The 9-inch screen of the CT-VM monitor ($175) shown here with Southwest's new CT-64 illustrates the terminal's 64-character lines, switchable control character printing, and word highlighting. At just $500 for both, these matching units provide a complete CRT terminal with full cursor control, 110-1200 Baud serial interface, and many other features.

**Now $325 buys a 64-character terminal kit**

Our new CT-64 terminal kit gives you scrolling, full cursor control, 128-character ASCII display (with both upper and lower case), and two 1K memory pages. It's usable with any 8-bit computer.

Add our optional fully assembled 12 MHz CT-VM monitor for another $175 and you'll have the best CRT terminal buy offered anywhere.

The CT-64 gives you full cursor control, home-up and erase, erase to end of line or end of frame, cursor on/off, screen reversal, scroll or page, solid or blinking cursor, page selection, and end-of-page warning beeper.

The CT-64's features include:
- 64 or 32 characters per line (16 lines)
- Premium display with both upper and lower case letters, and descenders (g, j, etc.)
- Two 1K pages of 8-bit memory
- Scrolling or page mode operation
- 32 control character decoding
- Prints control characters (selectable)
- 128-character ASCII set
- 110/220 Volt 50-60 Hz power supply
- Highlights words with reversed background
- Optional 9-inch monitor with matching cover available
- Complete with keyboard, power supply, 110-1200 Baud serial interface, and case
Meet the most powerful \( \mu C \) system available for dedicated work. Yet it’s only $595.*

Here’s the muscle you’ve been telling us you wanted: a powerful Cromemco microcomputer in a style and price range ideal for your dedicated computer jobs—ideal for industrial, business, instrumentation and similar applications.

It’s the new Cromemco Z-2 Computer System. Here’s some of what you get in the Z-2 for only $595:

- The industry’s fastest \( \mu P \) board (Cromemco’s highly regarded 4 MHz, 250-nanosecond cycle time board).
- The power and convenience of the well-known Z-80 \( \mu P \).
- A power supply you won’t believe (+8V @ 30A, +18V and -18V @ 15A — ample power for additional peripherals such as floppy disk drives).
- A full-length shielded motherboard with 21 card slots.
- Power-on-jump circuitry to begin automatic program execution when power is turned on.
- S-100 bus.
- Standard rack-mount style construction.
- All-metal chassis and dust case.
- 110- or 220-volt operation.

DEDICATED APPLICATIONS

The new Z-2 is specifically designed as a powerful but economical dedicated computer for systems work. Notice that the front panel is entirely free of controls or switches of any kind. That makes the Z-2 virtually tamper-proof. No accidental program changes or surprise memory erasures.

FASTEST, MOST POWERFUL \( \mu C \)

Cromemco’s microcomputers are the fastest and most powerful available. They use the Z-80 microprocessor which is widely regarded as the standard of the future. So you’re in the technical fore with the Z-2.

BROAD SOFTWARE/PERIPHERALS SUPPORT

Since the Z-2 uses the Z-80, your present 8080 software can be used with the Z-2. Also, Cromemco offers broad software support including a monitor, assembler, and a BASIC interpreter.

The Z-2 uses the S-100 bus which is supported by the peripherals of dozens of manufacturers. Naturally, all Cromemco peripherals such as our 7-channel A/D and D/A converter, our well-known BYTESAVER with its built-in PROM programmer, our color graphics interface, etc., will also plug into the S-100 bus.

LOW, LOW PRICE

You’ll be impressed with the Z-2’s low price, technical excellence and quality. So see it right away at your computer store—or order directly from the factory.

Z-2 COMPUTER SYSTEM KIT (MODEL Z-2K) (includes 4 MHz \( \mu P \) card, full-length 21-card-slot motherboard, power supply, one card socket and card-guide set, and front panel; for rack mounting) .............. $595.

Z-2 COMPUTER SYSTEM ASSEMBLED (MODEL Z-2W) (includes the above as well as all 21 sockets and card guides and a cooling fan; for rack mounting) . . . . . . . . . . . . . . . . . . . . . . . . . . $995.

*kit price
To make your computer more useful—a wide choice of memory, I/O, CPU

Your computer’s usefulness depends on the capability of its CPU, memories, and I/O interfaces, right? So here’s a broad line of truly useful computer products that lets you do interesting things with your Cromemco Z-1 and Z-2 computers. And with your S-100-compatible Altairs and IMSAls, too.

**CPU**
- **Z-80 MICROPROCESSOR CARD.** The most advanced µP card available. Forms the heart of our Z-1 and Z-2 systems. Also a direct replacement for Altair/IMSAI CPUs. Has 4-MHz clock rate and the power of the Z-80 µP chip. Kit (Model ZPU-K): $295. Assembled (Model ZPU-W): $395.

**MEMORIES**
- **16K RAM.** The fastest available. Also has bank-select feature. Kit (Model 16KZ-K): $495. Assembled (Model 16KZ-W): $795.
- **THE BYTESAVER—an 8K capacity PROM card with integral programmer.** Uses high-speed 2708 erasable PROMs. A must for all computers. Will load 8K BASIC into RAM in less than a second. Kit (Model BSK-K): $145. Assembled (Model BSW-W): $245.
- **16K CAPACITY PROM CARD.** Capacity for up to 16K of high-speed 2708 erasable PROM. Kit (Model 16KPR-K): $145. Assembled (Model 16KPR-W): $245.

**I/O INTERFACES**
- **FAST 7-CHANNEL DIGITAL-ANALOG I/O.** Extremely useful board with 7 A/D channels and 7 D/A channels. Also one 8-bit parallel I/O channel. Kit (Model D + 7A-K): $145. Assembled (Model D + 7A-W): $245.
- **DIGITAL INTERFACE (OUR NEW TU-ART).** Interfaces with teletype, CRT terminals, line printers, etc. Has not one but two serial I/O ports and two 8-bit parallel I/O ports as well as 10 on-board interval timers. Kit (Model TRT-K): $195. Assembled (Model TRT-W): $295.
- **JOYSTICK.** A console that lets you input physical position data with above Model D + 7 A/D card. For games, process control, etc. Contains speaker for sound effects. Kit (Model JS-1-K): $65. Assembled (Model JS-1-W): $95.

**PROFESSIONAL QUALITY**
You get first-class quality with Cromemco.

Here are actual quotes from articles by independent experts: “The Cromemco boards are absolutely beautiful”...“The BYTESAVER is tremendous”...“Construction of Cromemco I/O and joystick are outstanding”...“Cromemco peripherals ran with no trouble whatsoever.”

Everyone agrees. Cromemco is tops.

**STORES/Mail**
So count on Cromemco. Look into these Cromemco products at your store. Or order by mail from the factory.

We wish you pleasure and success with your computer.

---

Cromemco
incorporated
Specialists in computers and peripherals
2432 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400

Circle 2 on inquiry card.
In the Queue

Foreground

12 HOW TO COMPUTERIZE YOUR MODEL RAILROAD
   Applications—Brown
36 THIS CIRCUIT MULTIPLIES
   Arithmetic Hardware—Hall
52 THE TV OSCILLOSCOPE
   Hardware—Barbier
64 SPEECH RECOGNITION FOR A PERSONAL COMPUTER SYSTEM
   Peripherals—Boddie
92 A MACHINE CODE RELOCATOR FOR THE 8080
   Software—Zolman
118 HOW FAR—WHICH WAY?
   Applications—Pittet
122 ADD CURSOR CONTROL TO YOUR TVT II
   Hardware—McGahoe
126 GIVING KIM SOME FANCY JEWELS
   Hardware—Grater

Background

42 CONDENSED REFERENCE CHART FOR THE 6800
   Reference Materials—Borrmann
44 A TRAIN CONTROL DISPLAY USING THE LSI-11 MICROCOMPUTER
   Applications—Hart-Badger
72 HOW TO PICK UP A DROPPED BIT
   Error Theory—Maurer
78 GIVE YOUR MICRO A MEGABYTE
   System Design—Grappel
82 AN INTRODUCTION TO NUMBERS
   Software—Simmons
96 BASICALLY BASIC
   Software—Baker

Nucleus

4 In This BYTE 124 Classified Ads
6 This Elephant Never Forgets 128 PAPERBYTES Forum
10 Letters 142, 166 Programming Quickies
26 West Coast Computer Faire 151 Book Reviews
32, 115, 129 What's New? 155, 158 BYTE's Bits
34, 150 Technical Forum 156 PAPERBYTES: Floppy Disk
41 Ask BYTE 158 Object Code
57 BYTE's Bugs 160 Clubs, Newsletters
60 Desk Top Wonders: Jeu de
   NIM, Peut-Être? 176 BOMB
118 How Far—Which Way?
   Applications—Pittet
122 Add Cursor Control to Your TVT II
   Hardware—McGahoe
126 Giving Kim Some Fancy Jewels
   Hardware—Grater

Cover: COMPUTER ENGINEERING by Robert Tinney

BYTE is published monthly by BYTE Publications Inc., 70 Main St., Peterborough NH 03458. Address all mail except subscription to above address; phone (603) 924-7217. Address all editorial correspondence to the editor at the above address. Unacceptable manuscripts will be returned if accompanied by sufficient first-class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of BYTE. Address all subscriptions, change of address, complaints to BYTE Subscriptions, PO Box 361, Arlington MA 02174; phone (617) 646-3229.

Second class postage paid at Peterborough NH 03458 and at additional mailing offices—USPS Publication No. 102410. Subscriptions are $12 for one year, $22 for two years, and $32 for three years in the USA and its possessions. Add $5.50 per year for subscriptions to Canada and Mexico. $25 for a one year subscription by surface mail worldwide. Air delivery to selected areas at additional rates available upon request. $25 for a one year subscription by air delivery to Europe, or DM 30.00 — for a one year subscription through our European distributor in Darmstadt.

Single copy price is $1.00 in the USA and its possessions, $2 in Canada and Mexico, and $3 elsewhere, or DM 7.00 — through our European distributor in Darmstadt.

Foreign subscriptions and sales should be remitted in United States funds. Printed in United States of America. Entire contents copyright © 1977 by BYTE Publications Inc. All rights reserved.
Sometimes a bit of serendipity falls out of an application or project. Ken Barbier describes one such case in the form of his technique of using a television display circuit's ability to generate a raster with various synchronous patterns to debug and verify its operation. Read The TV Oscilloscope.

There are many ways to make a computer talk, but how do you get it to listen? Speech Recognition for a Personal Computer System discusses a topic which has fascinated and frustrated experimenters for years. Author James Boddie of Bell Labs (the pioneer researchers in the field) covers the history of the subject and presents a practical system which can be realized by a personal computing experimenter.

Real world systems may not be perfect, yet programming demands perfection. (If we drop a bit in one machine instruction, it becomes another instruction altogether.) As a background discussion of a large subject, W Douglas Maurer presents some information on How to Pick up a Dropped Bit using some of the techniques of error detection and correction.

What's a sure cure for memory megalomania? Why, Give Your Micro a Megabyte as outlined in Robert Grappel's short tutorial on some large memory design techniques which will prove adaptable to microcomputer systems. There should be virtually no reason to complain if his ideas were put into practice more universally.

An Introduction to Numbers, by Webb Simmons, serves as an introduction to the concepts of fixed, scaled and floating point numbers. Here you'll find some basic forms for each type, how the forms differ from each other, and how each can be used.

If you've ever been frustrated by the drudgery involved in relocating machine language programs with nothing but toggle switches, then Leor Zolman's A Machine Code Relocator for the 8080 is for you! Just enter six key pieces of information and the program does the rest automatically, even fixing up all your address references.

For beginners first learning about computers, we often get requests for some basic information. In BASICally BASIC, Robert Baker gives an informal introduction to the nature of the BASIC language and its uses in programming. Finding out what a typical BASIC can do is a good starting point in your personal evaluation of products available in the personal computing marketplace.

Given latitude and longitude of two points on the earth, how do you calculate the distance and bearing? If you use a little BASIC program by Rene Pittet, you can answer the question of How Far - Which Way? using a small processor.

If you have a Southwest Technical Products' TVT II, there is a simple circuit that you can add that will give you manual and computer control over the cursor's movement, erase and bell functions. In his article, Add Cursor Control to Your TVT II, Brother Thomas McGahee describes this simple circuit which can be attached to a TVT II.

Poor KIM. If one puts KIM-1 inside a fancy case, the built-in jewels of keyboard and displays get hidden. But Robert Grater comes to the rescue by Giving KIM Some Fancy Jewels, which consist of a remote set of displays connected to the processor by cable.
The Megabyte Memory

Until today, the largest memory you could fit and address in a single microcomputer CPU was 65K.

Now, IMSAI presents an incredible memory system for micros 16 times more powerful than yesterday's best.

Imagine, a full megabyte of power from sixteen 65K RAM boards.

And, to control all this, the IMSAI Intelligent Memory Manager (IMM), the super control board.

You can write protect blocks throughout the full megabyte. Or, map in 16K blocks.

Plus, preset 16 mapping configurations with protect for high speed transfer or rapid change.

All interrupts are fully vectored, and there's an interrupt if an attempt is made to write into protected memory.

There's even a real "time of day" clock.

65K, 32K and 16K RAM Boards

Until today, the most memory you could plug into a single slot was 16K.

Now, IMSAI presents memory boards in astonishing multiples of sixteen: 65K, 32K and 16K low power, dynamic RAM Boards. They can be used in any S-100 bus computer individually or in combination to form conventional systems up to 65K bytes.

Every board is fast. With "hidden refresh" and no "wait state".

The Complete Megabyte Microcomputer System

The IMSAI Megabyte Micro™ is only part of the story. The full system can include dual floppy disks, terminals, plotters, printers and tape cassettes.

IMSAI also offers the finest high level and peripheral software available: Paper tape and Tape Cassette I/O and super Disk Operating Systems. Plus, BASIC and Disk BASIC with more high level languages coming.

Until today, the microcomputer's potential was just something you talked about.

Now, you can put it to work. Powerfully.

Circle 3 on inquiry card.
This Elephant Never Forgets

By Carl Helmers

In a press release dated March 9 1977, Texas Instruments Inc made it official: a commercial bubble memory product is now available for general use, initially in sample quantities priced at $200 per chip. The device is the TBM0103, a single chip 92,304 bit storage element composed functionally of 144 shift registers with 641 bits per register. (Internally, according to TI, there are 157 registers, but the large size of the chip carries with it the attendant probabilities of defects, so discretionary manufacturing steps are used to connect 144 good registers per package.) The device has a complicated mechanical structure which includes two permanent magnet bias field elements, two orthogonal coils which modulate the bias field to produce a net rotating magnetic field in the garnet film bubble chip, and, of course, a magnetic shield to isolate the chip from external stray magnetic fields. The dynamic data characteristics of the chip are a bit transfer rate of 50,000 bits per second and an access time of 4 ms setup prior to transferring a 144 bit word of data into or out of the device. The procedure for writing a 144 bit word

Continued on page 58
NOW $2750.
AMERICA'S LOWEST-PRICED
PERSONAL COMPUTER SYSTEM
WITH COLOR VECTOR GRAPHICS.

By taking advantage of the new technologies available to the industry today, we've consistently been able to give you one of the best prices on the market. Now because of great response, we can give you the best price. You can now buy the Compucolor 8001 for the reduced price of $2750. A complete stand-alone system with expanded graphics software for plotting points, vectors and bargraphs on a 160 x 192 addressable grid—in color. Eight independent background and foreground colors.

The Compucolor 8001 has an Intel 8080 CPU, 34 I/O ports and a color display with an effective band width of 75 MHZ compared to 5 MHZ for standard TV sets. In fact the Compucolor is the only totally integrated system on the market which includes a color display. You can also have special options for the Compucolor 8001 right now, including: Mini Disk Drives for extra memory, light pens and a variety of special keyboard features.

BASIC 8001 Is Easy To Learn. Compucolor's BASIC 8001 is a conversational programming language which uses English-type statements and familiar mathematical notations. It's simple to learn and easy to use, too. Especially when it comes to intricate manipulations or expressing problems more efficiently. The BASIC 8001 Interpreter runs in ROM memory and includes 26 statement types, 18 mathematical functions, 9 string functions and 7 command types for executing, loading, saving, erasing, continuing, clearing or listing the program currently in core.

Expandable Memory To 64K. The Compucolor 8001 has 11K bytes of non-destructible read-only memory which handles the CPU and CRT operating systems as well as BASIC 8001. Sockets are in place for an additional 21K of EPROM/MROM memory. The Random Access Main Memory has 8K bytes for screen refresh and scratch pad, 8K bytes for user workspace and room for 16K bytes of additional user workspace. The Compucolor also comes complete with a convenient mass storage device, Floppy Tape Memory. It's an 8-track continuous loop tape system, with a Baud rate of 4800 and an extra storage capacity of up to 1024K bytes per tape.

Color Graphics At Alphanumeric Black And White Prices. That's what we're becoming famous for, and thanks to the tremendous response to the Compucolor 8001, we've been able to reduce our price even lower—to $2750. Look over our dealer listing on the adjacent page for the dealer nearest you. Then drop by for a demonstration. And while you're checking out the Compucolor 8001, check out your dealer's financing plan. He can help you turn a good deal into a good deal more.

Compucolor Corporation, P.O. Box 569, Norcross, Georgia 30091.

Compucolor Corporation

Circle 4 on inquiry card.
One Sol-20 equals three computers.

To do real work with any computer, big or small, it takes a complete system. That's one of the nice things about the Sol-20. It was built from the ground-up as the heart of three fixed price computer systems with all the peripheral gear and software included to get you up and on the air.

Sol System I costs just $1649 in kit form or $2129 fully burned in and tested. Here's what you get: a Sol-20 with the SOLOS personality module for stand alone computer power, an 8192 word memory, a 12" TV/video monitor, a cassette recorder with BASIC software tape and all necessary cables.

Sol System II has the same equipment plus a larger
capacity 16,384 word memory. It sells for $1883 in kit form; $2283 fully assembled.

For even more demanding tasks, Sol System III features Sol-20/SOLOS, a 32,768 word memory, the video monitor, Helios II Disk Memory System and DISK BASIC Diskette. Price, $4237 in kit form, $5037 fully assembled and tested.

And remember, though we call these small or personal computer systems, they have more power per dollar than anything ever offered. They provide performance comparable with mini-computer systems priced thousands of dollars more.

The Small Computer Catalog for the rest of the real computer system story.

Visit your local computer store for a copy of our fully illustrated 22 page catalog. Or you may write or call us if more convenient. Please address Processor Technology, Box B, 6200 Hollis Street, Emeryville, CA 94608, (415) 652-8080.
See Sol here...

ALABAMA
Birmingham:
The Byte Shop (205) 253-2828

ARIZONA
Phoenix:
Byte Shop Phoenix (602) 253-2828

CALIFORNIA
Berkeley:
The Byte Shop (415) 647-3828
Cupertino:
Micro Computer Systems (408) 572-3828

COLORADO
Boulder:
Byte Shop (303) 449-2222

CONNECTICUT
West Hartford:
Electronic Data Systems (203) 247-2222

FLORIDA
Miami:
The Byte Shop of Miami (305) 254-2923

COLORADO
Boulder:
Byte Shop (303) 449-2222

COLORADO
Denver:
Micro Computer Systems (303) 254-2923

CONNECTICUT
West Hartford:
Electronic Data Systems (203) 247-2222

FLORIDA
Miami:
The Byte Shop of Miami (305) 254-2923

OHIO
Columbus:
The Computer Store (614) 455-0867

ILLINOIS
Champaign:
The Numbers Racket (217) 322-3425

MISSOURI
Kansas City:
Atari Toaster (816) 222-3000

NEW YORK
East Meadow, L.I.:
The Computer Shop of Long Island (516) 794-1310

OHIO
Columbus:
Cybershop (614) 239-3223

OKLAHOMA
Oklahoma City:
High Technology (405) 322-3000

OREGON
Beaverton:
Byte Shop (503) 264-3266

Eugene:
The Real Oregon Computer Co. (503) 264-3266

PORTLAND
Byte Shop Computer Store (503) 223-3406

RHODE ISLAND
Warwick:
Computer Power, Inc. (401) 736-8477

SOUTH CAROLINA
Columbia:
Byte Shop (803) 771-824T

TEXAS
Houston:
Byte Shop (713) 977-6644

TENNESSEE
Kingsport:
Microproducts & Systems (615) 248-8081

VIRGINIA
Richmond:
The Micro Store (804) 231-1302

WASHINGTON
Bellevue:
Byte Shop Computer Store (206) 726-8934

WEST VIRGINIA
Wheeling:
The Byte Shop of Long Island (203) 254-2923

The Byte Shop of Miami (305) 254-2923

ONTARIO
Ottawa, Ontario:
Tridentine (613) 262-9767

TORONTO, Ontario:
Futurist Computer (416) 482-0090

CANADA
Ottawa, Ontario:
Tridentine (613) 262-9767

TORONTO, Ontario:
Futurist Computer (416) 482-0090

Burlington, Ontario:
Kodak Computer Store, Ltd. (416) 339-7717

THE COMPUTER PLACE (416) 339-7717

WHITESTONE, New York:
The Byte Shop (516) 771-8241

VANCOUVER, B.C.:
Pacific Computer Store (604) 438-3201

Circle 5 on inquiry card.

