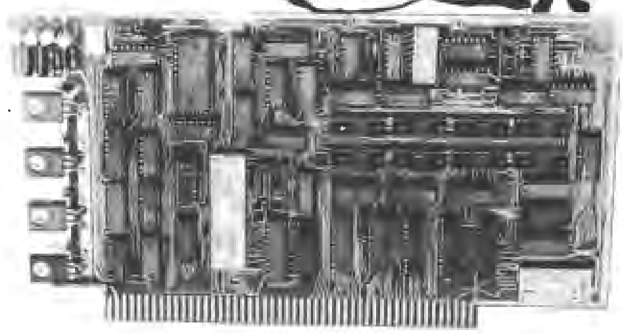
### *S-100 Plug-In • Parallel Keyboard Port*

On board 4K Screen Memory (Optional). On board Eprom (Optional) for Video Driver or Text Editor Software.

### *Up and Down Scrolling through Video Memory*

Reverse Video, Blinking Characters.

***Display:*** 128 ASC11 Characters 64 X 32 or 32 X 16 Screen format (Jumper Selectable). 7 by 11 Dot Matrix Characters.



***American or European TV Compatible (CRT Controls Programable)***

***Dealer Inquires Invited***

15 MHZ DUAL TRACE

## *Portable Scope*

BIG PRICE BREAKTHRU **$399.**

MODEL MS-215

• Battery or A.C. Operated • External and Internal Trigger • Time Base—.1 m Sec./Div. Into 21 Calibrated Ranges • 3% Accuracy • Input Impedence 1 M Ohms • Complete with Input Cables, Battery and Charger.

*OPTIONAL:* • Leather Case $45.00 • 10:1 Probe $27.00 (2 for $49.00)

### *MS-15 Single Trace Scope $299.*

## 32-K Static RAM $499. KIT

• S-100 Plug-In • Kit includes P.C. board, all parts and assembly manual • Uses 2114L, 450 nS.

I.C. sockets - $20.00

P.C. BOARD BY S-100 CO.

## 16-K Static RAM $249. KIT

• S-100 Plug-In Kit includes P.C. board, all parts and assembly manual. Uses 2114L 450 nS.

Sockets - $10.00

Add $40.00 for 300 nS (4MHz) RAMS

P.C. BOARD BY WAMECO

## Z-80 CPU $125. KIT

• S-100 Plug-In Kit includes P.C. boards, all parts and assembly manual.

**FEATURES:** 2MHz operation • S-100 plug-in • Power-on jump • On board provision for 2708 (optional at $12.95).

P.C. BOARD BY ITHACA AUDIO

## *ASCII Keyboard Kit $77.*

Assembled and Tested $93.00

• Single +5V Supply • Full ASCII Set (Upper and Lower Case) • Parallel Output • Positive and Negetave Strobe • 2 Key Rollover • 3 User Definable Keys • P.C. Board Size: 17-3/16" X 5" • Control Characters Molded on Key Caps • Optional Provision For Serial Output

*OPTIONAL:* Metal Enclosure $27.50 • Edge Con. $2.00 • Sockets $4.00 • Upper Case Lock Switch $2.50 • Shift Register (For Serial Output) $2.00

*Dealer Inquiries Invited*

## *Apple II I/O Board Kit*

Plugs into Slot of Mother Board

•1 8 Bit Parallel Output Port (Expands to 3 Ports) •1 Input Port • 15mA Output Current Sink or Source • Can be used for peripheral equipment such as printers, floppy discs, cassettes, paper tapes, etc. • 1 free software listing for SWTP PR40 or IBM selectric.

*PRICE:* 1 Input and 1 Output Port $49.00
1 Input and 3 Output Ports $64.00

*Dealer Inquiries Invited*

SHIPPING $3.50 / California residents add 6% sales tax

## ELECTRONICS WAREHOUSE Inc.

NEW ADDRESS

15820 Hawthorne Boulevard
Lawndale, CA 90260
(213) 370-5551

VISA

Circle 130 on inquiry card.

### Visible Computer Supply Offers Free General Catalog


Visible Computer Supply Corp, supplier of data processing supplies and accessories, is offering their 116 page 1979 illustrated catalog featuring more than 2800 products. Their product line covers binders and accessories for printout storage, systems and programming aids, a complete line of magnetic media and related handling and storage systems, minicomputer accessories, video terminal stands, keypunch furniture and accessories, word processing supplies, microform retention and retrieval systems, and pressure sensitive labels. For further information contact Visible Computer Supply Corp, 3626 Stern Dr, St Charles IL 60174.

Circle 648 on inquiry card.

### Apple Software Directory

Over 700 software programs for the Apple computer have been compiled into the *Apple Software Directory*. All programs are listed alphabetically so that the same type of program produced by several sources can be compared. Listings include description, memory requirements, price, format and the source. All sources are listed with addresses.

The directory is printed in two volumes. Volume 1 covers business and utility programs. Volume 2 covers games and entertainment programs. Each is priced at $4.95. For more information, write WIDL Video, 5325 N Lincoln, Chicago IL 60625.

Circle 649 on inquiry card.

### Publication Lists 32 BASIC Programs for the PET

*32 BASIC Programs for the PET Computer* by Tom Rugg and Phil Feldman is precisely that...32 fully documented programs that are ready to run on an 8 K byte Commodore PET 2001 computer. The reader does have the option of making changes to these programs. This 267 page book covers application, educational, game, graphic display, mathematical and miscellaneous programs. The book is priced at $15.95. For further information, contact Dilithium Press, POB 92, Forest Grove OR 97116.

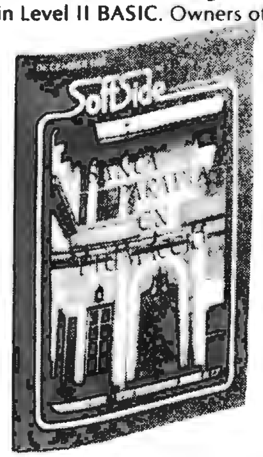
Circle 650 on inquiry card.

### Software Magazine Devoted to Radio Shack TRS-80

Owners of Level II Radio Shack computers will appreciate *SoftSide*, a magazine devoted to providing games and light application software in Level II BASIC. Owners of other personal computers using Microsoft BASIC will also find programs that can be readily converted for their systems.

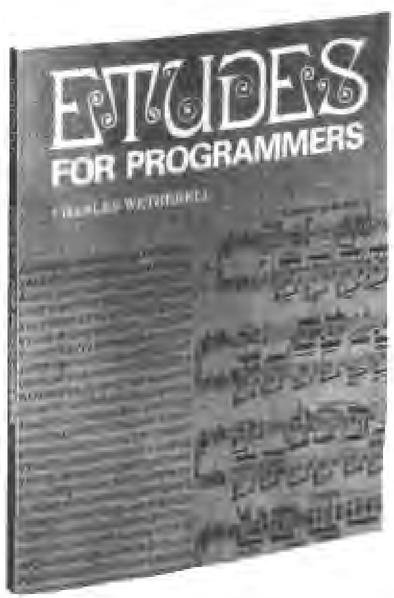
The particular emphasis of the magazine is simulation games. Readers of recent issues have been able to play football, race a clipper ship around Cape Horn, rule a fifteenth century Italian city-state or chase wild animals on a photographic safari. Light application programs are also published, and have included an income tax program and a personal finance program complete with graphic pictures of checks on the screen. Hints for TRS-80 programmers regularly appear in various places throughout the magazine.

*Softside* is published monthly and is available by subscription for an annual rate of $15. A special cassette edition which includes the magazine and all the monthly programs in machine readable form is available for $38 for a six month subscription. For further information contact SoftSide Publications, POB 68, Milford NH 03055.




Circle 651 on inquiry card.

### New Book Features Self-Contained Programming Problems




Circle 652 on inquiry card.

*Etudes for Programmers* by Charles Wetherell is a collection of large scale problems for learning by doing. Each problem includes a real world background discussion of appropriate programming techniques, detailed requirements for correct solution, extensions, and annotated bibliography. Two of the problems are completely solved by the author. The solutions concentrate on good programming techniques, measuring the quality of the program and the output, and possible extensions of the problem. They are models of what solutions to any programming job should be, and they contain many practical hints about writing good programs.

Additionally, this 200 page book offers references to sources for programming information and further reading about problem subjects. It includes a complete set of four projects for a programming language course: macro-interpreter, compiler, relocating loader and computer simulator.

*Etudes for Programmers* is priced at $12.95. For further information, contact Prentice-Hall Inc, Englewood Cliffs NJ 07632.

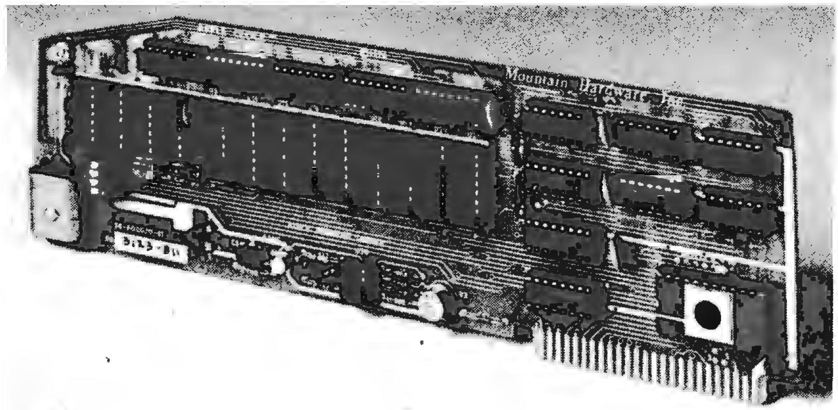

TO ORDER CALL TOLL FREE 800-223-7318
The COMPUTER FACTORY
COMMODORE "PET"
only $795
USED $559
• 14K ROM Operating system
• 8K RAM Memory
• 9" Video Monitor
• Built in Keyboard
• Digitally controlled tape
Today's best value in personal computers, along with the latest peripherals are always in stock at the COMPUTER FACTORY
NEW! PET BUSINESS SYSTEM
The PET is now a truly sophisticated Business System with the Floppy Disk and Printer which makes an ideal cost efficient business system for most professional and specialized fields: medicine, law, research, engineering, education, etc.
PET 2001 -
16K $995
32K $1295
• 16 or 32K Bytes Dynamic RAM • 14K Operating System
• Upper/Lower Case and Graphics
• Full Sized Business Keyboard
• Full Screen Editing
• Operating system will support multiple Languages (BASIC resident)
• Machine Language Monitor
• 8K ROM Expansion Sockets
• 9" CRT
PERIPHERALS FOR PET
• 24K Memory Expansion $499
• 16K Memory Expansion 399
• PET to RS232 Serial 169
• 2 Way Serial Communication 229
• Modem Board for PET 375
• Analog to Digital Board For 16 Devices 275
• Second Cassette Drive 95
• Parallel Printer Interface 199
PET PRINTERS
2023 $849
2022 $995 (Includes Tractors)
• Bottom and Rear Tractor Feed
• 8½" Paper Width
• 6504 Microprocessor Controlled
• 4K ROM • ½K RAM
• Upper/Lower Case and Graphics
• 7 x 6 Dot Matrix
PET MUSIC BOX
Add music and sound effects to your programs. Compose, play, and hear music on your pet. Completely self-contained (no wiring). Free 3 programs including Star Wars theme, sound effects, etc. $39.
DUAL DRIVE FLOPPY DISK 2040 $1295
• 360 K Bytes Storage
• High Speed Data Transfer
• Plugs into IEEE Port
• 6504 Microprocessor
• 8K ROM Operating System
• 8K ROM Encoding and Decoding
• 4K RAM
• Uses Single or Dual sided Diskettes
SINGLE DRIVE FLOPPY UNIT 2041 $895
• Intelligent mini floppy
• 171.5K net user storage
SALE
Come In Or Call For Special Low Prices
WE'RE MOVING TO LARGER QUARTERS AND MUST SELL EVERYTHING IN OUR INVENTORY
FREE
$1495 Complete!
16K Model add $200
32K Model add $500
Compucolor II
COMPUCOLOR II Disk-Based Model 3
Advanced hardware and software technology gives you
• 13" Color Display
• Advanced Color Graphics
• 51K Disk Built-In
• 16K ROM Operating System
• 8K RAM User Memory
• 4K RAM Refresh
• 8080A Microcomputer
• RS-232 I/O
Every unit comes with an extended DISK-BASIC that has full file management capability resident in the COMPUCOLOR II in 16K of ROM. Color is fantastic, but COMPUCOLOR II has the power to handle complex tasks and small business applications. An impressive software library supplements
NEW! from Eventide
AUDIO SPECTRUM ANALYZER
• Mounts inside the PET
• Third-Octave audio spective analysis
• Complete with software and documentation
• Replaces equipment costing thousands of dollars
$595
MARK SENSE CARD READER $750
• Automatic turn-on and card feed
• Ideal for marking test scores
• Accepts any length card
• Perfect for schools & business
apple
APPLE II
APPLE is a completely contained computer system with BASIC in ROM, full ASC II keyboard in a lightweight molded carrying case. Expandable to 48K
APPLE II Disk $595
APPLESOFT in ROM $200
Interface Cards $179
MICROPRODUCTS
• EPROM Prgmr $ 99.95
• EPROM Socket Adpt $ 14.95
• APPLE II Modem $129.95
BUSINESS COMPUTER
IMSAI
The low cost solution for all small business problems. A wide variety of software is available for all your needs
PCS series include dual floppies, 32K RAM, I.O, DOS, BASIC
• PCS-42 (400KB) $3295
• PCS-44 (780KB) $3995
VDP-42 series adds video terminal, keyboard and VIO to above
• VDP-42
• VDP-44 $5595
• VDP 80
SUPER SOFTWARE
Word Processing For PET
This super advanced full function program will allow you to create text from PET or terminal keyboard. • INSERT • DELETE • CENTER • UNDERLINE • FULL SCREEN EDITING • MOVE LINES OR BLOCKS • SAVE TEXT ON TAPE • AUTO PRINT FEATURE....$45
Apple's Moving Average Plot Program
This fantastic program disk allows the statistician, mathematician, trader in stocks, money or commodities, the ability to maintain 30 database series of up to 300 values and plot 3 different moving averages of a series at the same time, in 3 different colors. Files can be updated, deleted, changed, extended, etc.
A sure value disk at only $40!
Word Processing For Apple on disk...$50
SORCERER
by EXIDY
• Z-80 Processor
• Keyboard and Numeric Pad
• 8K RAM (up to 32K)
• Serial and Parallel I/O
• Dual Cassette I/O
• 30 x 64 Display
• S 100 Compatible
• 64 User Defined Characters
• 240 x 512 Resolution Graphics
Modular design allows use of both cassettes for programs and ROM PAC cartridges for future language introductions like, APL, COBOL, PILOT, FORTRAN, etc. Add mor itor and tape or disk for complete computer system
$895
SPECIAL
12" Video Monitor for SORCERER ($299 value)
ONLY
$125 with 8K unit
95 with 16K unit
65 with 32K unit
ANDERSON JACOBSON
ANDERSON JACOBSON
841 I/O Terminal
Ideal for word processing and small businesses.
• ASC 11 Code
• 15 cps Printout
• High Quality Selective Printing
• Use Keyboard for PET
• Reliable heavy duty Mechanism
• Completely Refurbished by A.J.
• Service in 15 Major Cities
NOW IN STOCK!
Parallel $1095
Serial $1195
RADIO SHACK • PET • SORCERER • APPLE • COMPUCOLOR • ETC.
PRINTERS • PRINTERS • PRINTERS
The COMPUTER FACTORY'S extensive inventory and wide selection of computer printers assures you of finding the printer best suited for your needs and specifications. The following printers work well with all known personal computers
CENTRONICS
AXIOM (Parallel)
AXIOM (Serial)
TRENDCOM
INTEGRAL DATA
QUME or DIABLO
NEW CENTRONICS 730
50 CPS - MICROPROCESSOR CONTROLLED!
Tractor & Friction Feed • Uses Single Sheets, Roll, Fanfold • Upper & Lower Case • Light Weight
Parallel $995
Serial $1045
master charge
Min Credit Card Order $75
VISA
Open Mon.-Fri. 10-6 Sat. 10-4
N.Y. residents add 8% sales tax • Same day shipment on prepaid and credit card orders • Add $5 shipping for computers, $3 for boards, $.25 each cassette tape
SPECIAL SALE
Buy any computer and take 10% off price of printer of your choice
TO ORDER CALL TOLL FREE 800-223-7318
The COMPUTER FACTORY
485 Lexington Avenue 750 Third Avenue New York, N.Y. 10017 (46th St.)
(212) 687-5001 (212) PET-2001 Foreign order desk - Telex 640055

### Answer and Originate Acoustic Coupler

The AC-312 answer/originate acoustic coupler operates at 300 bps. The device is 103 Western Electric compatible and is switch selectable between originate and answer modes. When in the answer mode, this unit will generate the answer tone necessary to communicate with 300 bps originate only couplers and modems. The AC-312 includes a single plugable printed circuit board for ease of field upgrading to 1200 bps operation and field service. Standard light emitting diode diagnostic indicators are also featured. The AC-312 answer/originate modem is priced at $295. For further information, contact Digicom Data Products Inc, 1440 Koll Circle, Suite 108, San Jose CA 95112.

Circle 653 on inquiry card.

### Real Time Calendar and Clock for Apple II Computer



Mountain Hardware has announced the Real Time Calendar and Clock for Apple II computers. The Apple Clock keeps time and data in 1 ms increments continuously for over one year. Calendar, clock, and event timer functions are easily accessed from BASIC using routines contained in on board read only memory. Some of the features of the Apple Clock include crystal control, on board rechargeable battery to keep the clock running during computer down times; software for calendar and clock routines as well as an event timer contained in on board read only memory; and an interrupt feature which can be programmed to make efficient use of computer time. Sample applications include programming a morning printout of appointments; date of transactions; creating games in which elapsed time is important; and time events. The Apple Clock can be added to Mountain Hardware's Introl Remote Control System for real time control and monitoring of remote devices over regular AC wiring.

The price of the Apple Clock is $199 assembled and tested. For further information, contact Mountain Hardware Inc, 300 Harvey W Blvd, Santa Cruz CA 95060.

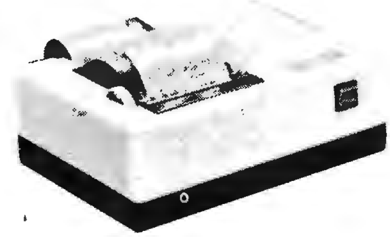
Circle 654 on inquiry card.

### Interface TRS-80 to Summagraphics' Bit Pad

Summagraphics Corp has announced the availability of an interface for the company's digitizer, the Bit Pad, which allows connection to the Radio Shack TRS-80 microcomputer. This new interface permits the entry and transfer of X,Y coordinate values for graphics and data entry applications from the Bit Pad to the TRS-80 computer.

The interface is priced at $175, and a cassette containing software is provided. Data is transferred from the Bit Pad in groups of five bytes. The interface is contained in a small separate box that connects to the Bit Pad and the TRS-80. The interface allows use of all other TRS-80 accessories. For further information, contact Summagraphics Corp, 35 Brentwood Ave, Fairfield CT 06430.

Circle 655 on inquiry card.



### Compact Low Cost Alphanumeric Printers

The DigiTec 6410 and 6420 small desktop printers print 20 columns of alphanumeric characters. Sixty-four different characters are produced in a 5 by 7 dot matrix. The printer can easily replace teletypewriter terminals in applications that don't need 80 column capability. An internal microprocessor makes these new printers reliable and easy to interface. The Model 6410 provides a serial interface to RS-232C and 20 mA current loop systems at 110 bps. The Model 6420 works with 8 bit parallel bus systems at up to 1000 characters per second. They both use the ASCII input format. The single unit price is $395. For further information and special OEM information, contact United Systems Corp, 918 Woodley Rd, Dayton OH 45403.

Circle 656 on inquiry card.

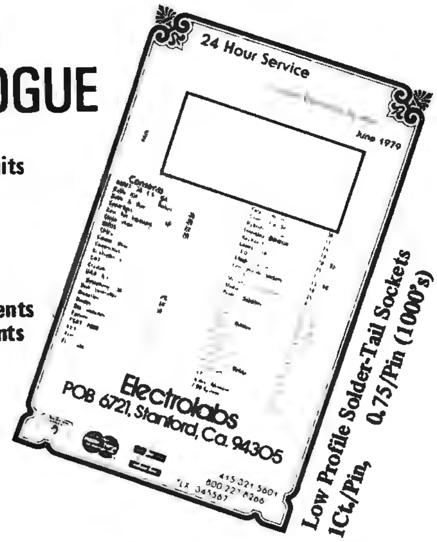
# Electrolabs

POB 6721, Stanford, Ca. 94305

*In California:* 415-321-5601
*Elsewhere:* 800-227-8266
TLX: 345567

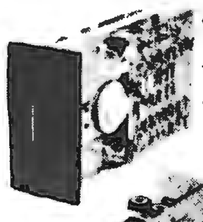
## NEW CATALOGUE

Crystals
Integrated circuits
Keyboards
Lasers
LSI-11
Media
RAMs
S-100 Components
Z-80 Components



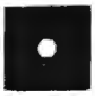

Low Profile Solder-Tail Sockets
1Ct./Pin, 0.75/Pin (1000's)

## FLOPPY SYSTEMS



**8" Siemens FDD120-8 Drive** All Siemen's options included in this drive which can be configured hard or soft and single or double density. (Others give only stripped unit) $399.00

**"Power One" Model CP206 Floppy Power Unit.** For two drives going full-out, and possably more on less severe service. 2.8A@24V, 2.5A@5V, 0.5A@–5V. Beautiful quality. $99.00



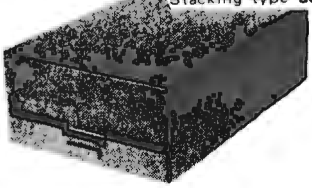
**DISKETTES** (Standard)

| | | |
|---|---|---|
| 8" | Boxed 10 | $39.00 |
| 5¼" | Boxed 10 | $34.95 |

**Tarbell ("It Works") Interface** (Includes cable set for 2 drives) $265.00 BUT ONLY $219.00 with purchase of two drives.

**Cable Kits** 10' with 50 cond. cable and connectors and also Molex connectors and power cable: For one drive: $27.50 For two drives: $33.95, and for three drives: $38.95

**CABINETS** for FDD120 and 801R Drives, or CP206 power supply. Matte finish in mar resistant black epoxy paint. Stacking type design. $29.99

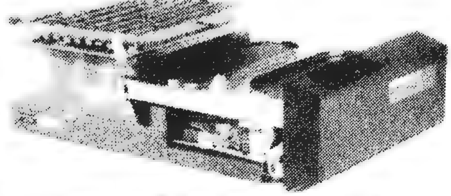


## ANNOUNCEMENT! SUPERDISK

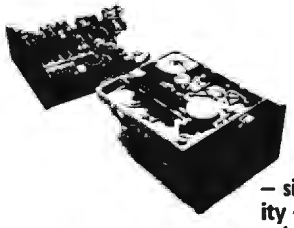
**ELECTROLABS is proud to announce appointment as DISTRIBUTOR by CII-Honeywell Bull.**

| | |
|---|---|
| PRICE BREAKTHROUGH on *SUPERDISK* 10 MBY! | $3495.00 |
| General purpose controller (requires 2 parallel I/O ports) | 1500.00 |
| S-100 Controller (DMA) | 995.00 |
| "RL–01/RK–05" surrogate (transparent to RT, RS, RX) | 1900.00 |

**SOFTWARE:** (CP/M Compatible)
SUPERDOS 1 (Z-80) ..... $695.00
MICRODOS 1 (TRS-80) ..... $199.00
Power supply (switching) .......... $395.00
Enclosure (desktop) .......... $ 99.00



Removeable Media Cartridge Drive



## 5¼" MINI-FLOPPY DRIVE $299.00

– single or double density – quick access time – high reliability & durability

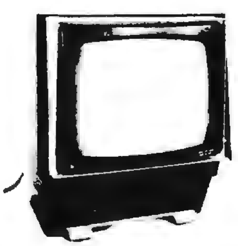
**Mini-floppy CABLE KIT:** for TRS-80 or your Tarbell controller. $24.95

## Daisy Wheel Printers
## Qume Sprint 3\45

New!

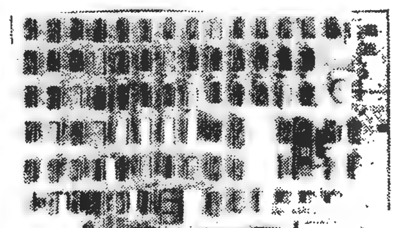
Print wheels $8.95 Ribbons $5.95

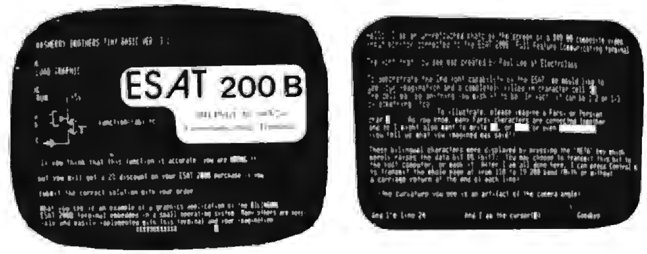
**OEM Style mechanism $1399.00**



Used Sylvania 12" Video Monitors. Composite video 15mhz, 115vac, 50/60hz New Tube. As shown $109 OEM style without case: $99, Anti-glare tube option add $12. Specify p4 or p39

## ESAT 200B






**BI-LINGUAL 80x24 COMMUNICATING TERMINAL**
**Scrolling, full cursor, bell, 8x8 metrix, 110-19,200 baud, Dual Font Applications. Arabic & Hebrew, Multilingual Data Entry, Forms Drawing, Music Instruction, Specialized Graphics (e.g. Games, Chemical Plants, Switchyards) $349.00** We carry keyboards, cases, power supplies, etc., enough to make an entire system.

### Associative Computer Memory Available from Semionics Associates

Content addressable or associative computer memory is available from Semionics Associates, 41 Tunnel Rd, Berkeley CA 94705. Called REM (recognition memory), it differs from conventional memory by eliminating serial searching. An item may be accessed simply by being named. REM can be written into and read from like ordinary memory, but has parallel processing functions, including six types of recognize and multiwrite. The recognition operations replace serial searching, while multiwrite allows the processor to write into multiple locations with a single instruction. Individual bit masking may be applied to all of the operations, including ordinary (location accessed) read and write. A data processing system with these functions is known as a CAPP (content addressable parallel processor). Ideal for pattern recognition and information retrieval applications, it is also capable of performing parallel arithmetic operations.

Semionics' first product is an add-in recognition memory for microcomputers having the S-100 bus. Called REM S-100, the board converts the microcomputer to a CAPP by adding new instructions to the instruction set of the processor. The board is organized to make these additional instructions possible without any alteration to the processor.

Recognition memory is organized in 8 bit words and 256 word REM records. It is a static memory with an access time of 200 ns for a single memory location, and recognize or multiwrite time, for all REM records of 4 μs. This time does not increase with size of memory. In a system with multiple REM boards, all of these are accessed in parallel during a recognize or multiwrite operation.

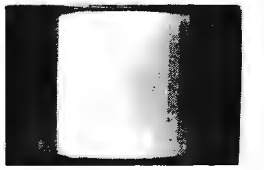
The REM S-100 add-in recognition memory board has a capacity of 4 K bytes and is priced at $345.

Circle 657 on inquiry card.

# SOLID STATE SALES . . . *Announces a Breakthrough in Computer Technology*

**A PICTURE MAY BE TAKEN BY OUR CAMERA, STORED IN A COMPUTER IN REAL TIME AND THEN DISPLAYED ON A CRT AT AN AFFORDABLE PRICE**

# VIDEO COMPUTER
## PROCESSING SYSTEM



GRAY LEVELS

THIS REMARKABLE VP-1 COMPUTER/INTERFACE KIT HAS THE FOLLOWING:

### FEATURES

- IT PRODUCES COMPOSITE VIDEO OUTPUT IN A 128 × 128 MATRIX FROM A DIRECT MONITOR CONNECTION USING 8K OF MEMORY
- THE SYSTEM USES A STANDARD S 100 BUSS
- WILL NOT TIE UP COMPUTER SOFTWARE WHEN NOT ADDRESSED
- IT DISPLAYS CONTINUOUSLY WHEN NOT ADDRESSED
- IT MAY PRODUCE PSEUDO COLOR AND/OR GRAPHICS (UP TO 16 GREY LEVELS, 4 BIT BINARY)

THE CAMERA WILL TAKE BETWEEN 15 AND 100 FRAMES/SECOND. THE CAMERA CONNECTS TO THE PROCESSOR WITH SEVEN LINES. THIS INCLUDES VIDEO AND TIMING SIGNALS

### APPLICATIONS

- CONTINUOUS SURVEILLANCE
- INSPECTION OF MOVING PARTS WITH PROPER STROBING
- VISUAL GRAPHIC INPUT TO A COMPUTER
- CHARACTER OR PATTERN RECOGNITION
- PICTURES MAY BE TAKEN DIRECTLY FROM A TV WITHOUT ELECTRICAL CONNECTIONS
- THE INTERFACE KIT MAY BE USED SEPARATELY AS A 128 × 128 16 LEVEL GRAPHIC DISPLAY

*OUR VP1 VIDEO SYSTEM CONSISTS OF THE FOLLOWING KITS:*

- CCD 202C SOLID STATE VIDEO CAMERA KIT ASSEMBLED & TESTED . . . . $499.00
- VP-1 COMPUTER/VIDEO INTERFACE SYSTEM (3 BOARDS) ASSEMBLED & TESTED . . . . . . . . . . . . . . . . . . . . $999.00
- ASSEMBLED 8K MEMORY BOARD (OPTIONAL) . . . . . . . . . . . . . . . . . $235.00

**THIS VIDEO COMPUTER KIT CAN WORK WITH THE GE, REDICON, OR ANY OTHER 128 × 128 SENSOR CAMERA**

### REGULATED POWER SUPPLIES

POWER SYSTEMS # PS1111
115-230V 50/60 cy. in 5v DC at 35A out.
6"x 16¾"x 15¼" 26 lbs. shipping weight $55.00

POWER SYSTEMS # PS1106
115-230V 50/60 cy. in 12v DC at 15A out.
5"x 16¾"x 5" 19 lbs. shipping weight. $49.00
(OV PROTECT)

### C/MOS (DIODE CLAMPED)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4001 | 20 | 4019 | 37 | 4049 | 40 | 74C73 | 65 |
| 4002 | 20 | 4020 | 90 | 4050 | 40 | 74C74 - | 45 |
| 4007 | 18 | 4021 | 90 | 4051 – | 1 00 | 74C88 | 40 |
| 4009 | 37 | 4022 | 90 | 4053 | 1 10 | 74C93 | 75 |
| 4010 | 37 | 4023 | 18 | 4055 | 1 25 | 74C151 | 1.40 |
| 4011 | .20 | 4024 | 75 | 4066 | 70 | 74C160 | 1 05 |
| 4012 | 20 | 4025 | 18 | 4069 – | .45 | 74C161 | 1·05 |
| 4014 | 75 | 4027 | 37 | 4071 | 20 | 74C174 | 1.05 |
| 4015 | 75 | 4028 | 80 | 4072 – | .21 | 74C175 | 1 05 |
| 4016 | 37 | 4029 | 95 | 4076 | 97 | 74C901 | 48 |
| 4017 | 1.05 | 4030 | 33 | 4518 | .95 | 74C902 | 48 |
| 4018 | 90 | 4035 | 97 | 4520 | 70 | 74C914 | 1 70 |
| | | 4042 | 65 | 74C00 | 22 | | |
| | | 4044 | 65 | 74C02 | 22 | | |
| | | | | 74C10 | 27 | | |

| | | | |
|---|---|---|---|
| 1488 RS232 INTERFACE | $1 40 | 2117 16K DYN RAM | 7 95 |
| 1489 RS232 INTERFACE | 1 28 | UPD410 1-4K STAT RAM | 9 95 |
| 2509 TRI STATE STAT SR | 1 35 | 4116 Z | 9 95 |
| 2513 CHARAC GEN UP | 6 75 | 5280 2107B 4K DYN RAM | 1 40 |
| 2518 HEX 32 BIT SR | 2 20 | TMS 4050L | 1 95 |
| 2522 STATIC SHIFT REG | 1 95 | 2704 4K PROM | 4 95 |
| 2708 8K EPROM (450 ns) | 12 50 | 87523 | 1 95 |
| TMS 4050NL | 2.95 | 825129 – | 2 95 |
| TMS 3409 NC 80 BITS DYN SR | 1 35 | 825126 | 3 25 |
| 2716 (TI) | 35 98 | AY5 1013 UART | 5 25 |
| 2102 1 (450 ns) | 98 | TR1 602B | 4 05 |
| 21L02 1 (450 ns) | 1 28 | 8700C TELEDYNE 8 BIT AD CONVERT | 9 95 |
| MM5270 4K X1 DYN | 3.45 | FT 1487 R PEAT | 8 95 |
| MK4008P | 1.95 | 8080A | 7 75 |
| 2101 1 256 x 4 STATIC | 7 45 | 8224 | 2 95 |
| 2111 1 256 x 4 STATIC | 3.45 | | |
| 2112 1 256 x 4 STATIC | 7 75 | | |
| 2114 4K STAT RAM (450 ns) | 8 50 | | |

8130 2 95 · 81380 2 25 · 8834 2 25 · 8836 2 25
8131 2 95 · 8830 2.95 · 8837 2 25

CRYSTALS $3 45 ea
2.000 MHz
4.000 MHz
5 000 MHz
6 000 MHz
8.000 MHz
10 000 MHz
20.000 MHz

RIBBON CABLE
FLAT (COLOR CODED)
#30 WIRE
26 cond. - .50/per foot
40 cond. - .75/per foot
50 cond. - .90/per foot

CTS 206-8 eight position dip switch . . . . . $1.60
CTS 206-4 four position dip switch. . . . . . . $1.45
LIGHT ACTIVATED SCR's to 18 200 V 1A. $.70

### SILICON SOLAR CELLS

2¼" diameter .4V at 500 ma $4.00
4" diameter .4V at 1 AMP . . . . . . . . . . . . $10.00

### LED READOUTS

FND 359 C.C. .4" $ .60
FCS 8024 4 digit C.C. 8" display $5 95
FND 503 C.C. .5" $ 85
FND 510 C.A. .5" $ 85
DL 704 .3" C.C $ 85
DL-704 C.A. .3" $ .75
DL 747 C.A 6" $1.50
HP3400 .8"CC $1.95
HP3405 .8"CA $1.95

### PRINTED CIRCUIT BOARD

4" x 6" DOUBLE SIDED EPOXY BOARD 1/16" thick
$.60 ea . . . . . . . . . . . . . . . . . . 5/$2.60

**7 WATT LD 65 LASER DIODE IR $8.95**

| | | |
|---|---|---|
| 2N 3820 P FET | | $ 45 |
| 2N 5457 N FET | | $ 45 |
| 2N2646 UJT | | $ 45 |
| ER 900 TRIGGER DIODES | 4 | $1 00 |
| 2N 6028 PROG UJT | | $ 65 |

MINIATURE MULTI-TURN TRIM POTS
100, 1K, 2K, 5K, 10K, 20K, 50K, 200K, 1Meg, 2Meg, $.75 each 3/$2.00

### CHARGED COUPLE DEVICES

CCD 201C 100x100 Image Sensor . . . . . $95.00
CCD 202C 100x100 Image Sensor . . . . $145.00

**VERIPAX PC BOARD . . . . . . . . . . . . . $16.95**
Our new Prototyping is a hi density 4½" x 6½" single sided 1/16" epoxy board. It will hold 40, 24, 16 (34 units), 14 + 8 pin IC's. There are three busses, +5V, ground and a floating buss. There is a pad for a TO 220 regulator. There is a 22 pin edge connector with .156" spacing.

FP 100 PHOTO TRANS . . . . . . . . $ .50
RED, YELLOW, GREEN LARGE LED's. 2" . . . . . . . . . 6/$1.00
TIL-118 OPTO-ISOLATOR . . . . . . $ .75
MCT-6 OPTO ISOLATOR . . . . . . . . $ .80
1 WATT ZENERS. 3.3, 4.7, 5.1, 5.6, 9.1, 10, 12, 15, 18, or 22V . . . . . . 6/$1.00
MCM 6571A 7 x 9 character gen . $ 10.75

### UNIVERSAL 4Kx8 MEMORY BOARD KIT
**$69.95**

32-2102-1 fully buffered, 16 address lines, on board decoding for any 4 of 64 pages, standard 44 pin buss, may be used with F-8 & KIM

### Silicon Power Rectifiers

| PRV | 1A | 3A | 12A | 50A | 125A | 240A |
|---|---|---|---|---|---|---|
| 100 | 06 | 14 | 35 | 90 | 3 70 | 5.00 |
| 200 | 07 | 20 | 40 | 1.30 | 4 25 | 6.50 |
| 400 | 09 | 25 | .65 | 1.50 | 6 50 | 9 50 |
| 600 | 11 | 30 | 80 | 2 00 | 8 50 | 12 50 |
| 800 | 15 | 35 | 1.00 | 2 50 | 10 50 | 16.50 |
| 1000 | 20 | 45 | 1.25 | 3 00 | 12 50 | 20 00 |

SAD 1024 a REDICON 1024 stage analog "Bucket Brigade" shift register. $14.95

IN 4148 (IN914) 15/$1.00
.1 uf 25V ceramic caps 16/$1.00, $5.00/100

**RS232 CONNECTORS**
DB 25P male $2 95
DB 25S female $3.50
HOODS . . . $1.50

### REGULATORS

323K - 5V 3A . . $ 5.75
309K . . . . . . . $1.10
723 . . . . . . . . . $ .50
320T - 5, 12, or 15 V . . . $ 1.10
340K - 12, 15 or 24 V. . . . $ 1.10
340T - 5, 6, 8, 12 15, 18 or 24V $ 1.10
78 MG . . . . . . $ 1.35
79 MG . . . . . . $ 1.35

### TRANSISTOR SPECIALS

| | |
|---|---|
| 2N6233-NPN SWITCHING POWER | $ 1.95 |
| MRF-8004 a CB RF Transistor NPN | $ .75 |
| 2N3772 NPN Si TO 3 | $ 1.00 |
| 2N1546 PNP GE TO-3 . . . . . . . . . | $ .75 |
| 2N4908 PNP Si TO 3 | $ 1 00 |
| 2N5086 PNP Si TO 92 | 4 $ 1 00 |
| 2N3137 NPN Si RF . . . | $ .65 |
| 2N3919 NPN Si TO 3 RF | $ 1 50 |
| 2N1420 NPN Si TO 5 . . . . . . . . . | 3/$ 1.00 |
| 2N3767 NPN Si TO 66 | $ 70 |
| 2N2222 NPN Si TO 18 | 5 $ 1 00 |
| 2N3055 NPN Si TO 3 | $ 50 |
| 2N3904 NPN Si TO 92 | 6/$ 1.00 |
| 2N3906 PNP Si TO 92 | 6/$ 1.00 |
| 2N5296 NPN Si TO 220 | $ 50 |
| 2N6109 PNP Si TO 220 | $ 55 |
| 2N3638 PNP Si TO 5 | 5 $ 1 00 |
| MPSA 13 NPN Si. . . . . . . . . . . | 4/$ 1.00 |

### TTL IC SERIES

| | | | | | |
|---|---|---|---|---|---|
| | | 7447 | 58 | 5411 | 70 |
| | | 7448 – | 69 | 75497 | 70 |
| 7400 | 15 | 7450 | 18 | 74154 | 94 |
| 7401 | 15 | 7472 | 25 | 74155 | 58 |
| 7402 | 15 | 7473 | 28 | 74157 | 55 |
| 7403 | 15 | 7474 | 28 | 74161 | 56 |
| 7404 | 18 | 7475 | .45 | 74162 | 80 |
| 7405 | 18 | 7476 – | 30 | 74163 | 56 |
| 7406 | 22 | 7480 | .31 | 74164 | 85 |
| 7408 | 18 | 7483 | 65 | 74165 | .95 |
| 7409 | 18 | 7485 – | .87 | 74173 | 1 20 |
| 7410 | 15 | 7486 | 28 | 74174 | 95 |
| 7411 | 18 | 7489 | 1 25 | 74175 | 85 |
| 7412 | 18 | 7490 | 42 | 74176 - | 75 |
| 7413 | 36 | 7491 – | 58 | 74177 | 75 |
| 7414 | 60 | 7492 – | 43 | 74180 | 65 |
| 7416 | 37 | 7493 | .43 | 74181 | 1 90 |
| 7417 | 25 | 7494 | .67 | 74190 | 1.00 |
| 7420 | 18 | 7495 | 65 | 74191 | .85 |
| 7425 | 30 | 7496 | .65 | 74192 | 79 |
| 7426 | 22 | 74107 | 28 | 74193 | 79 |
| 7427 | .28 | 74121 | 30 | 74194 | 80 |
| 7430 | 18 | 74122 | 40 | 74195 | 50 |
| 7432 | .22 | 74123 | 45 | 74196 | 86 |
| 7437 | 27 | 74125 - | 40 | 74197 - | 80 |
| 7438 | 27 | 74126 | 40 | 74279 - | 55 |
| 7440 | 18 | 74145 - | .85 | 74298 - | 90 |
| 7441 | 70 | 74148 | 1 10 | 74368 - | 62 |
| 7442 | 45 | 74150 - | 1 00 | 74393 | 1 25 |
| 7445 | 65 | 74151 - | .61 | 75322 - | 1.50 |
| | | 74153 | 61 | 74325 - | 1 60 |

DATA CASSETTES 1/2 HR $ .95

14 pin headers 3/$1.00

MM5387AA . . . . . . . . . . . . . . $5 95
M7001 . . . . . . . . . . . . . . . . $7 50
MM5369 . . . . . . . . . . . . . . . $3 75

NO 30 WIRE WRAP WIRE SINGLE STRAND 100' $1.40

ALCO SUBMINIATURE TOGGLE SWITCHES
MTA 106 SPDT $ .95
MTA 206 DPDT $ 1.70
MTA 206 P-DPDT CENTER OFF $ 1.85
MSD 206 P-DPDT CENTER OFF LEVER SWITCH $ 1.85

### Full Wave Bridges

| PRV | 2A | 6A | 25A |
|---|---|---|---|
| 100 | | | 1 30 |
| 200 | 75 | 1 25 | 2 00 |
| 400 | 95 | 1.50 | 3 00 |
| 600 | 1 20 | 1.75 | 4.00 |

### DIP SOCKETS

8 PIN .17 · 24 PIN .35
14 PIN .20 · 28 PIN .40
16 PIN .22 · 40 PIN .60
18 PIN .25

### SANKEN AUDIO POWER AMPS

Si 1010 G 10 WATTS $ 7.80
Si 1020 G 20 WATTS $15.70
Si 1050 G 50 WATTS $28.50

### TANTULUM CAPACITORS

| | | | | | |
|---|---|---|---|---|---|
| .22UF | 35V | 5/$1.00 | 6.8UF | 35V | 4/$1.00 |
| 47UF | 35V | 5/$1 00 | 10UF | 10V | $ 25 |
| 68UF | 35V | 5/$1.00 | 22UF | 25V | $ .40 |
| 1UF | 35V | 5/$1.00 | 15UF | 35V | 3/$1.00 |
| 2 2UF | 20V | 5/$1.00 | 30UF | 6V | 5/$1.00 |
| 3.3UF | 20V | 4/$1.00 | 33UF | 20V | $ .40 |
| 4 7UF | 15V | 5/$1.00 | 47UF | 20V | $.35 |
| | | | 68UF | 15V | $ .50 |

### 74LS SERIES

CALL FOR AVAILABILITY & PRICING

74LS00, 74LS02, 74LS03, 74LS04, 74LS05, 74LS08, 74LS09, 74LS10, 74LS11, 74LS13, 74LS14, 74LS15, 74LS20, 74LS21, 74LS22, 74LS26, 74LS27, 74LS28, 74LS30, 74LS32, 74LS37, 74LS38, 74LS40, 74LS42, 74LS51, 74LS54, 74LS73, 74LS75, 74LS76, 74LS86, 74LS90, 74LS92, 74LS93, 74LS109, 74LS112, 74LS113, 74LS114, 74LS124, 74LS125, 74LS126, 74LS132

74LS138, 74LS139, 74LS151, 74LS153, 74LS155, 74LS156, 74LS157, 74LS160, 74LS161, 74LS162, 74LS163, 74LS164, 74LS168, 74LS169, 74LS170, 74LS173, 74LS174, 74LS175, 74LS181, 74LS190, 74LS191, 74LS192, 74LS193, 74LS195, 74LS196, 74LS197, 74LS241, 74LS257, 74LS258, 74LS259, 74LS265, 74LS266, 74LS279, 74LS290, 74LS293, 74LS365, 74LS366, 74LS368, 74LS375, 74LS386, 74LS390, 74LS670

### LINEAR CIRCUITS

| | |
|---|---|
| LM 201 | – 75 |
| LM 301 748 | 25 |
| LM307 | – 30 |
| LM 308 | – 75 |
| LM 311 | – 75 |
| LM 318 | – 1 20 |
| LM 324 | – 95 |
| LM 339 | – 1 10 |
| LM 358 | – 70 |
| LM 370 | – 1 15 |
| LM 377 | – 1 60 |
| LM 380 | – 95 |
| LM 382 | – 1 25 |
| LM386 | – .80 |
| LM 387 | 1 25 |
| LM 1800 | 1.65 |
| CA 3080 | 95 |
| LM 537 | – 2 50 |
| LM 553 | – 2.50 |
| LM 555 | – 49 |
| LM 556 | – 85 |
| 560 | – 2 00 |
| 565 | – .95 |
| 566 | 1 10 |
| 567 | – 1 10 |
| 703 | – 90 |
| 733H | 75 |
| 709 | – 25 |
| 711CH | .40 |
| 741C or V | – .30 |
| 747 | – 50 |
| LM 1310 | – 2 50 |
| 1456 | – 95 |
| 1458 | 50 |
| 3900 | – 40 |
| 8038CC | – 3 90 |
| 791 | – 1 95 |
| LF356H | – 1 20 |

### TRIACS · SCR'S

| PRV | 1A | 10A | 25A | 1.5A | 6A | 35A |
|---|---|---|---|---|---|---|
| 100 | 40 | 70 | 1 30 | 40 | 50 | 1 20 |
| 200 | 70 | 1.10 | 1 75 | 60 | 70 | 1 60 |
| 400 | 1 10 | 1 60 | 2 60 | 1 00 | 1 20 | 2 20 |
| 600 | 1 70 | 2 30 | 3 60 | | 1 50 | 3 00 |

Terms: FOB Cambridge, Mass. Send Check or Money Order. Include Postage, Minimum Order $5.00, COD'S $20.00

Send 25¢ for our catalog featuring Transistors and Rectifiers
145 Hampshire St., Cambridge, Mass.

SOLID STATE SALES
P.O. BOX 74B
SOMERVILLE, MASS. 02143 TEL. (617) 547-7053

WE SHIP OVER 95% OF OUR ORDERS THE DAY WE RECEIVE THEM

### Disk Operating and File Management System for 6800 Microcomputers

This disk operating and file management system, called INDEX (Interrupt Driven Executive), executes faster than most systems because the console and other I/O (input/output) devices are serviced by interrupt requests instead of by polling. INDEX supports unlimited disk commands. A user can expand INDEX by adding utility commands and driver routines. These reside on disk and are loaded into memory only when needed. With INDEX, files can be assigned an activity value as a parameter of the file name, and the user may thereby list or display only the file directory entries at or above the activity level specified in the DIR (directory listing) command.

INDEX handles both ASCII and binary files, and disk files are automatically created, allocated and de-allocated. Files are referenced by names, and file name parameters are addended for name extension, drive number, directory level and a file protection flag.

The INDEX operating system software also features a routine for copying files onto a disk. The console interface segment of INDEX software supports any standard serial ASCII terminal. It features program interrupt for runaway programs; operator start, stop and skip display control; interrupt-serviced, typed-ahead character queue buffer; and a secondary line editing queue buffer.

INDEX versions are available for the PerCom LFD-400, Southwest Technical Products' MF-68, Smoke Signal Broadcasting Company's BFD-68 disk systems, and Motorola's EXORciser development system.