ALTAIR (S-100) BUS QUESTION

I am using an 8080 based microprocessor system to control a video imaging system. A number of circuit boards are in use or being built which plug into the Altair style S-100 bus and work with video. Some of these boards require video sync signals (horizontal drive, vertical drive, composite blanking, composite sync and subcarrier). I would like to buffer these and put them on the bus as TTL signals. Could you please describe the known uses to date of the uncommitted pins of the Altair style S-100? I am looking for suggestions of a standardized place to put these five signals.

Carl Geiger

Can any readers supply an answer? One place to ask your question would be to call the engineering departments of various companies which make peripheral for the Altair bus.

ON PRINTERS

On April 1, 1977, volume 2 number 4 arrived with the fine article by M. S. McNatt. I look forward to the next issue with the interfacing information. I truly believe that this is a most overlooked situation: the use of Baudot machines with the small computer. It would appear that many, many more people would be interested in the small computer if there was some method of retaining ASCII throughout the computer except in the use of the Baudot machine for input and output.

I must, however, take issue with the noise and smell of the Model 15. A Model 15 that is properly lubricated and adjusted, with its cover in place, should be something other than noisy; the cover should shield the owner from smell. If smoke comes out of the machine, something is wrong.

I note that Mr. McNatt did mention the Western Union strip printers; the model number does not come to mind, but I believe that they were 2-B. I saw an advertisement on this type of machine and have reason to believe that the keyboard of such a machine will not adapt the CRT or the video reperforator which usually (to my knowledge) comes without a keyboard.

Also, there is the Friden Flexowriter. My law office has a battery of those machines. The first one was acquired in 1958 and it clatters and bangs happily all day long. (Flexowriters are generally noisy, particularly the FL and SDS models.) Why is there not mention of
these machines? They are extremely sturdy and it is my understanding that one can still get maintenance for them. Many have input and output jacks or plugs.

Lucius B. Dabney Jr
POB 947
Vicksburg MS 39180

A way of ducking the issue of your last question would be to point to the title of the article: "A Guide to Burndot Teletypes." As for oil and smoke, Mr McNatt's caricature of the Model 15 certainly fits my reminiscences of using one on the amateur radio bands when I was in college... CH.

DOES ANYONE HAVE THE LATEST ON 8080 FORTRAN?

Nine months ago, when I began to get serious about my own microcomputer, everybody was promising a FORTRAN compiler "just around the corner." A major manufacturer promised a FORTRAN compiler "soon" which degrades to "early 1977." Now they don't mention FORTRAN in their advertisements. As time goes on we hear less and less of a FORTRAN compiler. Here I sit with 40 K, access to timesharing and a pile of FORTRAN programs I can't use. I am a user, not a hardware man or a systems man. I would like to compile these and other FORTRAN programs one way or another and get on with using my machine.

FORTRAN may be a pterodactyl as Mr Lashley suggests, but it is a language commonly used to solve problems by those more interested in problem solving than in technique. Some of the hardware nuts would have us scrap everything every time a new chip came along. In that event we would do nothing except construct new hardware. Mr Lashley would have us scrap years of training every time a new programming idea appeared. That's OK for someone whose main interest is learning new languages. I don't feel we must scrap FORTRAN any more than Ma Bell has scrapped the old phone system. Sure, there have been problems, but by insisting the new must work compatibly with the old, she has built a good system for the user. The purists working for Bell who wish to scrap everything every time a new idea is discovered probably have ulcers.

So let us continue to build upon what we have and not scrap everything every time something new comes up. If this is the way things are done, I'll be updating my system far into the future. Now, I'd like to see some way of converting that pile of FORTRAN programs.

Lawson T Pierce
4891 Songbird Dr
Columbus OH 43229

LASHING LASHLEY

I nominate P M Lashley (February 1977 BYTE, page 78) for the "Cybersnob of the Year" award for 1977. The year is young but I think he should get the nod; no one can possibly top his useless, self-serving, supercilious, unnecessary attack on another man's worthy efforts.

Roxton Baker
56 South Rd
Ellington CT 06029

APL INTEREST, AND CHARACTER GENERATION TUTORIAL NEEDED

I am writing in response to letters in the February 1977 issue of BYTE to say that I am building a microprocessor based APL capability. I plan to build an APL CRT terminal and will want a character generator chip. Since I plan to build my own but have been unsuccessful in finding a construction and theory article, BYTE could include such an article, or at least a bibliography, in a future issue.

Karl Gerhard
264 Taylor Rd
Portsmouth RI 02871

APL INFORMATION WANTED

Would you please send me some information concerning implementation of APL on microcomputers? Thank you for your consideration.

Gerry Wong
7930 Mercure
Brossard Quebec
CANADA J4Y 1A8

Watch future issues of BYTE.

COVER KUDOS

I had to write, though somewhat belatedly, to comment on the January 1977 cover painting by Robert Timney. It was, I must admit, a rather emotional moment when I slid the wrapper off to reveal a painting which depicts exactly the feelings and aspirations I have felt for some time now. The hope that the power of the computer, in the hands of all the right people, can somehow change the world from the smog filled grey and brown outside the window into the clear and beautiful vision seen through the terminal.

May I suggest that this cover be made available in poster size through BITS. While I did enjoy the Thomas Jefferson cover, I feel that we in the world of computing should be looking to the future, and that this particular painting

Continued on page 146

MULTIPLE DATA RATE INTERFACING FOR YOUR CASSETTE AND RS-232 TERMINAL

the CI-812

The Only S-100 Interface You May Ever Need

On one card, you get dependable "KC-standard"/biphase encoded cassette interfacing at 30, 60, 120, or 240 bytes per second, and full-duplex RS-232 data exchange at 300- to 9600-baud. Kit, including instruction manual, only $89.95*.

PERCOM DATA COMPANY, INC.
4021 RUNDLE--GARNETT, TEXAS 76322
(214) 276-1968

*Assembled and tested, $119.95. Add 5% for shipping. Texas residents add 5% sales tax, BAC/MC available.
Figure 1: A simple train layout for one train showing the manner in which electricity is transmitted to the motor of the train.

One of the many uses of computers, and especially personal computers, is in the area of real time process control operations. An enjoyable way of experimenting with this area would be to have your small computer control a model train layout. Those of us who attended the Personal Computing 76 convention held in Atlantic City NJ in August undoubtedly saw the Digital Equipment Corporation’s display of a train layout controlled by an LSI-11 computer. There is no reason why you could not do something similar with your own system.

Since many of you may not be familiar with the techniques used in model railroading, let’s start with a bit of background information. Figure 1 shows the simple oval layout most of us probably had as children in the US, where DC power has dominated the scene recently. Power is applied to each rail. One set of wheels on the engine picks up the positive side and feeds it to the motor; the other set of wheels picks up the negative. Unfortunately, this type of system can only handle one train running at a given time. If a second train were put on the tracks, we would not be able to control the two trains’ speeds independently. To solve this problem, we go to block control, shown in figure 2. As you can see, the negative rail is common, as it was in figure 1. The difference lies in using insulators on the positive rails and two power packs. Now we can use switches so that either power pack can control either section of track. This allows us to have a train on the smaller loop (block 1) and another train on the run around track to the right (block 2), and we can control each train’s speed independently through the two power packs. We still have to be a little careful when we operate this system, because if the train on block 2 runs onto block 1 while block 1 is occupied we will be right back where we started, with two trains running at the same speed and controlled by one power pack.

How to Computerize

David C Brown
1704 Manor Rd
Havertown PA 19083

12
Your Model Railroad

In reality, then, two blocks are insufficient. A moderate size model railroad is likely to have from 12 to 24 blocks. Just before a train enters a block, the operator would flip that block’s control to his power pack. When the train leaves a block, the operator would turn off that block so that the other operator could bring his or her train into the block if he or she wanted to.

The only other really important consideration in train layouts that we should discuss is the reverse loop. Suppose we had a loop as in figure 3, so that the train could go around the loop and end up on the same track but going the opposite direction. If you follow the polarities of the tracks around the loop, you will see that they end up as a short circuit, with “+” connected to “-” and vice versa. This condition requires the use of a special type of block, known as a reverse block or reverse loop. The reverse block is similar to the other blocks we have discussed, except that in this case, both tracks have insulators instead of just one. This is illustrated in figure 4. We have also added some other switches for reversing train directions. A normal power pack has a reverse switch that switches the polarity of its output to make the train run in the other direction. With reverse loops, we can no longer use this switch. Instead, we need a

Figure 2: A basic example of block wiring. The negative rail is common to both blocks. The positive rail is divided by insulators into two sections. Each block can run a train using a separate power pack.

Figure 3: An illustration of the type of difficulty that arises when a track is put through a reverse loop. Following polarities it will be seen that the negative side of the track will connect to the positive side at the end of the loop and vice versa.
Figure 4: Using insulators on both the positive and negative tracks a block is created that is totally isolated from the main track. Using a separate power pack will allow the train to run through the block, turn around, and return to the main track.

direction switch for the reverse loop and another direction switch for the rest of the track. By playing with these switches, we can keep the polarities correct and prevent a short circuit. This is easier to see in an illustration. Assume that the train is coming down the main track from the left of figure 4 and that the track switch is positioned to the right so that the train will end up going around the reverse block in a counterclockwise direction. The polarities of the main and reverse block on each side of the A insulators are the same, so there is no problem crossing the insulators. If we leave everything the way it is though, we will have a problem when the train hits the B insulators, since the polarities are reversed on opposite sides of the insulators. In this case, while the train is on the reverse block, we flip the main direction switch. This reverses the polarity of the main track so that it is the same as the reverse block's polarity at the B insulators. Now the train can leave the reverse block and go back onto the main track with no problems. Suppose the polarity of the reverse block had been reversed when the train was coming down the main track. In this case, we would have had no problem leaving the reverse block, but would not have been able to enter it in the first place. The solution would be to flip the reverse block direction switch while the train was on the main track. This would match up the polarities at the A insulators. Then, after the train got onto the reverse block, we would reverse the main direction as we did before so that the train could safely leave the reverse block.

Two items were omitted from figure 4 for the sake of simplicity in the prior discussion. At point C we would tie in the second power pack and the block control switches of figure 2. At point D we would need a 2 pole 2 position switch to allow either power pack to control the reverse loop.

So much for train layouts and block control! The only other major piece of equipment on a model railroad is the track switch, called turnouts in model railroad terminology. Turnouts allow you to vary the route that your train takes, which is certainly more interesting than watching your train do nothing but go around in a circle. A turnout represents the actual track that can be switched. It physically can sit in one of two states which are electrically controlled. The electrical components for moving the turnout are called switch machines. Figure 5 shows a typical turnout and switch machine. The turnout contains a movable section. In
figure 5, the train will go straight. If the movable section were pulled down, the train would go to the left relative to its motion, up the turnout track. The switch machine is really just a pair of solenoid coils. A soft metal bar with an S shaped groove slides between the coils. The movable section of the turnout has a lever connected to it. The other end of this lever rides in the groove of the sliding bar of the switch machine. In figure 5, if the reverse switch was closed, coil B would be energized and the soft bar would slide to the right. The lever would ride in the groove and be pulled down, thus pulling down the movable section of the turnout. Since the coils draw close to 1 A when energized, they should only be energized long enough to move the turnout. If left energized for more than a few seconds, they could overheat and burn out. For this reason, push button switches are normally used for controlling the switch machines. The switch machines normally run off an alternating current accessory power pack and have no relation to the block system. They represent a separate, isolated set of circuits on the model railroad.

All of these electrical requirements may sound somewhat confusing. Linn Westcott has an excellent, very inexpensive paperback book entitled How to Wire Your Model Railroad from Kalmbach Publishing Company, 1959. This book, or one like it, should be available at any hobby shop. Another very good set of paperbacks is published by Atlas Tool Company, a leading manufacturer of model railroad equipment. Atlas has also simplified the wiring for you. They sell sealed switch units containing printed circuitry for all of the wiring I have discussed. The Atlas selectors contain block control circuitry, the Atlas controllers contain direction control and reverse loop circuitry, and the Atlas switch controllers contain the turnout control switches. These devices are inexpensive and have the ability to be plugged together for handling any size train system. All you do is connect your power packs, block wires and switch machine wires to the Atlas devices and your wiring is complete. The Atlas books also diagram track layouts, usually 6 to 8, from simple to complex, give construction details, and show how to wire them with the Atlas devices. What more could you ask for?

Now that I have you hooked, or at least interested in model railroading, let's take a look at what we need to control this system by computer. All of the principles discussed so far still apply. The only additional item that we need on the train layout itself is some way of knowing where the trains are.

The simplest approach would be to use magnetic reed switches. As shown in figure 6, these switches are nothing but two contacts, very close together, and sealed in a glass envelope. If we place these switches at intervals along the tracks and glue a small magnet to the bottom of our trains, the train going over the glass capsule will close the switch. These switches, by the way, are common in burglar alarm systems and other proximity sensing devices. I will refer to these switches along the tracks as markers in the remainder of this article.

The train control system to be discussed is currently under development. So far, the development has concentrated on the software electronics.

Figure 5: The basic workings of a turnout and switch machine. The two coils of wire move a piece of soft metal with a groove back and forth between them. The switch machine rides in the groove. When coil B is activated, by using the reverse mode, the metal is pulled into the B coil and the switch machine slides down in the groove changing the position of the switch machine.

Figure 6: This is a magnetic reed switch which can be used as a marker along the track. The switch consists of two contacts that are enclosed within a glass envelope. The two metal contacts are activated magnetically when a car with a magnet attached passes over.
ware aspects, and most of the hardware details are yet to be worked out. These I must leave up to your ingenuity at the moment. You will need a way of controlling the switch machines. The October 1975 issue of BYTE, page 87, showed one way this could be done with a flip flop, several 2N3766 transistors, and a oneshot multivibrator. You will need a way of controlling the block switches, allowing you to turn a block off or to route power to it from any of your power packs. You will need a way of switching the direction controls, both main and reverse loop, as discussed with figure 4. Along with this, you will need a way of reading the polarity between the current block and the next block, probably using the marker number as the address. Fortunately, you only need this capability at entrances and exits from reverse blocks. You will need a way of controlling speed. This could be done with a digital to analog converter, or you could adapt the speed controller shown in the January 1975 issue of Popular Electronics, page 58. Last, but definitely not least, you will need the marker system.

The markers, as mentioned earlier, are the switches located along the track for sensing the location of the trains. When the train passes over a marker, an interrupt could be generated. The interrupt software must be able to read in a binary number identifying which marker was hit. This could be accomplished on an 8080 based system by cascading together Intel 8214 Priority Interrupt ICs, as shown in figure 7. Ideally, we would
like a large number of markers. Although I haven't the room to discuss this aspect, a graphics display could be used so that the track layout is displayed and each marker would light up as the train passed it. For this, we want many markers. The circuit of figure 7 will allow up to 40, but needs five 8214 chips, at $15 each. Another possibility would be to use a 256 word by 8 bit read only memory and have the markers control the address lines into the memory, which would be programmed with the marker numbers. I would be interested in hearing from anyone who has a cheap way of debouncing and encoding 256 switches into a byte of data that could be read in by an interrupt routine. For that matter, I would be interested in hearing from people on any of these hardware requirements, since I haven't worked them out myself yet.

So much for hardware. The software should actually be easier to explain than model railroading theory. The entire software package is table driven. This makes it easier to understand and easier to adapt to a wide variety of train layouts. The four major tables used are the marker table, the block table, the switch table and the train table. The marker table really controls most of the system. Each marker on the track has a corresponding entry in the marker table. When the train hits a marker, an interrupt is generated. Interrupts are probably preferable to a polled system, since it is crucial that none be missed and unless the marker event is latched, a busy processor could miss it. If your program is processing a console command and a train passes a marker before you get to poll it, you could have real chaos. The marker interrupt routine, then, would read in the marker number and look it up in the marker table. Table 1 shows the layout of an entry in the marker table. If a graphics display is used, MXY would be used to update the display, showing the train at the new marker location. MBLOCK would be the pointer to the block table entry for the block which contains this marker. MSETUP indicates that this is the last marker in the block. The software would go to the block table, figure out the next block that the train will enter, and issue a command to apply power to that block (remember figure 23). MFREE tells the software that the marker is the first one in the block and power should be removed from the prior block. This is the reverse of MSETUP.

MFA and MFD are fractional accelerators and fractional decelerators. Fractionals represent a small change in the speed of the train. Any marker can be set up to add or subtract a fractional to the train's current speed. This would be used to speed up the train as it goes up a hill, called a grade, or slow it down on a downgrade. It could really be used any place you wanted the train's speed to automatically change. Fractionals can also be used for programmed stops. The MSTOP indicator in the marker table entry tells the system that the train should stop. However, a sudden stop would be unrealistic, so we can use the fractional decelerator. If the operator requests a stop at a particular stop number and that stop number equals MSTOPNO, the system will look at MSTOP and MFDS. If MFDS is on, the fractional decelerator will be applied, slowing down the train. MFDS also will prevent the fractional decelerator from being applied if a stop was not requested. Eventually the train will hit a marker that has the correct MSTOPNO and has the MSTOP indicator on, and the train will be stopped completely.

Each block on the track layout has a corresponding block table entry, as shown in table 2. The block table entry contains indicators for what train is in the block and whether the block is a reverse block (figure 3 and 4 discussion). The reverse indicator will be used by the marker processing. When power is to be applied to the next

<table>
<thead>
<tr>
<th>Label</th>
<th>Byte</th>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMARKER</td>
<td>1 to 2</td>
<td>1 to 8</td>
<td>marker number</td>
</tr>
<tr>
<td>MBLOCK</td>
<td>3</td>
<td>1 to 8</td>
<td>block table entry displacement</td>
</tr>
<tr>
<td>MSTOPNO</td>
<td>4</td>
<td>1 to 8</td>
<td>stop number</td>
</tr>
<tr>
<td>MSETUP</td>
<td>5</td>
<td>1</td>
<td>setup next block indicator</td>
</tr>
<tr>
<td>MSTOP</td>
<td>5</td>
<td>2</td>
<td>stop indicator</td>
</tr>
<tr>
<td>MFDS</td>
<td>5</td>
<td>7</td>
<td>apply fractional decelerator on stop only</td>
</tr>
<tr>
<td>MFREE</td>
<td>5</td>
<td>8</td>
<td>release prior block indicator</td>
</tr>
<tr>
<td>MFA</td>
<td>6</td>
<td>1 to 4</td>
<td>fractional accelerator value</td>
</tr>
<tr>
<td>MFDF</td>
<td>6</td>
<td>5 to 8</td>
<td>fractional decelerator value</td>
</tr>
<tr>
<td>MXY</td>
<td>7 to 8</td>
<td>1 to 8</td>
<td>XY coordinates for graphics display routine</td>
</tr>
</tbody>
</table>

Table 1: Marker Table Information. The purpose of the marker table is to organize the current information associated with each marker input from the model railroad layout. The list of information shown here is described in the text in more detail. The use of a graphic display of the layout is assumed by putting in display coordinate fields (MXY), which might be omitted otherwise. The basic marker entry information is repeated in this format for each marker.

<table>
<thead>
<tr>
<th>Label</th>
<th>Byte</th>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBLOCK</td>
<td>1</td>
<td>1 to 8</td>
<td>block number</td>
</tr>
<tr>
<td>BTRON</td>
<td>2</td>
<td>1 to 4</td>
<td>which train is currently in the block</td>
</tr>
<tr>
<td>BREV</td>
<td>2</td>
<td>8</td>
<td>reverse block indicator</td>
</tr>
<tr>
<td>BPROR</td>
<td>3</td>
<td>1 to 8</td>
<td>prior block ID</td>
</tr>
<tr>
<td>BNEXT</td>
<td>4</td>
<td>1 to 8</td>
<td>next block ID</td>
</tr>
</tbody>
</table>

Table 2: Block Table Information. The purpose of the block table is to organize the current information about each block in the model railroad layout. The block is identified by its number. (Although this field might be omitted if numbers and position in the table are made identical.) If the block currently has a train, its number is identified. The topology of the track network is reflected in the prior block and next block pointers.
Table 3: Switch Table Information. The switch table entries describe each switch in relation to the railroad layout by means of pointers to the “from,” “through” and “turnout” blocks. This information, which is relatively fixed for a given layout, is used to modify the prior and next block information in the block table according to the settings of the switch. One such entry is needed for each switch in the layout. (A roundhouse would have to be a special case ...)