INDEX is supplied on two 5 inch disks together with a users manual for $99.95. For further information contact PerCom Data Co, 318 Barnes, Garland TX 75042. Circle 658 on inquiry card.

### New Self-Merchandising Software From ComputerLand

ComputerLand is now making personal computing software available through all participating ComputerLand stores. SoftSpot is a custom designed, self-merchandising fixture offering off-the-shelf programs for personal finance, time budgeting, education, games, stock analysis, stock portfolio evaluation, and more. SoftSpot programs start at $7.95.

Customer education about personal computers and their applications are offered through MainBrain. Books, self-study cassettes and video tapes, and "in person" lecture programs are available from well known publishers. MainBrain has self-service instructions to assist customers in making their own choice of multimedia educational products.

For further information contact ComputerLand, 14400 Catalina St, San Leandro CA 94577. Circle 525 on inquiry card.

# 7400 TTL

| | | | | | |
|---|---|---|---|---|---|
| SN7400N | .16 | SN7470N | .29 | SN74160N | .89 |
| SN7401N | .18 | SN7472N | .29 | SN74161N | .89 |
| SN7402N | .18 | SN7473N | .35 | SN74162N | 1.95 |
| SN7403N | .16 | SN7474N | .35 | SN74163N | .89 |
| SN7404N | .18 | SN7475N | .49 | SN74164N | .89 |
| SN7405N | .20 | SN7476N | .35 | SN74165N | .89 |
| SN7406N | .29 | SN7479N | 5.00 | SN74166N | 1.25 |
| SN7407N | .29 | SN7480N | .50 | SN74167N | 1.95 |
| SN7408N | .20 | SN7482N | .99 | SN74170N | 1.59 |
| SN7409N | .20 | SN7483N | .59 | SN74172N | 6.00 |
| SN7410N | .18 | SN7485N | .79 | SN74173N | 1.25 |
| SN7411N | .25 | SN7486N | .35 | SN74174N | .89 |
| SN7412N | .25 | SN7489N | 1.75 | SN74175N | .79 |
| SN7413N | .40 | SN7490N | .45 | SN74176N | .79 |
| SN7414N | .70 | SN7491N | .59 | SN74177N | .79 |
| SN7416N | .25 | SN7492N | .43 | SN74179N | 1.95 |
| SN7417N | .25 | SN7493N | .43 | SN74180N | .79 |
| SN7420N | .20 | SN7494N | .65 | SN74181N | 1.95 |
| SN7421N | .29 | SN7495N | .65 | SN74182N | .79 |
| SN7422N | .39 | SN7496N | .65 | SN74184N | 1.95 |
| SN7423N | .25 | SN7497N | 3.00 | SN74185N | 1.95 |
| SN7425N | .29 | SN74100N | .89 | SN74186N | 9.95 |
| SN7426N | .29 | SN74107N | .35 | SN74188N | 3.95 |
| SN7427N | .25 | SN74109N | .59 | SN74190N | 1.25 |
| SN7429N | .39 | SN74116N | 1.95 | SN74191N | 1.25 |
| SN7430N | .20 | SN74121N | .35 | SN74192N | .79 |
| SN7432N | .25 | SN74122N | .39 | SN74193N | .79 |
| SN7437N | .25 | SN74123N | .49 | SN74194N | .89 |
| SN7438N | .25 | SN74125N | .49 | SN74195N | .69 |
| SN7439N | .25 | SN74126N | .49 | SN74196N | .89 |
| SN7440N | .20 | SN74132N | .75 | SN74197N | .89 |
| SN7441N | .89 | SN74136N | .75 | SN74198N | 1.49 |
| SN7442N | .49 | SN74141N | .79 | SN74199N | 1.49 |
| SN7443N | .75 | SN74142N | 2.95 | SN74S200 | 4.95 |
| SN7444N | .75 | SN74143N | 2.95 | SN74251N | 1.79 |
| SN7445N | .75 | SN74144N | 2.95 | SN74279N | .79 |
| SN7446N | .69 | SN74145N | .79 | SN74283N | 2.25 |
| SN7447N | .59 | SN74147N | 1.95 | SN74284N | 3.95 |
| SN7448N | .79 | SN74148N | 1.29 | SN74285N | 3.95 |
| SN7450N | .20 | SN74150N | .89 | SN74365N | .69 |
| SN7451N | .20 | SN74151N | .59 | SN74366N | .69 |
| SN7453N | .20 | SN74152N | .59 | SN74367N | .69 |
| SN7454N | .20 | SN74153N | .59 | SN74368N | .69 |
| SN7459A | .25 | SN74154N | .99 | SN74390N | 1.95 |
| SN7460N | .20 | SN74155N | .79 | SN74393N | 1.95 |
| | | SN74156N | .79 | | |
| | | SN74157N | .65 | | |

# C/MOS

| | | | | | |
|---|---|---|---|---|---|
| CD4000 | .23 | CD4028 | .89 | CD4070 | .55 |
| CD4001 | .23 | CD4029 | 1.19 | CD4071 | .23 |
| CD4002 | .23 | CD4030 | .49 | CD4072 | .49 |
| CD4006 | 1.19 | CD4035 | .99 | CD4076 | 1.39 |
| CD4007 | .25 | CD4040 | 1.19 | CD4081 | .23 |
| CD4009 | .49 | CD4041 | 1.25 | CD4082 | .23 |
| CD4010 | .49 | CD4042 | .99 | CD4093 | .99 |
| CD4011 | .23 | CD4043 | .89 | CD4098 | 2.49 |
| CD4012 | .25 | CD4044 | .89 | MC14409 | 14.95 |
| CD4013 | .39 | CD4046 | 1.79 | MC14410 | 14.95 |
| CD4014 | 1.39 | CD4047 | 2.50 | MC14411 | 14.95 |
| CD4015 | 1.19 | CD4048 | 1.35 | MC14419 | 4.95 |
| CD4016 | .49 | CD4049 | .49 | MC14433 | 19.95 |
| CD4017 | 1.19 | CD4050 | .49 | MC14506 | .75 |
| CD4018 | .99 | CD4051 | 1.19 | MC14507 | .99 |
| CD4019 | .49 | CD4053 | 1.19 | MC14562 | 14.50 |
| CD4020 | 1.19 | CD4056 | 2.95 | MC14583 | 3.50 |
| CD4021 | 1.39 | CD4059 | 9.95 | CD4508 | 3.95 |
| CD4022 | 1.19 | CD4060 | 1.49 | CD4510 | 1.39 |
| CD4023 | .23 | CD4066 | .79 | CD4511 | 1.29 |
| CD4024 | .79 | CD4068 | .39 | CD4515 | 2.95 |
| CD4025 | .23 | CD4069 | .45 | CD4518 | 1.29 |
| CD4026 | 2.25 | | | CD4520 | 1.29 |
| CD4027 | .69 | | | CD4566 | 2.25 |

# 74C00

| | | | | | |
|---|---|---|---|---|---|
| 74C00 | .39 | 74C85 | 2.49 | 74C163 | 2.49 |
| 74C02 | .39 | 74C90 | 1.95 | 74C164 | 2.49 |
| 74C04 | .39 | 74C93 | 1.95 | 74C173 | 2.60 |
| 74C08 | .49 | 74C95 | 1.95 | 74C192 | 2.49 |
| 74C10 | .39 | 74C107 | 1.25 | 74C193 | 2.49 |
| 74C14 | 1.95 | 74C151 | 2.90 | 74C195 | 2.49 |
| 74C20 | .39 | 74C154 | 3.00 | 74C922 | 5.95 |
| 74C30 | .39 | 74C157 | 2.15 | 74C923 | 6.25 |
| 74C42 | 1.95 | 74C160 | 2.49 | 74C925 | 8.95 |
| 74C48 | 2.49 | 74C161 | 2.49 | 74C926 | 8.95 |
| 74C73 | .89 | | | 80C95 | 1.50 |
| 74C74 | .89 | | | 80C97 | 1.50 |

# LINEAR

| | | | | | |
|---|---|---|---|---|---|
| 78MG | 1.75 | LM340K-18 | 1.35 | LM710N | .79 |
| LM106H | .99 | LM340K-24 | 1.35 | LM711N | .39 |
| LM300H | .80 | LM340T-5 | 1.25 | LM723N/H | .55 |
| LM301CN/H | .35 | LM340T-6 | 1.25 | LM733N | 1.00 |
| LM302H | .75 | LM340T-8 | 1.25 | LM739N | 1.19 |
| LM304H | 1.00 | LM340T-12 | 1.25 | LM741CN/H | .35 |
| LM305H | .60 | LM340T-15 | 1.25 | LM741-14N | .39 |
| LM307CN/H | .35 | LM340T-18 | 1.25 | LM747N/H | .79 |
| LM308CN/H | 1.00 | LM340T-24 | 1.25 | LM748N/H | .39 |
| LM309H | 1.10 | LM358N | 1.00 | LM1310N | 2.95 |
| LM309K | 1.25 | LM370N | 1.95 | LM1458CN/H | .59 |
| LM310CN | 1.15 | LM373N | 3.25 | MC1488N | 1.39 |
| LM311N/H | .90 | LM377N | 4.00 | MC1489N | 1.39 |
| LM312H | 1.95 | LM380N | 1.25 | LM1496N | .95 |
| LM317K | 6.50 | LM380CN | .99 | LM1556V | 1.75 |
| LM318CN/H | 1.50 | LM381N | 1.79 | MC1741SCP | 3.00 |
| LM319N | 1.30 | LM382N | 1.79 | LM2111N | 1.95 |
| LM320K-5 | 1.35 | NE501N | 8.00 | LM2901N | 2.95 |
| LM320K-5.2 | 1.35 | NE510A | 6.00 | LM3053N | 1.50 |
| LM320K-12 | 1.35 | NE529A | 4.95 | LM3065N | 1.49 |
| LM320K-15 | 1.35 | NE531H/V | 3.95 | LM3900N(3401) | .49 |
| LM320K-18 | 1.35 | NE536T | 6.00 | LM3905N | .89 |
| LM320K-24 | 1.35 | NE540L | 6.00 | LM3909N | 1.25 |
| LM320T-5 | 1.25 | NE544N | 4.95 | MC5558V | .59 |
| LM320T-5.2 | 1.25 | NE550N | 1.30 | 8038B | 4.95 |
| LM320T-8 | 1.25 | NE555V | .39 | LM75450N | .49 |
| LM320T-12 | 1.25 | NE556N | .99 | 75451CN | .39 |
| LM320T-15 | 1.25 | NE560B | 5.00 | 75452CN | .39 |
| LM320T-18 | 1.25 | NE561B | 5.00 | 75453CN | .39 |
| LM320T-24 | 1.25 | NE562B | 5.00 | 75454CN | .39 |
| LM323K-5 | 5.95 | NE565N/H | 1.25 | 75491CN | .79 |
| LM324N | 1.80 | NE566CN | 1.75 | 75492CN | .89 |
| LM339N | .99 | NE567V/H | .99 | 75493N | .89 |
| LM340K-5 | 1.35 | NE570N | 4.95 | 75494CN | .89 |
| LM340K-6 | 1.35 | LM703CN/H | .69 | RC4136 | 1.25 |
| LM340K-8 | 1.35 | LM709N/H | .29 | RC4151 | 2.85 |
| LM340K-12 | 1.35 | | | RC4194 | 5.95 |
| LM340K-15 | 1.35 | | | RC4195 | 4.49 |

# 74LS00TTL

| | | | | | |
|---|---|---|---|---|---|
| 74LS00 | .29 | 74LS47 | .89 | 74LS138 | .89 |
| 74LS01 | .29 | 74LS51 | .29 | 74LS139 | .89 |
| 74LS02 | .29 | 74LS54 | .29 | 74LS151 | .89 |
| 74LS03 | .29 | 74LS55 | .29 | 74LS155 | .89 |
| 74LS04 | .35 | 74LS73 | .45 | 74LS157 | .89 |
| 74LS05 | .35 | 74LS74 | .45 | 74LS160 | 1.15 |
| 74LS08 | .29 | 74LS75 | .59 | 74LS161 | 1.15 |
| 74LS09 | .29 | 74LS76 | .49 | 74LS162 | 1.15 |
| 74LS10 | .29 | 74LS78 | .45 | 74LS163 | 1.15 |
| 74LS11 | .75 | 74LS83 | .89 | 74LS164 | 1.25 |
| 74LS13 | .59 | 74LS85 | 1.25 | 74LS175 | .99 |
| 74LS14 | 1.25 | 74LS86 | .45 | 74LS181 | 2.49 |
| 74LS15 | .35 | 74LS90 | .59 | 74LS190 | 1.15 |
| 74LS20 | .29 | 74LS92 | .75 | 74LS191 | 1.15 |
| 74LS21 | .35 | 74LS93 | .75 | 74LS192 | 1.15 |
| 74LS22 | .35 | 74LS95 | .99 | 74LS193 | 1.15 |
| 74LS26 | .35 | 74LS96 | 1.15 | 74LS194 | 1.15 |
| 74LS27 | .35 | 74LS107 | .45 | 74LS195 | 1.15 |
| 74LS28 | .35 | 74LS109 | .45 | 74LS253 | .99 |
| 74LS30 | .29 | 74LS112 | .45 | 74LS257 | .89 |
| 74LS32 | .35 | 74LS123 | 1.25 | 74LS258 | 1.75 |
| 74LS37 | .45 | 74LS125 | .89 | 74LS260 | .69 |
| 74LS40 | .35 | 74LS132 | .99 | 74LS279 | .75 |
| 74LS42 | .89 | 74LS136 | .49 | 74LS367 | .75 |
| | | | | 74LS368 | .75 |
| | | | | 74LS670 | 2.49 |

# EXCITING NEW KITS

## NEW JE600 HEXADECIMAL ENCODER KIT

FEATURES:
- Full 8 bit latched output for microprocessor use
- 3 User Define keys with one being bistable operation
- Debounce circuit provided for all 19 keys
- LED readout to verify entries
- Easy interfacing with standard 16 pin IC connector
- Only +5VDC required for operations

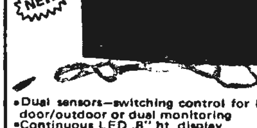
FULL 8 BIT LATCHED OUTPUT—19 KEYBOARD

The JE600 Encoder Keyboard provides two separate hexadecimal digits produced from sequential key entries to allow direct programming for 8 bit microprocessor or 8 bit memory circuits. Three (3) additional keys are provided for user operations with one having a bistable output available. The outputs are latched and monitored with LED readouts. Also included is a key entry strobe

**JE600 .......... $59.95**

## Digital Thermometer Kit

NEW



- Dual sensors—switching control for indoor/outdoor or dual monitoring
- Continuous LED .8" ht. display
- Range: -40°F to 199°F / -40°C to 100°C
- Accuracy: ±1° nominal
- Set for Fahrenheit or Celsius reading
- Sim. walnut case - AC wall adapter incl.
- Size: 3-1/4"H x 6-5/8"W x 1-3/8"D

**JE300 ....... $39.95**

# DISCRETE LEDS

| | | |
|---|---|---|
| **.200" dia.** | | |
| XC556R | red | 5/$1 |
| XC556G | green | 4/$1 |
| XC556Y | yellow | 4/$1 |
| XC556C | clear | 4/$1 |
| **.200" dia.** | | |
| XC22R | red | 5/$1 |
| XC22G | green | 4/$1 |
| XC22Y | yellow | 4/$1 |
| **.170" dia.** | | |
| MV108 | red | 4/$1 |
| **.085" dia.** | | |
| MV50 | red | 6/$1 |
| **.125" dia.** | | |
| XC209R | red | 5/$1 |
| XC209G | green | 4/$1 |
| XC209Y | yellow | 4/$1 |
| **.185" dia.** | | |
| XC526R | red | 5/$1 |
| XC526G | green | 4/$1 |
| XC526Y | yellow | 4/$1 |
| XC526C | clear | 4/$1 |
| **.190" dia.** | | |
| XC111R | red | 5/$1 |
| XC111G | green | 4/$1 |
| XC111Y | yellow | 4/$1 |
| XC111C | clear | 4/$1 |

INFRA-RED LED 1/4"x1/4"x1/16" flat 5/$1

TIMEX T1001
**LIQUID CRYSTAL DISPLAY CLASS II**
FIELD EFFECT

4 DIGIT — .5" CHARACTERS
THREE ENUNCIATORS
2.00" X 1.20" PACKAGE
INCLUDES CONNECTOR

| | |
|---|---|
| T1001-Transmissive | $7.95 |
| T1001A-Reflective | 8.25 |

# DISPLAY LEDS

| TYPE | POLARITY | HT | PRICE | TYPE | POLARITY | HT | PRICE |
|---|---|---|---|---|---|---|---|
| MAN 1 | Common Anode-red | .270 | 2.95 | MAN 6730 | Common Anode-red ± 1 | .560 | .99 |
| MAN 2 | 5 x 7 Dot Matrix-red | .300 | 4.95 | MAN 6740 | Common Cathode-red-D.D. | .560 | .99 |
| MAN 3 | Common Cathode-red | .125 | .25 | MAN 6750 | Common Cathode-red ± 1 | .560 | .99 |
| MAN 4 | Common Cathode-red | .187 | 1.95 | MAN 6760 | Common Anode-red | .560 | .99 |
| MAN 7G | Common Anode-green | .300 | 1.25 | MAN 6780 | Common Cathode-red | .560 | .99 |
| MAN 7Y | Common Anode-yellow | .300 | .99 | DL701 | Common Anode-red ± 1 | .300 | .99 |
| MAN 72 | Common Anode-red | .300 | .99 | DL704 | Common Cathode-red | .300 | .99 |
| MAN 74 | Common Cathode-red | .300 | 1.25 | DL707 | Common Anode-red | .300 | .99 |
| MAN 82 | Common Anode-yellow | .300 | .99 | DL728 | Common Cathode-red | .500 | 1.49 |
| MAN 84 | Common Cathode-yellow | .300 | .99 | DL741 | Common Anode-red | .600 | 1.25 |
| MAN 3620 | Common Anode-orange | .300 | .99 | DL746 | Common Anode-red ± 1 | .630 | 1.49 |
| MAN 3630 | Common Anode-orange ± 1 | .300 | .99 | DL747 | Common Anode-red | .600 | 1.49 |
| MAN 3640 | Common Cathode-orange | .300 | .99 | DL749 | Common Cathode-red ± 1 | .630 | 1.49 |
| MAN 4610 | Common Anode-orange | .400 | .99 | DL750 | Common Cathode-red | .600 | 1.49 |
| MAN 4640 | Common Cathode-orange | .400 | .99 | DL338 | Common Cathode-red | .110 | .35 |
| MAN 4710 | Common Anode-red | .400 | .99 | FND70 | Common Cathode | .250 | .69 |
| MAN 4730 | Common Anode-red ± 1 | .400 | .99 | FND358 | Common Cathode ± 1 | .357 | .99 |
| MAN 4740 | Common Cathode-red | .400 | .99 | FND359 | Common Cathode | .357 | .75 |
| MAN 4810 | Common Anode-yellow | .400 | .99 | FND503 | Common Cathode(FND500) | .500 | .99 |
| MAN 4840 | Common Cathode-yellow | .400 | .99 | FND507 | Common Anode (FND510) | .500 | .99 |
| MAN 6610 | Common Anode-orange-D.D. | .560 | .99 | 5082-7730 | Common Anode-red | .300 | 1.30 |
| MAN 6630 | Common Anode-orange ± 1 | .560 | .99 | HDSP-3400 | Common Anode-red | .800 | 2.10 |
| MAN 6640 | Common Cathode-orange-D.D | .560 | .99 | HDSP-3403 | Common Cathode red | .800 | 2.10 |
| MAN 6650 | Common Cathode-orange ± 1 | .560 | .99 | 5082-7300 | 4 x 7 sgl Digit-RHDP | .600 | 19.95 |
| MAN 6660 | Common Anode-orange | .560 | .99 | 5082-7302 | 4 x 7 Sgl Digit-LHDP | .600 | 19.95 |
| MAN 6680 | Common Cathode-orange | .560 | .99 | 5082-7304 | Overrange character (±1) | .600 | 15.00 |
| MAN 6710 | Common Anode-red-D.D. | .560 | .99 | 5082-7340 | 4 x 7 Sgl Digit-Hexadecimal | .600 | 22.50 |

# RCA LINEAR

| | | | |
|---|---|---|---|
| CA3013T | 2.15 | CA3082N | 2.00 |
| CA2023T | 2.56 | CA3083N | 1.60 |
| CA3035T | 2.48 | CA3086N | .85 |
| CA3039T | 1.35 | CA3089N | 3.75 |
| CA3046N | 1.30 | CA3130T | 1.39 |
| CA3059N | 3.25 | CA3140T | 1.25 |
| CA3060N | 3.25 | CA3160T | 1.25 |
| CA3080T | .85 | CA3401N | .49 |
| CA3081N | 2.00 | CA3600N | 3.50 |

# CALCULATOR CHIPS/DRIVERS

| | |
|---|---|
| MM5725 | $2.95 |
| MM5738 | 2.95 |
| DM8864 | 2.00 |
| DM8865 | 1.00 |
| DM8887 | .75 |
| DM8889 | .75 |
| 9374 7 seg. C.A. LED driver | 1.50 |

# CLOCK CHIPS

| | |
|---|---|
| MM5309 | $4.95 |
| MM5311 | 4.95 |
| MM5312 | 4.95 |
| MM5314 | 4.95 |
| MM5316 | 6.95 |
| MM5318 | 9.95 |
| MM5369 | 2.95 |
| MM5387/1998A | 4.95 |
| MM5841 | 9.95 |

# MOTOROLA

| | |
|---|---|
| MC1408L7 | $4.95 |
| MC1408L8 | 5.75 |
| MC1439L | 2.95 |
| MC3022P | 2.95 |
| MC3061P | 3.50 |
| MC4016(74416) | 7.50 |
| MC4024P | 3.95 |
| MC4040P | 6.95 |
| MC4044P | 4.50 |

# IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS

| | 1-24 | 25-49 | 50-100 | | 1-24 | 25-49 | 50-100 |
|---|---|---|---|---|---|---|---|
| 8 pin LP | $.17 | .16 | .15 | 22 pin LP | $.37 | .36 | .35 |
| 14 pin LP | .20 | .19 | .18 | 24 pin LP | .38 | .37 | .36 |
| 16 pin LP | .22 | .21 | .20 | 28 pin LP | .45 | .44 | .43 |
| 18 pin LP | .29 | .28 | .27 | 36 pin LP | .60 | .59 | .58 |
| 20 pin LP | .34 | .32 | .30 | 40 pin LP | .63 | .62 | .61 |

## SOLDERTAIL STANDARD (TIN)

| | 1-24 | 25-49 | 50-100 | | 1-24 | 25-49 | 50-100 |
|---|---|---|---|---|---|---|---|
| 14 pin ST | $.27 | .25 | .24 | 28 pin ST | $.99 | .90 | .81 |
| 16 pin ST | .30 | .27 | .25 | 36 pin ST | 1.39 | 1.28 | 1.15 |
| 18 pin ST | .35 | .32 | .30 | 40 pin ST | 1.59 | 1.45 | 1.30 |
| 24 pin ST | .49 | .45 | .42 | | | | |

## SOLDERTAIL STANDARD (GOLD)

| | 1-24 | 25-49 | 50-100 | | 1-24 | 25-49 | 50-100 |
|---|---|---|---|---|---|---|---|
| 8 pin SG | $.30 | .27 | .24 | 24 pin SG | $.70 | .63 | .57 |
| 14 pin SG | .35 | .32 | .29 | 28 pin SG | 1.10 | 1.00 | .90 |
| 16 pin SG | .38 | .35 | .32 | 36 pin SG | 1.65 | 1.40 | 1.28 |
| 18 pin SG | .52 | .47 | .43 | 40 pin SG | 1.75 | 1.59 | 1.45 |

## WIRE WRAP SOCKETS (GOLD) LEVEL #3

| | 1-24 | 25-49 | 50-100 | | 1-24 | 25-49 | 50-100 |
|---|---|---|---|---|---|---|---|
| 8 pin WW | $.39 | .38 | .31 | 22 pin WW | $.95 | .85 | .75 |
| 10 pin WW | .45 | .41 | .37 | 24 pin WW | 1.05 | .95 | .85 |
| 14 pin WW | .39 | .38 | .37 | 28 pin WW | 1.40 | 1.25 | 1.10 |
| 16 pin WW | .43 | .42 | .41 | 36 pin WW | 1.59 | 1.45 | 1.30 |
| 18 pin WW | .75 | .68 | .62 | 40 pin WW | 1.75 | 1.55 | 1.40 |