<table>
<thead>
<tr>
<th>Label</th>
<th>Byte</th>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSWITCH</td>
<td>1</td>
<td>1 to 8</td>
<td>switch number</td>
</tr>
<tr>
<td>SPOS</td>
<td>2</td>
<td>1</td>
<td>through or turnout indicator</td>
</tr>
<tr>
<td>SFROM</td>
<td>3</td>
<td>1 to 8</td>
<td>from block ID</td>
</tr>
<tr>
<td>STHRU</td>
<td>4</td>
<td>1 to 8</td>
<td>through block ID</td>
</tr>
<tr>
<td>STURN</td>
<td>5</td>
<td>1 to 8</td>
<td>turnout block ID</td>
</tr>
<tr>
<td>SNMKTHRU</td>
<td>6</td>
<td>1 to 8</td>
<td>setup next marker ID for through block</td>
</tr>
<tr>
<td>SNMKTURN</td>
<td>7</td>
<td>1 to 8</td>
<td>setup next marker ID for turnout block</td>
</tr>
<tr>
<td>SPMKTHRU</td>
<td>8</td>
<td>1 to 8</td>
<td>release prior marker ID for through block</td>
</tr>
<tr>
<td>SPMKTURN</td>
<td>9</td>
<td>1 to 8</td>
<td>release prior marker ID for turnout block</td>
</tr>
</tbody>
</table>

![Diagram](image)

**Figure 8a:** An example of a marker being between two blocks. In this type of array the computer must only determine whether the train will go into block 2 or 3.

**Figure 8b:** This figure is slightly more complex than figure 8a. In this type of array the train may enter block 3 or block 2. The computer must determine which block the train will enter. If it is block 3 the information for block 3 must be updated for the particular train as soon as the train has passed marker 1. If the train will enter block 2 next, however, the information for block 2 must not be updated until the train has passed marker 2.

block, the software will check to see if either the current or the next block has the reverse block indicator on. If neither block is a reverse block, power is applied to the next block. However, if either block is a reverse block, we must do additional processing. Power is applied, but we have to worry about the polarity. Therefore, we read in the polarity between the two blocks, probably using the marker number as the address for the polarity interface. We cannot use the block number, since the polarity will be different at the two ends of the reverse block. If, after reading the polarity between the two blocks, we find that they are equal, we are done. If they are different and the current block is a reverse block, then we must reverse the polarity of the main track. If they are different and the current block is not a reverse block, we must reverse the polarity of the reverse block. These are the same operations that we discussed with figure 4, only the computer will do them rather than the operator. The BPRIOR and BNEXT fields in the block table entry are pointers to the block table entries for the prior and next block on the track layout. These pointers are maintained and modified by the switch processing.

When the operator requests that a particular switch position be changed, the software will locate the switch entry in the switch table. Table 3 shows the layout of a switch table entry. SPOS, the position indicator, will be toggled and the graphics display, if used, will be updated to show the new switch position. The new value of SPOS will be output to the switch interface. SFROM, STHRU, and STURN are used to update the prior and next block fields of the block table entries so that each block entry points to its correct successor and predecessor. SNMKTHRU, SNMKTURN, SPMKTHRU, and SPMKTURN are used to turn on and off the MSETUP and MFREE indicators in the marker table. For example, if the switch is set for turnout, MSETUP in the marker pointed to by SNMKTHRU will be turned off and will be turned on in the marker pointed to by SNMKTURN. SPMKTHRU and SPMKTURN do the same with the MFREE indicator in the marker table. The reason for this may not be apparent, so let’s look at an example. In figure 8a, this processing would not be needed, MSETUP in marker 1 would always be on, and MFREE could be left on in both marker 2 and marker 3 since there is no other way into block 2 or 3 other than from block 1. However, figure 8b is a different case. Here we have the switch in the middle of a block. If the switch were set for the
turnout, the train would leave block 1 and enter block 3. Marker 1 would have MSETUP on and marker 4 would have MFREE on. Now let’s change the switch position. The train will no longer go into block 3. The switch processing would have updated block 1’s block table entry so that block 2 was the next block. If we try to set up the next block (block 2) according to marker 1, we will be premature. Maybe there is another train in block 2 at the moment, since there could be quite a bit of track between the switch and the beginning of block 2. Likewise, if a train comes down block 3 and tries to free the prior block according to marker 4, we might have a similar problem. The switch processing, then, must turn off MSETUP in marker 1 and turn off MFREE in marker 4 if the switch is set for through and turn them on if the switch is set for turnout. In this particular case, markers 2 and 5 could have MSETUP on all the time, markers 3 and 6 could have MFREE on all the time, SNMKTHRU and SPMKTHRU will be zero, and SNMKTURm and SPMKTURN will have a 1 and a 4 respectively.

The train table of table 4 has one entry for each train that you are running. This table entry is used primarily for keeping track, pardon the pun, of where each train is. It specifies what block the train is in, the last marker hit, the train’s direction (forward or reverse), the train’s current speed, and the port address for the train’s power pack interface. This interface would control the speed, through digital to analog converters or similar, and the main and reverse block direction controls. The train table entry would be updated by marker processing and by speed and direction commands issued by the operator.

So far, we have discussed primarily the automatic functions of the software. Now let’s look at the functions controlled by the operator. We already discussed switches and stops. That leaves us with direction and speed. Suppose we want to reverse a train. First, this command would not be allowed if the train was moving, since we might cause a derailment if we suddenly reversed a fast moving train. Other than that, it’s fairly simple. We store an indicator in TDIR of the train table entry to indicate that the train is going in the reverse direction and then issue the command to reverse either the main polarity or the reverse block polarity, depending on whether or not the train is currently in a reverse block. From then on, we reverse logically the meaning of the MSETUP and MFREE indicators in the marker table entries and we reverse logically the meanings of the prior and next block pointers in the block table entries. Therefore, we set up the next block when MFREE is on and free the prior block when MSETUP is on. BNEXT points to the prior block and BPRIOR points to the next block.

Train speed control can be somewhat more difficult. There are basically two ways to control speed. The easiest would be to use a digital to analog converter. Trains run on 0 to 16 VDC, so a converter would be needed capable of providing this range at approximately 1 A per train to be run at a time. Therefore, if you plan on running two trains at the same time, you need a 0 to 16 VDC supply rated at 2 A minimum. This idea works well with the fractional system outlined earlier, but is not very realistic. The operator would have to enter on the console the voltage to be applied and the software would output to the converter the correct value. To gradually speed up a train would keep the operator busy for a while entering speed commands.

A more realistic approach would be to enter an accelerate command that would gradually speed up the train at a uniform rate. When the proper speed is reached, the operator would enter a hold command to maintain the current speed. To slow down, a decelerate command would cause the train to gradually slow down, and the hold command would again be used to stop decelerating. This type of control could be accomplished by adapting the circuit on page 58 of the January 1975 Popular Electronics to operate as an IO device rather than by push buttons. This complicates the fractional system, since the fractionals were intended to represent actual voltage quantities to be applied to the train’s speed.

A similar approach could be to still use digital to analog converters, but to have the software periodically increment or decrement the value applied to the converter if an accelerate or decelerate command was in effect. This method will satisfy both realism and the operation of the fractionals, but depends upon the construction of the soft-

<table>
<thead>
<tr>
<th>Label</th>
<th>Byte</th>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTRAIN</td>
<td>1</td>
<td>1 to 4</td>
<td>train number</td>
</tr>
<tr>
<td>TDIR</td>
<td>1</td>
<td>B</td>
<td>direction</td>
</tr>
<tr>
<td>TBLOCK</td>
<td>2</td>
<td>1 to 8</td>
<td>current block ID</td>
</tr>
<tr>
<td>TMARKER</td>
<td>3 to 4</td>
<td>1 to 8</td>
<td>current marker ID</td>
</tr>
<tr>
<td>TSPEED</td>
<td>5</td>
<td>1 to 8</td>
<td>train’s current speed</td>
</tr>
<tr>
<td>TPORT</td>
<td>6</td>
<td>1 to 8</td>
<td>IO port address for controller</td>
</tr>
</tbody>
</table>

Table 4: Train Table Information. Each train in the layout must also be followed in the software as it runs its route. The information can be formatted and referenced with a structure such as this. The details vary for each train with time, as the train progresses through the layout.
Marker Interrupt Processing Routine

The M algorithm is used for marker interrupt processing. It reads in the marker number, performs operations on it, including block processing (B algorithm) if necessary, determines any change in speed and then returns from the interrupt.

M1. Read the marker number input and look it up in Marker Table.
M2. Use MBLOCK of Marker Table entry to point to Block Table entry.
M3. Use BTRAIN of Block Table entry to point to Train Table entry.
M4. Store marker and Block Table pointers in TMARKER and TBLock.
M5. If TDIR = 0 and MSETUP is on then go to B1.
M6. If TDIR = 1 and MFREE is on then go to B1.
M7. If TDIR = 0 and MFREE is on then go to B11.
M8. If TDIR = 1 and MSETUP is on then go to B11.
M9. Send MXY to graphics display if it is used.
M10. If MFA > 0 then add MFA to TSPEED and send TSPEED output to the speed controller.
M11. If MFD > 0 and MFDS is off then subtract MFD from TSPEED and send TSPEED output to the speed controller.
M12. If MFD > 0 and MFDS is on and stop has been requested at MSTOPNO then subtract MFD from TSPEED and send TSPEED output to the speed controller.
M13. If MSTOP is on and a stop has been requested at MSTOpNO then store 0 in TSPEED and send TSPEED output to the speed controller.
M14. Return from interrupt processing routine.

Switch Command Routine

The S algorithm performs the switch command processing by resetting all of the markers that are being affected by a particular train. It is called by an opposite command.

S1. Look up the switch number in the Switch Table.
S2. Reverse value of SPOS (0-through, 1-turnout).
S3. If SPOS = 0 then use STHRU as the pointer to next block and go to S5.
S4. Use STURN as the pointer to the next block.
S5. Store the pointer to the next block in BNEXT of the Block Table entry pointed to by SFROM.
S6. If SPOS = 1 then go to S14.
S7. Turn off MSETUP in Marker Table entry pointed to by SMKTURN.
S8. Turn on MSETUP in Marker Table entry pointed to by SMKTHRU.
S9. Turn off MFREE in Marker Table entry pointed to by SMKTHRU.
S10. Turn on MFREE in Marker Table entry pointed to by SMKTHRU.
S11. Send SCOS output to the switch controller using SSWITCH as address.
S12. If the graphics display is used then update it to show new switch position.
S13. Return to operator command processor.
S14. Return to operator command processor.
S15. Turn off MSETUP in the Marker Table entry pointed to by SMKTHRU.
S16. Turn on MFREE in the Marker Table entry pointed to by SMKTHRU.
S17. Turn on MFREE in the Marker Table entry pointed to by SMKTHRU.
S18. Go to S11.

Reverse Direction Command Routine

The R algorithm performs the reverse direction command processing. If the train is not moving, the polarity of the block will be reversed and TDIR will be logically inverted.

R1. If TSPEED in the current Train Table entry = 0, then indicate invalid request and return to command processor.
R2. Reverse the present value of TDIR in the current Train Table entry.
R3. Use TBLock of the current Train Table entry to point to the current Block Table entry.
R4. If BREV of the current Block Table entry = 1 then go to R7.
R5. Send command output to reverse the polarity of the main track.
R6. Reverse to operator processor.
R7. Send command output to reverse polarity of reverse block.
R8. Return to operator processor.
ware to increment or decrement at a reasonable pace.

Table 5 shows a list of operator commands that might be used on a typical computerized model railroad. The remainder of this article is devoted to some algorithms for the more important processes we have discussed. Algorithms, if the term is unfamiliar, are simply flowcharts expressed in words rather than in blocks and arrows. Since this article is not presenting actual code and is more directed to theory than to bit by bit details, generalized algorithms seem more appropriate. I hope this article has increased your interest in the area of model railroading and has given you some ideas on how you can simulate, on a small scale, the types of real time process control systems that could be used on a large scale to control an actual railroad.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Command meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Tx</td>
<td>accelerate train x</td>
</tr>
<tr>
<td>D Tx</td>
<td>decelerate train x</td>
</tr>
<tr>
<td>H Tx</td>
<td>hold train x at current speed</td>
</tr>
<tr>
<td>S Tx</td>
<td>stop train x immediately</td>
</tr>
<tr>
<td>S Tx Sy</td>
<td>stop train x at stop number y</td>
</tr>
<tr>
<td>FA xxx My</td>
<td>set fractional accelerator value of xxx into marker y</td>
</tr>
<tr>
<td>FD xxx My</td>
<td>set fractional decelerator value of xxx into marker y</td>
</tr>
<tr>
<td>F0 xxx My</td>
<td>same as above but also turn on MFDS</td>
</tr>
<tr>
<td>R Tx</td>
<td>reverse train x</td>
</tr>
<tr>
<td>Sx</td>
<td>toggle position of switch x</td>
</tr>
<tr>
<td>Ex My1,y2,---,yn</td>
<td>establish stop x at markers y1,y2,---,yn and turn on MSTOP in marker yn</td>
</tr>
<tr>
<td>Dx</td>
<td>delete stop x</td>
</tr>
<tr>
<td>K</td>
<td>kill system; emergency power off</td>
</tr>
<tr>
<td>C</td>
<td>close down system; order stop of trains and power off</td>
</tr>
</tbody>
</table>

Table 5: Definition of a Command Language. Here is mnemonic set of commands for control of the trains on a model railroad layout from a terminal keyboard. When a general purpose keyboard is used, this sort of sequence is required — although the particular commands defined here are not necessarily the only set possible. Algorithms S and R (see separate boxes) are command processing routines for switch toggling and reversal of trains, two of the more complex commands in this list.

GLOSSARY

Algorithm: A method used to express the steps to be taken in solving a problem. It may be expressed in words, as a flowchart of blocks, lines and arrows, or in other forms.

Block control: Sectioning up a railroad layout so that multiple power packs can be used to run more than one train on the tracks at a time.

Block table: A table used in the software for controlling a model train layout. Each block on the tracks has a corresponding entry in the block table.

Digital to analog converter: Commonly abbreviated as DAC, the device takes as input a digital value, typically an "n" bit binary number, and outputs an analog signal in the form of a variable voltage corresponding to the digital input value.

Fractions: As used in this article, a fractional is a 4 bit binary number representing a small change in the speed of the train. It is used for automatic speed changes under program control.

Graphics display: A video display device that can display lines, curves and drawings rather than just characters. In a train system, a graphics display could be used to show a real time diagram of the train layout and where each train is located.

Insulators: In a train layout, insulators are used to isolate sections of track in order to use block control. Since power is supplied to the train through the tracks, insulators are placed between sections of track so that each section can be controlled independently.

Markers: Switches located on the track that are activated by the passing of a train. These markers are used by the software to keep track of where the trains are.

Marker table: A table used in the software, containing an entry for each marker on the train layout.

Polarity: As used in this article, polarity refers to which track is positive and which track is negative. Normally, this is not really important, except in the case of reverse loops, where the polarity between two adjacent blocks on the layout might be different, resulting in a short circuit when crossed by the train.

Reverse loop: Or reverse block; this is where a section of a track will allow a train to go around in a loop and end up on the same track but going in the opposite direction. Special handling of these cases is needed.

Solenoid: A coil of wire which acts as an electromagnetic. When current is applied to the coil, a bar of soft metal will be attracted toward the magnet. In model railroading, solenoids are part of a switch machine.

Switch machine: A solenoid activated mechanism attached to a turnout, used to move a section of track so that the route of the train can be modified.

Switch table: A table used in the software, containing an entry for each switch machine and attached turnout on the track.

Train table: A table containing an entry for each train that is running on the train layout (also, the physical structure on which your train layout is built).

Turnout: A movable section of track, used to modify the route of the train. A turnout allows the train to go in one of two possible directions. A switch machine is used to move the turnout. A turnout has a normal position, which is the direction most commonly taken, and a reverse position, which is the less frequently used position. In this article, I have also used turnout to refer to the reverse position, and through to refer to the normal position.
Introducing Apple II.
You've just run out of excuses for not owning a personal computer.

Clear the kitchen table. Bring in the color TV. Plug in your new Apple II, and connect any standard cassette recorder/player. Now you're ready for an evening of discovery in the new world of personal computers. Only Apple II makes it that easy.

cassette interface, so you can swap with other Apple II users.

You can create dazzling color displays using the unique color graphics commands in Apple BASIC. Write simple programs to display beautiful kaleidoscopic designs. Or invent your own games. Games like PONG—using the game paddles supplied. You can even add the dimension of sound through Apple II's built-in speaker.

But Apple II is more than an advanced, infinitely flexible game machine. Use it to teach your children arithmetic, or spelling for instance. Apple II makes learning fun.

Apple II can also manage household finances, chart the stock market or index recipes, record collections, even control your home environment.

Right now, we're finalizing a peripheral board that will slide into one of the eight available motherboard slots and enable you to compose music electronically. And there will be other peripherals announced soon to allow your Apple II to talk with another Apple II, or to interface to a printer or teletype.

Apple II is designed to grow with you as your skill and experience with computers grows. It is the state of the art in personal computing today, and compatible upgrades and peripherals will keep Apple II in the forefront for years to come.

Write us today for our detailed brochure and order form. Or call us for the name and address of the Apple II dealer nearest you. (408) 996-1010. Apple Computer Inc., 20863 Stevens Creek Boulevard, Bldg. B3-C, Cupertino, California 95014.

---

Apple II is a completely self-contained computer system with BASIC in ROM, color graphics, ASCII keyboard, light-weight, efficient switching power supply and molded case. It is supplied with BASIC in ROM, up to 48K bytes of RAM, and with cassette tape, video and game I/O interfaces built-in. Also included are two game paddles and a demonstration cassette.

SPECIFICATIONS
- Microprocessor: 6502 (1 MHz).
- Video Display: Memory mapped, 5 modes—all Software-selectable:
  - Text—40 characters/line, 24 lines upper case.
  - Color graphics—40h x 48v, 15 colors
  - High-resolution graphics—280h x 192v; black, white, violet, green (12K RAM minimum required)
- Both graphics modes can be selected to include 4 lines of text at the bottom of the display area.
- Completely transparent memory access. All color generation done digitally.
- Memory: up to 48K bytes on-board RAM (4K supplied)
  - Uses either 4K or new 16K dynamic memory chips
  - Up to 12K ROM (8K supplied)
- Software:
  - Fast extended BASIC in ROM with color graphics commands
  - Extensive monitor in ROM
- I/O:
  - 1500 bps cassette interface
  - 8-slot motherboard
  - Apple game I/O connector
  - ASCII keyboard port
  - Speaker
  - Composite video output

Apple II is also available in board-only form for the do-it-yourself hobbyist. Has all of the features of the Apple II system, but does not include case, keyboard, power supply or game paddles. $598.

PONG is a trademark of Atari Inc.

*Apple II plugs into any standard TV using an inexpensive modulator (not supplied).
Order your Apple II now.

Use this order form to get your Apple II fast. As a special offer for those who order now, we will include free a custom vinyl carrying case (a $50 value). And we will also pay shipping charges to anywhere in the continental United States.

Apple II Price List.

<table>
<thead>
<tr>
<th>RAM Complement</th>
<th>Apple II System</th>
<th>Calif. Residents Add</th>
<th>Apple II Board-only</th>
<th>Calif. Residents Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>4K</td>
<td>$1,298.00</td>
<td>$84.37</td>
<td>$598.00</td>
<td>$38.87</td>
</tr>
<tr>
<td>8K</td>
<td>1,398.00</td>
<td>90.87</td>
<td>698.00</td>
<td>45.37</td>
</tr>
<tr>
<td>12K</td>
<td>1,498.00</td>
<td>97.37</td>
<td>798.00</td>
<td>51.87</td>
</tr>
<tr>
<td>16K</td>
<td>1,698.00</td>
<td>110.37</td>
<td>978.00</td>
<td>63.57</td>
</tr>
<tr>
<td>20K</td>
<td>1,778.00</td>
<td>115.57</td>
<td>1,078.00</td>
<td>70.07</td>
</tr>
<tr>
<td>24K</td>
<td>1,878.00</td>
<td>122.07</td>
<td>1,178.00</td>
<td>76.57</td>
</tr>
<tr>
<td>32K</td>
<td>2,158.00</td>
<td>140.27</td>
<td>1,458.00</td>
<td>94.77</td>
</tr>
<tr>
<td>36K</td>
<td>2,258.00</td>
<td>146.77</td>
<td>1,558.00</td>
<td>101.27</td>
</tr>
<tr>
<td>48K</td>
<td>2,638.00</td>
<td>171.47</td>
<td>1,938.00</td>
<td>125.97</td>
</tr>
</tbody>
</table>

Memory is offered at a 20% savings when ordered with the system—or board—as reflected in the prices above.

Additional RAM can be easily added-in at a later date as your needs develop.

One set 4K chips (4K bytes) $125
One set 16K chips (16K bytes) $600

Prices and specifications subject to change without notice.

Order Form

☐ Please send me an Apple II System
☐ Board Only
  with ______ K bytes of RAM (4K minimum) at $ ______
  California Residents add 6.5% tax
  Total $ ______

Name ____________________________
Address ____________________________
City __________________ State ________ Zip ________
Phone ____________________________
Shipping Address (if different) ________

☐ Cashier’s check or money order enclosed.
  (Please allow 2 additional weeks for personal checks.)
  Please charge to my
  ☐ BankAmericard
  ☐ VISA
  ☐ Master Charge
  Card Number ____________________________
  Expiration Date ____________________________
  Signature ____________________________

Mail to: Apple Computer Inc., 20863 Stevens Creek Blvd., B3-C, Cupertino, California 95014

(408) 996-1010

Circle 7 on inquiry card.
Random Observations and Conversations
at the First West Coast Computer Faire

Notes and Photos by Lawrence F. Willard

The kids monopolized the computer games most of the time, and were better in their scoring than most of the adults. Said one man to another:

"Those kids have been at the keyboard for 15 minutes; shall we push 'em out of the way?"

"No," said the other. "They're doing a helluva lot better than I know I could do."

There were several variations of Star Trek in evidence. At the Processor Technology booth one frustrated kid kept getting this printout whenever he entered a command:

"Captain, I'm afraid your last command made no sense."

He was lucky. A kid about 12 years old playing Star Trek at the Vector Graphic Inc booth apparently made a king sized error and got this printout:

"Thanks to your bungling the galaxy has been destroyed by the Klingon War Fleet. You are demoted to cabin boy. Try again."

The boy turned around and saw me watching him.

"Don't say anything," he growled. These games reveal a lot about people to dedicated people like me. At the National Semiconductor booth a computer was set up to plot, on a screen, a graph of the player's physical, emotional and intellectual cycles. You were asked to key in your birth date and to indicate over how many days you wanted the chart plotted. I noted, curiously, that some people used arbitrary dates or somebody else's birth date rather than their own. Are some people reluctant

Larry Willard, who writes regularly for Yankee magazine (Dublin NH), acted as BYTE's roving eye at the First West Coast Computer Faire held in San Francisco's Civic Auditorium on April 16 and 17 of this year. Here are Larry's notes on Random Observations and Conversations at the event.
Rich Man, Poor Man, Merchant, Physician, Teacher, Lawyer, Student, Musician....

There's an iCOM Floppy Disk System for Everyone!

More Uses
People from every walk of life are adding iCOM® Floppy Disks to their microcomputers for such diverse tasks as payroll, inventory control, mailing lists, game playing, record keeping, parts ordering... We've uncovered some innovative applications, too: The sailboat architect who puts equations and algorithms on an iCOM disk to test his nautical theories; the student who has automated a bowling alley; the iCOM dealer who designed an environmental control system for a university.

More Speed
These users have found iCOM floppies to be much faster and more versatile than cassette or paper tape. With iCOM, programs can be loaded in seconds; files updated in minutes; hundreds of programs can be stored on a single disk.

More Models
iCOM has Frugal Floppies™, Dual Floppies, Microfloppies™ (using the new 3 ½ '' diskette), and other new approaches to floppy disk systems. Each is hardware and software compatible with Altair™, IMSAI, Poly 88, Sol-20 and other microcomputers using the Altair S-100 bus format.

More Software
Then there's iCOM's famous software: Powerful field-proven FDOS-II with macro-assembler, string-oriented text editor, and file manager. Plus easy-to-use compatible 6K Disk BASIC. Each with super features such as: named variable length files, automatic file create, open and close, multiple merge and delete... and more.

More Backup
We've been building floppies for microcomputers for more than 3 years. Long before the rest. Thousands of systems are operating perfectly in the field. And we're part of Pertec Computer Corporation, one of the largest manufacturers of peripherals, microsystems, data entry products and data processing systems. We'll be around whenever you need us.

More Dealers
Maybe not in quantity, but in quality. We've chosen our dealer network carefully to assure you of assistance every step of the way. Our prices are right. Our delivery is fast. Our dealers are experienced and knowledgeable.

Must Reading
Our free booklet, "What a Floppy Disk Can Do for You" is must reading. Send for yours today or visit your dealer.
There’s an iCOM for Everyone at these Computer Stores...

ARIZONA
Byte Shop
Tempe
(602) 384-1128

CALIFORNIA
Orange Computer Ctr.
Costa Mesa
(714) 468-0221
Byte Shop
Lawndale
(213) 371-2412
Byte Shop
Mira Loma
(714) 858-9020
Computer Mart
Orange
(714) 323-1222
Byte Shop
Pasadena
(213) 884-3311
Byte Shop
Santa Barbara
(805) 969-2638
Byte Shop
Santa Clar
(408) 249-4221
Byte Shop
San Diego
(714) 586-8008
Byte Shop
San Rafael
(415) 457-9311
Computer Room
San Jose
(408) 226-8384
Byte Shop
Tarzana
(213) 743-3919
Tech Mart
Tarzana
(213) 344-0153
Byte Shop
Thousand Oaks
(805) 497-9595
Byte Shop
Woodland Hills
(818) 992-6600
Byte Shop
Westminster
(714) 894-9131

COLORADO
Prime Radix
Denver
(303) 573-5942

FLORIDA
Byte Shop
Miami
(305) 264-2993
The Computer Store
Jacksonville
(904) 725-9158

ILLINOIS
Itty Bitty Machine
Evanston
(312) 328-6800
Numbers Racket
Champaign
(217) 352-5435
Champaign Computer
Champaign
(217) 359-5883

INDIANA
Byte Group
Indianapolis
(317) 524-1712

KENTUCKY
Cybertronics
Louisville
(502) 499-1551

MASSACHUSETTS
American Used
Computers
Boston
(617) 267-2100

MINNESOTA
Microprogramming
Burnsville
(612) 894-1129

NEW HAMPSHIRE
Computer Mart
Nashua
(603) 883-2386

NEW JERSEY
Computer Mart
Iselin
(201) 283-0600

NEW YORK
Synchro Sound
Hollis
(212) 686-7007
Computer Mart
NYC
(212) 686-7923

NORTH CAROLINA
Digital Dynamics
Charlotte
(704) 374-1527

PENNSYLVANIA
Byte Shop
Bryn Mawr
(215) 525-7712

SOUTH CAROLINA
Carolina Computers
Columbia
(803) 758-6524

TEXAS
Micro Store
Richardson
(214) 231-1095

WISCONSIN
Madison Computer
Madison
(608) 255-5552

CANADA
Computer Mart
Toronto
(416) 484-9708
Computer Place
Toronto
(416) 589-0262

Photo 3: The kid at the keyboard of this IMSAI 8080 under the watchful eye of Lynn Stone is Bob Bowdige, a fifth grader at Rollingwood School in San Bruno. At this school, 200 students each week study BASIC programming. The school was invited to become part of the exhibit at the Action Audio Electronics booth at the show.

to know what their cycles are, or are they reluctant to have other people see when their highs and lows occur, a privacy thing, perhaps? Anyway, I saw a pretty young lady who couldn’t be out of her 20s key in a

Photo 4: Victoria Van Bushirk, 10, with her father Thomas, a programmer at Lockheed in Sunnyvale CA, in the Recreational Computer Centers booth. Tom, with his interests in computer games and home accounting problems, plans to acquire a computer.
s he's Spear of NorthWest Computer Cinema. (For more details on the plotter kit, see BYTE January 1977 page 85.) Dick has a 6502 based computer similar to the KIM which the Penns are using in this setup; he wants to use it for computer controlled graphics.

birth date of October 9 1912. An arbitrary date? The birth date of a parent? Only she knows. The computer took a fair amount of time to figure it all out, but finally traced the patterns on the screen for a 20 day period.

At the Advanced Technology Research Associates booth there were flashing space jewels and Plexiglas ray guns that lit up with a pulsing bright colored light when the trigger was pulled. These fascinated kids and grownups alike, only at 24 bucks and on up the kids weren't buying. The older people were, though. I asked one guy what he planned to use his space ray gun for.

"I'm a magician," he said. "I'll find a use for it on stage."

Another guy wanted to know if the space jewels were waterproof. He thought it might be interesting to drop one of the little flashing pendants into his tropical fish tank, or into a pitcher of his favorite iced beverage. They aren't waterproof; the battery compartment would be flooded.

Like a lot of other people, I had a computer portrait made—you stand in front of a TV camera, your image is held on the TV

Photo 5: The husband and wife team of Ellen and Allen Penn (he's president, she's VP and secretary of SylvanHills Laboratory Inc, 1 Sylvanway, Box 239, Strafford MO 65757) demonstrate computer controlled plotters to Dick Spear of Northwest Computer Cinema. (For more details on the plotter kit, see BYTE January 1977 page 85.) Dick has a 6502 based computer similar to the KIM which the Penns are using in this setup; he wants to use it for computer controlled graphics.
VECTOR 1 is based on the 8080A Microprocessor and S-100 Bus -
EASY TO ASSEMBLE, EASY TO ACCESSORIZE, EASY TO BUY AND VERY EASY TO OPERATE.
screen until it can be scanned for a computer printout. In about a minute the printer rolls it out and it is torn off and handed to you. The guy in front of me in line made a production out of it. He was bearded, and he wanted a profile shot with a pipe in his mouth and his hand up to it in a studied pose. He explained that he wanted it for his office door.

There were a couple of booths playing computer music and the sound was great, even if the repertoire was a bit limited. About the 15th or so time you've heard a particular tune during the day you can draw the waveforms of each note. But the performance was pretty good. Solid State Music's booth was playing such tunes as "The Flight of the Bumblebee," Bach's "Little Fugue in G Minor," "Blues in F," "It's a Small World," and "Maple Leaf Rag." Try those on your computer.

Footweary after awhile, I went into the cafeteria, got some coffee, and sat down at a table and got into a conversation with two men sitting there. They were Manabu Uyehara and George M Hirota, who have the Radio Shack franchise in Honolulu. They had flown to San Francisco especially for the Faire. Both men have had previous computer experience, and they are anxious to carry computers in their store.

I moved to another table and met Scott M Dickson, a computer programmer in the dental school of the University of Michigan. He wants a computer to experiment with. He thinks dentists could use them for patients' records and for billing.

Coming from the cafeteria I stopped at a booth offering a selection of fine prints and etchings done in sort of a science fiction-fantasy style. One of them, I thought, would look especially fine in my study. Before I could ask how much it was, somebody else did.

"Six-fifty," said the young artist.

Real reasonable, I thought, and moved forward to buy one.

The man who had asked how much, then asked:

"Six hundred and fifty dollars?"

The artist said that was correct.

I walked away.

That's more than most of the computers were selling for.
IS YOUR DOLLAR BUYING AS MUCH POWER & FLEXIBILITY AS TDL'S XITAN SYSTEMS PROVIDE?

OUR CUSTOMERS SAY THINGS LIKE THIS ABOUT TDL PRODUCTS:

"...the best CPU board I've put together... excellent parts... worked right out without trouble shooting." IRG Cambridge, Mass.

"Great product." KMM Bella Vista, Ark.

"...high quality components, good engineering & complete documentation...up and running without any problems." WP Seattle, Wash.

"Excellent." Col. DWW Santa Maria, Calif.

"Very impressed with superb quality." Sk  Boston, Mass.

When we combined our highly praised ZPU board and our System Monitor Board, we defined the standard for the industry: we integrated more power and flexibility in two slots of our motherboard than most other systems can muster using five or more boards. When we put this setup into our rugged aluminum case we created the first XITAN system, the alpha 1. By adding a CRT terminal and/or teleprinter you will have a complete computer system.

KIT: $769 ASSEMBLED & TESTED: $1039

By adding a 216 memory module and our PACKAGE A software to the alpha 1 we created a second XITAN system, the alpha 2. Thus, a complete and extremely powerful micro-computer system emerges well worthy of you who are operating at the most sophisticated levels. The XITAN alpha 2 provides you with 18K of RAM, 2K of ROM, 2 serial I/O ports, 1 parallel I/O port, our 1200 baud audio cassette interface as well as our extraordinarily powerful software package which includes 8K Basic, the Text Output Processor, the Zapple Text Editor and the Relocating Macro-Assembler. Add your own I/O device and GO...with the most powerful and flexible micro-computer package ever offered.

KIT: $1389 ASSEMBLED & TESTED: $1749

IF YOU ARE A BEGINNER, YOU WON'T EASILY OUTGROW THE XITAN SYSTEM.
IF YOU ARE AN ADVANCED USER, YOU WILL DISCOVER XITAN IS EXACTLY WHAT YOU NEED.

Circle 10 on inquiry card.

Write for descriptive brochure on the XITAN alpha series and system software. When you ask at your dealer, say "ZY-TAN."

ORDERING INFORMATION: Send check, money order or BankAmericard, Master Charge current number and expiration date. Shipping is usually made via UPS or UPS Blue Label. Specify other arrangements if you wish. Prepaid orders are shipped postpaid.
This system, available for $2770 in kit form or $3800 assembled and tested, features the 16 bit TMS9900 processor manufactured by Texas Instruments Inc. What EBNEK has done is to design a finished package around the TMS9900 with the following characteristics:

- TMS9900 processor board.
- 16 K bytes of programmable memory on two 8 K byte boards.
- 8 K bytes of EROM socket capacity on one board, with 2 K bytes of operating system software supplied (ie: room for 6 K bytes of user defined EROM programs).
- Video display board with television monitor.
- IO interface board with keyboard interface, eight bits of parallel IO, serial data port with data rate generator, and cassette modem.
- Cabinet and power supply motherboard.
- Phi-Deck tape transport with firmware support.
- ASCII keyboard and case.

The design of the system is modular, with an addressing space limitation of 64 K bytes, and a design limit of 128 K bytes using memory bank switching. (Power adequate for a full 128 K bytes is provided as a standard feature according to the company.)

The video display provides 15 lines of either 32 or 64 characters, generated with a 7 by 9 dot matrix. The display also has a point graphics mode allowing a 256 by 240 dot display driven from an 8 K byte segment of main memory. The display and processor are interleaved so that operation of the processor does not break up the picture with visual hash, allowing programmed updates to occur while information is being displayed.

This product looks like a significant one for the personal computing user who wants the ultimate in microprocessor execution speed and performance. The TMS9900 architecture includes 16 general purpose work areas at arbitrary memory locations which use a short form of addressing. It has the unique characteristic of built-in multiply and divide operations which take (worst case without wait states, using 3 MHz clock) 17 and 41 usec to evaluate 32 bit results of 16 bit operands. Individuals with real time computational applications (as in voice recognition, scene analysis, electronic music generation) will find the 9900 architecture quite useful. In addition to the physical configuration, EBNEK's announcement of the EBNEK 77 describes availability of an assembler, editor, utilities and a user's software library.

Circle 631 on inquiry card.

Polyphonic Music Generator Board

A new interface board designed for music generation called the SRS-320 has been announced by Stillman Research Systems, POB 14036, Phoenix AZ 85063, for $175. The board, which can serve as the basis for a computerized music synthesizing system, is Altair bus compatible and can generate up to four musical notes simultaneously over a four octave range. This is sufficient to generate most elementary chords and inversions. The SRS-320 is priced at $175, and a demonstration cassette of the board is available for $3, as well as a variety of other music generation electronics.

Circle 632 on inquiry card.

New Vector Catalog Available

Vector Electronic Company Inc's 1977 catalog is now available, listing prices and descriptions of the entire Vector product line, including printed circuit cards, terminals and pins, breadboarding kits, wiring panels and so on. Of special interest to the computer experimenter is the company's new cordless rechargeable wire wrap tool called Wrap and Strap. The catalog is available free from Vector Electronic Company Inc, 12460 Gladstone Av., Sylvan CA 91342, and no serious experimenter would want to be without it.

Circle 633 on inquiry card.
PolyMorphic Systems now offers the complete, assembled, personal computer system—the POLY 88 System 16. A full 16K system with high speed video display, alphanumeric keyboard, and cassette program storage. A BASIC software package providing the most advanced features available in the personal computing market. Features like PLOT and TIME, which utilize our video graphics and real-time clock. Others like VERIFY, so that you know your tape is good before you load another. Or input type-ahead so you can tell your program to run while the tape is still loading (it stores up to 64 characters of commands or question responses to be executed). All these plus a complete package of scientific functions, formatting options, and string capabilities. With the POLY 88 System 16 you can amaze your timesharing friends the very first night!


Scientific Functions: Sine, cosine, log, exponential, square root, random number, x to the y power.

Formatted Output • Multi-line Function Definition • String Manipulation and String Functions • Real-Time Clock • Point-Plotting on Video Display • Array dimensions limited by memory • Cassette Save and Load of Named Programs • Multiple Statements per Line • Renumber • Memory Load and Store • 8080 Input and Output • If Then Else • Input type-ahead.

Commands: RUN, LIST, SCR, CLEAR, REN, CONTINUE

Statements: LET, IF, THEN, ELSE, FOR, NEXT, GOTO, ON, EXIT, STOP, END, REM, READ, DATA, RESTORE, INPUT, GOSUB, RETURN, PRINT, POKE, OUT.

Built-in Functions: FREE, ABS, SGN, INT, LEN, CHR$, VAL, STRS, ASC, SIN, COS, FND, LOG, TIME, WAIT, EXP, SORT, CALL, PEEK, INP, PLOT.

Systems Available. The POLY 88 is available in either kit or assembled form. It is suggested that kits be attempted only by persons familiar with digital circuitry.

System 2: is a kit consisting of the POLY 88 chassis, CPU, video circuit card, and cassette interface. Requires keyboard, TV monitor, and cassette recorder for operational. $735

System 16: consists of an assembled and tested System 2 with 16K of memory, keyboard, TV monitor, cassette recorder, 11K BASIC and Assembler on cassette tapes. $2250.

System 0: The circuit cards an S-100 mainframe owner needs to be compatible with the POLY 88 software library. System 0 consists of the central processor card with monitor ROM, the video circuit card, and cassette interface, all in kit form. $525.

Prices and specifications are subject to change without notice. California residents add 6% sales tax.

460 Ward Drive
Santa Barbara, Ca. 93111
(805) 967-2351

PolyMorphic Systems
A Proposed Microprocessor Software Standard

Many people are concerned about the need for standardization in the microprocessor industry. One obvious source of standards is people who are seriously designing and supporting microprocessors. The following text was provided by Mostek to document one interchange standard intended for the Mostek version of the Z-80 processor. The text was written by Peter Formaniak and David Leitch and was distributed without copyright. The standard is intended to be an upwards compatible extension of the original Intel 8080 object code format, with provision for relocatable programs and linkage of separate assembles.

Object Output Definition

Each record of an object module begins with a delimiter (colon or dollar sign) and ends with carriage return and line feed. A colon (:) is used for data records and end of file indicator for Intel compatibility. A dollar sign ($) is used for records containing relocation information and linking information. An Intel loader will ignore such information and allow loading of non relocatable, non linkable programs. All information is encoded in ASCII format.

Each record is identified by a “type.” The type appears in the eighth and ninth bytes of the record and can take five values. In the following text the record type begins each heading.

- Type 00: Data Record

<table>
<thead>
<tr>
<th>Byte number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Colon (:) delimiter.</td>
</tr>
</tbody>
</table>

ALB003212A defines last reference to external symbol 'LAB'

| 00          | FFFF 1.                            |
| 00 200E      | 2.                                 |

- Type 01: End-of-file Record

<table>
<thead>
<tr>
<th>Byte number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Colon (:) delimiter.</td>
</tr>
<tr>
<td>2, 3</td>
<td>ASCII zero.</td>
</tr>
<tr>
<td>4, 5</td>
<td>Most significant byte of start address of data.</td>
</tr>
<tr>
<td>6, 7</td>
<td>Least significant byte of start address of data.</td>
</tr>
<tr>
<td>8, 9</td>
<td>ASCII zeros. This is the &quot;record type&quot; for data.</td>
</tr>
</tbody>
</table>

Checksum of all bytes except the delimiter, carriage return, and line feed. The checksum is the negative of the binary sum of all bytes in the record.