# 1/4 WATT RESISTOR ASSORTMENTS – 5%

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ASST. 1 | 5 ea. | 10 OHM, 27 OHM | 12 OHM, 33 OHM | 15 OHM, 39 OHM | 18 OHM, 47 OHM | 22 OHM, 56 OHM | 50 PCS | $1.75 |
| ASST. 2 | 5 ea | 68 OHM, 180 OHM | 82 OHM, 220 OHM | 100 OHM, 270 OHM | 120 OHM, 330 OHM | 150 OHM, 390 OHM | 50 PCS | 1.75 |
| ASST. 3 | 5 ea | 470 OHM, 1.2K | 560 OHM, 1.5K | 680 OHM, 1.8K | 820 OHM, 2.2K | 1K, 2.7K | 50 PCS | 1.75 |
| ASST. 4 | 5 ea | 3.3K, 8.2K | 3.9K, 10K | 4.7K, 12K | 5.6K, 15K | 6.8K, 18K | 50 PCS | 1.75 |
| ASST. 5 | 5 ea | 22K, 56K | 27K, 68K | 33K, 82K | 39K, 100K | 47K, 120K | 50 PCS | 1.75 |
| ASST. 6 | 5 ea | 150K, 390K | 180K, 470K | 220K, 560K | 270K, 680K | 330K, 820K | 50 PCS | 1.75 |
| ASST. 7 | 5 ea | 1M, 2.7M | 1.2M, 3.3M | 1.5M, 3.9M | 1.8M, 4.7M | 2.2M, 5.6M | 50 PCS | 1.75 |

**ASST. 8R** Includes Resistor Assortments 1-7 (350 PCS.) **$9.95 ea.**

**$10.00 MINIMUM ORDER — U.S. Funds Only** **Spec Sheets — 25¢**
**California Residents — Add 6% Sales Tax** **1979 Catalog Available—Send 41¢ stamp**

FREE A 1979 CATALOG

**Jameco ELECTRONICS**

PHONE ORDERS WELCOME (415) 592-8097

*MAIL ORDER ELECTRONICS – WORLDWIDE*
1021 HOWARD AVENUE, SAN CARLOS, CA 94070
ADVERTISED PRICES GOOD THRU JULY

# TELEPHONE/KEYBOARD CHIPS

| | | |
|---|---|---|
| AY-5-9100 | Push Button Telephone Dialler | $14.95 |
| AY-5-9200 | Repertory Dialler | 14.95 |
| AY-5-9500 | CMOS Clock Generator | 4.95 |
| AY-5-2376 | Keyboard Encoder (88 keys) | 14.95 |
| HD0165 | Keyboard Encoder (16 keys) | 7.95 |
| 74C922 | Keyboard Encoder (16 keys) | 5.95 |

# ICM CHIPS

| | | |
|---|---|---|
| ICM7045 | CMOS Precision Timer | 24.95 |
| ICM7205 | CMOS LED Stopwatch/Timer | 19.95 |
| ICM7207 | Oscillator Controller | 7.50 |
| ICM7208 | Seven Decade Counter | 19.95 |
| ICM7209 | Clock Generator | 6.95 |

# NMOS READ ONLY MEMORIES

| | | |
|---|---|---|
| MCM6571 | 128 X 9 X 7 ASCII Shifted with Greek | 13.50 |
| MCM6574 | 128 X 9 X 7 Math Symbol & Pictures | 13.50 |
| MCM6575 | 128 X 9 X 7 Alphanumeric Control Character Generator | 13.50 |

# MISCELLANEOUS

| | | |
|---|---|---|
| TL074CN | Quad Low Noise bi-fet Op Amp | 2.49 |
| TL494CN | Switching Regulator | 4.49 |
| TL496CP | Single Switching Regulator | 1.75 |
| 11C90 | Divide 10/11 Prescaler | 19.95 |
| 95H90 | Hi-Speed Divide 10/11 Prescaler | 11.95 |
| 4N33 | Photo-Darlington Opto-Isolator | 3.95 |
| MK50240 | Top Octave Freq. Generator | 17.50 |
| DS0026CH | 5Mhz 2-phase MOS clock driver | 3.75 |
| TIL308 | .27" red num. display w/integ. logic chip | 10.50 |
| MM5320 | TV Camera Sync. Generator | 14.95 |
| MM5330 | 4½ Digit DPM Logic Block (Special) | 3.95 |
| LD110/111 | 3½ Digit A/D Converter Set | 25.00/set |

**LITRONIX ISO-LIT 1**
Photo Transistor Opto-Isolator (Same as MCT 2 or 4N25)
**2/99¢**

**SN 76477**
SOUND GENERATOR
Generates Complex Sounds
Low Power - Programmable
**3.95 each**

**TV GAME CHIP AND CRYSTAL**
AY-3-8500-1 and 2.01 MHZ Crystal (Chip & Crystal includes score display, 6 games and select angles, etc. **7.95/set**

# EXAR

| | | | | | |
|---|---|---|---|---|---|
| XR205 | $8.40 | JE2206KA | 14.95 | XR2242CP | 1.50 |
| XR210 | 4.40 | JE2206KB | 19.95 | XR2264 | 4.25 |
| XR215 | 4.40 | XR1800 | 3.20 | XR2556 | 3.20 |
| XR320 | 1.55 | XR2206 | 4.40 | XR2567 | 2.99 |
| XR-L555 | 1.50 | XR2207 | 3.85 | XR3403 | 1.25 |
| XR555 | .39 | XR2208 | 5.20 | XR4136 | 1.25 |
| XR556 | .99 | XR2209 | 1.75 | XR4151 | 2.85 |
| XR567CP | .99 | XR2211 | 5.25 | XR4194 | 4.95 |
| XR567CT | 1.25 | XR2212 | 4.35 | XR4202 | 3.60 |
| XR1310P | 1.30 | XR2240 | 3.45 | XR4212 | 2.05 |
| XR1468CN | 3.85 | | | XR4558 | .75 |
| XR1488 | 1.39 | | | XR4739 | 1.15 |
| XR1489 | 1.39 | | | XR4741 | 1.47 |

# DIODES

| TYPE | VOLTS | W | PRICE | TYPE | VOLTS | W | PRICE |
|---|---|---|---|---|---|---|---|
| 1N746 | 3.3 | 400m | 4/1.00 | 1N4002 | 100 PIV | 1 AMP | 12/1.00 |
| 1N751 | 5.1 | 400m | 4/1.00 | 1N4003 | 200 PIV | 1 AMP | 12/1.00 |
| 1N752 | 5.6 | 400m | 4/1.00 | 1N4004 | 400 PIV | 1 AMP | 12/1.00 |
| 1N753 | 6.2 | 400m | 4/1.00 | 1N4005 | 600 PIV | 1 AMP | 10/1.00 |
| 1N754 | 6.8 | 400m | 4/1.00 | 1N4006 | 800 PIV | 1 AMP | 10/1.00 |
| 1N757 | 9.0 | 400m | 4/1.00 | 1N4007 | 1000 PIV | 1 AMP | 10/1.00 |
| 1N759 | 12.0 | 400m | 4/1.00 | 1N3600 | 50 | 200m | 6/1.00 |
| 1N959 | 8.2 | 400m | 4/1.00 | 1N4148 | 75 | 10m | 15/1.00 |
| 1N965 | 15 | 400m | 4/1.00 | 1N4154 | 35 | 10m | 12/1.00 |
| 1N5232 | 5.6 | 500m | .28 | 1N4305 | 75 | 25m | 15/1.00 |
| 1N5234 | 6.2 | 500m | .28 | 1N4734 | 5.6 | 1w | .28 |
| 1N5235 | 6.8 | 500m | .28 | 1N4735 | 6.2 | 1w | .28 |
| 1N5236 | 7.5 | 500m | .28 | 1N4736 | 6.8 | 1w | .28 |
| 1N5242 | 12 | 500m | .28 | 1N4738 | 8.2 | 1w | .28 |
| 1N5245 | 15 | 500m | .28 | 1N4742 | 12 | 1w | .28 |
| 1N456 | 25 | 40m | 6/1.00 | 1N4744 | 15 | 1w | .28 |
| 1N458 | 150 | 7m | 6/1.00 | 1N1183 | 50 PIV | 35 AMP | 1.60 |
| 1N485A | 180 | 10m | 5/1.00 | 1N1184 | 100 PIV | 35 AMP | 1.70 |
| 1N4001 | 50 PIV | 1 AMP | 12/1.00 | 1N1185 | 150 PIV | 35 AMP | 1.70 |
| | | | | 1N1186 | 200 PIV | 35 AMP | 1.80 |
| | | | | 1N1188 | 400 PIV | 35 AMP | 3.00 |

# SCR AND FW BRIDGE RECTIFIERS

| | | | |
|---|---|---|---|
| C36D | 15A @ 400V | SCR(2N1849) | $1.95 |
| C38M | 35A @ 600V | SCR | 1.95 |
| 2N2328 | 1.6A @ 300V | SCR | .50 |
| MDA 980-1 | 12A @ 50V | FW BRIDGE REC. | 1.95 |
| MDA 980-3 | 12A @ 200V | FW BRIDGE REC. | 1.95 |

# TRANSISTORS

| | | | | | |
|---|---|---|---|---|---|
| C106B1 | .50 | | | 2N3904 | 4/1.00 |
| MPSA05 | .30 | 2N3055 | .89 | 2N3905 | 4/1.00 |
| MPSA06 | 5/1.00 | MJE3055 | 1.00 | 2N3906 | 4/1.00 |
| TIS97 | 6/1.00 | 2N3392 | 5/1.00 | 2N4013 | 3/1.00 |
| TIS98 | 6/1.00 | 2N3398 | 5/1.00 | 2N4123 | 6/1.00 |
| 40409 | 1.75 | PN3567 | 3/1.00 | PN4249 | 4/1.00 |
| 40410 | 1.75 | PN3568 | 4/1.00 | PN4250 | 4/1.00 |
| 40673 | 1.75 | PN3569 | 4/1.00 | 2N4400 | 4/1.00 |
| 2N918 | 4/1.00 | MPS3638A | 5/1.00 | 2N4401 | 4/1.00 |
| 2N2219A | 2/1.00 | MPS3702 | 5/1.00 | 2N4402 | 4/1.00 |
| 2N2221A | 4/1.00 | 2N3704 | 5/1.00 | 2N4403 | 4/1.00 |
| 2N2222A | 5/1.00 | MPS3704 | 5/1.00 | 2N4409 | 5/1.00 |
| PN2222 Plastic | 7/1.00 | 2N3705 | 5/1.00 | 2N5086 | 4/1.00 |
| 2N2369 | 5/1.00 | MPS3705 | 5/1.00 | 2N5087 | 4/1.00 |
| 2N2369A | 4/1.00 | 2N3706 | 5/1.00 | 2N5088 | 4/1.00 |
| MPS2369 | 5/1.00 | MPS3706 | 5/1.00 | 2N5089 | 4/1.00 |
| 2N2484 | 4/1.00 | 2N3707 | 5/1.00 | 2N5129 | 5/1.00 |
| 2N2906 | 4/1.00 | 2N3711 | 5/1.00 | PN5134 | 5/1.00 |
| 2N2907 | 5/1.00 | 2N3724A | .65 | PN5138 | 5/1.00 |
| PN2907 Plastic | 7/1.00 | 2N3725A | 1.00 | 2N5139 | 5/1.00 |
| 2N2925 | 5/1.00 | 2N3772 | 2.25 | 2N5210 | 5/1.00 |
| MJE2955 | 1.25 | 2N3823 | 1.00 | 2N5449 | 3/1.00 |
| 2N3053 | 2/1.00 | 2N3903 | 5/1.00 | 2N5951 | 3/1.00 |

# CAPACITOR CORNER

50 VOLT CERAMIC DISC CAPACITORS

| | 1-9 | 10-99 | 100- | | 1-9 | 10-99 | 100- |
|---|---|---|---|---|---|---|---|
| 10 pf | .05 | .04 | .03 | .001µF | .05 | .04 | .035 |
| 22 pf | .05 | .04 | .03 | .0047µF | .05 | .04 | .035 |
| 47 pf | .05 | .04 | .03 | .01µF | .05 | .04 | .035 |
| 100 pf | .05 | .04 | .03 | .022µF | .06 | .05 | .04 |
| 220 pf | .05 | .04 | .03 | .047µF | .06 | .05 | .04 |
| 470 pf | .05 | .04 | .035 | .1µF | .12 | .09 | .075 |
| **100 VOLT MYLAR FILM CAPACITORS** | | | | | | | |
| .001mf | .12 | .10 | .07 | .022mf | .13 | .11 | .08 |
| .0022 | .12 | .10 | .07 | .047mf | .21 | .17 | .13 |
| .0047mf | .12 | .10 | .07 | .1mf | .27 | .23 | .17 |
| .01mf | .12 | .10 | .07 | .22mf | .33 | .27 | .22 |
| **+20% DIPPED TANTALUMS (SOLID) CAPACITORS** | | | | | | | |
| .1/35V | .28 | .23 | .17 | 1.5/35V | .30 | .26 | .21 |
| .15/35V | .28 | .23 | .17 | 2.2/25V | .31 | .27 | .22 |
| .22/35V | .28 | .23 | .17 | 3.3/25V | .31 | .27 | .22 |
| .33/35V | .28 | .23 | .17 | 4.7/25V | .32 | .28 | .23 |
| .47/35V | .28 | .23 | .17 | 6.8/25V | .36 | .31 | .25 |
| .68/35V | .28 | .23 | .17 | 10/25V | .40 | .35 | .29 |
| 1.0/35V | .28 | .23 | .17 | 15/25V | .63 | .50 | .40 |
| **MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS** | | | | | | | |
| Axial Lead | | | | Radial Lead | | | |
| .47/50V | .15 | .13 | .10 | .47/25V | .15 | .13 | .10 |
| 1.0/50V | .16 | .14 | .11 | .47/50V | .16 | .14 | .11 |
| 3.3/50V | .14 | .12 | .09 | 1.0/16V | .15 | .13 | .10 |
| 4.7/25V | .15 | .13 | .10 | 1.0/25V | .16 | .14 | .11 |
| 10/25V | .15 | .13 | .10 | 1.0/50V | .16 | .14 | .11 |
| 10/50V | .16 | .14 | .12 | 4.7/16V | .15 | .13 | .10 |
| 22/25V | .17 | .15 | .12 | 4.7/25V | .15 | .13 | .10 |
| 22/50V | .24 | .20 | .18 | 4.7/50V | .16 | .14 | .11 |
| 47/25V | .19 | .17 | .15 | 10/16V | .14 | .12 | .09 |
| 47/50V | .25 | .21 | .19 | 10/25V | .15 | .13 | .10 |
| 100/25V | .24 | .20 | .18 | 10/50V | .16 | .14 | .12 |
| 100/50V | .35 | .30 | .28 | 47/50V | .24 | .21 | .19 |
| 220/25V | .32 | .28 | .25 | 100/16V | .19 | .15 | .14 |
| 220/50V | .45 | .41 | .38 | 100/25V | .24 | .20 | .18 |
| 470/25V | .33 | .29 | .27 | 100/50V | .35 | .30 | .28 |
| 1000/16V | .55 | .50 | .45 | 220/16V | .23 | .17 | .16 |
| 2200/16V | .70 | .62 | .55 | 470/25V | .31 | .28 | .26 |

Circle 200 on inquiry card.

## save more than 20%!
## NORTH STAR——XITAN INTERTUBE

**the smartest computers at the smartest price**
**DOUBLE DENSITY**

HORIZON-1-16K kit, list $1599 ...$1279
Assembled & tested, list $1899 ...$1519
HORIZON-2-32K kit, list $2249 ...$1799
Assembled & tested, list $2549 ...$2039
PASCAL for NORTH STAR on Disk .......$49
Powerful NORTH STAR BASIC.........FREE

XITAN computers--NOW--With QUAD DENSITY DISKS!
Famous XITAN Software & BASIC......FREE
HORIZON & XITAN are S-100, Z-80--THE BEST
INTERTUBE II Terminal, list $995 .....$780

NEW: Our VIDEO BOARD CODE on Disk!
Now you can run our computers on a TV!
SAVE HUNDREDS $$$$

TV Code on DISK...$10 LISTING...FREE
Business Software, Terminals, Printers,
Computers in stock & special-ordered
Other brands at good discounts. Ask!
Which Computers are best? BROCHURE...FREE

**AMERICAN SQUARE COMPUTERS**
KIVETT DR, JAMESTOWN NC 27282
(919) 883-1105

Circle 5 on inquiry card.

## MAXIMUM VALUE FOR YOUR DOLLAR

**NORTH STAR COMPUTER PRODUCTS**

HORIZON1 16K KIT........$1275.00
16K RAM BOARD KIT ......$ 250.00
32K RAM BOARD KIT ......$ 475.00

**VERBATIM DISCS FOR NORTH STAR**
BOX OF 10 .........$29. POST PAID

**COMPLETE SYSTEMS AVAILABLE**
CUSTOM SOFTWARE FOR NORTH STAR SYSTEMS

**CASIO CALCULATORS AT DISCOUNT PRICES**

MANY OTHER SUPER VALUES
WRITE OR CALL:

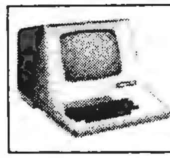
**A.E.I.**
3851 HACKETT AVE.
LONG BEACH, CALIF. 90808
(213) 421-4815 (213) 429-0535

Circle 21 on inquiry card.

## HAZELTINE 1400

only

**$649.95!**



- Verbatim Mini Diskettes ... ... $3.70 each (boxes of 10)
- Intertube . . $784.00
- TRS-80 16K Level II Expansion Kit . . . . . $89.95
- Centronics 779 tractor . . . . . . . . . . . . . . . . . $1050.00
- Horizon II ass. . . . $1999.00

VISA

Mail Order Only.

TORA SYSTEM INC.
29-02 23rd Avenue
Astoria NY 11105
(212) 932-3533

## TRS-80 16K MEMORY EXPANSION KIT

INCLUDES 8 TESTED & GUARANTEED M5K 4116 3 16K RAMS. PROGRAMMING PLUGS & EASY-TO-FOLLOW INSTRUCTIONS.
$72.00 PER KIT

## 6800 64K BYTE RAM SET AND CONTROLLER CHIP SET

*MAKE 64K BYTES OF MEMORY FOR YOUR 6800. THE CHIP SETS INCLUDE:*
32 M5K 4116-3 16K RAMS.
1 MC3480L MEMORY CONTROLLER.
1 MC3242AP MEMORY ADDRESS MULTIPLEXER/COUNTER.
DATA & APPLICATION SHEETS
PARTS TESTED & GUARANTEED.

$295.00 PER COMPLETE SET.

**16K DYNAMIC RAMS**
M5K 4116-3 200NSEC ACCESS TIME/375NSEC CYCLE TIME. TESTED & BURNED-IN
$8.50 EACH/MIN. QTY. 8

**4K STATIC RAMS**
EQUIV. TO TMS40L44-30 300NSEC ACCESS TIME/CYCLE TIME FOR 4MHZ Z-80 OPERATION.
$7.50 EACH/MIN. QTY. 8
TESTED & GUARANTEED

*QUANTITY DISCOUNTS AVAILABLE*
ALL ORDERS POSTPAID. U.S. FUNDS. CHECK OR MONEY ORDER. VISA, BA, MASTERCHARGE — SEND ACCOUNT NO., EXPIRATION DATE, INTERBANK NO., & SIGNED ORDER.
PHONE ORDERS: 714/633-4460

MEASUREMENT SYSTEMS & CONTROLS, INC.
MEMORY DEVICES DIVISION, DEPT. B3
867 NORTH MAIN ST., ORANGE, CA 92668

Circle 216 on inquiry card.

## TRS - 80

**SPECIAL PROMOTION SALE**

**SAVE** 10%, 15% or more on **ALL** Computers, Peripherals, Software, and **ALL** other fine Radio Shack® products.

**NO TAXES** on out-of-state shipments.

**FREE** Surface delivery in U.S.

**WARRANTIES** will be honored by your local Radio Shack® store.

Offered exclusively by
**Radio Shack® Authorized Sales Center**
1117 Conway
Mission, Texas 78572
**(512) 585-2765**

VISA

Circle 314 on inquiry card.

## SURPLUS ELECTRONICS

ASCII

ASCII

®

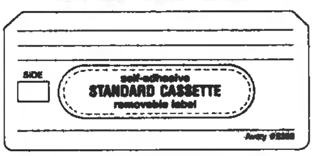
**IBM SELECTRIC BASED I/O TERMINAL WITH ASCII CONVERSION INSTALLED $645.00**

- 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.**
130 NORTHEASTERN BLVD.
NASHUA, N.H. 03060
Phone orders accepted using VISA or MC. Toll Free 1-800-258-1036
In N.H. 603-889-7661

Circle 391 on inquiry card.

## CASSETTE AND FLOPPY DISC LABELS.




Avery offers a complete line of labels for cassettes and floppy discs—all with removable adhesive. Including these standard sizes:

Cassette Label (1⅝" x 3½") #5250
Floppy Disc Label (1¾" x 5½") #5252

Write for more information and free samples to:

Avery Label
777 East Foothill Blvd.
Azusa, CA 91702

**Avery Label**
An Avery International Company

Circle 20 on inquiry card.

## Scotch® BRAND Diskettes

Order Direct:
2 for $11.75
5 for $26.45
10 for $49.75 (Hard Box)

**Price Includes Shipping**
(Hawaii, Alaska & Canada add $1.00 shipping)
VISA, Master Charge, Check, Money Order

740-0 - 8" Soft Sector
740-32 - 8" Hard (32) Sector
744-0 - 5" Soft Sector
744-10 - 5" Hard (10) Sector
744-16 - 5" Hard (16) Sector

*Mass. residents add 5% sales tax.*

**cpu** COMPUTER PACKAGES UNLIMITED
99 Reservoir St. Holden, MA 01520
(617) 829-2570 Div. of SCB Inc.

## ••CRT INTERFACES••
## black • white/color

Monitors * Combination Rcvr/monitor sets * Modulator kits * B-W Cameras * Color Cameras * Audio Subcarrier kits * Parts

**WRITE or PHONE for DETAILS & PRICING.**

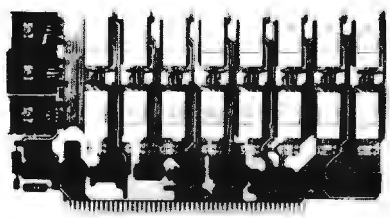
DIAL: 402-987-3771

Dealers welcomed. Well established program.

13-B Broadway — **ATV Research** — Dakota City, NE. 68731

Circle 19 on inquiry card.

## 16K EPROM CARD-S 100 BUSS



**$59.95 KIT**

**OUR BEST SELLING KIT!**

**USES 2708's!**

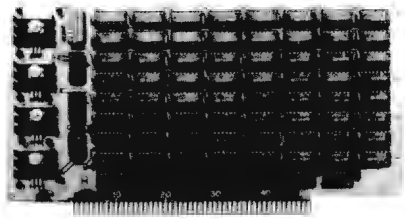
Thousands of personal and business systems around the world use this board with complete satisfaction. Puts 16K of software on line at **ALL TIMES!** Kit features a top quality soldermasked and silk-screened PC board and first run parts and sockets. All parts (except 2708's) are included. Any number of EPROM locations may be disabled to avoid any memory conflicts. Fully buffered and has WAIT STATE capabilities.

**OUR 450NS 2708'S ARE $8.95 EA. WITH PURCHASE OF KIT**

**ASSEMBLED AND FULLY TESTED ADD $25**

## 8K LOW POWER RAM KIT-S 100 BUSS

## 250 NS SALE!



**ADD $5 FOR 250NS!**

**$129 KIT**

**(450 NS RAMS!)**

Thousands of computer systems rely on this rugged, work horse, RAM board. Designed for error-free, NO HASSLE, systems use.