CRLF Carriage return, line feed.

- Type 02: Internal Symbol

<table>
<thead>
<tr>
<th>Byte number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dollar sign ($) delimiter.</td>
</tr>
<tr>
<td>2, 3</td>
<td>Up to 6 ASCII characters of the internal symbol name. The name is left justified, blank filled.</td>
</tr>
<tr>
<td>8, 9</td>
<td>Record type 02.</td>
</tr>
<tr>
<td>10, 13</td>
<td>Address of the internal symbol, most significant byte first.</td>
</tr>
<tr>
<td>14, 15</td>
<td>Binary checksum.</td>
</tr>
</tbody>
</table>

CRLF Carriage return, line feed.

Continued on page 62
Fantastic Software

This LIBRARY is a complete do it yourself kit. Knowledge of programming not required. EASY to read and USE.

Written in compatible BASIC immediately executable in ANY computer with at least 4K, NO other peripherals needed.

This Library is the most comprehensive work of its kind to date. There are other software books on the market but they are dedicated to computer games. The intention of this work is to allow the average individual the capability to easily perform useful and productive tasks with a computer. All of the programs contained within this Library have been thoroughly tested and executed on several systems. Included with each program is a description of the program, a list of potential users, instructions for execution and possible limitations that may arise when running it on various systems. Listed in the limitation section is the amount of memory that is required to store and execute the program.

Each program's source code is listed in full detail. These source code listings are not reduced in size but are shown full size for increased readability. Almost every program is self instructing and prompts the user with all required running data. Immediately following the source code listing for most of the programs is a sample executed run of the program.

The entire Library is 1100 pages long, chocked full of program source code, instructions, conversions, memory requirements, examples and much more. ALL are written in compatible BASIC executable in 4K MITS, SPHERE, IMS, SWTPC, PDP, etc. BASIC compilers available for 8080 & 6800 under $10 elsewhere.

This Library is destined to become one of the reference bibles for the small computer field, due to its versatility and uniqueness and the ease of operation of the programs it contains. These volumes are deductible as a business expense when purchased by a company. Send your remittance for prompt delivery, while supplies last. Volume discounts are available to qualified dealers.

FUTURE ADDITION TO THE "BASIC SOFTWARE LIBRARY"

Volume VI (A Complete Business System $49.95) General Ledger System - Taxes, Payroll, W-2's, Inventory, Depr., Financial Statements, etc. AVAILABLE MID SUMMER

Available at most computer stores.
Add $1.50 per volume for postage and handling.

SCIENTIFIC RESEARCH

1712-B FARMINGTON COURT
CROFTON MD 21114

Phone Orders call (800) 638-9194
Information and Maryland Residents Call (301) 721-1148
We all know that algorithms for multiplication can be implemented perfectly well in software. But such programmed approaches to arithmetic operations can often prove quite slow. There are a number of possible approaches to gaining speed in operations involving multiplication. These may vary all the way from just tightening up the code to putting a full hardware floating point arithmetic unit into the processor. The latter is often expensive and difficult to build. One approach which is midway between these extremes is to implement the “inner loop” of a software algorithm as some special purpose hardware. An example of this technique is this simple binary multiplier which takes two bytes of information and supplies a 2 byte product value, leaving all the other functions of a full arithmetic package to software. By speeding up the multiplication operation, the entire arithmetic software will run much faster whenever it involves multiplication.

### How To Do Binary Multiplication

To examine the operation of a shift and add algorithm for multiplication, let’s look at the multiplication of two 4 bit bytes:

\[
\begin{array}{c|c}
0 & 1110 \\
1 & 1010 \\
\end{array}
\]

From inspection it can be seen that the result can be achieved in the same number of steps as there are bits in the byte of the multiplier, and that the answer requires as many bits as the number bits in the two bytes. In other words, two 4 bit bytes generate an 8 bit product, and the multiplication of two 8 bit bytes generates a 16 bit product. Now, by further inspection, it is possible to visualize the necessary implementation of the algorithm with hardware.

First, note that the answer requires eight bits. Let’s start off then with an 8 bit register (P) set to all zeroes:

\[
00000000
\]

Then we’ll note that when the first bit to be operated within the multiplier is a zero,

\[
1010
\]

the result is a zero; so no adding is necessary and we merely shift one place for the next operation. Now the bit to be operated with is a one,

\[
1010
\]

and one times a number is that number; so we just add the multiplicand into the P register as follows:

\[
10001100 = \text{product (P)}
\]

Then we shift over one and select the next bit to be operated on:

\[
1010
\]

Since it is a zero, only a shift is necessary. Then we operate on the last bit of the multiplier:

\[
1010
\]
Figure 2: Control logic of multiplier. The control logic for the multiplier generates a series of nine clock pulses on ACLK. The first pulse coincides with the high state of the MSTART signal, and is used by the arithmetic logic to load the X operand into the low order part of the 17 bit output shift register while clearing the high order nine bits, P15 thru P7. On successive clock cycles, the arithmetic logic of figure 1 either shifts the product right one bit, or adds the Y operand with the resulting sum being loaded with a 1 bit right shift. The operation of a typical 8 bit multiplication is shown in the state table, with the circled numbers referring to clock transitions on the timing diagram.
This is a one; so we again add the multiplicand into the P register into the position indicated by the shift:

\[
\begin{array}{c}
0, 0, 0, 1, 1, 1, 0, 0 \\
+ 1, 1, 1, 0
\end{array}
\]

\[
1, 0, 0, 0, 1, 1, 0, 0 = \text{final product in P register}
\]

Thus, the product is generated.

How It Works

The multiplication logic described for the 4 bit case is generalized to 8 bit operands with a 16 bit product in the circuit of figures 1 and 2. The multiplier logic of this design accepts two 8 bit bytes, X and Y, from which it generates the 16 bit product P. After the start signal, a mere ten clock periods suffice to generate the product value. Since TTL devices are utilized, it is possible to generate the result in less than one microsecond by using a fast enough clock. Practically speaking though, with the 1 MHz speed of the typical microprocessor clocks, a result in about ten microseconds will be quite acceptable, especially when comparison is made to the equivalent pure software approach.

The multiplier unit provides optional read and busy lines as status bits for the processor, and requires a free running clock plus a start signal from the processor. The ready and busy lines can be ignored entirely in most computers, by simply making sure that one or two instructions are executed, taking ten clock periods total between start of a multiply and reference to the product value.

The Multiplication Cycle

Let us assume that the unit is idle, waiting for an input. This idling situation is shown in the timing diagram of figure 2 at the left hand side. At idle, the BUSY output will be low and the READY output will be high. To start the multiplication, a high level is placed on the START input after the X and Y bytes are set up and presented to the unit. At the low to high edge transition of the next clock pulse after the START line is raised, the READY and BUSY lines will change state as flip flop IC6a is clocked. Within eight clock times of these transitions, the START level should be brought low again to keep the unit from restarting the multiplication upon completion of operation. These transitions raise the MSTART line to a high level, which causes the control counter IC1 to be loaded with the value 8 (binary 1000 on the D, C, B, A inputs). At this time, the X input byte is loaded into shift register IC2, and the high order product bits in shift register IC3 and flip flop IC6b are all cleared. All this action happens on the ACLK transition, labelled 1 in the timing diagram of figure 2.

The shift and add multiplication algorithm occurs during the next 8 clock transitions, labelled 2 thru 9 in the timing diagram. At the low to high transition of these clocks the binary counter IC1 is incremented, and a shift right occurs in the shift register IC2 which holds P0 to P6 of the final product and starts out with the value of X. If the shift carry output (SC) of IC2 is low, the transition will simply shift the high order portions of the product (IC6b and IC3) right by one bit, since binary 0 has no effect on the sum. But if an X bit was 1, SC will be high, which sets up control lines so that Y is added to the previous value of the product P and loaded into P with a right shift 1 bit position. (This right shift is in the wiring of the adder as shown in figure 1, where, for example, the old bit P15 plus the high order bit Y7 of Y are added together to form bit P14 of the new P value, and the new P15 is taken from the carry out of the addition process.) The operation of the algorithm is shown in the state table of figure 2 for the example 25 \times 150 = 3750.

The multiplication operation is terminated when the binary counter reaches a count of 15, thus generating a ripple carry output which resets the flip flop IC6a to a READY state at the final high to low transition of the ACLK clock during a multiplication. If the external circuitry tests READY, this indicates end of the multiplication and a valid product on the P output lines. Note that the binary counter performs one more increment to a zero count, then inhibits itself until the next START signal.

A Memory Address Space Interface

One good way to interface the multiplier is through the wiring of two addresses with appropriate output latches and decoding logic. Each address has a possible input and output significance, corresponding to whether the processor is reading or writing data to the “memory” location which is the multiplier:

<table>
<thead>
<tr>
<th>Address</th>
<th>Write Action</th>
<th>Read Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Load X operand latch</td>
<td>Read P15 to P8</td>
</tr>
<tr>
<td>n+1</td>
<td>Load Y operand latch and start multiply.</td>
<td>Read P7 to P0</td>
</tr>
</tbody>
</table>

This method of interfacing avoids using IO instructions, and can take advantage of the double byte load and store instructions of the 8080 or 8086. (In certain cases, such as array indexed address calculation, the 8080's double byte addition operation can prove useful.)
Announcing a Computer that thinks in BASIC for only $298.

Ohio Scientific's new Model 500 computer has full mini-computer BASIC in ROM and a minimum of 4K RAM for user programs. BASIC is always there when the computer is turned on.

Make your terminal intelligent!
If your company or school has remote computer terminals, consider the Model 500-1. It fits in the communications line between the terminal and modem or system. When the 500-1 is off, the terminal talks to the modem. When it is on, the terminal talks to the Model 500. Use the Model 500 for short and medium sized programs in BASIC. Use its immediate mode as an ultra-powerful scientific calculator!

Computers for Students
Couple a Model 500 to a low cost purchased or leased terminal such as the popular ASR-33 teletype for student instruction. The Model 500’s BASIC is a perfect instructional language for students. Since the computer's machine code is always accessible, the Model 500 can challenge the advanced student with other optional software such as our Interactive Assembler/Editor.

Personal Computing
The Model 500 has eliminated all of the grief of the first generation personal computers. This computer comes fully assembled, tested, and guaranteed.
Most importantly, it is easy to use since the BASIC is always there.

Flexibility and Expandability
The Model 500 uses Ohio Scientific’s standard 48 line bus so that it is fully compatible with our Model 400 kits and Challenger™ product lines. Any of thirteen accessory boards including RAM, PROM, parallel, serial, A/D, D/A, cassette, and video graphics can be added as well as peripherals including floppy disks, line printers, color graphics, and more. The Model 500 is offered here in three forms to meet every application and pocket book.

Specifications
Model 500 Board $298.00
8” x 10” fully assembled board including 6502 microprocessor running at 1MHz, 512 bytes of PROM, 6192 bytes of ROM containing 8K BASIC and 4096 bytes of RAM for user programs. The board contains an ACIA based serial interface which is jumper selectable for RS-232C or 20ma loop at 110, 300, 1200, 2400, or 4800 baud.
The 8K BASIC features: full floating point math including transcendental functions, N dimension arrays, multiple letter variables, full string functions, logical operators, PEEK, POKE, USR, and lots more!
The Model 500 board requires +5 volts at 2 amps, –9 volts at 500ma, an external reset switch, and an ASCII serial terminal for operation.
Model 500-1 $429.00
Fully enclosed 500 board with power supply, reset switch, and two 25 pin EIA standard terminal connectors for loop through operation. The 500-1 is 12” x 15” x 4”.
Model 500-8 $629.00
This unit is a 500 board in an eight slot Challenger™ case allowing seven slots for expansion. The unit has a heavy-duty UL recognized power supply and is 15” x 17” x 10”.

ORDER FORM
Order directly from: Ohio Scientific; Hiram, Ohio 44234
This introductory offer expires August 31, 1977. Please allow up to 60 days for delivery.
PRINT OR TYPE CLEARLY.
Name ____________________________ ____________________________
Shipping Address ____________________________ ____________________________
City ____________________________ State ______ Zip ______
Payment by: BAC (VISA) ______-____ MC ______-____ Money Order ______-____
Credit Card Account # ____________________________ Exp __________
Interbank # (Master Charge) ____________________________
________ Model 500 Boards @ $298.00
________ Model 500-1 Computer @ $429.00
________ Model 500-8 Computer @ $629.00
________ Additional 4K RAM Memories for 500 $6 @ $14.00
________ Sales Tax (Applies to Ohio Residents only)
TOTAL CHARGED OR ENCLOSED ____________________________
All orders shipped insured UPS unless otherwise requested.

Circle 12 on inquiry card.

Ohio Scientific
Hiram, Ohio 44234
The S-100 bus, so called, is the name used by "everybody else" who manufactures a MITS Altair compatible peripheral, processor or board. MITS Inc defined the bus by designing the Altair 8800, and during 1976 various manufacturers who make Altair compatible peripherals and processors with Altair-like backplanes began calling it the "S-100" bus in advertisements. So for detailed information, purchase the documentation of an Altair 8800, or one of the other central processors which copied this arrangement. Soon or later, we expect to print an article detailing this information.

MULTIPLICATION

... In the April 1977 BYTE you published some information on the 808. From looking at the instruction set published I don't see a multiplication instruction. Do any microprocessors have a hardware multiply instruction, or is it always in software?

Joseph H Stockton Jr
Rt 6, Caroline Dr
Stockbridge GA 30281

The 808 was the first microprocessor to be both widely marketed and of sufficient capability to make a decent general purpose computer. Since it came out, in the early years of this decade, designs have evolved considerably, but few designs to date (1977, first quarter) include hardware multiply and divide functions. The Digital Equipment Corporation's LSI-11 has a PDP-11 instruction set which does not include multiply or divide operations in its basic set, but which has an optional extended arithmetic option which provides 16 bit fixed point arithmetic operations and 32 bit floating point arithmetic operations. The Texas Instruments TMS-9900 processor has multiply and divide operations for 16 bit integer arithmetic built into its design. But the popularly used 8080, Z-80, 6800 and 6302 processors do not incorporate this feature and the user must do multiplication and division in software.
Condensed Reference Chart

Here is a quick reference chart that describes the opcodes for the Motorola 6800 microprocessor. This chart gives all of the hexadecimal codes, including those that are unimplemented. The codes are given in a four line format as shown in figure 1.

The first line has three values: a, b and NN. NN is the hexadecimal op code; a is the length, one, two or three bytes, of the instruction; b is the length of execution in machine cycles for each instruction.

The third line contains the condition flags that are affected by the execution of the instruction. The notation used for the condition flags is summarized in table 2. If a flag is set to a specific value by the instruction, the flag and value are shown on line 4. If the command is a branch instruction then this line contains the conditions under which the branch will occur in parentheses. The letter “u” represents an operand interpreted as an unsigned 8 bit binary number, “s” represents an operand interpreted as a signed two’s complement number. The “u” and “s” conditions are correct only if the last flag affecting instruction executed before the branch is SBA, CBA, SUB or CMP.

The fourth line contains a summary of the operations of the instruction in the AHPL language. AHPL stands for A Hardware Programming Language and is discussed by Hill and Peterson in Digital Systems: Hardware Organization and Design which is published by John Wiley and Sons Inc. The notation is taken from page 112 of that book. The upward and downward pointing arrows stand for various types of shifts. The upward pointing arrow is a circular shift left and the downward pointing arrow is a circular shift right. An arrow with a circle at its beginning denotes a shift with a zero moving into the vacated end. The arrow with the ditto mark (‘) is my own invention to denote a shift with the vacated end bit maintained. The @ symbol is used to denote the exclusive OR function. The left pointing arrow is the APL standard assignment symbol. If a condition flag is set to a particular value by the instruction, the value is indicated. Any condition flags that do not have a particular notation follow the ordinary rules. Primes on the flag condition, such as in the LSR, ROR and ASR instruction, denote a special rule for setting or clearing that flag.

A nonstandard notation concerns the pushing or pulling of the stack. Pulling from the stack is accompanied by an increase in the value of the stack pointer. The value is increased by 1 in the PULA and PULB instructions; by 2 in the RTS instruction; and by 7 in the RTI instruction. Similarly, pushing into the stack is accompanied by a decrease in the value of the stack pointer by the appropriate amount.

Editor's Note... The Holes in Opcode Space

Looking at this chart of the 6800 (or a similar chart for any processor) one sees numerous “holes” in operation code space. These holes are the yellow boxes. An interesting challenge for the owner of a microprocessor (or any computer) is to try and figure out what happens when the forbidden opcodes are executed. Will these exotica turn out to have any uses? Or will they simply be another set of NOPs? For example, look at the strange grouping of the operation codes 4X, 5X, 6X, 7X where X is any hexadecimal digit. 40, 41, 42, 44 are all negatives, 43, 45, 46, 47 are all complements, and various arithmetic operations occur for other values. But the operation codes 41, 51, 61 and 71 are nominally undefined as are 4B, 5B, 6B, 7B and several others. Could it be that those are operation codes that weren't documented in the literature because the masks of the LSI chip had errors? Could it be that they are supposedly unimplemented but are really just not documented so that a "next generation" processor can be achieved by simply releasing the paperwork and selling "new" chips? It's an area of personal research which could prove quite useful: Whenever you find a new processor, examine it for missing opcodes and see what wasn't documented - then try to figure it out by writing programs which execute "unimplemented" instructions under carefully controlled conditions.

Robert J. Borrmann
Electrical Engineering Dept.
Manhattan College
Riverdale NY 10471

Table 1: Summary of the addressing modes that are used in the chart. These modes are found in blue on the chart.

A = Accumulator A.
B = Accumulator B.
# = Immediate.
d = Direct.
x = Indexed.
e = Extended.
r = Relative.
f = Indirect (no symbol).

Table 2: Summary of the condition flag codes that are used in the chart.

CCR = xXxhizuc.
h = Half carry.
i = Interrupt mask.
n = Negative.
z = Zero.
v = Overflow.
c = Carry.
x = Don't Care, always 1.
for the 6800
A Train Control Display
Using the LSI-11 Microcomputer

In order to demonstrate the application of a microcomputer to real world situations, a model train system was developed which is controlled by a Digital Equipment Corporation LSI-11 microcomputer. Two model trains were used for the demonstration - one adhered to a fixed schedule while the other was directed along a path to make it avoid the first train.

HO model railroad components were used. (HO is currently the most popular size for model trains and accessories among enthusiasts in the United States; the scale is 1/64th of actual size, but this can vary.) The layout includes some 75 feet of track and 18 switches plus detectors to determine train locations and a programmable power supply to control train speed and direction. Control interfaces are used between the computer and the track for power. In addition, switches are used to control train position and both station indicators and billboards check train positions and provide atmosphere. A 60 Hz clock is used for a time base, and peripheral devices are used for data entry, status reporting and transferring commands to the train system.

As in many other such systems, the hardware and software elements of this system can be analyzed separately, but it is important to keep the interrelations between the two in mind. Thus, when a detector informs the computer that a train is in a certain location, the control program must update the

Figure 1: Model railroad track layout used in authors' system. During a typical demonstration, both trains leave their sidings under control of a Digital Equipment Corporation LSI-11 microcomputer and are routed along the tracks according to a control program in the computer. Feedback sensors are used to verify train locations at strategic times.
system's files so that the new location can be related to the location of the other train.

Take the A Train

The operation of this electric train system is straightforward (see figures 1 and 2). Upon startup, the trains are moved forward until they trip the location sensors in their respective sidings. These location sensors are actually reed switches which will close in the presence of a magnetic field. Each train is provided with a magnet for this purpose (the next section describes these switches in greater detail). After both siding sensors have been tripped, the trains (called A and B for convenience) are individually brought clear of the siding area.

The lead train A is switched to a station and train B is diverted to the outer track. The software used to control the trains and keep them separated is then employed throughout the remainder of the LSI-11 demonstration.