**KIT FEATURES:**

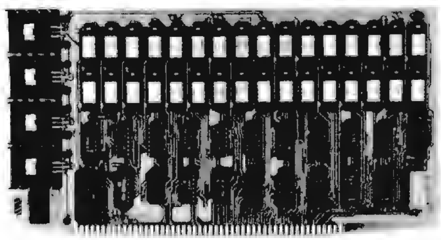
1. Doubled sided PC Board with solder mask and silk screen layout. Gold plated contact fingers.
2. All sockets included.
3. Fully buffered on all address and data lines.
4. Phantom is jumper selectable to pin 67.
5. FOUR 7805 regulators are provided on card.

Blank PC Board w/Documentation **$29.95**
Low Profile Socket Set...**13.50**
Support IC's (TTL & Regulators) **$9.75**
Bypass CAP's (Disc & Tantalums) **$4.50**

**ASSEMBLED AND FULLY BURNED IN ADD $30**

## 16K STATIC RAM KIT-S 100 BUSS

**$295 KIT**



**FULLY STATIC, AT DYNAMIC PRICES**

**WHY THE 2114 RAM CHIP?**

We feel the 2114 will be the next industry standard RAM chip (like the 2102 was). This means price, availability, and quality will all be good! Next, the 2114 is FULLY STATIC! We feel this is the **ONLY** way to go on the S-100 Buss! We've all heard the HORROR stories about some Dynamic Ram Boards having trouble with DMA and FLOPPY DISC DRIVES. Who needs these kinds of problems? And finally, even among other 4K Static RAM's the 2114 stands out! Not all 4K static Rams are created equal! Some of the other 4K's have **clocked** chip enable lines and various timing windows **just as critical** as Dynamic RAM's. Some of our competitor's 16K boards use these "tricky" devices. But not us! The 2114 is the ONLY logical choice for a trouble-free, straightforward design.

KIT FEATURES:

1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 2 amps TYPICAL from the +8 Volt Buss
10. Blank PC Board can be populated as any multiple of 4K

BLANK PC BOARD W/DATA—**$33**
LOW PROFILE SOCKET SET—**$12**
SUPPORT IC'S & CAPS—**$19.95**
ASSEMBLED & TESTED—**ADD $30**
2114 RAM'S—**8 FOR $69.95**

## TM990 BUSS PROTOTYPE & WIREWRAP BOARD

For use with the Texas Instrument Series of 16 Bit Micro-computer Module. Fully buss compatible. An inexpensive and quick way to expand the capacity of your TI computer. Made of G-10 Epoxy PC material. Gold plated contact finger, all plated through holes. High density, up to over 100 DIP's. Fully documented.

**$70 each** **(OEM Discounts Available)**

## 16K DYNAMIC RAM CHIP

16K X 1 Bits. 16 Pin Package. Same as Mostek 4116-4. 250 NS access. 410 NS cycle time. Our best price yet for this state of the art RAM. 32K and 64K RAM boards using this chip are readily available. These are new, fully guaranteed devices by a major mfg. **VERY LIMITED STOCK!**

**8 FOR $89.95**

**NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.**

## 450 NS! 2708 EPROMS

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.~~ **$9.95** ~~4 FOR $50.00~~

**PRICE CUT**

**NATIONAL SEMICONDUCTOR** ***NEW!***

## CAR CLOCK MODULE - #MA6008

**$6.99 each**

Originally used by HYGAIN to indicate time and channel on an expensive C.B. Mini size, self contained module. Not a Kit. Four digits plus flashing indicator for seconds. Includes MM5369 and 3.58 MHZ crystal for super accurate time base. With hookup data.

**INCLUDES CRYSTAL TIMEBASE! WORKS ON 12 VDC!**

**MFGR's CLOSEOUT LIMITED QTY.**

## EXPERIMENTER'S HEATING PLATE

Large Manufacturers Surplus. 5¼x10½ in. Made of 3/8 in. tempered glass with heating element laminated on back. Works off 120 VAC. Protected by thermostat and two thermal fuses. Rated 120 Watts. Use for any heating applications. Perfect for heating ferric chloride to increase PC Board etching efficiency. Units are brand new, non-submersible.

**WHILE THEY LAST—$2.99 each**

## MALLORY COMPUTER GRADE CAPACITOR

30,000 MFD 15 WVDC
Small: 3 x 2 Inches
**$1.99 ea. 3 For $4.99**

**New! REAL TIME Computer Clock Chip**

N.S. MM5313. Features BOTH 7 segment and BCD outputs. 28 Pin DIP. **$4.95 with Data**

## Z-80 PROGRAMMING MANUAL

By MOSTEK, or ZILOG. The most detailed explanation ever on the working 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. **$12.95**

**GENERAL INSTRUMENT**

FULL WAVE BRIDGE
4 AMP 600 PIV
LOOK!
3/4 IN. SQUARE - WITH LUGS

**75¢ ea. 3 FOR $2** #LM-1

**"THE COLOSSUS"**

**FAIRCHILD SUPER JUMBO LED READOUT**

A full .80 inch character. The biggest readout we have ever sold! Super efficient. Compare at up to $2.95 each from others! YOUR CHOICE

FND 843 Common Anode
FND 850 Common Cathode **$1.49 ea** **(6 for $6.95)**


# Digital Research Corporation
**(OF TEXAS)**

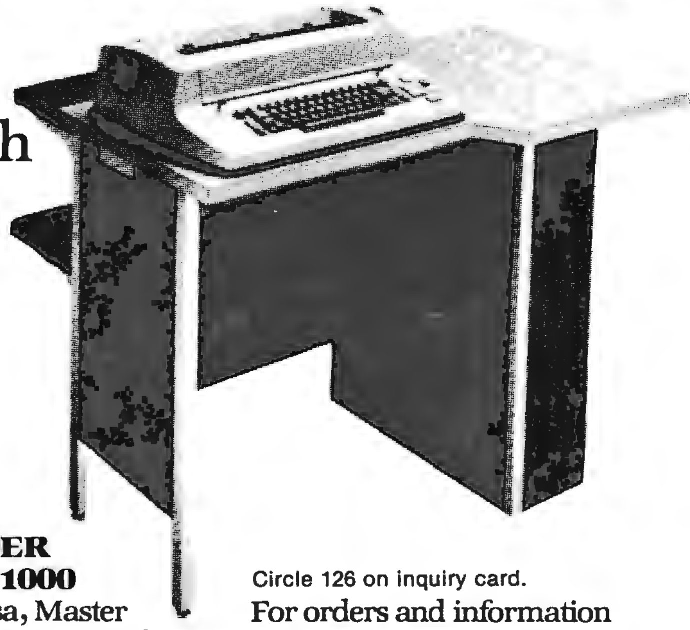
**P.O. BOX 401247Y GARLAND, TEXAS 75040 • (214) 271-2461**


**TERMS:** Add 30¢ postage, we pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa, MasterCharge, and American Express cards. Tex. Res. add 5% Tax. Foreign orders (except Canada add 20% P & H. 90 Day Money Back Guarantee on all items.

Circle 100 on inquiry card.

# The DATA-TRANS 1000

## A completely refurbished **IBM** Selectric Terminal with built-in **ASCII** Interface.



# $1395

**Features:**

- 300 Baud
- 14.9 characters per second printout
- Reliable heavy duty Selectric mechanism
- RS-232C Interface
- Documentation included
- 60 day warranty - parts and labor
- High quality Selectric printing Off-line use as typewriter
- Optional tractor feed available
- 15 inch carriage width

**HOW TO ORDER DATA-TRANS 1000**

1. We accept Visa, Master Charge. Make cashiers checks or personal check payable to:

**DATA-TRANS**

2. All orders are shipped F.O.B. San Jose, CA
3. Deliveries are immediate

Circle 126 on inquiry card.

For orders and information

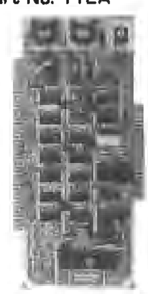
**DATA-TRANS**
2154 O'Toole St.
Unit E
San Jose, CA 95131
Phone: (408) 263-9246

**ASCII KEYBOARD** . . . . . . . . Mounted to DECWRITER Panel

By Cherry Products . . . ASSEMBLED 89.95

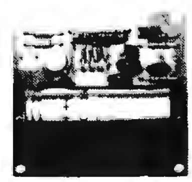
**TIDMA**✲

- Tape Interface Direct Memory Access • Record and play programs without bootstrap loader (no prom) has FSK encoder/decoder for direct connections to low cost recorder at 1200 baud rate, and direct connections for inputs and outputs to a digital recorder at any baud rate • S-100 bus compatible • Board only $35.00 Part No. 112, with parts $110 Part No. 112A

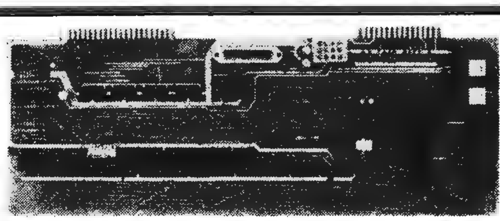


**T.V. INTERFACE**

- Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple • Power required is 12 volts AC C.T., or +5 volts DC • Board only $7.60 part No. 107, with parts $13.50 Part No. 107A



(Illegal where prohibited by law.)



**ASCII to Correspondence code converter**

This bidirectional board is a direct replacement for the board inside the Trendata 1000 terminal. The on board connector provides RS-232 serial in and out. Sold only as an assembled and tested unit for $330.00. Part No. TA 1000C

apple II

With 16K $1024.00

**To Order:** Mention part number, description, and price. In USA, shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Outside USA add 10% for air mail postage and handling, no C.O.D.'s. Checks and money orders must be payable in US dollars. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. Prices are in US dollars. No open accounts. To eliminate tariff in Canada boxes are marked "Computer Parts." Dealer inquiries invited. 24 Hour Order Line: (408) 226-4064

✲ Circuits designed by John Bell

For free catalog including parts lists and schematics, send a self-addressed stamped envelope.

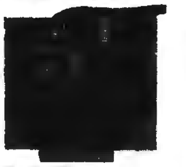
**ELECTRONIC SYSTEMS** Dept. B, P. O. Box 21638, San Jose, CA USA 95151



Circle 125 on inquiry card.

## TRS-80 E.S. SERIAL I/O

• Can input into basic • Can use LLIST and LPRINT to output, or output continuously • RS-232 compatible • Can be used with or without the expansion bus • On board switch selectable baud rates of 110, 150, 300, 600, 1200, 2400, parity or no parity odd or even, 5 to 8 data bits, and 1 or 2 stop bits. D.T.R. line • Requires +5, -12 VDC • Board only $19.95 Part No. 8010, with parts $59.95 Part No. 8010A, assembled $79.95 Part No. 8010 C. No connectors provided, see below.



EIA/RS-232 connector Part No. DB25P $6.00, with 8', 8 conductor cable $10.95 Part No. DB25P8.

3' ribbon cable with attached connectors to fit TRS-80 and our serial board $19.95 Part No. 3CAB40.

## MODEM✲

• Type 103 • Full or half duplex • Works up to 300 baud • Originate or Answer • No coils, only low cost components • TTL input and output-serial • Connect 8 Ω speaker and crystal mic. directly to board • Uses XR FSK demodulator • Requires +5 volts • Board only $7.60 Part No. 109, with parts $27.50 Part No. 109A

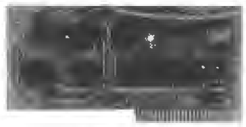
## DISKETTES

BOX OF 10



5" $29.95
8" $39.95

## APPLE II✲ SERIAL I/O INTERFACE



Baud rate is continuously adjustable from 0 to 30,000 • Plugs into any peripheral connector • Low current drain. RS-232 input and output • On board switch selectable 5 to 8 data bits, 1 or 2 stop bits, and parity or no parity either odd or even • Jumper selectable address • SOFTWARE • Input and Output routine from monitor or BASIC to teletype or other serial printer • Program for using an Apple II for a video or an intelligent terminal. Also can output in correspondence code to interface with some selectrics. • Also watches DTR • Board only $15.00 Part No. 2, with parts $42.00 Part No. 2A, assembled $62.00 Part No. 2C

## 8K EPROM PIICEON

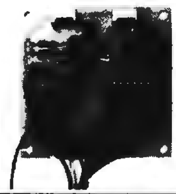
Saves programs on PROM permanently (until erased via UV light) up to 8K bytes. Programs may be directly run from the program saver such as fixed routines or assemblers. • S-100 bus compatible • Room for 8K bytes of EPROM non-volatile memory (2708's). • Onboard PROM programming • Address relocation of each 4K of memory to any 4K boundary within 64K • Power on jump and reset jump option for "turnkey" systems and computers without a front panel • Program saver software available • Solder mask both sides • Full silkscreen for easy assembly. Program saver software in 1 2708 EPROM $25. Bare board $35 including custom coil, board with parts but no EPROMS $139, with 4 EPROMS $179, with 8 EPROMS $219.

## T.V. TYPEWRITER

• Stand alone TVT • 32 char/line, 16 lines, modifications for 64 char/line included • Parallel ASCII (TTL) input • Video output • 1K on board memory • Output for computer controlled curser • Auto scroll • Non-destructive curser • Curser inputs: up, down, left, right, home, EOL, EOS • Scroll up, down • Requires +5 volts at 1.5 amps, and -12 volts at 30 mA • All 7400, TTL chips • Char. gen. 2513 • Upper case only • Board only $39.00 Part No. 106, with parts $145.00 Part No. 106A

## TAPE ✲ INTERFACE

• Play and record Kansas City Standard tapes • Converts a low cost tape recorder to a digital recorder • Works up to 1200 baud • Digital in and out are TTL-serial • Output of board connects to mic. in of recorder • Earphone of recorder connects to input on board • No coils • Requires +5 volts, low power drain • Board only $7.60 Part No. 111, with parts $27.50 Part No. 111A



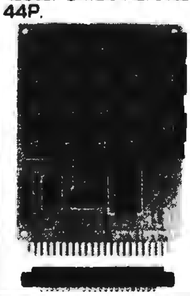
## UART & BAUD RATE GENERATOR✲

• Converts serial to parallel and parallel to serial • Low cost on board baud rate generator • Baud rates: 110, 150, 300, 600, 1200, and 2400 • Low power drain +5 volts and -12 volts required • TTL compatible • All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity. • All connections go to a 44 pin gold plated edge connector • Board only $12.00 Part No. 101, with parts $35.00 Part No. 101A, 44 pin edge connector $4.00 Part No. 44P

## HEX ENCODED KEYBOARD

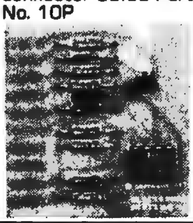
E.S.

This HEX keyboard has 19 keys, 16 encoded with 3 user definable. The encoded TTL outputs, 8-4-2-1 and STROBE are debounced and available in true and complement form. Four onboard LEDs indicate the HEX code generated for each key depression. The board requires a single +5 volt supply. Board only $15.00 Part No. HEX-3, with parts $49.95 Part No. HEX-3A. 44 pin edge connector $4.00 Part No. 44P.



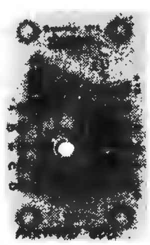
## RS-232/ TTL✲ INTERFACE

• Converts TTL to RS-232, and converts RS-232 to TTL • Two separate circuits • Requires -12 and +12 volts • All connections go to a 10 pin gold plated edge connector • Board only $4.50 Part No. 232, with parts $7.00 Part No. 232A 10 Pin edge connector $3.00 Part No. 10P



## RS-232/ TTY✲ INTERFACE

• Converts RS-232 to 20mA current loop, and 20mA current loop to RS-232 • Two separate circuits • Requires +12 and -12 volts • Board only $4.50 Part No. 600, with parts $7.00 Part No. 600A



## WAMECO PRODUCTS WITH ELECTRONIC SYSTEMS PARTS

**FDC-1** FLOPPY CONTROLLER BOARD will drive shugart, pertek, remex 5" & 8" drives up to 8 drives, on board PROM with power boot up, will operate with CPM (not included). PCBD ........ $42.95

**FPB-1** Front Panel. (Finally) IMSAI size hex displays. Byte or instruction single step. PCBD ........ $42.95

**MEM-1A** 8Kx8 fully buffered, S-100, uses 2102 type RAMS. PCBD ........ $24.95, $168 Kit

**QMB-12** MOTHER BOARD, 13 slot, terminated, S-100 board only ....... $34.95 / $89.95 Kit

**CPU-1** 8080A Processor board S-100 with 8 level vector interrupt PCBD .. $25.95 / $89.95 Kit

**RTC-1** Realtime clock board. Two independent interrupts. Software programmable. PCBD ........ $25.95, $60.95 Kit

**EPM-1** 1702A 4K EPROM card PCBD ........ $25.95 / $49.95 with parts less EPROMS

**EPM-2** 2708/2716 16K/32K EPROM card PCBD ........ $24.95 / $49.95 with parts less EPROMS

**QMB-9** MOTHER BOARD. Short Version of QMB-12. 9 Slots PCBD ....... $30.95 / $67.95 Kit

**MEM-2** 16Kx8 Fully Buffered 2114 Board PCBD ............ $25.95, $269.95 Kit

## S-100 BUS ✲ ACTIVE TERMINATOR

Board only $14.95 Part No. 900, with parts $24.95 Part No. 900A

## DC POWER SUPPLY✲

• Board supplies a regulated +5 volts at 3 amps., +12, -12, and -5 volts at 1 amp. • Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps. • Board only $12.50 Part No. 6085, with parts excluding transformers $42.50 Part No. 6085A

**To Order:** Mention part number, description, and price. In USA, shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Outside USA add 10% for air mail postage and handling, no C.O.D.'s. Checks and money orders must be payable in US dollars. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. Prices are in US dollars. No open accounts. To eliminate tariff in Canada boxes are marked "Computer Parts." Dealer inquiries invited. 24 Hour Order Line: (408) 226-4064

✲ Circuits designed by John Bell

For free catalog including parts lists and schematics, send a self-addressed stamped envelope.

**ELECTRONIC SYSTEMS** Dept. B, P. O. Box 21638, San Jose, CA USA 95151

# a PERCOM SAMPLER


**For your SS-50 bus computer — the CIS-30+**

- Interface to data terminal and *two* cassette recorders with a unit only 1/10 the size of SWTP's AC-30.
- Select 30, 60, or 120 bytes per second cassette interfacing, 300, 600 or 1200 baud data terminal interfacing.
- Optional mod kits make CIS-30+ work with *any* microcomputer. (For MITS 680b, ask for Tech Memo TM-CIS-30+—09.)
- KC-Standard/Bi-Phase-M (double frequency) cassette data encoding. Dependable self-clocking operation.
- Ordinary functions may be accomplished with 6800 Mikbug™ monitor.
- Prices: Kit, $79.95; Assembled, $99.95.

Prices include a comprehensive instruction manual. Also available: Test Cassette, Remote Control Kit (for program control of recorders), IC Socket Kit, MITS 680b mod documentation, Universal Adaptor Kit (converts CIS-30+ for use with any computer). MIKBUG® Motorola, Inc.

**In the Product Development Queue . . .**

**Coming PDQ. Watch for announcements.**

**6809 Processor Card** — With this SS-50 bus PC board, you'll be able to upgrade with the microprocessor that Motorola designers describe as the "best 8-bit machine so far made by humans."

**The Electric Crayon™** — This color graphics system includes its own $\mu$P and interfaces to virtually any microcomputer with a parallel I/O port.

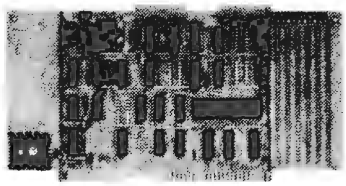
**Printer Interface** — For your TRS-80™. Interface any serial RS232 printer to your TRS-80™ with this system.

™ELECTRIC WINDOW, ELECTRIC CRAYON, Pilon-30 and Pilon-10 are trademarks of Percom Data Company, Inc.
TRS-80 is a trademark of Tandy Corporation and Radio Shack which has no relationship to Percom Data Company.

**Orders may be paid by check or money order, or charged to Visa or Master Charge credit account. Texas residents must add 5% sales tax.**


**For your data storage — Pilon-30™ and Pilon-10™ data cassettes**

- Orders-of-magnitude improvement in data integrity over ordinary audio cassettes.
- Pilon-coated pressure pad eliminates lint-producing felt pad of standard audio cassettes.
- Smooth pilon coating minimizes erratic tape motion.
- Foam pad spring is energy absorbing. Superior to leaf spring mounted pad which tends to oscillate and cause flutter.
- Five-screw case design virtually precludes deformation during assembly.
- Price: $2.49.



**For your S-100 computer — the CI-812**

- Both cassette and data terminal interfacing on one S-100 bus PC board.
- Interfaces *two* recorders. Record and playback circuits are independent.
- Select 30, 60, 120, or 240 bytes per second cassette interfacing, 110 to 9600 baud data terminal interfacing.
- KC-Standard/Bi-Phase-M (double frequency) encoded cassette data. Dependable self-clocking operation.
- Optional firmware (2708 EPROM) Operating System available.
- Prices: kit, $99.95; assembled, $129.95.

Prices include a comprehensive instruction manual. In addition to the EPROM Operating System, a Test Cassette, Remote Control Kit (for program control of recorders), and an IC Socket Kit are also available.

## CASSETTE SOFTWARE

**For 8080/Z-80 $\mu$Cs . . .**

**BASIC ETC** — Developed by the co-authors of the original Tiny BASIC, BASIC ETC is easy to use yet includes commands and functions required for powerful business and scientific programs as well as for hobby applications. 9.5K bytes of RAM. 1200-baud cassette and 42-page user's manual .............. $35.00

**Cassette Operating System** — EPROM (2708) COS for the Percom CI-812 dual peripheral interfacing PC card .. $39.95

If you're programming on a 6800 $\mu$C, you'll want these development and debugging programs written by Ed Smith of the Software Works:

**Disassembler/Source Generator** — Disassembles SWTP Resident Assembler, TSC Mnemonic Assembler/Text Editor or Smoke Signal Mnemonic Assembler/Text Editor and produces compacted source code suitable for re-editing. Prints or displays full assembly-type output listing. 4K bytes of RAM.
(Order M68SG) ............. $25.00

**Disassembler/Trace** — Use to examine (or examine and execute) any area of RAM or ROM. "Software-single-step" through any program, change the contents of CPU or memory location at any time, trace subroutines to any depth. 2.3K bytes of RAM.
(Order M68DT) ............. $20.00

**Support Relocator Program** — Supplied on EPROM, this program relocates a program in any contiguous area of RAM or ROM to anywhere in RAM. Use to assemble and test programs in RAM, adjust programs for EPROM operating addresses and then block move to your EPROM burner address. 952 bytes of RAM. Loads at hex 1000.
(Order M68EP) ............. $20.00

**Relocating Assembler & Linking Loader** (M68AS) .................. $50.00

**Relocating Disassembler & Segmented Source Text Generator** (M68RS) $35.00

**Americana Plus** — 14 tunes for the Newtech Model 68 Music Board in machine language ready to load and run. Cassette compatible with Percom CIS-30+ and SWTP AC-30. Order MC-1SW .. $15.95

## HARDWARE

**Newtech Model 68 Music Board** — Produces melodies, rhythms, sound effects, morse code, etc. from *your* programs. Includes manual with BASIC for writing music scores and assembly language routine to play them. Installs in SWTP I/O slot. Assembled & tested ..... $59.95

**The Percom ELECTRIC WINDOW™** — Memory-resident and programmable, this video display character generator board for your SS-50 bus displays up to 24 80-character lines. Features dual character generators, dual-intensity high-lighting. One programmable register controls scrolling. Compatible with standard video monitors ..... $249.95

**SS-50 Prototype Cards:**
Large card (up to 70 40-pin ICs) $24.95
I/O size card ............... $14.95

PERCOM
PERCOM DATA COMPANY, INC.
DEPT. B
211 N. KIRBY • GARLAND, TX. 75042

PERCOM™ *'peripherals for personal computing'*

**To order products or request additional literature, call Percom's toll-free number: 1-800-527-1592. For detail technical information call (214) 272-3421.**

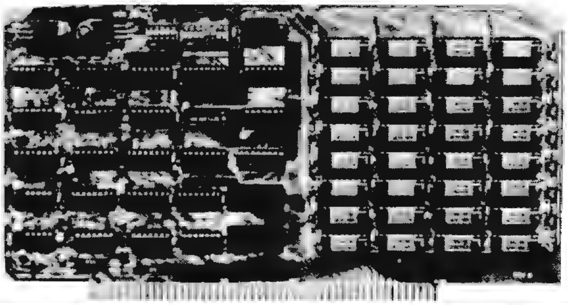
Visit our new retail location!