Figure 2: Block diagram of the LSI-11 control system. The data bus is connected to two terminals used to input and display information at the operator's control station. It is also connected to a special interface which is based on two hardware latches called parallel line units, or PLUs. These PLUs store command bytes coming from the microcomputer which in turn drive flip flops and relay drivers used to power the track and operate track switches. Commands from the computer are in the form of 16 bit words. Each section of the track has been assigned a specific bit, and only those bits of the word necessary to change the track sections desired will change. Information from reed switches is also latched by the PLUs and sent back to the computer to be used in closing the feedback control loop.
HI! I'M AN LSI-11 MICROCOMPUTER
I'M RUNNING THIS RAILROAD TO DEMONSTRATE MY REAL TIME
MULTITASK FOREGROUND, BACKGROUND CAPABILITY. I MONITOR 27
INPUTS AND CONTROL 32 OUTPUTS AND STILL HAVE MOST OF MY
POWER UNUSED.

<table>
<thead>
<tr>
<th>STATION</th>
<th>STOP CODE</th>
<th>ARRIVE TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOSTON</td>
<td>1</td>
<td>DEPARTED *(1.)</td>
</tr>
<tr>
<td>NEW YORK</td>
<td>2</td>
<td>+32 SECONDS *(1., 2.)</td>
</tr>
<tr>
<td>LOS ANGELES</td>
<td>3</td>
<td>SCHEDULED *(1.)</td>
</tr>
<tr>
<td>SAN FRANCISCO</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: A typeset example of the type of display generated by the authors' terminal. Such pertinent train information as arrival times and estimated delays is available to the operator. "Arrival Time" messages vary between "ARRIVED," "DEPARTED," "+XX SECONDS," "DELAYED," "NEXT STOP," and "SCHEDULED."

Operator instructions such as specifying where the trains are to stop and requests from the computer for instructions are transmitted by an RT02 remote data terminal. The terminal has a 30 character keyboard input and can display as many as 32 alphanumeric characters. These characteristics provide all the capabilities needed for full interaction between the operator and the system. A full video terminal, the VT52, is used to display train information as well as advertisements and time of day. The terminal is serially connected to the central processor and operates at 9,600 bps. Multiple messages can be displayed on the screen simultaneously (see figure 3).

Software for the train demonstration can be called "multitask." Tasks are assigned to be either a foreground or a background priority status. There are 12 possible tasks required to run the train system, eight of which are in background. Foreground jobs take priority over background jobs by means of interrupts. Since most of the jobs are executed on an "as needed" basis, the primary job of the background is to act as a "wheel spinner" which cycles in a wait loop until called upon.

Photo 1: Authors Hart (foreground) and Badger performing a different kind of "engineering" with their LSI-11 controlled model railroad. The system was on display at the IEEE Electro 76 conference in Boston. The display was also seen by many of our readers at the Personal Computing 76 conference in Atlantic City N.J. in August 1976.
Software execution in this system is controlled by a real time clock program module which creates a reference base to coordinate all program operations. This particular clock makes use of a standard 60 Hz power line frequency which is passed through a low pass noise filter and then transformed into a train of square waves at TTL voltage level. These square waves are next fed to the BUS event line of the LSI-11 where they create processor interrupts at each positive transition of the waveform. Every 16.7 ms, then, control is transferred to the clock job, which regulates the “time of day” and “train arrival” displays on the video terminal.

**Keeping Track of Your Trains**

Neither train can be successfully controlled unless its location is known with a great deal of certainty at any given time. This is done by processing information from the various reed switches and extrapolating times of arrival at key points (eg: when train A is scheduled to arrive at a station).

As mentioned earlier, each train has a magnet mounted on it to activate the reed switch sensors (see figure 4). Each sensor is polled periodically to determine if a train is present. From this information it is possible to compare train position against the schedule and the time base to quickly tell if a train is on time or late; this information can be displayed on the video screen. The reed switches are connected to the input side of parallel line unit number one (PLU 1, which serves as a data storage buffer to and from the computer) via a TTL translator which converts the switches’ status to the appropriate TTL voltage levels needed by the PLU. PLU 1’s output is used to energize the billboard signs and sections of track under control of the LSI-11.

**Powering Up**

When two trains are to be controlled simultaneously, as in this case, the simplest technique is to apply power to isolated sections of track at the right times. The computer can differentiate between trains A and B by checking position and schedule and

---

**Figure 4: Reed switch circuitry used to send feedback information to the microcomputer.** When a train passes by, the magnetic field causes the switch to close, pulling the signal line down to ground potential; this sends a signal back to the interface (and finally to the computer) that a train is passing.
then creating the proper track energization sequence to control the trains. There is one problem in using PLU 1's output to energize the tracks: its TTL output power is too low to drive the HO trains being used. This problem can be solved by the use of relays (see figure 5).

Relaying Information

Two types of relays are employed. For those sections of track which are simply to be turned on or off, a single pole relay is sufficient. When power is required, the appropriate output word is transferred to PLU 1, which causes the desired relay to be energized. This in turn applies power to the track. When power is no longer required, PLU 1 is updated with a new word and the relay is de-energized. Some tracks, however, have reversing capabilities which require not only power but also one of two polarities for track voltage. Figure 5 shows how two relays are used to do this: relay 1 chooses the polarity of the track based on a command from PLU 1, while relay 2 applies power to the track, again on a command from PLU 1. Each time a command is given, a single 16 bit word is written into the output data buffer.

Figure 5: Computer controlled track power circuitry. Inverted logic levels coming from PLU 1 drive two types of relays. The first type (represented by relay 1 in this illustration) chooses the polarity of the voltage going to the track, while the second type (relay 2 here) applies power to the track. Only certain strategic areas of the track require polarity switching.

Figure 6: A typical fork (or “turnout”) in the track. In this illustration, the turnout is positioned to divert the train to the left leg. These solenoid controlled mechanisms act as their own sensors. One leg of the switch is either grounded or ungrounded depending on its position. This is used to provide a logical 0 or 1 output to the computer for confirmation of the switch’s position. The solenoid is illustrated in figure 7.
PLU 1. Each section of the track has been assigned a specific bit and only those bits of the word necessary to alter the status of the track section in question will change.

A Successful Turnout

Another aspect of control is the "turnout" switch (see figure 6). Turnout is a railroading term for a fork in the track which divides it into two branches. This term is used to avoid confusion with the electrical switches. The turnout settings will obviously have to change as a function of the two trains' positions and the programmed routes. The current status of a turnout must be determined by the computer before a train can pass through. A derailment will occur, for example, if the train proceeds from either leg of a turnout against the current setting. The turnout mechanisms in this system have been modified to act as their own sensors. One of the two moving legs of the turnout is electrically isolated in one position and grounded in the other. It is connected to a +5 VDC level (TTL) through a pullup resistor so that its output, which goes to parallel line unit number 2 (PLU 2), is either high or low depending on the switch's position.

A software routine controls the turnout mechanisms by indexing the track table in memory to determine which track switches must be thrown to get both trains to their next checkpoints. It then moves the turnouts to the required position, waits until the mechanical motion of the switches is com-

Figure 7: A turnout solenoid. This type of solenoid can be activated in either direction because of its center tapped coil. Application of voltage to the desired side of the coil determines the direction of throw.

Figure 8: Coil activation circuitry. The coils used to energize the track solenoids cannot be energized 100% of the time (a 100% duty cycle) without burning up. This circuit uses flip flops and delay elements to provide a one second pulse to a relay where it is converted to 17 VAC, sufficient to activate the solenoid. This illustration, like the rest in this article, is intended to illustrate a concept rather than to serve as a basis for actual construction.
completed, and checks the new switch position. Each track switch is moved by a solenoid as shown in figure 7. This type of solenoid can be actuated in either direction because of its center tapped coil: application of voltage to the desired side of the coil determines the direction of throw.

Mortal Coils

Activating each solenoid requires a fairly elaborate arrangement, since PLU 2 has only 16 bits in its output word, while 32 control signals are actually required for all left and right solenoid positions. Further, the solenoids cannot merely use a level converter to increase the output control signal of PLU 2: such solenoids cannot be constantly energized at the required level without burning up. Figure 8 shows a circuit which converts the incoming signal to a level of 17 VAC used to energize the solenoid. The initial signal is sent to IC1, a hardware latch with both inverting and noninverting outputs. This latch is used to buffer the incoming data. A 75 ns delay is used between the inverted output of the latch and the clock of flip flop IC2 to allow the data level to settle to its correct value before being clocked in.

A logical 1 signal from PLU 2 will cause IC2 to go high and activate the coil. Another delay circuit from the inverted output of IC2 feeds back to the clear line and turns the relay coil off after one second. All of this causes the turnout switch to move to its right hand position. Shifting the switch to the left involves the same type of procedure. To insure the correct initial conditions, a software routine in the LSI-11 places all turnout switches in the left hand position following startup.

Off the Beaten Track

An emergency (eg: a train derailment) will occasionally occur which requires an immediate halt to all operations. A provision on the RT02 data terminal keyboard permits "panic stops" in such circumstances. To bring a more normal, orderly halt to the system, an appropriate command is entered into the data terminal: this sends train A to station 1 and train B to station 4. When train A reaches its station, power is reversed on the tracks so that it will back into siding 1. Train B is similarly backed into siding 2, which signals the end of a very tractable system demonstration!
Deal Yourself in...

Atlantic City, N.J.
August 27th-28th

What it's all about!
Software Development
Micro Computers
Hardware Development
Disc Memories
Computer Comparisons
Interfacing
Program Implementation
AMSAT
Computerized Music
Video Terminals
Kit Construction
Printers
Computer Games
Digital Tapes

- Seminars and Technical talks by leading electronic equipment manufacturers
- Major Exhibits from all over the country
- Demonstrations in many areas including Home and Personal Computing
- Door Prizes, Free Literature and Free Mementos
- All this plus Sun and Surf - Fun and Excitement - Relaxation and Leisure

Personal Computing

SPECIAL GROUP RATES FOR CLUBS AND ORGANIZATIONS TRAVELING FROM THE WEST COAST AND MID-AMERICA.

EXHIBITION BOOTHs STILL AVAILABLE - CALL (609) 653-1188
The TV Oscilloscope

Photo 1: A "typical" display. This shows the completed display in operation. The large block letters are reverse video spaces. The use of blanking prevents generation of any signals at the top or bottom of the screen and keeps the visible portion of the display within the linear region of the TV set.

Although conceived and designed independently, the display circuitry in my CRT terminal is virtually identical to that of CW Gantt in his article "Build a Television Display" on page 16 of June 1976 BYTE. The major difference is that I did not use the MM5320 sync generator, and my display is therefore not interlaced and its horizontal sweep is not exactly at the usual TV horizontal rate. My display, shown in operation in photo 1, consists of 24 lines of 64 characters, with a frame consisting of 270 raster lines repeated 60 times per second. With this formulation I have a time allotment for 27 character lines with ten raster scan lines per character. The extra three character line positions represent overhead for vertical sync.

In a TV set the interlacing is produced not by the hardware within the set, but by the nature of the incoming sync signal. By using a decode of what would have been character line 26 (raster lines 251 thru 260) as the vertical sync, and a decode of character column 72 as the horizontal sync, the generator produces a horizontal sync frequency of 16,200 Hz. The TV set is happy with these small deviations from its usual sync diet, requiring no readjustment when used for its original purpose.

Another difference between the two approaches is in the size of the display. I wanted no fewer than 64 characters per line in order to be compatible with existing software written for a Teletype terminal. And to be consistent with more characters per line, using more lines per frame preserves the character's aspect ratio: the ratio of its height to width.

With 64 character spaces each seven dots wide, plus sync and retrace time allowances of about 20%, I found my dot rate would be

Or, Some Notes on Building a Display

Kenneth Barbier
POB 1042
Socorro NM 87801

Figure 1: Block diagram of a television display. This version is very similar to the display described by C.W. Gantt in June 1976 BYTE, page 18. The points indicated by circled numbers are used as described in the text to generate the various test patterns shown in photos 2, 3, 4 and 6.
something over 9 MHz. How to push all that through a commercial TV set front end? No way! I never even tried. So I had to invent my own TV set interface which was described in my short article in July 1976 BYTE, page 38.

A simplified block diagram of the timing portion of my terminal is shown in figure 1. Since it varies only slightly from that of Gantt's, I have not included a complete detailed circuit diagram or explanation. At the block diagram level it can be seen that an oscillator drives a dot counter and decoder to produce a scan of the five dots making up each line of a character, followed by two dots worth of spacing. The seventh dot is used to increment the character counter, which counts the 64 characters to be displayed in each line. The end of the line is a count corresponding to what would have been character 72; this is used to increment the scan counter. The 65th thru 72nd character positions are timing overhead allowing time for horizontal sync.

The scan counter counts ten scan lines: one blank line above the characters, the seven raster lines comprising the characters, a blank line below the characters, and a line on which an underline or cursor position marker can appear. At the end of the tenth scan line, the character line counter advances. There are 24 displayable character lines numbered 0 thru 23. Line number 24 is for spacing at the bottom of the frame, line number 27 is reserved for the vertical sync and retrace time, and line number 26 is a blank line at the top of the screen. (Lines are numbered 0 to 26 consistent with the 27 binary states of the line counter.)

The outputs of the character counter (0 thru 64) and the line counter (0 thru 23) are used to address the display's programmable memory of 2048 bytes on two boards of 1024 bytes (1 K bytes) each. Only 1536 bytes of this memory are used in the display; the remainder is available to the processor for normal programming purposes. The processor can both write into the programmable memory area and read data from this memory. However, these operations should take place during vertical retrace time to avoid visible glitches. The processor needs to access the display memory in order to move the cursor around without disturbing the stored characters, and in order to implement scrolling when writing more than 24 lines. Being able to read and write the display memory makes it possible for these functions to be implemented entirely in software.

Enabling the processor as well as the display to read the memory also makes the hardware more complex than Gantt's. His approach was to implement a display only, mine was to build a smart video terminal. Both are built of the same foundation blocks: a sync generator, a memory address counter, and a character generator.

Whatever approach you might take, once you have built a composite sync generator and interfaced it to a TV set you have a powerful tool for developing the rest of the system, the main theme of this article. By breaking the video line shown on Gantt's

![Figure 2: (a) A modification of the Gantt television display circuit, of page 18, June 1976 BYTE, to use this test instrument technique. The line from IC7 pin 12 to the video level potentiometer is broken so that input to the video mixer, and thus to the TV, is taken from some arbitrary point under test rather than the usual source. (b) The author's video output circuitry, with a similar clip lead jumper normally in place for display purposes, but free to be moved around for testing purposes.](image-url)

About the Author

Ken Barbier is an electronics technician employed by National Radio Astronomy Observatory, Socorro NM. In support of the construction of the largest radio telescope in the world, he has been designing and building microcomputer controlled automatic test equipment for the project's Digital Control System. This gives him the opportunity to divide his time between hardware design and programming. His other hobbies include building a microprocessor controlled bridge game, and flying, preferably without an engine.
schematic (June 1976 BYTE, page 18) between the output of IC7 pin 12 and the level control and bringing it off the printed circuit board through the edge connector, we can insert an alligator clip lead in series as seen in figure 2a. The equivalent connection modifying my own interface, published in July 1976 BYTE, page 38, is shown in figure 2b. Now by using the clip lead as a test probe, any TTL level signal within the system can be displayed on the TV set to facilitate system development and debugging.

This TV oscilloscope will display signals as intensity modulation within the display field area, instead of as the vertical deflection on a real oscilloscope. Since the hardware we are building operates in synchronism with the field produced by the composite sync, and since our signals are all digital with a TTL logical 0 showing as white and logical 1 as black, this is not a real disadvantage.

For example, probing around in my system I can display the output of the next to least significant bit of the address to the character storage memory as seen in photo 2, test point 1 on figure 1. This output should change state for every other character position in the line. Counting the number of bars in this picture, I saw that not all of the 64 characters were going to be visible, since there are less than 32 bars showing in this display. This told me that some adjustments had to be made in the relationship between the character counter, the horizontal sync, and the time allowed for retrace. Also, as seen at the top of the raster in photo 2, the horizontal sync is imperfect following the vertical retrace. This came about in my system since the widths of the sync and equalization pulses were not made the same as TV standards. Since none of the 24 lines to be displayed are affected, I concluded that there was no need to cure this “problem.”

Photo 3 is another of the address lines to the memory, in this case the bit which will change state once every other character line (test point 2 in figure 1). The line counts for several positions are shown on this picture.

A single dot time (one of seven per character) as displayed on the TV oscilloscope is shown in photo 4 from test point 3, figure 1. Examining this photo we might expect some smearing of the characters to occur when the character generator is connected, since the “dots” are wider than they are high. This is a result of the TV set trying to display a 9 MHz dot rate.

Up to this point we have been able to use the TV oscilloscope to test the portions of
ORDER FORM

[Write in each box the quantity of each part required]

<table>
<thead>
<tr>
<th>KIT ASSEMBLED</th>
<th>KIT ASSEMBLED</th>
<th>KIT ASSEMBLED</th>
</tr>
</thead>
<tbody>
<tr>
<td>4K ROM</td>
<td>8KSC-Z</td>
<td>68KSC</td>
</tr>
<tr>
<td>WWC</td>
<td>88 EXT</td>
<td>68EXT-S</td>
</tr>
<tr>
<td>BB WC</td>
<td>68EXT-L</td>
<td>BB WC</td>
</tr>
<tr>
<td>ASSEMBLY AND OPERATING MANUAL $4.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NAME __________________________

ADDRESS ______________________________________

PLEASE PRINT OR TYPE

CITY ___________________ STATE ______ ZIP __

SEND CHECK • MONEY ORDER • COD’S ACCEPTED • CREDIT CARDS

SEALS ELECTRONICS, INC.
P.O. BOX 11651
KNOXVILLE, TN 37919

MOST ORDERS SHIPPED WITHIN 10 WORKING DAYS

4KROM [READ ONLY MEMORY]
- Will accept up to 16 ea. 1702-A or 5203 EProm Providing up to 4096 words of non-volatile memory for Boot Loads to Complete Programs.
- Programming Available at Factory for $3.00 per EProm when accompanied by binary formatted tapes.
- Each 1702-A has its own Vcc clocked for Low Power Consumption.
- Will work with the weakest power supply based 5-100 bus systems.
- Switched Selected Address in 4K Blocks.
- Switch selected wait states so that even the slowest 1702-A can work in your system. 0-3 wait states.
- Solder Masked on both sides of PC Board.
- Component Screened on Component Side of PC Board.

Kit Price: . . $119.00
Assembled Price: . . . $179.00

68KSC [STATIC MEMORY CARD FOR SWTPC 6800]
- 8192 Words of Static Memory
- Access Time: 500 nsec. (250 nsec on request)
- Memory Chip 91102 APC
- Battery Standby
- Address Selected 8 Ea. SPST Dip Switch
- Low Power
- All Lines Buffered.
- All IC’s with sockets
- Solder Masked on both sides of PC Board.

Kit Price: . . $269.00
Assembled Price: . . . $359.00

8KSC-Z [STATIC MEMORY CARD]
- ALTAIR/IMSAI and S-100 bus compatible.
- Access Time: 250 nsec max.
- Zilvo Speed Compatible up to 4 mhz.
- Memory Chip: 2102L and 2102AL-Z
- Battery Standby: 1.5 to 4 volts
- Wait States: None
- Current Reg.: Less than 200 ma per 1K
- All Address, Control, and Data out lines fully buffered.
- All IC’s supplied with IC Sockets
- Solder Masked on Both Front and Back of P.C. Board.

Kit Price: . . $295.00
Assembled Price: . . . $395.00

OTHER SEALS ELECTRONIC PARTS AVAILABLE

<table>
<thead>
<tr>
<th>Item No.</th>
<th>Description</th>
<th>Kit Price</th>
<th>Assembled Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>BKSC</td>
<td>8K Static Memory Card 500 nsec</td>
<td>$269.00</td>
<td>$369.00</td>
</tr>
<tr>
<td>WWC</td>
<td>Wire Wrap Card</td>
<td>$37.50</td>
<td>$47.50</td>
</tr>
<tr>
<td>68 EXT</td>
<td>BB Extender Card</td>
<td>$29.00</td>
<td>$38.00</td>
</tr>
<tr>
<td>68 Ext-S</td>
<td>Extender Card (Small)</td>
<td>$19.00</td>
<td>$25.00</td>
</tr>
<tr>
<td>68 Ext-L</td>
<td>Extender Card (Large)</td>
<td>$29.00</td>
<td>$39.00</td>
</tr>
<tr>
<td>68 WWC</td>
<td>Wire Wrap Card (MACR-5200 Computers)</td>
<td>$35.00</td>
<td>$45.00</td>
</tr>
<tr>
<td>BB UC</td>
<td>Battery Back-Up Card</td>
<td>$55.00</td>
<td>$68.00</td>
</tr>
</tbody>
</table>

ORDER DELIVERY: FROM STOCK TO 10 DAYS
Photo 6: A map of the preferred states of one memory chip. By tapping into just one of the memory data lines with the display probe, it is possible to generate a map of the data. This data is identical to that of photo 5, but we are now looking directly at one bit, instead of generating characters from 8 bit byte groupings of several chips. This same technique of looking at one bit can be done with normal programmed displays, too, rather than the turn on pattern used here.

The system that scans the dots making up the characters and the scanning of the memory addresses. When all is well to this point, we could connect the programmable memory address lines to the data inputs of the character generator and display one of each character, as Gantt did on page 17 of June 1976 BYTE.

The next step is the connection of the display memory to the address lines, with the display memory data going into the character generator. Turning on the system with the probe on the character generator output, we find that the memory contents will initially be an arbitrary pattern, producing a screen full of garbage as in photo 5. Note how the horizontal “pulling” at the top of the screen distorts the characters in the top line. As this is actually line 26, it is not used in the display, and will disappear after the “blank” signal is connected as in figure 2b.

The arbitrary pattern is not totally so, since a virtually identical pattern will appear with each turn on. This is due to built-in asymmetries in the memory chips, with each bit having a preferred state to come up in when power is initially applied. We can connect our clip lead to one bit of the memory data as in photo 6 and see that each memory chip in the 2 K by 8 bit array has a unique pattern visible in these preferred states.

In photo 6, notice the obvious division in the pattern about two thirds of the way down the display. This corresponds to the division between the lower 1 K (lines 0 thru 15) and the upper 1 K (lines 16 thru 23) of memory, which use different chips.

After we connect the processor to the input side of the memory, we can use this same TV oscilloscope connection to examine the setting and resetting of each bit at each address under program control. Setting the memory to all zeros, each bit we turn on will appear as one black block in the display. The TV oscilloscope can thus be used to insure that the processor and display address...
correspond correctly, and that data is written correctly. (Of course, we could just as well have put some message on the display, too.)

Photo 7 shows the hardware in my system. The clip lead can be seen connected to the display memory line out of the timing board which was used to produce the display in photo 3. Since all of the dot signals, scan line signals, memory addresses, and the memory read and write data buses all appear on the backplane wiring of the system, virtually everything in the system is accessible to the TV oscilloscope test instrument.

For even more versatility in the TV oscilloscope, it might be useful to add an AND gate and another clip lead, so that the exact timing relationships between two signals could be determined. Alternately, one of the signals could be added into the video at a different level to produce a gray image so that the two signals could be differentiated. If for looking at programmed behavior, the same technique can still be used if the vertical or horizontal sync signal is used as a timing cue. Much more could obviously be written on this subject as readers adopt Ken’s idea to personal use... CH/

---

**BYTE's Bugs**

**A Structure Glitch**

---

Jack Emmerichs called upon his receipt of April 1977 BYTE and reported a couple of errors in the execution of figure 3 on page 66. (As Murphy’s law would have it, figure 3 was moved to the first part of the article at the last moment at Jack’s request, and he did not get to see author proofs for that one figure.)

The major glitch is corrected by noting the above segment of the diagram in its proper form. Two minor errors are an extraneous "level 3" at the lower left of the diagram, and an extraneous "Compare Strings" box in the "General Service Routines" shown at the lower right of the diagram.
consists of serially loading the 144 bits into the major loop, then transferring all 144 bits into the minor loops; reading is done in reverse order by copying the currently positioned bits of all 144 minor loops into the major loop then serially reading the major loop. The result is a nonvolatile file storage device which has 641 blocks of 18 bytes per block, a total of 11,538 bytes in one 14 pin dual in line package measuring 1.0 by 1.1 by 0.4 inches (2.5 by 2.8 by 1.0 cm). Using a volume parameter for memory density, this elephant of a memory stores 200,000 bits per cubic inch (13,000 bits per cubic centimeter) in a nonvolatile magnetic medium.

With this short introduction to the existence of a new low cost, high density memory system, what are the ramifications of such a part for the personal computing user? I’ll confine my comments to two major classes of application for the memory, applications which take advantage of its nonvolatility, low cost of,22 cents per bit in sample quantities, and the large size of each part’s memory. The first class of applications for the bubble memory is as nonvolatile file storage in minimum size personal computing systems analogous to pocket calculators; the second class of application is as on line file storage for larger general purpose desk top or console personal computers. Superficially, the difference between the two classes of application is in the number of these bubble memory parts used with the product.

What Can You Do with Just One?

Consider first the pocket programmable calculator. At the April 6, 1977 meeting of the New England Computer Society, Bev Pettit of the Hewlett-Packard calculator sales office in Lexington MA gave an excellent presentation of the functional capabilities of the new HP-67 and HP-97 calculators, two products which represent the ultimate in pocket calculator performance to date. These machines can be used to perform fairly sophisticated problems, with overlay structures for data and program material swapped using the magnetic card medium incorporated in the design. But, the machines are limited by the fact that there is a finite set of data registers (26, if I recall correctly) and 224 programming steps. What would be the impact of adding a bulk program storage facility implemented with the TI bubble memories or equivalent? Viewed only as a calculator, with 8 bit command codes for each function, just one such bub-
A more expensive console or tabletop personal computing system, the same characteristics of nonvolatile electronically controlled bulk storage which expand the calculator concept can be applied to the general purpose computer system concept. Basically, the bubble memory makes possible a very large totally electronic line buffer storage memory, at a price compatible with personal computing economics. Since the memory is

This dynamite new microcomputer system in a kit moves data like nothing else on the market. Run it alone or plug it into an $100 bus Alter/MSAI. For solo performance, all you need is an unregulated power supply and an I/O device. Plugged in, QUAY 80AI is a CPU, ROM, SIO, and RAM board—run any $100 compatible device. BUT MORE THAN THAT, QUAY 80AI's Z-80 CPU operates challenging new areas of personal computing.

Features

- 5100 bus compatible. Plugs in one slot of your Alter or MSAI.
- Z-80 w/2.5 MHz clock.
- 1 K static RAM.
- 512 byte (ROM) monitor. Comes with running, printing, printer, dump, and load memory, set breakpoints, jump to user program. Handles serial I/O or keyboard input, including setting baud rate.
- 4 UVEPROM (2708) sockets.
- Serial I/O, RS-232 and 20 ma interface.
- Parallel keyboard interface. Accepts standard ASCII keyboard.
- UVEPROM programmer. Program 2708 type UVEPROM.
- 2 phase clock and sync. Run 5100 compatible peripherals.
- 158 instructions. All 76 3000 instructions plus 80 new powerful instructions.
- On board voltage regulators.

QUAY 80AI in a kit is $450; factory assembled, $600. Send for complete details. Or for fast action call 201-681-8700.

Articles Policy

BYTE is continually seeking quality manuscripts written by individuals who are applying personal systems, or who have knowledge which will prove useful to our readers. Manuscripts should have double spaced type-written tests with wide margins. Numbering sequences should be maintained separately for figures, tables, photos and listings. Figures and tables should be provided on separate sheets of paper. Photos of technical subjects should be taken with uniform lighting, sharp focus and should be supplied in the form of clear, glossy black and white or color prints (if you do not have access to quality photography, items to be photographed can be shipped to us in many cases). Computer listings should be supplied using the darkest ribbons possible on new (not recycled) blank white computer forms or bond paper. Where possible, we would like authors to supply a short statement about their background and experience.

Articles which are accepted are typically acknowledged with a binder check 4 to 8 weeks after receipt. Honorariums for articles are based upon the technical quality and suitability for BYTE's readership and are $50 to $50 per typed page. We recommend that authors record their name and address information redundantly on materials submitted, and that a return envelope with postage be supplied in the event the article is not accepted.

Market and Sales Department accepted. COD with 1/3 deposit. N.J. residents add 3% sales tax. Price does not include shipping and handling.

Dealer inquiries invited.

P.O. Box 386, Freehold, N.J. 07728
Phone: 201-681-8700
Circle 17 on inquiry card.
Given a choice for your home computer investment, you'd want totally electronic there is no problem with mechanical design considerations, an advantage which is offset by the constraint of not having removable media as is the case for a floppy disk. As the price of the parts goes down with eventual mass production, the cost per installed bit has the potential of dropping well below that of the removable media floppy disk storage devices.

The implications of this bubble memory for the "appliance" computer are most interesting. Let's consider what the new "minimum" personal computing system which incorporates the bubble memory medium might look like. First, the system includes a typical microprocessor design with eight of the contemporary 16 K dynamic memory parts for main memory, with perhaps 8 K to 16 K of systems software in read only memory. Peripherals built-in include video display (interface or built-in monitor depending on the manufacturer), and alphanumeric keyboard, with a high speed audio cassette interface for long term program storage. For filing of current programs and data this minimum system incorporates eight of the TBM0103 bubble memory units, giving a capacity of 92,304 bytes of data or program file capacity. With an appropriate operating system and high level language built into the read only memories, the symbolically named files in this bubble memory region can be rolled in or out of the 16 K volatile programmable memory region as desired. Using the audio tape interface at the end of a programming session, the file memory could be checkpointed automatically onto tape while the user attends to some other noncomputer related activity like eating, sleeping, etc. (Assuming an audio tape interface at 100 bytes per second, it would take 923 seconds or about 15 minutes to dump the state of the file memory, plus about three minutes to dump the programmable volatile memory of 16 K bytes.)

What would such a conception cost? Perhaps $2000 to $3000, assuming production quantities of the bubble memory cost significantly less than $200, that the support circuitry is not excessive, and that the rest of the system uses existing technology. This would be the minimum configuration of such an "appliance computer circa 1977-1978." The logical next improvement in performance would be to incorporate one Shugart style minifloppy disk file, at perhaps a $500 to $800 increment in price. The bubble film memory is functionally

You'd want 8 or 16-bit digital computers with fully expandable memories and a variety of I/O interfaces. Such versatility lets you optimize the system for your hobby, education or small business application.
You'd want an 8-bit computer featuring an intelligent front panel with octal keyboard entry and display for fast readout, a resident monitor with built-in bootstrap for one-button program entry or storage. Or a powerful 16-bit computer with resident monitor.

TI happens to be the first to commercially market the bubble memories. (AT&T has already been using them for several months in production phone equipment, but these parts are not available to general commercial users.) If we use the example of the Shugart minifloppy disk drives as a prototype for the delay from announcement to proven end user product for a system of similar complexity, then we can expect to see end user deliveries of mass storage subsystems for present day personal computer systems sometime late this coming fall. (This editorial was written April 8, 1977.) As for complete computer systems which make the bubble memory an integral part, I would tend to expect the first such systems in the first half of 1978 or sooner depending upon the cleverness and resources of the companies which ultimately produce such products. In the meantime, perhaps some technologically enterprising reader will purchase the early sample versions of the bubble memory chips along with documentation, and design a homebrew computer interface which we can publish for the benefit of all BYTE readers. The bubble memory, a long awaited promise, has finally become a technological and commercial reality available for use in personal computing products.

very similar to a floppy disk. The bubble memory's data transfer rate, at 50 kbps, is comparable to the transfer rates of the small floppies, and its access time to blocks of data is approximately one to two orders of magnitude faster (12.8 ms versus typically 250 to 500 ms head positioning time on a floppy disk). But the bubble memory chip capacity of 92,000 bits is about the same as the capacity of the typical 5 inch floppy disk. Using one floppy disk drive for its removable media attributes, and the standard permanently "mounted" simulated floppy disk of the bubble memory, this appliance computer concept now has the ability to "instantly" copy disk files, sort data within the constraints of two 92,000 byte file regions, and do many of the "memory intensive" tasks usually associated with large interactive systems.

When Will We Users See Products?

The announcement by Texas Instruments, accompanied by extensive advertisements in the electronics trade press for the bubble memory, a 64-K charge coupled device, volatile memory and large 16 K dynamic programmable random access memories, is aimed primarily at designers of new products.
Continued from page 34

• Type 03: External Symbols

<table>
<thead>
<tr>
<th>Byte number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dollar sign ($) delimiter.</td>
</tr>
<tr>
<td>2, 7</td>
<td>Up to 6 ASCII characters of the external symbol name. The name is left justified, blank filled.</td>
</tr>
<tr>
<td>8, 9</td>
<td>Record type 03.</td>
</tr>
<tr>
<td>10, 13</td>
<td>Last address which uses the external symbol. This is the start of a link list which is described below. The most significant byte is first.</td>
</tr>
<tr>
<td>14, 15</td>
<td>Binary checksum.</td>
</tr>
<tr>
<td>CRLF</td>
<td>Carriage return, line feed.</td>
</tr>
</tbody>
</table>

The Mostek SDB-80 assembler outputs the external symbol name and the last address in the program where the symbol is used. The data records which follow contain a link list pointing to all occurrences of that symbol. This is illustrated in figure 1.

1. The external symbol record shows the label (LAB) and the last location in the program which uses the symbol (212AH).
2. The object code at 212A has a pointer which shows where the previous reference to the external symbol occurred (200FH).
3. This backward reference list continues until a terminator ends the list. This terminator is FFFFH.

This method is easy to generate and decode. It has the advantage of reducing the number of bytes of object code needed to define all external references in a program. Intel defines each external reference explicitly.

• Type 04: Relocating Record

The addresses in the program which must be relocated are explicitly defined in these records. Up to 16 addresses (64 ASCII characters) may be defined in each record.

<table>
<thead>
<tr>
<th>Byte number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dollar sign ($) delimiter.</td>
</tr>
<tr>
<td>2, 3</td>
<td>Number of sets of 2 ASCII characters. Where two sets define an address.</td>
</tr>
</tbody>
</table>
4 - 7 ASCII zeros.
8, 9 04
10 Addresses which must be relo-
cated.
Last 2
bytes Binary checksum.
CRLF Carriage return, line feed.

• Assembler Object Output Sequence

The SDB-80 Assembler will output object records in the following sequence:

1. Internal symbols (type 02) in alphabetical order.
2. External symbols (type 03) in alphabetical order.
3. Data records (type 00), interspersed with relocation records (type 04).
4. End of file record (type 01).

Internal symbol records will be outputted only if internal symbols are defined in the source program. The same is true of external symbol records. Relocation records always will be outputted by the assembler.

It's All Yours This Fall!
Because the world-leader in quality electronic kits has researched the needs of the computer hobbyist and developed some of the most sophisticated and affordable products around.

WATCH FOR THE
HEATHKIT®
COMPUTER PRODUCTS!
They're the ones you've been waiting for!

6. A value/price Breakthrough
You'd want all hardware systems in stock in easily affordable and buildable kit form from the world's largest and most experienced kit manufacturer. Systems designed to give you more and better features for your computer dollar. And all backed by a "We won't let you fail" promise.

GLOSSARY
Object module: The output from an assembler is called the object module. One or more object modules can be placed into memory by a loader. The output from the loader is called a "load module."

Relocatable program: A program which has extra information in the object module is referred to as relocatable. The extra information allows the loader to place the program anywhere in memory. A nonrelocatable program can be loaded only in one place in memory in order to work properly.

Position independent program: A program which can be placed anywhere in memory is called position independent. It does not require relocating information in the object module.

Internal symbol: A symbol which is defined in a program and whose location is made known to all other programs is called an internal symbol. It is also called "public" (Intel), "defined," "global," or "common."

External symbol: A symbol which is used in a program but which is not defined in the program is called an external symbol. Such symbols must be defined as internal symbols in other programs.

Linkable program: A program with extra information in the object module which defines internal and external symbols is called a linkable program. The loader uses this information to connect, or link, external references to internal symbols.
Speech Recognition for a Personal Computer System

The August 1976 issue of BYTE presented to the personal computer user an introduction to half of the problem of man-machine communication by speech: machine generation of speech. Speech synthesis has been investigated since at least the 1800s. High quality speech can be produced by a computer if it decodes utterances that have been efficiently coded and stored in memory. Poorer quality, but sometimes acceptable speech can be synthesized by electronically simulating the human speech producing mechanism.

Speech communication in the other direction, machine recognition of speech, is usually considered to be a harder task. The purpose of this article is to present some of the problems and to introduce some solutions that may be realized by the personal computer enthusiast.

The ideal speech recognizer would be a machine that could accept casually spoken sentences from many speakers with different dialects and transcribe them in real time into written text or translate them into commands for controlling some function. Unfortunately, there are at least five major problem areas in speech recognition that are still incompletely solved.

- **The Continuous Speech Problem.** In ordinary conversational speech our words tend to "run together" acoustically. Say "How are you" and "How (pause) are (pause) you." The continuous version is uttered almost as a single word. What is the best procedure for segmenting such an utterance into words?
- **The Multiple Speaker Problem.** People say the same words differently. Can a machine allow for these differences or must it be "tuned" to each individual user?
- **The Limited Vocabulary Problem.** Error rates, recognition rates and memory requirements increase with vocabulary size. How many words can be reliably handled by a machine?
- **The Vocabulary Entry Problem.** The words in the vocabulary have to be described to the computer in some way. What is the most efficient way?
- **The Noise Problem.** Speech is filtered by some transmission process or corrupted by noise in transmission media such as telephone or radio. What is the best signal processing technique for these conditions?

These are the problems that concern many industrial and academic research laboratories using the most sophisticated techniques and equipment available. However, there are some realistic goals for a speech recognizer in a personal computer system. That is, it should be possible to build a system which can accept isolated words from a limited vocabulary of ten to 20 words spoken by a speaker whom the machine has been trained to understand. These goals are well within the capabilities of most contemporary microprocessors.

Several such systems have already been designed using a modest amount of discrete
logic. (See the first six references in the bibliography at the end of this article.) Each of these systems can be easily duplicated with a small amount of hardware such as a microphone, amplifier, filters, an analog to digital converter and a microcomputer. This paper will show how one of these well known recognizers can be implemented. But first, a few basic facts about human speech production.

The Speech Code

In order to construct a system for decoding speech you should know how the acoustic signal is encoded with the desired message. One way of looking at the process is shown in figure 1. The message from the brain may be thought of as going through a series of transformations which modify it by many factors. The word idea or linguistic information is first translated into speech sound units called phonemes. This “code” is a collection of the descriptors of the sounds necessary to say the words. In English there are approximately 40 basic types of sounds which are used to construct our entire vocabulary. The phonemic code is translated by the central nervous system into neuromuscular signals for controlling the physical speech making apparatus or articulators. The articulators “modulate” the sound generated by the vocal cords and produce the radiated acoustic signal.

This process has sources of error. The phonemic codes for various dialects are different from one another. A speaker may say the same word many different ways depending on context or physical and emotional state. Differences in the sizes of people’s articulators account for variations between speakers. Day to day changes in the articulators (due to a head cold, for example) can cause variations in the way a given person sounds. Finally, the acoustic signal itself may be corrupted by noise or filtering.

The message coding process produces a signal which contains much information that is not significant to the intended message and which makes any direct comparison of speech waveforms difficult if not impossible. The extent of the coding inefficiency can be seen by a simple example. The following sentence:

“The objective of this section is to develop these techniques.”

contains 60 characters and can be spoken in about three seconds. Using a 5 bit code for alphabetic characters, real time transmission at speech rates would require only 100 bps. If the utterance were transmitted with a string of 42 phonemes coded with six bits each, then 84 bps would be necessary. However, if the speech signal itself were band-limited at 3 kHz, sampled at 6 kHz, and quantized to 256 levels or eight bits, real time transmissions would require a rate of 48,000 bps. If a computer could analyze this digitized speech in real time it would have to process one 8 bit speech sample every 160 μs. A microcomputer with a cycle time of 500 ns would have only 320 cycles between speech samples. That does not allow time for very many instructions to analyze and match the data with reference waveforms. The direct storage of the waveform would require 6,000 bytes of memory for every second of speech. It should be clear that it is desirable to have a method of rapidly extracting the useful information, thereby reducing the amount of data storage and computation time. A look at the speech production mechanism provides some clues for data analysis and reduction.

Speech Parameters

Speech sounds originate from two sources. So-called “voiced” sounds are made by the vocal cords and make up vowel-like sounds. The vocal cords produce periodic bursts of air through the vocal tract at a repetition rate of about 125 to 200 Hz.