# SD EXPANDORAM

*The Ultimate S-100 Memory*

The EXPANDORAM is available in versions from 16K up to 64K, so for a minimum investment you can have a memory system that will grow with your needs. This is a dynamic memory with the invisable on-board refresh, and IT WORKS!

- Bank Selectable
- Phantom
- Power 8VDC, ±16VDC, 5 Watts
- Lowest Cost Per Bit
- Uses Popular 4116 RAMS
- PC Board is doubled solder masked and has silk-screen parts layout.

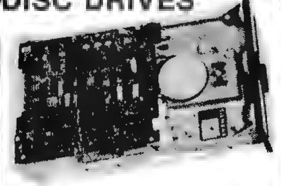


- Extensive documentation clearly written
- Complete Kit includes all Sockets for 64K
- Memory access time: 375ns, Cycle time: 500ns.
- No wait states required.
- 16K boundries and Protection via Dip Switches
- Designed to work with Z-80, 8080, 8085 CPU's.

**EXPANDO 64 KIT (4116)**

| | |
|---|---|
| 16K | $245.00 |
| 32K | $310.00 |
| 48K | $375.00 |
| 64K | $440.00 |

## DISC DRIVES



Sugart SA400 5¼"
with attractive metal case
**$295.00**

*Sugart 801*
with attractive metal case
**$495.00**

*Siemens FDD 200-8 8" double-sided double density*
***$599.00***

## DISC CONTROLLER

**SD "VERSAFLOPPY" Kit**

The Versatile Floppy Disk Controller — Only $159.00

FEATURES: IBM 3740 Soft Sectored Compatible. S-100 BUS Compatible for Z-80 or 8080. Controls up to 4 Drives (single or double sided). Directly controls the following drives:
1. Shugart SA400/450 Mini Floppy
2. Shugart SA800/850 Standard Floppy
3. PERSCI 70 and 277
4. MFE 700/750
5. CDC 9404/9406
6. GSI/Siemans FDD120-8.

34 Pin Connector for Mini Floppy. 50 Pin Connector for Standard Floppy. Operates with modified CP/M operating system and C-Basic Compiler. The new "Versafloppy" from S.D. Computer Products provides complete control for many of the available Floppy Disk Drives. Both Mini and Full Size. FD1771B-1 Single Density Controller Chip. Listings for Control Software are included in price.

## SAVE $100.00

## DM2700S DISK & CABINET with POWER SUPPLY

DM2700S includes Siemans FD120-8" Disk Drive with the following features:

- Single or Double Density
- Hard or Soft Sector
- Door Interlock
- Write Protect
- Hard Sector Detection
- 500 KB/S Transfer
- 800 KB unformated
- Bit density 6536 BP1
- Sugart 800 Series Compatable

Cabinet includes:
- 110V to 125V 60 Hz power supply
- Data Cable
- Fan
- Accepts per SCI, Shugart, Siemans 8" Drives

SHOP & COMPARE!

**DM2700S** Disk Drive & Cabinet **$650.00**
REG. $750 SALE PRICED

**DM2700** Cabinet, less Drive ~~$249.00~~ **$225.00**

CONTINENTAL SPECIALTIES CORPORATION

## Logic Probes and Digital Pulsers

SALE

### LOGIC PROBES

CSC logic probes are the ultimate tool for breadboard design and testing. These hand-held units provide an instant overview of circuit conditions. Simple to use; just clip power leads to circuit's power supply, set logic family switch to TTL/DTL or CMOS/HTL. Touch probe to test node. Trace logic levels and pulses through digital circuits. Even stretch and latch for easy pulse detection. Instant recognition of high, low or invalid levels, open circuits and nodes. Simple, dual-level detector LEDs tell it quickly, correctly. HI (Logic "1"); LO (Logic "0"). Also incorporates blinking pulse detector, e.g., HI and LO LEDs blink on or off, tracking "1" or "0" states at square wave frequencies up to 1.5 MHz. Pulse LED blinks on for ⅓ second during pulse transition. Choice of three models to meet individual requirements; budget, project and speed of logic circuits.

**MODEL LP-1**

Hand-held logic probe provides instant reading of logic levels for TTL, DTL, HTL or CMOS. **Input Impedance:** 100,000 ohms. **Minimum Detectable Pulse:** 50 ns. **Maximum Input Signal (Frequency):** 10 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored.

CSC Model LP-1 Logic Probe—Net Each ........ ~~$44.95~~ **$42.70**

**MODEL LP-2**

Economy version of Model LP-1. Safer than a voltmeter. More accurate than a scope. **Input Impedence:** 300,000 ohms. **Minimum Detectable Pulse:** 300 ns. **Maximum Input Signal (Frequency):** 1.5 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** None.

**CSC Model LP-2 Logic Probe—Net Each** ........ ~~$24.95~~ **$23.70**

**MODEL LP-3**

High speed logic probe. Captures pulses as short as 10 ns. **Input Impedance:** 500,000 ohms. **Minimum Detectable Pulse:** 10 ns. **Maximum Input Signal (Frequency):** 50 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored.

CSC Model LP-3 Logic Probe—Net Each ........ ~~$59.95~~ $56.45

SALE

**DIGITAL PULSER**

The ultimate in speed and ease of operation. Simply connect clip leads to positive and negative power, then touch DP-1's probe to a circuit node; automatic polarity sensor detects circuit's high or low condition. Depress the pushbutton and trigger an opposite polarity pulse into the circuit. Fast troubleshooting includes injecting signals at key points in TTL, DTL, CMOS or other popular circuits. Test with single pulse or 100 pulses per second via built-in dual control push-button; button selects single shot or continuous modes. LED indicator monitors operating modes by flashing once for single pulse or continuously for a pulse train. Completely automatic, pencil-size lab/field pulse generator for any family of digital circuits. **Output:** Tri-state. **Polarity:** Pulse-sensing auto-polarity. **Sync and Source:** 100 mA. **Pulse Train:** 100 pps. **LED Indicator:** Flashes for single pulse; stays lit for pulse train

**CSC Model DP-1 Digital Pulser—Net Each** ........ ~~$74.95~~ **$71.20**

## S100 PLUG BOARDS

Vector

8807

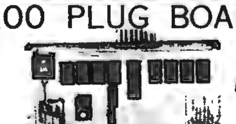
DEC LSI-11, PDP8, PDP11, Heath H-11, P Pattern Epoxy Glass, Plug Board 8.43"x5.187". Dual 36 pin DEC/HEATH Connectors.

| 1-4 | 5-9 | 10-24 |
|---|---|---|
| 19.95 | 17.96 | 15.96 |

8801-1

Plain no etched circuitry except contacts. Produces maximum flexibility.

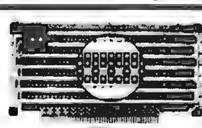
| 1-4 | 5-9 | 10-24 |
|---|---|---|
| 14.95 | 13.46 | 11.96 |



8802-1

Pad per 2 holes. Two-holE pads allow tack soldering of socket, plus second hole for component leads.

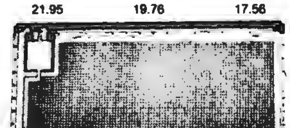
| 1-4 | 5-9 | 10-24 |
|---|---|---|
| 21.95 | 19.76 | 17.56 |



8804

"ANY DIP" has full power and ground planes back to back. Board accommodates 3, 4, 6, 9" Dips.

| 1-4 | 5-9 | 10-24 |
|---|---|---|
| 21.95 | 19.76 | 17.56 |



8801

Individual tinned square pads surround most holes. Ideal for mounting components by "tack soldering" Top of board pod free for mounting I/O connectors.

| 1-4 | 5-9 | 10-24 |
|---|---|---|
| 19.95 | 17.95 | 15.96 |

Vector

4608

Is form and size compatible with INTEL SBL80 Series and NATIONAL BLC 80 Series microcomputer boards. Power and Ground buses on both sides.

| 1-4 | 5-9 | 10-24 |
|---|---|---|
| 45.00 | 40.50 | 36.00 |

**4608-1**

Same as 4608, except plain less power buses.

| 104 | 5-9 | 10-24 |
|---|---|---|
| 34.00 | 30.60 | 27.20 |

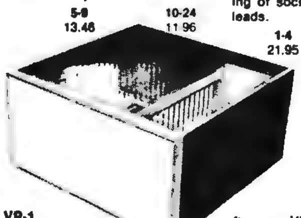
## VECTOR-PAK ASSEMBLED MICROCOMPUTER CASES

Adjustable packaging system for S-100 bus microcomputers, compatible with Altair 8800 and IMSAI 8080 size cards.

- Smart looking, deluxe cases unmarred by unsightly screws or fasteners.
- Finished in dark blue textured vinyl.
- Instantly accessible interiors with slip out covers.
- Removable recessed rear and front panels.
- Fully adjustable interior mounting systems for any card or card spacing within size limitations. No cutting or drilling necessary.
- Perforated bottom cover for cooler operation.

**DESCRIPTION**

Assembled case with perforated bottom cover. Installed mounting struts for card guides and receptacles or mother board. Cards top loaded, spanning front to back. Card guide (12 pair) and chassis plate supplied uninstalled.



VP-2

Vector

**VP-1**
**$163.00**
Shipping Weight 25 lbs.

**VP-2**
**$159.00**
Same as VP1 except cards oriented side to side.

**PRIORITY ONE ELECTRONICS** ©

16723B Roscoe Blvd, 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. Prepaid U.S. orders less than $75.00 include 5% shipping and handling, minimum $2.50. Excess refunded. Just in case . . . please include your phone no.
Prices subject to change without notice.
We will do our best to maintain prices thru June 1979.

**phone orders welcome** (213) 894-8171, (800) 423-5633

**OEM and Institutional Inquiries invited.**

ORDER TODAY 1-800-423-5633

Visit our new retail location!



Visit our new retail location!

Circle 312 on inquiry card.

HICKOK LX303 $74.95
HICKOK LX303 $74.95
HICKOK LX303 $74.95
HICKOK LX303 $74.95
HICKOK LX303 $74.95
HICKOK LX303 $74.95

We have 5V 2716's in Stock. We have the Best Prices on 2102's, 2114's, 4116's.

We have the Best Prices on 2102's, 2114's, 4116's. We make Custom Ribbon Cable Assemblies while you wait!

We make Custom Ribbon Cable Assemblies while you wait! We have the Best Prices on 2102's, 2114'2, 4116's.

## S-100 BUS EDGE CONNECTORS

**S100WWG** 50/100 Cont. .125 ctrs. 3 LEVEL WIRE WRAP .025 sq. posts on .250 spaced rows GOLD PLATED

| 1-4 | 5-9 | 10-24 |
|---|---|---|
| $4.00 | $3.75 | $3.50 |

**S100ALT** 50/100 Cont. .125 ctrs DIP SOLDER TAIL on .140 spaced rows for ALTAIR motherboards. GOLD plated.

| 1-4 | 5-9 | 10-24 |
|---|---|---|
| $4.00 | $3.75 | $3.50 |

**S100-STG** 50/100 Cont. .125 ctrs DIP SOLDER TAIL on .250 spaced rows for VECTOR and MASI motherboards GOLD plated.

| 1-4 | 5-9 | 10-24 |
|---|---|---|
| $3.50 | $3.25 | $3.00 |

**S100SE** 50/100 Cont .125 ctrs. PIERCED SOLDER EYELET tails GOLD

| 1-4 | 5-9 | 10-24 |
|---|---|---|
| $5.00 | $4.50 | $4.25 |

## OTHER POPULAR EDGE CONNECTORS

All Edge Card Connectors are GOLD PLATED (not Gold Flash) Bodies are non brittle, Solvent res., G.E. Valox Contacts are Bifurcated; Phos/Bronze: GOLD over Nickel.

ABBREVIATIONS: SE = Solder Eyelet WW = 3 Level Wire Wrap ST = Solder Tail

### .100" Contact Center Connectors

| PART NO. | TYPICAL APPLICATION | PRICE 1-4 | 5-9 | 10-24 |
|---|---|---|---|---|
| D1326-1SE | Imsai M10, SIO | 2.60 | 2.40 | 2.20 |
| D2244-1WW | Vector Plugboards | 4.00 | 3.80 | 3.60 |
| D2250-1SE | Imsai P10, Intel Multibuss | 3.70 | 3.50 | 3.40 |
| D2250-1ST | Imsai P10, Intel Multibuss | 3.50 | 3.30 | 3.10 |
| D2040-1SE | TRS-80 | 3.20 | 3.05 | 2.90 |
| D2040-1ST | TRS-80 | 3.00 | 2.85 | 2.70 |
| D2040-1WW | TRS-80 | 3.30 | 3.15 | 3.00 |
| D3060-1WW | Intel Multibuss | 4.10 | 3.90 | 3.70 |
| D3672-1SE | Vector Plugboards | 5.00 | 4.75 | 4.50 |
| D3672-1ST | Vector Plugboards | 4.95 | 4.70 | 4.45 |
| D3672-1WW | Vector Plugboards | 4.90 | 4.65 | 4.40 |
| D4080-1SE | PET | 5.95 | 5.70 | 5.45 |
| D4080-1ST | PET | 5.00 | 4.75 | 4.50 |
| D4080-1WW | PET | 5.20 | 4.95 | 4.70 |
| D4386-1SE | Cos ELF | 5.60 | 5.35 | 5.05 |
| D4386-1ST | Cos ELF | 5.40 | 5.15 | 4.90 |
| D4386-1WW | Cos. ELF | 5.50 | 5.25 | 5.00 |
| D50100-1WW | | 5.95 | 5.75 | 5.55 |

### .125" Contact Center Connectors

| PART NO. | TYPICAL APPLICATION | PRICE 1-4 | 5-9 | 10-24 |
|---|---|---|---|---|
| D3672-2WW | | 5.25 | 5.00 | 4.75 |
| D4080-2WW | Vector 4350 | 5.95 | 5.65 | 5.35 |
| S100-STG | S-100, Imsai, Vector, Cromenco, Motherboards | 3.50 | 3.25 | 3.00 |
| S100-WWG | S-100 Wire Wrap | 4.00 | 3.75 | 3.40 |
| S100-ALT | Altair | 4.50 | 4.25 | 4.00 |
| S100SE | S100 Solder Eyelet | 5.00 | 4.50 | 4.00 |

### .156 Contact Centers Connectors

| PART NO. | TYPICAL APPLICATION | PRICE 1-4 | 5-9 | 10-24 |
|---|---|---|---|---|
| S6X-5SE | Pet. NSC CLK Modules | 1.40 | 1.30 | 1.20 |
| D612-5SE | Pet. NSC CLK Modules | 1.60 | 1.50 | 1.40 |
| D1224-5SE | Pet | 2.40 | 2.30 | 2.20 |
| D2224-5ST | Pet | 2.30 | 2.20 | 2.10 |
| D1530-5SE | Vector Plugboards, GRI Keybrds | 2.50 | 2.35 | 2.10 |
| D1530-5ST | Vector Plugboards, GRI Keybrds | 2.40 | 2.25 | 2.05 |
| D1530-5WW | Vector Plugboards, GRI Keybrds | 2.60 | 2.40 | 2.15 |
| D1836-5SE | | 3.00 | 2.80 | 2.60 |
| D2244-5SE | Vector, Kim, etc | 3.00 | 2.80 | 2.20 |
| D2244-5ST | Vector, Kim, etc | 3.00 | 2.80 | 2.50 |
| D2244-5WW | Vector Kim, etc | 3.95 | 3.70 | 3.40 |
| D3672-5SE | Vector Plugboards | 5.50 | 5.30 | 5.00 |
| D3672-5ST | Vector Plugboards | 5.45 | 5.25 | 5.00 |
| D3672-5WW | Vector Plugboards | 5.60 | 5.40 | 5.10 |
| DE4386-5SE | Mot 6800, Intel Multibuss, NSC pacer | 6.00 | 5.75 | 5.25 |
| D4386-5ST | Mot 6800, Intel Multibuss, NSC pacer | 5.90 | 5.65 | 5.15 |
| D4386-5WW | Mot 6800, Intel Multibuss, NSC pacer | 6.50 | 6.30 | 5.90 |
| CG-1 | Imsai Style Card Guides | 5/1.00 or 100/10.00 | | |

### RS232 & "D" TYPE CONNECTORS

P = Plug-Male S = Socket-Female C = Cover-Hood

| PART NO. | DESCRIPTION | PRICE 1-4 | 5-9 | 10-24 |
|---|---|---|---|---|
| DE-9P | 9 Pin Male | 1.50 | 1.30 | 1.10 |
| DE-9S | 9 Pin Female | 1.95 | 1.75 | 1.45 |
| DE-9C | 9 Pin Cover | 1.50 | 1.30 | 1.10 |
| DA15P | 15 Pin Male | 2.00 | 1.80 | 1.55 |
| DA15S | 15 Pin Female | 2.90 | 2.70 | 2.45 |
| DA15C | 15 Pin Cover | 1.80 | 1.60 | 1.30 |
| DB-25P | 25 Pin Male | 2.50 | 2.20 | 2.06 |
| DB-25S | 25 Pin Female | 3.50 | 3.10 | 2.96 |
| DB51212-1 | 1 pc. Grey Hood | 1.65 | 1.35 | 1.20 |
| DB1226-1A | 2 pc. Black Hood | 1.80 | 1.60 | 1.35 |
| DB110963-3 | 2 pc. Grey Hood | 1.70 | 1.40 | 1.25 |
| DC37P | 37 Pin Male | 3.95 | 3.75 | 3.50 |
| DC37S | 37 Pin Female | 5.50 | 5.25 | 4.90 |
| DC37C | 37 Pin Cover | 2.00 | 1.80 | 1.60 |
| DD50P | 50 Pin Male | 5.00 | 4.75 | 4.60 |
| DD50S | 50 Pin Female | 6.50 | 6.00 | 5.75 |
| DD50C | 50 Pin Cover | 2.50 | 2.30 | 2.20 |
| D20418-S | Hardware Set (2 pair) | 1.00 | .80 | .70 |

MALE

FEMALE

Connector for CENTRONICS 700 SERIES:
Amphenol 57-30360 for back of Centronics 700 Series printers 1-4—$9.00 5-up—$7.50

### DIP PLUGS

| Part # | No. of Pins | PRICE 1-24 | 25-99 | 100-499 |
|---|---|---|---|---|
| P08P02 | 8 | .41 | .36 | .29 |
| P14P02 | 14 | .48 | .42 | .34 |
| P16P02 | 16 | .55 | .47 | .38 |
| P18P02 | 18 | .67 | .57 | .46 |
| P22P02 | 22 | .75 | .67 | .63 |
| P24P02 | 24 | .79 | .71 | .66 |
| P28P02 | 28 | 1.10 | .93 | .81 |
| P40P02 | 40 | 1.26 | 1.07 | .94 |

### GOLD SOLDERTAIL STANDARD IC Sockets

| | 1-24 | 25-49 | 50-99 |
|---|---|---|---|
| 8STG | .30 | .27 | .24 |
| 14SG | .35 | .32 | .29 |
| 16STG | .38 | .35 | .32 |
| 18STG | .52 | .47 | .43 |
| 20STG | .60 | .56 | .52 |
| 22STG | .70 | .63 | .57 |
| 24STG | .70 | .63 | .57 |
| 28STG | 1.10 | 1.00 | .90 |
| 40STG | 1.75 | 1.55 | 1.45 |

### TIN SOLDERTAIL · LOW PROFILE IC Sockets

| | 1-24 | 25-49 | 50-99 |
|---|---|---|---|
| 8CS2 | .25 | .16 | .15 |
| 14CS2 | .25 | .18 | .16 |
| 16CS2 | .25 | .20 | .18 |
| 18CS2 | .29 | .26 | .27 |
| 20CS2 | .34 | .32 | .30 |
| 22CS2 | .37 | .36 | .35 |
| 24CS2 | .38 | .37 | .36 |
| 28CS2 | .45 | .44 | .43 |
| 40CS2 | .63 | .62 | .61 |

### 3 LEVEL GOLD WIRE WRAP SOCKETS

Sockets purchased in multiples of 50 per type may be combined for best price.

| | 1-9 | 10-24 | 25-99 | 100-249 | 250-999 |
|---|---|---|---|---|---|
| 8 pin* | .40 | .36 | .34 | .31 | .27 |
| 14 pin* | .45 | .39 | .37 | .34 | .32 |
| 16 pin* | .50 | .42 | .40 | .36 | .34 |
| 18 pin | .70 | .60 | .55 | .50 | .45 |
| 20 pin | .90 | .80 | .75 | .65 | .62 |
| 22 pin* | .95 | .85 | .80 | .70 | .65 |
| 24 pin | .95 | .85 | .80 | .70 | .65 |
| 28 pin | 1.25 | 1.15 | 1.00 | .95 | .90 |
| 40 pin | 1.65 | 1.45 | 1.35 | 1.20 | 1.10 |

All sockets are GOLD 3 level closed entry *End and side stacable 2 level, Solder Tail, Low Profile, Tin Sockets and Dip Plugs available CALL FOR QUOTATION

July Clearance SALE

## PORTABLE OSCILLOSCOPES

**MS-215 Dual Trace**
Reg. $435.00
Sale Priced
**$389.00***

NLS

**MS-15 Single Trace**
Reg. $318.00
Sale Priced
**$289.00***

**With Rechargeable Batteries and Charger Unit**

- 15 megahertz band width
- External and internal trigger
- Time base - 1 micro sec. to 0.5 sec/div. 21 settings
- Battery or line operation
- Viewing area 1.1" x 1.35" case size 2.75"H x 6.4"W x 7.5"D
- Parts & labor guaranteed 1 year, made in USA!
- Automatic or line sync modes
- Power consumption less than 15W.
- Verticle Gain - 0.01 to 50 Volts/div. 12 settings.
- Weight is only 3 pounds

| | |
|---|---|
| 41-140 Leather Carrying Case | $45.00 |
| 41-141 10:1 Probe | $27.00 |
| 41-3495 10:1, 1:1 Combo Probe | $36.00 |

* Sale Prices Valid for PREPAID or CREDIT CARD ORDERS ONLY.

Why Cut?
Why Strip?
Why Slit?
WHY NOT . . .
JUST WRAP™

**NEW**

- AWG 30 Wire
- .025" Square Posts
- Daisy Chain or Point To Point
- No Stripping or Slitting Required . . . JUST WRAP TM . . .
- Built in Cut off
- Easy Loading of Wire
- Available Wire Colors: Blue, White, Red & Yellow

**JUST WRAP TOOL WITH ONE 50 FT. ROLL OF WIRE**

| COLOR | PART NO. | U.S. LIST PRICE |
|---|---|---|
| BLUE | JW-1-B | 14.95 |
| WHITE | JW-1-W | 14.95 |
| YELLOW | JW-1-Y | 14.95 |
| RED | JW-1-R | 14.95 |

**REPLACEMENT ROLL OF WIRE 50 FT.**

| COLOR | PART NO. | U.S. LIST PRICE |
|---|---|---|
| BLUE | R-JW-B | 2.98 |
| WHITE | R-JW-W | 2.98 |
| YELLOW | R-JW-Y | 2.98 |
| RED | R-JW-R | 2.98 |

**JUST WRAP KIT CONTAINS**
- JUST WRAP Tool
- Roll of Blue Wire, 50 ft.
- Roll of White Wire, 50 ft.
- Roll of Yellow Wire, 50 ft.
- Roll of Red Wire, 50 ft.
- Unwrapping Tool

JWK-6, JUST WRAP KIT
**$24.95**

## DIP JUMPERS

**FLAT RIBBON CABLE ASSEMBLIES WITH DIP CONNECTORS**

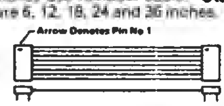
- Available with 14, 16, 24 and 40 contacts.
- Mate with standard IC sockets.
- Fully assembled and tested.
- Integral molded-on strain relief.
- Line-by-line probeability.

A P DIP Jumpers are the low-cost, high-quality solution for jumpering within a PC board; interconnecting between PC boards, backplanes and motherboards, interfacing Input/Output signals; and more.

All assemblies use rainbow cable. Standard lengths are 6, 12, 18, 24 and 36 inches.

WOW

DOUBLE-ENDED DIP JUMPERS




| No. Contacts | Length 6" | Length 12" | Length 18" | Length 24" | Length 36" |
|---|---|---|---|---|---|
| 14 | 924106 6 R $2.41 | 924106-12 R $2.61 | 924106 18-R $2.82 | 924106 24-R $3.02 | 924106 36-R $3.43 |
| 16 | 924116 6 R $2.65 | 924116 12-R $2.88 | 924116-18-R $3.11 | 924116-24-R $3.34 | 924116 36-R $3.80 |
| 24 | 924126 6 R $4.15 | 924126 12 R $4.50 | 924126 18-R $4.85 | 924126 24-R $5.20 | 924126 36 R $5.90 |
| 40 | 924136 6 R $6.93 | 924136 12 R $7.52 | 924136 18-R $8.11 | 924136 24-R $8.70 | 924136 36-R $9.88 |

ORDER TODAY
1-800-423-5633

VISA


**PRIORITY ONE ELECTRONICS**©
16723B Roscoe Blvd. 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. Prepaid U.S. orders less than $75.00 include 5% shipping and handling. minimum $2.50. Excess refunded. Just in case . . . please include your phone no.
Prices subject to change without notice. OEM and Institutional
We will do our best to maintain prices thru July 1979. Inquiries invited.

**phone orders welcome (213) 894-8171, (800) 423-5633**

We have the Best Prices on 2102's, 2114's, 4116's. We have 5V 2716's in Stock.



Circle 312 on inquiry card.

Circle 384 on inquiry card.





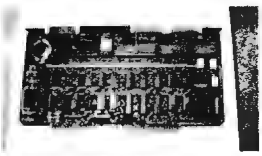
Circle 30 on inquiry card.

# asap computer products, inc.

11542-1 KNOTT STREET
GARDEN GROVE, CALIFORNIA 92641
(714) 891-2663

## MICROBYTE Z80/I-O

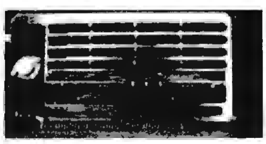
- A complete single board Z80A CPU and serial/paralle I/O system
- Fully S-100 Bus compatible, IMSAI, ALTAIR
- Z80A CPU (4MSs version of the Z80)
- 158 instructions — superset of and upward compatible from the 8080's 78 instructions
- 1K (2708), 2K (2716), or 4K (2732) EPROM onboard with Z80 Monitor program
- On board EPROM can be hardware and/or software deselected
- 2MHs or 4MHs operation is switch selectable
- 0 or 1 wait state for all cycles is switch selectable
- Full vectored interrupt capability (7 bits) with NMI (1 bit)
- 2 RS-232C serial ports with 8251 USARTs
- Serial baud rates switch selectable
- 1 8-bit parallel port with 2 control lines with Z80A-PIO
- Gold Contacts for higher reliability
- Power requirements: +8V @ 800mA, +16V, @ 86mA, - 16V @ 100mA
- Operating temperature 0°-55°C
- Will operate with or without IMSAI/ALTAIR front panel
- Low power shotttky tri-state buffers on all address and data lines
- Fully warranted for 120 days from date of shipment



$360.00

## MICROBYTE 16K STATIC RAM BOARD

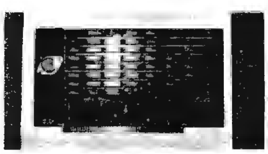
- Fully S100 Bus Compatible, IMSAI, SOL, ALTAIR, ALPHA MICRO
- Uses National's Low Power 5257 4K x 1 Static Rams
- 2 MHz or 4 MHz operation
- On board single 5 amp regulator
- Thermally designed heat sink (board operating temperature 0° — 70°C)
- Inputs fully low power Shottky Schmitt Trigger buffered on all address and data lines
- Phantom is jumper selectable to pin 67
- Each 4K bank addressable to any 4K slot with in a 64K boundary.
- 4K hardware or software selectable
- Selectable port address
- 4K banks can be selected or disabled on power on clear or reset
- Will operate with or without front panel
- Compatible with ALPHA MICRO, with extended memory management for selection beyond 64K
- No DMA restriction
- Low power consumption 1.3 amp
- Fully warranted for 120 days from date of shipment



450 NS $340.00
300 NS $360.00

## MICROBYTE 32K STATIC RAM BOARD

- Fully S100 Bus Compatible, IMSAI, SOL, ALTAIR, ALPHA MICRO
- Uses National's Low Power 5257 4K x 1 Static Rams
- 2 MHz or 4 MHz operation
- On board single 5 amp regulator
- Thermally designed heat sink (board operating temperature 0° — 70°C)
- Inputs fully low power Shottky Schmitt Trigger buffered on all address and data lines
- Phantom is jumper selectable to pin 67
- Each 4K bank addressable to any 4K slot with in a 64K boundary.
- 4K hardware or software selectable
- One on board 8-bit output port enables or disables the 32K in 4K blocks
- Selectable port address
- 4K banks can be selected or disabled on power on clear or reset
- Will operate with or without front panel
- Compatible with ALPHA MICRO, with extended memory management for selection beyond 64K
- No DMA restriction
- Low power consumption 2.3 — 2.5 amps
- Fully warranted for 120 days from date of shipment



450 NS $650.00
300 NS $680.00

## MICROBYTE DISK CONTROLLER

- IBM 3740 Soft Sectored Compatible
- Z80 or 8080 compatible on S-100 Bus
- Single density runs both mini and full size drives, runs CPM, on Shugart, Persci, Memorex etc.
- Selectable port/address
- On bord 2708/2716 for bootstrap or monitor program
- No hardware jumpers, uses plug in modules for different drives
- Uses 1771B-01 controller chip
- Assembled and tested

$250.00

## MICROBYTE MOTHERBOARD

- Active Diode termination
- Slot for IMSAI front panel
- Terminal block connection for easy hook-up
- Extra wide ground plane
- Silk screen and solder mask
- Assembled and tested

9 SLOT $135.00
20 SLOT $200.00

## 8251

PROGRAMMABLE/U-ART
TESTED @ 4 MHZ
**$6.00 each**

## IMSAI CONN.

100 PIN-SOLDERTAIL
GOLD CONTACTS
**$3.25 each or 10 / 2.75 ea.**

## 2708's

LOW POWER
450 NS.
**$9.00 each**
**8 for $69.00**

## SA400

DISK DRIVE INCLUDES CABINET, NO PWR SUPPLY CUTOUTS FOR SWITCH, FUSE, & INTERFACE CABLE
Mfg. By Lobo Drive
**$325.00**

## TRS - 80

FLOPPY DISK DRIVE WITH CABINET & PWR. SUPPLY COMPATIBLE WITH RADIO SHACK INTERFACE ASSEMBLED & TESTED WITH 1 YR. WARRANTY ON PARTS & LABOR
Mfg. By Lobo Drive **$385.00**

## 2716

5 VOLT ONLY
LOW POWER
HIGH SPEED

**$45.00**

Limited Qty.

## REGULATORS

| | 1-9 | 10-49 | 50 up |
|---|---|---|---|
| 320 T-5 | .90 | .85 | .75 |
| 320 T-12 | .90 | .85 | .75 |
| 340 T5 | .85 | .80 | .70 |
| 340 T-12 | .85 | .80 | .70 |
| 78 H05 | 6.50 | 6.00 | 5.50 |

## MM 5257

4K X 1 STATIC RAM

| | 1-16 | 17-48 | 49 up |
|---|---|---|---|
| 300 NS | 6.50 | 6.05 | 5.65 |
| 450 NS | 6.25 | 5.85 | 5.50 |

## SOCKETS—LO PROFILE

(tin)

| | 1-24 | 25-99 | 100-499 | 500 up |
|---|---|---|---|---|
| 14 PIN | .18 | .17 | .15 | .13 |
| 16 PIN | .19 | .18 | .16 | .14 |
| 18 PIN | .24 | .23 | .20 | .18 |
| 20 PIN | .29 | .28 | .26 | .25 |
| 24 PIN | .34 | .33 | .32 | .30 |
| 40 PIN | .60 | .59 | .58 | .56 |

## SPECIAL

.1 @ 12 VOLTS
CERAMIC CAP
**9¢ each**
**100 for $8.00**

## CALL FOR INFO ON OUR NEW PROM/ PROGRAMMER

PROGRAMS 2708 / 2716 / 2732

## SHUGART 801-Disk Drive

WITH CABINET & POWER SUPPLY
ASSEMBLED & TESTED
1 YR PARTS & LABOR
Mfg. By Lobo Drive
**$585.00**

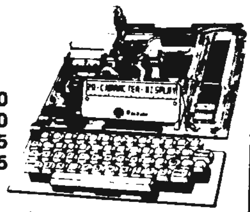
**ORDERING INFORMATION:**
Name, Address, Phone
Ship by: UPS or P.P.
Shipping Charge: Add $2.50 up to 5 lbs., all excess shipping charges will be refunded. Credit cards will be charged appropriate freight.

**TERMS:**
We accept cash, check, money orders, Visa, and Master Charge cards.
COD's: on approval only
Open Acct's: companies may inquire for net terms.
Tax: add 6% for Calif. residents only

Circle 11 on inquiry card.

# JADE Computer Products

## Rockwell AIM-65: The Head-Start in Microcomputers



A KIM-1 compatible machine with on-board printer and a real keyboard!

**$375.00 w/1K RAM**
**$450.00 w/4K RAM**

| | |
|---|---|
| 4K assembler/editor in ROM: | $ 80.00 |
| 8K BASIC in ROM: | $100.00 |
| Power supply: | $ 59.95 |
| Case for AIM-65: | $ 49.95 |

**Special Package Price: $599.00**
AIM-65 (4K), Power Supply, Case, and 8K BASIC ROM

## THE EXIDY SORCERER

Flexibility is the key. The Sorcerer Computer gives you the flexibility of using ready-to-run, pre-packaged programs or doing your own thing and personalizing the programs for yourself. Which ever you choose, the Sorcerer is the personal computer that speaks your language.

The Sorcerer also provides full graphics capablities. Each character, formed by an 8 x 8 dot cell, can be programmed as a graphic symbol set. High resolution (512 x 240 addressable points) gives a total of 122,880 locations for super animation and extremely tight plotting curves. The alphanumeric set gives 64 x 30 characters on the video screen

With 16K of memory ..................... $1150.00

## JADE MEMORY EXPANSION KITS For TRS—80, Apple, & Exidy 4116's

TRS-80*APPLE*SORCERER*TRS-80*APPLE*SORCERER*TRS-80*APPLE*SORCERER*TRS-80*APPLE*SORCERER

Everything a person needs to add 16K of memory. Chips come neatly packaged with easy to follow directions. In minutes your machine is ready for games and more advanced software.

**$82.00**

## THE SYM-1 $245

6502 — based single board computer with keyboard/display, KIM-1 hardware compatible, complete documentation.

SYM—1 CASE $39.95

## THE KIM 1 $179

KIM-1 Module monitor, programs stored in 2048 ROM Bytes, User Manual, Wall size schematic, Hardware manual, Programming manual, Programmers reference card, and Keyboard display.

## SD SYSTEMS Z-80 STARTER KIT

Based on the powerful Z80 CPU, this kit is an ideal introduction to microprocessors. It has an on-board keyboard and display, plus cassette tape interface and expansion provisions for two S-100 connectors. This "Do-it-all" Board will also program the 2716 2K EPROM.

Kit............................ $249.95
Assembled and Tested,.......... $399.95

## JADE VIDEO INTERFACE

S-100 Compatible Serial Interface with Sockets Included.

| | |
|---|---|
| Kit | $117.95 |
| Assembled & Tested | $159.95 |
| Bare Board w/manual | $ 35.00 |

## MEM-2 16 K Static RAM Board

| | |
|---|---|
| Kit- (450ns) | $250.00 |
| Kit- (250ns) | $285.00 |
| Assembled- (450ns) | $325.00 |
| Assembled- (250ns) | $350.00 |

## JADE PROTO BOARD

Includes gold plated fingers, S-100 size, holds 72-16 pin dips, accomodates all 8 thru 40 pin dip packages.

$19.95

## DYNAMIC RAM BOARDS EXPANDABLE TO 64K

### 32K VERSION • KITS

Uses 4115 (8Kx1, 250ns) Dynamic RAM's, can be expanded in 8K increments up to 32K:

8K $159.00 24K $249.00
16K $199.00 32K $299.00

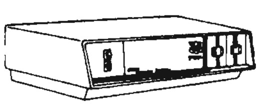
**4115 SALE 8 for $39.95**

### 64K VERSION • KITS

Uses 4116 (16Kx1, 200ns) Dynamic RAM's, can be expanded in 16K increments up to 64K:

16K $249.00 48K $469.00
32K $369.00 64K $569.00

## THE PIGGY MAINFRAME



THE PIGGY IS HERE!

This sleak new mainframe is neatly trimmed to hold six S-100 boards, two mini-floppy drives, and is available in five colors. Power requirements: 115/220 VAC, 50/60 HZ. Weight: 27 lbs. (with drives). Dimensions: 21.375" Wide X 8.4" High X 15.875" Deep. Power Supply: 8 volts at 18 Amps unregulated, + 16 volts at 3 amps uregulated, - 16 volts at 3 amps unregulated, + 5 volts at 3 amps regulated, + 12 volts at 3 amps regulated.

THE PIGGY (Without Drives).... $475.00

## FLOPPY DISK INTERFACE

JADE FLOPPY DISK (Tarbell board)

| | |
|---|---|
| Kit | $195.00 |
| Assembled & Tested | $250.00 |

## S.D. Computer Products VERSA-FLOPPY

| | |
|---|---|
| Kit | $159.95 |
| Assembled & Tested | $239.00 |

## JADE PARALLEL/SERIAL INTERFACE

S-100 compatible, 2 serial I/O ports, 1 parallel I/O.

| | |
|---|---|
| Kit JG-P/S | $124.95 |
| Assembled & Tested: JG-P/SA | $179.95 |
| Bare Board w/Manual | $ 30.00 |

## ★ STATIC RAM SPECIALS ★

| | 1-15 | 16-99 | 100 + |
|---|---|---|---|
| **2114's, low power (1024x4)** | | | |
| 450ns | 8.00 | 6.95 | 5.50 |
| 300ns | 9.00 | 8.00 | 6.50 |
| **TMS4044/MM5257, low power** | | | |
| 450ns | 8.00 | 7.50 | 6.50 |
| 300ns | 9.95 | 8.75 | 8.00 |
| **4200A (4Kx1, 200ns)** | 9.95 | 8.50 | 8.00 |
| **410D (4K x 1, 200 ns)** | 8.25 | 7.00 | 6.75 |

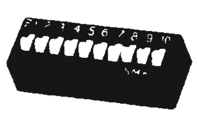
## LEEDEX MONITOR

- 12" Black and White
- 12MHZ Bandwidth
- Handsome Plastic Case

**$139.00**

## DIP SWITCHES

| Part No. | positions 1-9 | |
|---|---|---|
| SWD-103 | 3 | $1.18 |
| SWD-104 | 4 | $1.20 |
| SWD-105 | 5 | $1.24 |
| SWD-106 | 6 | $1.28 |
| SWD-107 | 7 | $1.30 |
| SWD-108 | 8 | $1.34 |
| SWD-109 | 9 | $1.36 |
| SWD-110 | 10 | $1.38 |



## Solid State Music's I/O 4

2- Serial & 2- Parallel I/O Ports. S-100 with full handshaking.

KIT .............. $149.95
Assembled ....... $199.95
Bare Board ........ $29.95

## VECTOR PLUG BOARDS

8800V
Universal/Microcomputer/Processor Plugboard S-100 Bus. Complete With Heat Sink & Hardware.
5.3" x 10" x 1/16"
$19.95

8801-1
Same as 8800V Except Plain, Less Power Buses & Heat Sink.
$15.95

P Pattern Plugboards For I.C.'s
Epoxy Glass 1/18" 44 Pin
Connector Space .156
3662 6.5" x 4.5"......................... $7.65
3662-2 9.6" x 4.5"...................... $11.45.

Hi-Density Dual-In-Line Plugboard For Wire Wrap With Power & Grd. Bus 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

Gen Purpose D.I.P. Boards With Bus Pattern For Solder Or Wire Wrap. Epoxy Glass 1/16" 44 Pin Con. Space .156
3677 9.6" x 4.5"......................... $10.90
3677-2 6.5" x 4.5........................ $9.74

3690-12
Card Extender
Card Extender Has 100 Contacts 50 Per Side ON .125 centers. Attached Connector is Compatible With S-100 Bus Systems
$25.83
3690 6.5" 22/4 Pin .158 Centers Extenders. $13.17

## SD SYSTEMS SBC-100

An S-100 single board computer. Z-80 CPU with 1024 bytes of RAM, 8 to 32K bytes of PROM, Serial I/O port.

Kit .............. $239.95
Assembled ....... $369.95

## TARBELL Cassette Interface

Plugs into your IMSAI or ALTAIR, 4 extra status lines, 37 page manual included, 4 extra control lines.

| | |
|---|---|
| Kit | $99.95 |
| Assembled | $175.00 |
| Bare Board | $40.00 |
| Manual | $8.00 |

## STATIC RAM BOARDS

**JADE 8K**

| | | |
|---|---|---|
| Kits: | 450ns | $125.95 |
| | 250ns | $149.75 |
| Assembled & Tested: | 450ns | $139.75 |
| | 250ns | $169.75 |
| Bare Board: | | $ 25.00 |

**16K — Uses 2114's (low power)**
Assembled & Tested:
RAM 16 (250ns) $375.00
RAM 16B (450ns) $325.00

**16K with memory management**
Assembled & Tested:
RAM 65 (250ns) $390.00
RAM 65B (450ns) $350.00

**32K Static**
Assembled & Tested:
250ns $795.00
450ns $725.00
250ns Kit $575.00

Circle 195 on inquiry card.

PLACE ORDERS TOLL FREE: 800/421-5809 Continental U.S. 800/262-1710 Inside California

## *THE BIG Z*

### THE NEW Z-80

### CPU BOARD FROM JADE

Features Include: ■ S-100 Compatible, available in 2MHz or 4MHz versions. ■ On-board 2708, 2716, 2516, or 2532 EPROM can be addressed on any 1K, 2K, or 4K boundary, with power-on jump to EPROM. ■ On-board EPROM may be used in SHADOW mode, allowing full 64K RAM to be used. ■ Automatic MWRITE generation if front panel is not used. ■ On-board USART for synchronous or asynchronous RS232 operation (on-board baud rate generator). ■ Reverse-channel capability on USART allows use with buffered peripherals or devices with "not-ready" signal.

2MHz- Kit: CPU-30200K, 2 lbs .......... $149.95
Assembled and Tested:
CPU-30200A, 2 lbs .......... $199.95

4 MHz- Kit: CPU-30201K, 2 lbs .......... $159.95
Assembled and Tested
CPU-30201A, 2 lbs .......... $209.95

## JADE'S DOUBLE DENSITY

DD™

KIT: $249.00
Assmb. & Tstd: $299.00

- Single or Double Density Recording
- Full Size or Mini Floppy
- CP/M Compatible in either density
- Programmed Data Transfer, no DMA
- Controls up to 8 drives
- IBM format in either density
- Software Selectable Density

■ This controler utilizes the proven reliability of the IBM standard format as well as the lastest phase-locked-loop for data separation ■ All clocks are generated from an on-board crystal oscillator ■ Right precompensation is used to enhance data recovery reliability in the double density mode ■ Density selection is entirely transparent to the user ■ Single and double density diskettes can be mixed on the same system.

| MICROPROCESSORS | |
|---|---|
| F8 | $16.95 |
| Z80 (2MHz) | $10.95 |
| Z80A (4MHz) | $14.95 |
| CDP1802CD | $19.95 |
| 6502 | $11.95 |
| 6800 | $9.75 |
| 6802 | $14.00 |
| 8008-1 | $15.95 |
| 8035 | $24.00 |
| 8035-8 | $24.00 |
| 8080-A | $10.00 |
| 8085 | $23.00 |
| TMS9900TL | $49.95 |
| **8080A SUPPORT DEVICES** | |
| 8212 | $2.90 |
| 8214 | $4.65 |
| 8216 | $2.75 |
| 8224 (2MHz) | $4.30 |
| 8224-4 (4MHz) | $9.95 |
| 8226 | $2.75 |
| 8228 | $6.40 |
| 8238 | $6.40 |
| 8243 | $8.00 |
| 8251 | $7.50 |
| 8253 | $20.00 |
| 8255 | $6.40 |
| 8257 | $18.00 |
| 8259 | $18.00 |
| 8275 | $51.20 |
| 8279 | $17.70 |
| **USRT** | |
| S2350 | $10.95 |
| **UARTS** | |
| AY5-1013A | $5.25 |
| AY5-1014A | $8.25 |
| TR1602B | $5.25 |
| TMS6011 | $5.95 |
| IM6403 | $9.00 |
| **BAUD RATE GENERATORS** | |
| MC14411 | $10.00 |
| 14411 Crystal | $4.95 |
| **6800 PRODUCT** | |
| 6821P | $5.25 |
| 6828P | $9.50 |
| 6834P | $16.95 |
| 6850P | $4.80 |
| 6852P | $5.25 |
| 6860P | $9.25 |
| 6862P | $12.00 |
| 6875L | $7.30 |
| 6880P | $2.50 |
| **CHARACTER GENERATORS** | |
| 2513 Upper (1-12 + 5) | $6.75 |
| 2513 Lower (1-12 + 5) | $6.75 |
| 2513 Upper (5 volt) | $9.75 |
| 2513 Lower (5 volt) | $10.95 |
| MCM6571 up scan | $10.95 |
| MCM6571A down scan | $10.95 |
| **PROMS** | |
| 1702A | $5.00 |
| 2708 | $12.95 |
| 2716 (5 · 12) | $49.00 |
| 2716 (5v) | $49.00 |
| 2758 (5v) | $30.00 |
| **DYNAMIC RAMS** | |
| 416D/4116 (200ns) | $12.50 |
| 2104/4096 | $4.00 |
| 2107B-4 | $3.95 |
| TMS4027/4096 | $4.00 |

| STATIC RAMS | 1—15 | 16—100 |
|---|---|---|
| 21L02 (450ns) | $1.50 | $1.20 |
| 21L02 (250ns) | $1.75 | $1.50 |
| 2101-1 | $2.95 | $2.60 |
| 2111-1 | $3.25 | $3.00 |
| 2112-1 | $2.95 | $2.65 |

| FLOPPY DISK CONTROLLERS | |
|---|---|
| 1771801 | $39.95 |
| 1791 | $49.95 |
| **KEYBOARD CHIPS** | |
| AY5-2376 | $13.75 |
| AY5-3600 | $13.75 |
| MM5740 | $18.00 |

### POWER SUPPLIES

PSD-249A: For a Single 5 1/4" Disk Drive. By Power-One or Alpha Power: +/- 5V at .7A, +/- 12V at 1.1A $52.00

PSD-205A: For Single 8" Disk Drive. By Power-One. +5V at 1A, -5V at .5A, +24V at 1.5A .......... $89.95

PSD-206A: For Two 8" Disk Drives. By Power-One or Alpha Power. +5V at 2.5A, -5V at .5A, +/-24V at 3A .......... $125.00
Rockwell: Aim-65 Power Supply
PSX-030A .......... $59.95
KIM-1 or SYM-1 Power Supply:
PSX-020A .......... $59.95

### ★ DISK DRIVES ★

**B51 5¼" $295.00**
by Micro Peripherals, Inc. Operates in either single density (125KB, unformatted) or double density (250KB, unformatted) modes, up to 40 tracks, with a track-to-track access time of only 5 ms.

**SA801R $575.00**
by Shugart Single-sided 8" floppy disk drive.

**FD8-100 $395.00**
GSI/Siemens. Runs cooler and quieter than 801 (8")

**SA400 $325.00**
Single density 5¼, 35-Track drive.
Cabinet and power supply available

1791 B01
Dual Density Controller Chip
**$49.95**

## Vista V80 Mini Disk
### *SYSTEM FOR TRS—80*
### $395.00

Includes disk drive, power supply, regulator board, and compact case. The V-80 offers 23% more storage capacity. Simply take it out of the box, plug in the cable, and it's ready to run. Requires 16K, Level II, expansion interface.

Signal Cable .......... $24.95

### JADE CABLE ASSEMBLIES

Mini-Disk Cable Kit: 5 1/4" interface to 2 Shugart or compatible drives. Cable is 5' long with 34 pin edge connectors
WCA-3431K .......... $34.95

8" Disk Cable Kit No. 1: 34 pin assembly WCA-5031K .......... $38.45

8" Disk Cable Kit No. 2: 50 pin assembly (great for Tarbell disk controller) WCA-5032K .......... $38.45

Signal Cables: 6 feet long, 34 pin edge connectors at each end .......... $24.95
WCA-3421A

### JADE ISO—BUS MOTHERBOARDS

Comes in either 6, 12, or 18 slot sizes. These boards with a special ground plane assures a silent operation

| JADE 6 Slot | |
|---|---|
| Kit | $49.95 |
| Assembled | $59.95 |
| Bare Board | $24.95 |
| **JADE 12 Slot** | |
| Kit | $89.95 |
| Assembled | $99.95 |
| Bare Board | $39.95 |
| **JADE 18 Slot** | |
| Kit | $129.95 |
| Assembled | $149.95 |
| Bare Board | $59.95 |

- Two Drives Siemens/GSI 8" Floppy
- Power Supply for Above
- Jade Double Density Board (KIT)
- CP/M Operating System with Basic E
- Package of 10 Blank 8" Diskettes (Double Density)
- Includes Interface Cables

Price if Purchased Separately
$1544.95
Jade Special Package Deal
*$1225.00*

## JADE *Computer Products*

4901 W ROSECRANS AVENUE
Department "F" 3
HAWTHORNE CALIFORNIA 90250
U S A

Telephone
(213) 679-3313
(800) 421-5809 Continental U S
(800) 262-1710 Inside California

Cash, checks, money orders, and credit cards accepted. Minimum order: $10.00, California residents add 6% sales tax. Minimum shipping and handling charge: $2.50. Discounts available at OEM quantities.

**WRITE FOR OUR FREE CATALOG**
All prices subject to change without notice.

### 3M or *VERBATIM* FLOPPY DISKS

5¼ in. Minidiskettes
Soft sector, 10 sector, or 16 sector
$4.40 each or
Box of ten for $37.50
8 in. Standard Floppy Disks
$4.75 each or
Box of ten for $35.00

Circle 195 on inquiry card.

Circle 354 on inquiry card.







Circle 296 on inquiry card.

# Unclassified Ads

FOR SALE: Digital Group system dual Phi-Decks, dress cabinets, Hitachi video display, DG keyboard, 34 K programmable memory, lots of software. Will sell for $2200. Optional Selecterm printer, $1475 with WOPROC. Irv, 14522 Acacia, Tustin CA 92680, (714) 731-4850

FOR SALE Digital Group system 34 K programmable memory, dual Phi Decks, lots of software. Homebrew mainframe cabinet with front panel All documentation. This is a reliable system. Worth $1700, asking $1100. Larry, 3069 Copenhagen, Riverside CA 92504, (714) 687 6725

FOR SALE: TRS 80, Level 2 BASIC, 16 K, complete with video monitor, power supply, keyboard, cassette unit. Level 1 and 2 manuals, game software, all in perfect condition, I pay shipping $750 Bruce Grunewald, 7065 Granada Dr, Flint MI 48504, (313) 733-1730 during business hours.

FOR SALE. IMSAI 8080 with 44 K programmable memory, iCOM FD-3712 dual drive floppy; Lear Siegler ADM 3 video display, Centronics 101 A 132 column printer; National Multiplex CC7A digital cassette recorder; SOLA 1 5 KW constant voltage transformer. iCOM FDOS II-III Relocating Assembler, Text Editor, DEBBI (MITS Disk BASIC); business software; more All manuals included All hardware operating perfectly for 1 5 years. Desire to sell as unit A bargain today for $8000. Ship at my expense John Berry, 1520 Aberdeen Av, Baton Rouge LA 70808, (504) 344 8760

FOR SALE. Heathkit microprocessor course and trainer computer. Works perfectly Teaches microprocessor design and machine language programming. Many experiments in interface circuitry. Check November 1978 BYTE for more information. First check or money order for $225 gets it and I will ship it anywhere free. John Maurer, 3583 Vernal, Merced CA 95340.

COMPUTERIZED COMMODITY: and stock market is my interest I have developed three systems and am interested in exchange with others oriented and successful in this area. E Trachtenberg, POB 407, Little Neck NY 11363.

WANTED: Catalogs from hardware software dealers for the Level II TRS-80. Bob Martin, 911 Buckingham, Windsor Ontario, CANADA N8S 2C9.

FOR SALE: Free games with purchase of a Micropolis 1042 macrofloppy disk drive system. For $595 you can have an assembled and tested package complete with drive, cabinet, power supply, top-notch system software, blank diskettes and full documentation, an $835 value. Like new I'll ship UPS. David or Robert Mason, 503 Timber Ter, Houston TX 77024, (713) 681 3427.

FOR SALE. Four ASR33 Teletypes All are in very good working order and have been under continuous maintenance contract. All four have the DEC paper tape modification and can be shipped with this modification enabled or disabled. Will sell each for $400. You ship. F Mitchell POB 86, Due West SC 29639, (803) 379-8816 days.

FOR SALE: IMSAI boards 32 K programmable memory, $650 16 K programmable memory, $350. 8080 microprocessor a, $100. PIC-8, $80. I have recently purchased a Z-80 and cannot use these boards. Must sell; will accept any reasonable offer Send SASE for correspondence John Paul Daley, 7545 S Beloit, Bridgeview IL 60455.

FOR SALE: Computer terminals Portable timeshare or direct connection terminals Contain 80 column quiet printer, full ASCII keyboard, built-in modem and acoustic coupler, dual speed 110 and 300 bps, RS-232 and 20 mA and DAA interfaces, in a small suitcase size enclosure. Brand new condition with guarantee. Send $0.30 SASE for manuals, reprint and pictures. $849.95 plus shipping. S Stoddard, 12 Kathy Dr, Poquoson VA 23662

FOR SALE: Going to school, must sell SwTPC CT-1024 video terminal. Cursor and serial boards, TI keyboard, power supply, scrolling, $85 GT 61 graphics Power supply, MP L, PPG J joystick, software. $150. PR-40 printer. New head, MP-L, smoked cover $250. 4 K memory boards $60 Morrow's S 100 Wunderbuss mother board with ten edge connectors. $100. All in sockets and running. If phone disconnected, please write Chip Kroll, 4136 Tennyson, Houston TX 77005, (713) 667 6318.

FOR SALE: HP-67 programmable calculator, all standard accessories Plus an extra recharger and battery pack and 100 blank program cards. Compare at $485 (plus tax) Everything goes immediately via UPS for $365 certified check or money order Michael Craig, 105 E Edward St, Endicott NY 13760.

FOR SALE. Apple computer, 32 K, $750 or trade for new HP 67 and $200. Send card to Rob Dinnell, 600 Park Av 12A, Capitola CA 95010.

FOR SALE TRS-80, Level II BASIC, expansion interface, 48 K of user programmable memory, dual disk drives, DOS version 2 1. Editor Assembler, Line Renumber and 12 disks containing game and statistical software Original cost, $3000. Current Radio Shack cost, $2700. Will sell for $2000 or best offer, I pay the freight Joseph Boykin, 1897 N Fourth St Apt A, Columbus OH 43201.

FOR SALE· 4 K programmable memory chips. 16 UPD414D chips from converting my 8 K Exidy Sorcerer to 32 K Make me an offer. Tom Kreuzer, 310 Pine Av, Canby MN 56220.

FOR SALE. MITS 4 K dynamic programmable memory assembled, tested, burned in. Also 2SIO board (only one port implemented) assembled, tested, including MITS documentation Best offers or trades (will accept parts) T Shaw, 5F Nutmeg Knoll, Cockeysville MD 21030.

FOR SALE. Two Ithaca Audio 300 ns low power 8 K programmable memory boards (S 100), $150 each Electronic Systems television typewriter board, stand alone, assembled but never used, $130. First money order or cashier's check takes any or all, all others returned. I pay shipping. R Schuchman, 6221 Del Paso, San Diego CA 92120.

FOR SALE· Why buy a kit? Two fully static, 500 ns 8 K programmable memory boards (one SSM MB-6, one Logos 1) $125 apiece. One IMSAI UCRI cassette board, $30. Henry Kapteyn, 1175 W Baseline Rd, Claremont CA 91711

FOR SALE: Heath H8 compatible dot matrix printer 5-by 7, 80 or 96 column, operates in serial or parallel mode from 110 to 1200 bps. Made by practical automation, this unit is a current production machine, not a discontinue or surplus. Plenty of documentation. It is similar to the IPS 745. Make offer. R C Jackson, 201 Mortimer Av, Rutherford NJ 07070, (201) 939-2421.

### NEW UNCLASSIFIED POLICY

*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 must be clearly noncommercial, typed double spaced on plain white paper, contain 75 words or less, 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. 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: BYTE magazine issues volume 1, issue 1 thru January 1977 inclusive Also have FOR SALE May 1977 issue Best offer will be contacted M M Kashani, 9862 La Jolla Farms Rd, La Jolla CA 92037.

FOR SALE: BYTE Issues 1, 2 and 3; 5 thru 16 (1976 complete); Vol II Issues 2 thru 12 (1977 except January), Vol III Issues 2 thru 4 (1978 February thru April). Best offer. Jim Taylor, 6720 N 75th St, Milwaukee WI 53223, (414) 353 1977.

FOR SALE SYM 1 microcomputer brand new, used one day, in box with warranty and manuals $200. Jim Riffle, 175 Piedra Loop, White Rock NM 87544, (505) 672 9541

WANTED: Digital Group equipment Needed are one each of the following. Z 80 processor card, TVC 64 TV interface, MB 4 mother board and an I O F parallel interface card. Will accept assembled or unassembled boards in good condition and in working order with documentation Name your price. Gregory P Cramer, POB 31, APO San Francisco CA 96555

FOR SALE. PDP 8 A equipped with programming console, options one and two, and 16 K core memory. ASR33 Everything works. Asking price for everything $3000. J Woodbury, Sleepy Hollow Rd, Atkinson NH 03811, (603) 362 4256 after 7 PM.

FOR SALE. Technico Super System 16. TI TMS9900 16 bit processor with RS 232C, erasable read only memory program, 2 K read only memory, 2 K programmable memory and 2 K eraseable read only memory plus additional 32 K programmable memory Also chassis, power supply, digital cassette tape drive and software. Factory assembled and tested Only one year old Asking $1600 John M Sullivan, 721 E Harding Av, Appleton WI 54911, (414) 734 3239

FOR SALE. Fully operational ELF II computer system with system cabinet and video-radio frequency modulator plus program manuals and *1802 Club* Newsletter $175 and shipping cost. Shaji Jacob, 827 Lincoln, Fort Morgan CO 80701

WANTED: TRS 80 Radio Shack computers Any condition, any quantity. Immediate cash available. Portacom briefcase ASCII printing terminals with integral modem $595 ASR33 Teletypes $495 and $595. DEC PDP 8 E and all peripherals. Also repair, buy and custom design DEC equipment peripherals. Tom Perera, POB 632, W Caldwell NJ 07006, (201) 226 9185.

FOR SALE: Micro Term ACT IVB with 20 mA cl and printer port options. Excellent condition, 500 hours use. Includes number pad, full editing, 80 by 24 display, block send, more. Full documentation. Originally $900, sell $800 (I pay shipping) Don Markuson, 56 Dellwood Av, Chatham NJ 07928, (201) 635-9615

FOR SALE: HP 67 like new, out of warranty, 100% operational, includes 120 new magnetic cards, two solutions manuals, programming pads, two boxes of paper roll, standard accessories, etc. $500 will ship UPS in US. Brewer Pedin, 121 Hearthstone Dr #2, Blacksburg VA 24060, (703) 951 0539.

# wirewrap

## PRECUT WIRE

### #30 WIRE KITS

| #1 $7.95 | | | | #2 $19.95 | | | |
|---|---|---|---|---|---|---|---|
| 250 | 3" | 100 | 4½" | 250 | 2½" | 250 | 5" |
| 250 | 3½" | 100 | 5" | 500 | 3" | 100 | 5½" |
| 100 | 4" | 100 | 6" | 500 | 3½" | 250 | 6" |
| | | | | 500 | 4" | 100 | 6½" |
| | | | | 250 | 4½" | 100 | 7" |

| #3 $24.95 | | | | #4 $44.95 | | | |
|---|---|---|---|---|---|---|---|
| 500 | 2½" | 500 | 4½" | 1000 | 2½" | 1000 | 4½" |
| 500 | 3" | 500 | 5" | 1000 | 3" | 1000 | 5" |
| 500 | 3½" | 500 | 5½" | 1000 | 3½" | 1000 | 5½" |
| 500 | 4" | 500 | 6" | 1000 | 4" | 1000 | 6" |

Choose One Color or Random Assortment:
Red, Blue, Green, Yellow, White, Orange, Black.

* #26 Prices on Request

#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.

| In. | 100 | 500 | 1000 | 5000 |
|---|---|---|---|---|
| 2½ | 1.04 | 2.98 | 5.16/K | 4.67/K |
| 3 | 1.08 | 3.22 | 5.65/K | 5.06/K |
| 3½ | 1.13 | 3.46 | 6.14/K | 5.46/K |
| 4 | 1.18 | 3.20 | 6.62/K | 5.87/K |
| 4½ | 1.23 | 3.95 | 7.12/K | 6.25/K |
| 5 | 1.28 | 4.20 | 7.61/K | 6.62/K |
| 5½ | 1.32 | 4.48 | 8.10/K | 7.03/K |
| 6 | 1.37 | 4.72 | 8.59/K | 7.43/K |
| 6½ | 1.60 | 5.37 | 9.84/K | 8.48/K |
| 7 | 1.66 | 5.63 | 10.37/K | 8.91/K |
| 7½ | 1.73 | 5.89 | 10.91/K | 9.33/K |
| 8 | 1.78 | 6.15 | 11.44/K | 9.79/K |
| 8½ | 1.82 | 6.41 | 11.97/K | 10.19/K |
| 9 | 1.87 | 6.76 | 12.51/K | 10.62/K |
| 9½ | 1.92 | 6.93 | 13.04/K | 11.05/K |
| 10 | 1.99 | 7.26 | 13.57/K | 11.48/K |
| Addl In. | 15 | 60 | 1.20/K | 1.00/K |

### EDGE CARD CONNECTORS

| | | | |
|---|---|---|---|
| 44 pin Solder Tail | $1.95 | $17.50/10 | ALL are Gold |
| 100 pin Solder Tail | $3.95 | $35.00/10 | 100 pin are IMSAI |
| 100 pin Wire Wrap | $3.95 | $35.00/10 | spacing. |

## WIRE WRAP TOOLS

### HOBBY WIRE WRAP TOOL


| | |
|---|---|
| **BW 630 (Back Force)** | **$34.95** |
| BT 30 Extra Bit | 2.95 |
| BT 2628 #26 Bit | 7.95 |
| Batteries & Charger | 11.00 |

### INDUSTRIAL WIRE WRAP TOOL


| | |
|---|---|
| **BW 928** | **$49.95*** |
| BW 928BF (Back Force) | 52.95* |
| #30 Bit & Sleeve | 29.50 |
| #26 Bit & Sleeve | 29.50 |
| Batteries & Charger | 11.00 |

### ELECTRICAL INDUSTRIAL WIRE WRAP TOOL


| | |
|---|---|
| **EW 7D** | **$85.00*** |
| EW 7D BF (Back Force) | 92.90* |
| #30 Bit & Sleeve | 29.50 |
| #26 Bit & Sleeve | 29.50 |

*Industrial Tools do not include Bit & Sleeve
Spring Loaded bit on Back Force models.

### 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 |

### OK PRODUCTS

| | | |
|---|---|---|
| WD 30 | 50 ft. Wire Dispenser Red, White, Blue, or Yellow | $3.75 |
| WD-30-TRI | TRI Color Dispenser | 5.50 |
| R-30-TRI | Refill for TRI Color | 3.75 |
| INS 1416 | 14 & 16 pin Insertion Tool | 3.25 |
| MOS 40 | 40 pin Insertion Tool | 7.50 |
| EX-1 | IC Extractor Tool | 1.49 |
| H-PCB-1 | Hobby PC Board | 4.99 |
| WSU 30 | Hand Wrap/Unwrap/Strip Tool | 6.25 |
| WSU 30M | Same as WSU30 with Modified Wrap | 7.50 |

### SOLDERLESS BREADBOARDS

**SK 10 $16.50**

2.2" x 6.5"


**MB10 $18.95**

**SK10 mounted on board W74 Binding Posts & Rubber Feet**

### WIRE WRAP SOCKETS

| | 1-9 | 10-24 | 25-99 | 100-249 | 250-999 |
|---|---|---|---|---|---|
| 8 pin* | .40 | .36 | .34 | .31 | .27 |
| 14 pin* | — | .39 | .37 | .34 | .32 |
| 16 pin* | — | .42 | .40 | .36 | .34 |
| 18 pin* | .70 | .60 | .55 | .50 | .45 |
| 20 pin | .90 | .80 | .75 | .65 | .62 |
| 22 pin* | .95 | .85 | .80 | .70 | .65 |
| 24 pin | .95 | .85 | .80 | .70 | .65 |
| 25 pin strip | 1.25 | 1.15 | 1.00 | .95 | .90 |
| 28 pin | 1.25 | 1.15 | 1.00 | .95 | .90 |
| 40 pin | 1.65 | 1.45 | 1.35 | 1.20 | 1.10 |

**Gold 3-Level Closed Entry Design** ***End & Side Stackable**

2-Level Sockets Available

### LOGIC PROBE

**PRB-1**

**$34.95**


- Compatible with all Logic Families
- 10 Nsec pulse response

## PAGE DIGITAL ELECTRONICS

135 E. Chestnut St. #5 Monrovia, CA 91016 (213) 357-5005

**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

Catalog available on request.

Dealer Inquiries Invited

Circle 298 on inquiry card.

# 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 the list. Add a 15-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. This helps us bring you a bigger BYTE.*



**Correspond directly with company.*

## BOMB—BYTE's Ongoing Monitor Box

| Article No. | ARTICLE | PAGE |
|---|---|---|
| 1 | **Smith: Graphic Input of Weather Data** | 18 |
| 2 | **Ciarcia: Sound Off** | 34 |
| 3 | **Albus: A Model of the Brain for Robot Control** | 54 |
| 4 | **Schmucker: The Mathematics of Computer Art** | 105 |
| 5 | **Dvorak: Sonic Anemometry for the Hobbyist** | 120 |
| 6 | **Powers: The Nature of Robots** | 134 |
| 7 | **Howe: Creativity in Computer Music** | 158 |
| 8 | **Chaffee: Quest** | 176 |
| 9 | **Helmers: Photo Essay: Physical Hardware of a New Computer Backplane** | 194 |
| 10 | **Grogono: Mouse, A Language for Microcomputers** | 198 |
| 11 | **Maurer: Subroutine Parameters** | 226 |

### April BOMB Reaches for the Stars

Results of the April BOMB put simulation at the top of the charts.

The first place prize of $100 went to Mark Dahmke for "A Simulated View of the Galaxy," page 66. Two second place winners, Mark Zimmerman for "Simulation of Physical Systems," page 26 and Randy Smith for "Smart Memory," Part 1, page 54, will receive $50 each. "Microcomputer Time Sharing," page 224, by Kenneth J Johnson placed third.

# There's A World of Difference In Computer Software!

## Graham-Dorian's Integrated, On-Line Programs Are Fast, Efficient, And Easy To Use.

Rely on Graham-Dorian, a full-line computer software manufacturer, for sophisticated programs — the most detailed on the market today. They're ready to go to work immediately or to be tailored for even more specific needs.

On-line capabilities enable you to make a single entry and update all affected files. An inquiry into a file at any time provides up-to-date information — no batching or sorting of input data.

The programs are easy to use. Messages on the video display guide you each step of the way. Programs make use of indexed sequential and chained files for fast and convenient retrieval of data with efficient use of disk space.

Order in standard eight-inch disk either double or single density, or various mini-floppy formats. Each program contains a free CBASIC-2, plus user's manual and hard copy source listing.

- **Accounts Receivable** — Records invoices, prepares statements and trial balance reports, etc. Automatically reports aging of accounts in periods of 30, 60, and 90 days, with each item listed separately.
- **Accounts Payable** — Vendor lookup and change, entering vendor invoices, writing checks (many options), cash flow analysis, accounts payable check register, and vendor list. Ideal for analyzing expenditures by vendor and by due date.
- **General Ledger** — Includes lookup and change, making journal entries, trial balance, transaction register, chart of accounts, financial statements, and monthly closing.
- **Job Costing** — Provides work order lookup, enters labor transactions, material set-up, progress report of hours, labor distribution report, weekly labor reset, actual versus estimated cost per job.
- **Inventory** — Can be connected with cash register for point of sale inventory control. Number of on-line items limited only by disk space available.
- **Cash Register** — Creates daily sales reports containing information on gift certificates, paidouts, overrings, refunds, and how much in each category a salesperson sold.
- **Payroll** — Handles 100% of all necessary payroll functions including state income tax tables for your state. Ideally suited for both large and small companies.
- **Apartment** — Said one user, "Obviously, this was developed by apartment owners." The package fills virtually all the needs of apartment owners and managers. Ideal for projects with 75 units or more.
- **CBASIC-2** — The most comprehensive and powerful commercially oriented BASIC available today. Enhancements over CBASIC-1: integer variables, multiple line functions, CHAINing with COMMON variables, additional predefined functions, etc. Comes free with any package, or purchase separately.

**Compatible with many computers:** Northstar, IMSAI, Altos, Cromenco, Industrial Micro Systems, Radio Shack TRS-80, SD Systems, Digital Microsystems, Dynabyte DB8/2, Micropolis MOD II, Vector MZ, and other 8080, 8085, and Z-80-based systems.

**See your GDSS dealer or send for information packet and sample runs.**

GDE

**Graham-Dorian Software Systems**
A Division of Graham-Dorian Enterprises
211 N. Broadway / Wichita, KS 67202 / (316) 265-8633

Circle 152 on inquiry card.

# OHIO SCIENTIFIC DOES IT AGAIN

Ohio Scientific has taken its standard Challenger III computer and married it to the new Shugart 29 Megabyte Winchester Drive. The result is the C3-C. This new microcomputer now fills the vacuum that existed for computer users who need more mass storage capability than floppies can offer – yet until now, could not justify the additional cost of a larger capacity hard disk computer such as our C3-B 74 Megabyte disk system.

### Winchester Technology

Winchester hard disk drives offer small business and professional computer users the logical solution to mass storage problems that are beyond the capability of floppy disks. In addition, Winchester disks feature a track seek-time that is much better than floppies and because they spin at eight times the rate of floppies, Winchesters have a shorter latency. Both of these points reflect one remarkable speed advantage Winchester disks have over floppies.

### Coupled to the Challenger III Computer

Ohio Scientific's award winning Challenger III computer is a classic. It is the only computer series that utilizes the three most popular microprocessors – 6502A, 68B00 and Z-80. This tremendous processor versatility enables one to utilize a seemingly endless selection of quality programs available from Ohio Scientific's software library as well as from many independent suppliers.

### And Advanced Software

For instance, there are single user, multi-user and network operating systems. A complete turnkey small business package, OS-AMCAP provides accounts receivable, accounts payable, disbursements, cash receipts, general ledger, etc. OS-CP/M offers a complete FORTRAN and COBOL package. And there is WP-2, a complete word processing system. For information management, OS-DMS, features an advanced file handling system and program library that simplifies information storage and recall and routinely performs tasks which usually require special programming on other systems.

### Yields the Microcomputer of the Future

With an eye toward the future, the C3-C, like all other Challenger III's was designed with provisions for future generation 16 bit microprocessors via plug-in options.

There are ten open slots for lots of I/O and multiuser operation. Truly, the Ohio Scientific C3-C is a computer with a future.

## The new C3-C computer with 29 Megabyte Winchester Hard Disk.


**OHIO SCIENTIFIC**

1333 S. CHILLICOTHE RD., AURORA, OHIO 44202 (216) 562-3101

OEM pricing available

Circle 290 on inquiry card.