“Unvoiced” sounds or “fricatives” are
Voiced
AE father
AI bite
AW bought
AY bay
EE bet
ER bird
IX fit
OU boast
UX book
UY boot
UX boot
WX win
YX yes
RX rip
SX lit
MX man
NX not
NG ring

Fricatives
FX fan
TH bath
SX sip
SH ship
CH church
HX hand

Combinations of Voiced and Unvoiced Sounds
VX van
DH than
ZX zip

Stops
JH jump
PX pan
TX tan
KX can

Voiced Stops
BX ban
DX dan
GX gab

Table 1: Speech phonemes. In English there are approximately 40 basic types of sounds which are used to construct the entire vocabulary. These sounds are called "phonemes." The phonemes are not pronounced literally, but rather serve as symbols for their respective sounds. The phonemic code is translated by the central nervous system into neuromuscular signals for controlling the physical speech making apparatus or "articulators." "Voiced" phonemes are the vowel-like sounds which are produced in the vocal cords as periodic bursts of air from 125 to 200 times a second. "Unvoiced" phonemes or "fricatives" are noise-like sounds made by forcing air through constrictions in the mouth. Some phonemes are combinations of voiced and unvoiced sounds. "Stops" result from momentary blockage of air flow (and sound).

noise-like and are made by forcing air through constrictions in the mouth. If the air flow is completely blocked for a moment, the sound is called a "stop." At times we make both voiced and unvoiced sounds. Table 1 classifies various phonemes into these source types. Phonemes have special symbols that are often used, but this table uses a machine readable format.

The vocal tract is like a pipe from the vocal cords to the lips which has several natural resonant frequencies called "formants." This pipe acts like a filter that most readily passes sounds whose frequencies are the same as the formants. The three most prominent formants are centered around 500, 1500 and 2500 Hz but they vary with changes in position of the tongue, lips and mouth. The voiced and unvoiced sounds can be thought of as "carrier" signals that are "modulated" by the resonances in the vocal tract.

It is well known that there is a correspondence between phonemes and the formant frequencies. In fact, it is only necessary to know the first two formant frequencies in order to identify many vowel sounds. Figure 2 shows a map of several vowel sounds in a plane formed by the first two formants. For example, if you knew that for some interval of a speech utterance the first formant was at 800 Hz and the second at 2500 Hz, then you might guess that the speech sound was the phoneme "EH."

A nice property of formant data is that it changes rather slowly. It is necessary to sample the formant frequencies only every 10 to 20 ms in order to keep up with changing phonemes. This means that the motion of two formants could be stored in only 100 bytes per second of speech as opposed to 6000 bytes per second for the waveform itself.

Naturally, we would like a technique for easily measuring the first two formant frequencies by looking at the waveform data. The most accurate methods require either a lot of high speed computation or special purpose hardware. Fortunately there is a very simple way to get an estimate of the formants which is most suitable for microcomputer processing.

Zero Crossing Analysis

Since the late 1940s it has been known that the amplitude of the speech waveform carries little information. Most of the useful information can be obtained from just the knowledge of the number of times the signal waveform crosses the zero axis. This is called zero crossing analysis. [This method would lend itself perfectly to Walsh transform analysis, . . . CM] If you count the number of times the signal changes sign in a fixed interval, say every 10 to 20 ms, you can get numbers which closely follow the changes in the first formant frequency. If you first emphasize the higher frequencies by high pass filtering, then the zero crossing rate tracks the second formant frequency.

Figure 2: Vowel phonemes on the formant plane. "Formants" are the natural resonant frequencies of the vocal tract. These resonances are similar to the resonant vibrations of organ pipes or the characteristic sound of an empty bottle when air is blown across it. The three most prominent formants are centered around 500, 1500 and 2500 Hz, but they vary with changes in position of the tongue, lips and mouth.
Although the zero crossing rates do not give accurate estimates of formant frequencies, they do contain enough information to partially separate the vowels for recognition purposes. Figure 3 shows a map of vowels on a high pass versus low pass zero crossing rate plane.

There are many advantages in using zero crossing analysis techniques in digital speech recognition. One of them is that the necessary hardware is simple. Since only the sign of the waveform is important, only one bit per sample is needed, which eliminates a requirement for an analog to digital converter in the system. The analysis can easily be done in real time since no multiplications are involved – only counting. The zero crossing parameters can be measured with simple hardware or software as the signal comes in, which eliminates the necessity of storing the waveform. Because the amplitude information is not used, the analysis is independent of voice volume and inflection. This makes zero crossing measurements somewhat less dependent upon characteristics of individual speakers.

**Pattern Matching**

After the acquisition of the zero crossing parameters, the identification of the word can be done using one of two simple processes: in a one step approach, the parameter data is directly compared with reference patterns (sometimes called templates) for each word in the vocabulary. The closest match is chosen as the most probable utterance. In a two step method, the parameter data is first converted into a string of phonemes. Then this string is compared with a phonetic representation of the words in the vocabulary.

Both methods have problems. The one step comparison suffers if the word is spoken at a different rate from the words in the sample set unless some technique is used to find the best time alignment. The time

---

**Figure 3:** Vowel formants on the zero crossing plane. For comparison purposes the formants in figure 1 are plotted here on the zero crossing plane. The term “zero crossing rate” refers to the number of times that the voice signal changes its polarity per unit time (see figure 4a).

**Figure 4a:** A plot of the word “SIX” showing the zero crossing rate as a function of time. The word is shown split into its four phonemes. In this case, the signal is first processed through two bandpass filters. The number of zero crossings is then determined by the computer. Results for both filters are shown. As the graph indicates, there is a very short period of silence during the pronunciation of the KX phoneme which can be verified by slowly pronouncing the word “six.”

**Figure 4b:** A diagrammatic representation of the word “SIX” on the zero crossing plane using a form of state diagram. In this graph the high and low pass functions from figure 4a are plotted against one another in stylized fashion to better show how the zero crossing rates change as the word is pronounced. (Note that the word begins and ends with the same phoneme, SX.) The signal is split into two bandpass regions because these particular regions reveal key attributes of English words which can act as types of “fingerprints” of the words for later identification.
Figure 5: Speech filtering circuitry. A voice signal is picked up by the microphone and preamplified. It is then sent through a low pass filter made up of C1 and R1 whose cutoff frequency is 3 kHz. The signal is next passed through a 1 kHz high pass filter and a 1 kHz low pass filter as shown. The two signals then go to the microprocessor interface for further processing. IC1, which forms the basis for this circuit, is a quad LM3900 op amp. The diode-like symbols shown on the four amplifiers indicate the use of "current mirrors" for the noninverting inputs. For a detailed discussion see National Semiconductor's book, Linear Integrated Circuits, page 2-250 and following.

Figure 6: In order to implement the speech recognition algorithm, 60 samples of speech over roughly 1 second's duration are sequentially processed through the circuit illustrated in figure 5. For each of the 60 intervals the number of high and low pass zero crossings per second is counted and stored in a software array corresponding to the regions in this figure. (This figure is not to scale.) After this, a straightforward statistical correlation is performed on the array to compare it with a series of word models in the memory and to arrive at the most likely match.

dependence can be eliminated by considering the pattern of the data plotted on a zero crossing plane. Figure 4 shows how the word "six" might be plotted. The one step method has the double disadvantage that a lot of memory is required for template storage and that the matching may be more time consuming. Two step systems use less memory in the template storage but depend on the performance of an imperfect phoneme identifier. The phoneme matcher must be able to tolerate errors such as missing, mislabeled or extraneous phonemes in the hypothesized string. For example, the output from a phoneme identifier given in the utterance "six" (or "SX IX KS SX") might be "SX SX EE IX IX EH XX SX SX."

A Practical Speech Recognizer

You have probably realized by now that there are many possible approaches to the analysis and recognition of speech. With this in mind, we can now consider in detail an isolated word recognizer that can be implemented on most microcomputer systems. This algorithm is based on one of the earliest successful word recognition ex-
The book you’ve waited for is here!

UNDERSTANDING MICROCOMPUTERS
AND SMALL COMPUTER SYSTEMS

What’s an accumulator? How do you use relative addressing? Indexed addressing? Boolean logic operations? Does a program counter count programs? What's an I/O port? What are EPROMS? What is paging? A stack? The difference between a dot and vector CRT display? How do you copy programs using an ordinary tape recorder? What can a microcomputer really do? What kind of personal computer system should I buy?

These and hundreds of other questions about microcomputers are answered in this profusely illustrated, easy-reading “must” book. Here are the fundamental concepts behind the operation of virtually all microcomputers... written in simple English so that anyone can gain the extra knowledge that will let him or her read and understand computer magazines and manufacturers’ literature and feel “at home” around computers. Here are just some of the items discussed...

- How a CPU is organized and how it follows sequences of directives to solve problems.
- Illustrates basic instructions from almost every class of microprocessor.
- Discusses common memory addressing modes — illustrates typical uses.
- What you must know to tell a computer what to do when using machine language programming methods.
- Explains use of flow charts; program worksheets; hand assembly of source codes into object codes; memory maps; purpose of an Editor, Assembler, Monitor.
- Benefits of using a high level language — introduction to BASIC.
- How a computer communicates. Types of commonly used input/output devices and their operational concepts.
- Practical aspects of selecting a small computer system.
- Plus, hundreds of other practical facts and information.

If you're at all curious about small computers, you must own this 300 page no-nonsense text that reads like a simple book, and includes an easy-to-use glossary of key microcomputer oriented words. UNDERSTANDING MICROCOMPUTERS. The name says it all. And, it all costs only $9.95. Order your copy today!

SCELBI COMPUTER
CONSULTING INC.
PO Box 133 — P O STN
Milford, CT 06460

Price shown for North American customers. Master Charge, Postal and Bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing, specifications, availability subject to change without notice. SCELBI Books are available in many fine Computer Stores.
FOR J = 1 TO 60
F1(J) = 0
F2(J) = 0
FOR I = 1 TO 100
[Delay for 160 μs loop time]
: ;INPUT X
X1 = XOR (X, XO)
IF X1 > 1 THEN F2(J) = F2(J) + 1
IF X1 = 1 OR X1 = 0 THEN F1(J) = F1(J) + 1
XO = X
NEXT I
NEXT J

Listing 1: The input algorithm. Arrays F1 and F2 store the number of zero crossing counts in each 16 ms interval for the low pass and high pass channels respectively. This listing, like the following listings, is written in BASIC for illustrative purposes, but it must be translated into machine language to achieve real time performance.

FOR I = 1 TO 30
P(I) = 0
NEXT I
FOR J = 1 TO 60
FOR I = 1 TO 5
IF F1(I) > F1(J) THEN 100
NEXT I
1: I = 6
100: FOR I = 1 TO 4
IF F1(I) > F1(J) THEN 200
NEXT I
200: I = (I-1)*6+1
R(I) = P(I) + 1
NEXT J

Listing 2: The classification and accumulation routine. This routine takes the information from arrays F1 and F2 and stores it in array P, which represents the zero crossing plane.

FOR J = 1 TO L ; Correlate
R(J) = 0
FOR I = 1 TO 30
R(J) = R(J) + P(I)*T(I, J)
NEXT I
NEXT J
J1 = 1 ; Pick best
R1 = R(J1)
FOR J = 2 TO L
IF R(J) > R1 THEN R1 = R(J) ; J1 = J
NEXT J
PRINT J1

Listing 3: The correlation routine. This routine compares the completed array from listing 2 with various word models in memory and uses standard statistical methods to find a best match. For a ten word vocabulary the entire speech recognition algorithm can be run in less than 69 ms.

INPUT "TEMPLATE NUMBER:" , L
FOR I = 1 TO 30
P(I) = 0
NEXT I
FOR K = 1 TO 8
INPUT "READY?": Q
[Call routine for sampling speech and computing P]
FOR I = 1 TO 30
P(I) = P(I) + P(I)
NEXT I
NEXT K
FOR I = 1 TO 30
T(I, L) = P(I)/8
NEXT I

Listing 4: The template generation routine. A "template" is a word model in the form of an array. This routine, which can be run off line (in BASIC if desired, since there are no time restrictions), is used to generate all of the word models in the vocabulary desired. Eight samples of a given word are uttered by the operator and averaged together to create each word template.

The next step is to divide the low pass versus high pass (F1, F2) plane into 30 regions as shown in figure 6. Then for each of the 60 intervals, the zero crossing counts in F1 and F2 are located on this plane and associated with one of the regions. A 30 element array P keeps up with the number of points in each region. This distribution of points in the F1, F2 plane is used as the input pattern to be compared with the reference patterns.

Listing 2 illustrates the classification and accumulation routine. The G1 and G2 arrays contain the information about the division of the F1, F2 plane. They should be initialized (in BASIC or equivalent machine code) as follows:
The calculation of index $I$ involves a multiplication by a constant 8. This can be most quickly done by coding:

$$I = (12-1)^*2 + (12-1)^*4 + 1$$

This BASIC statement can easily be translated into machine language using two left shifts and addition operations.

The pattern matching is done by correlating the $P$ array with precomputed reference patterns for each word in the vocabulary. The word with the best match is picked as the most likely utterance.

Listing 3 gives the pattern matching routine, expressed in BASIC. $L$ is the number of words in the vocabulary. $R$ is an array that contains the calculated correlation values for each word. $T$ is a 30 by $L$ array of reference templates. The number corresponding to the template which best matches the input is printed as $J1$. The calculation of $R$ requires 30 times $L$ multiplications, which can be very time consuming. For a ten word vocabulary and an 8080 processor (one 8 by 8 bit software multiply in 230 $\mu$s) the $R$ calculation can be done in 69 ms. Also note that while the numbers in $P$ are always positive, this does not apply to the numbers in $T$. $R$ should be a double precision (2 byte) number to avoid overflow.

The templates are generated by averaging several sample patterns for each word. Much of this task can be done offline rather than in real time, so it can be coded in BASIC or other interpretive languages. A direct machine language coding is relatively straightforward, however.

Listing 4 shows how eight patterns are averaged to form a template. The program uses the code of listings 1 and 2 to obtain the pattern in array $P$. The accumulation of the $P1$ array should be done with 2 byte arithmetic. The division by 8 is accomplished by a simple right shift three bit places.

Finally, a little data massaging is necessary. To make this type of correlation pattern matching work, each template should have the same statistical properties. Specifically, each template should have a mean value of zero and a normalized standard deviation. This is easily done with the program in listing 5. The removal of the mean is simplified by the fact that the mean for each template is the same (60 intervals divided by 30 regions gives a fixed mean value of 2). The normalization of the standard deviation involves computing a square root. If the square root function subroutine is not available, then it can be computed manually with a pocket calculator and entered by hand (PRINT S : INPUT S2). This is not too painful since it needs to be done only once for each word in the vocabulary. $S$ should be a 2 byte integer, and the division by $S2$ should be done using double precision arithmetic.

The templates can be saved and used indefinitely without alteration, but new templates will be necessary for different speakers. Composite templates can be made by averaging the reference patterns of several talkers, but they do not work as well as individualized ones.

This very simple speech recognizer is only one of many systems that can be adapted to microcomputers. Much more information on the digital analysis and synthesis of speech can be found in the IEEE Transactions on Acoustics, Speech and Signal Processing, which is available in most engineering libraries.

### BIBLIOGRAPHY


---

An Application for This Speech Recognition System

An excellent application for Dr Boddie's system would be to use it as a quick way to enter hexadecimal code without toggling or typing. The vocabulary required could be kept to a minimum if the operator pronounces each digit separately (i.e.: "seven, seven," not "seventy seven") so that the various compound forms do not have to be learned. The computer would then verify the verbal entry by outputting the code to a printer or display screen. Better yet, use this system in conjunction with a computerized speech synthesizer such as D Lloyd Rice's Computalker [BYTE, August 1976, page 16] to obtain a verbal interaction. A more sophisticated system could process verbal op code mnemonics and assemble them directly in hexadecimal code on your display screen for verification.
How to Pick up a Dropped Bit

The oxide coating of a tape is sometimes unevenly distributed, particularly when the tape is old and has been used many times.

The phenomenon of the dropped bit causes difficulties in two distinct areas of computer technology: in the recording of data on tape (or disk, or the like), and in the transmission of data from one place to another. Suppose, for example, that we are recording one hundred 32-bit words on a tape. Out of the 3200 bits that are to be written on the tape, there is a nonzero chance that at least one of them will be wrong. Either it will be recorded as a zero, when it should have been a one (a dropped bit) or it will be recorded as a one, when it should have been a zero (an added bit). Even if all 3200 bits are recorded correctly, there is still a nonzero chance that the next time we read this tape, we will read at least one of the one bits from the tape as if it were a zero, or one of the zero bits as if it were a one. In such a case we again speak of dropping a bit, or adding a bit. Often both dropped and added bits are referred to, generically, as dropped bits, and we shall continue to do so in this paper.

In a similar way, suppose we are transmitting a message which consists of bits. (The message does not have to involve computers at all; it may, for example, simply be a message from one Teletype to another.) In a long message there is, again, a very good chance that at least one bit which is transmitted will be received in the wrong way. It might be received as a zero, when it is supposed to be a one, or vice versa. Again we speak of bits being dropped in transmission. (One slightly confusing piece of terminology here is that the entire collection of bits, out of which a very few are dropped, is very often referred to as a "message," even when we are not transmitting it, but rather recording it on tape or disk.)

There are many possible sources of dropped bits. Tapes often have tiny dust particles on them which interfere with the reading and writing of data. The oxide coating of a tape is sometimes unevenly distributed, particularly when the tape is old and has been used many times. The same considerations, of course, apply to floppy disk memory, or any other kind of memory involving an oxide coating. In transmitting messages from one place to another, noise in the channel and receiver can very easily degrade the quality of the reception.

In order to solve the problems created by dropped bits, we can proceed in two general classes of ways. The first is to improve our hardware so that dropped bits do not occur: We can clean our tapes. We can throw away our old tapes. We can transmit messages at a slow rate, and so on. The other approach is what may be called "picking up" the dropped bits. The idea is to send a message that is longer than the original one, and that is so designed that, even if certain bits are dropped, the original message can be recovered. (As we mentioned above, the word "message" is being used here in a general sense; it may, in particular, be a record written on tape or disk.)

Picking up a dropped bit is referred to, more precisely, as "error correction," a term which must be carefully distinguished from "error detection." In error detection, we simply detect the fact that some bit has been dropped; we cannot tell which bit is the
wrong one. If we could tell that, then we could correct the error, because if a bit is "one," and it is wrong, then it must really be "zero," and vice versa. Knowing which bit is wrong is what enables us to do error correction.

Both error detection and error correction are affected by the question of how many bits in a message are wrong. If we can tell which bit is wrong, assuming that there is only one wrong bit, then we have single error correction — even though we might not be able to correct more than one error in a message (or we might correct these errors in the wrong way). In the same way, if we can tell, for example, that something is wrong whenever one or two (but no more than two) bits in a message are wrong, then we have double error detection, even though there might be some patterns of more than two wrong bits which are such that we cannot tell that anything is wrong.

A simple way to perform error detection is by means of what is commonly called parity checking. In large systems, 8-bit bytes recorded on an IBM standard tape are always accompanied by a ninth bit, the parity bit. This bit is so chosen that the total number of one bits among the nine bits (the original eight, plus the parity bit) is an odd number (1, 3, 5, 7 or 9). Sometimes it is done the other way around, that is, to give even parity instead of odd parity; but we shall ignore this alternative for the moment.) Now let us suppose that no more than one of these nine bits has been dropped. When the bits are read again, if the total number of one bits is an even number, we know that there has been a dropped bit. We do not know, however, which of the eight bits was dropped. (Actually, for all we know, it might be the ninth bit — the parity bit itself — that was dropped.)

Parity checking is extended, on IBM standard tapes, to provide for the possibility of error correction as well as error detection. This is done by performing "two-dimensional" parity checking, as shown in figure 1. Each column has a parity bit, and each row also has a parity bit. Now suppose that exactly one of the bits in the message of figure 1 was dropped. If that bit was in (say) the fifteenth column and the second row, then the parity in the fifteenth column will be wrong — there will be an even number of one bits in it, rather than an odd number — and the parity in the second row will also be wrong. If these two parities are wrong, it is then a simple matter to find the bit in the fifteenth column and the second row and change it (from a zero to a one or from a one to a zero).

The assumption that is made in this parity scheme is that dropped bits will be infrequent enough that, in a message or a record on tape) of this size, only one bit, at the most, will be dropped. For this reason the two-dimensional parity checking scheme of figure 1 is said to provide single error correction, but not double error correction. Suppose that the bit in the seventeenth column and the first row is also dropped. This means that the fifteenth column, the seventeenth column, the first row and the second row will all have wrong parity. In this case, which two bits are wrong? Let us denote by (x, y) the bit in column number x and row number y. In this case, it is bits (15, 2) and (17, 1) that were dropped. But it could just as easily have been bits (15, 1) and (17, 2), and exactly the same erroneous behavior would have occurred. In other words, we can't tell, when there is a double error, where the double error is if this scheme is used, and thus, we have no way of correcting it.

The two-dimensional parity checking scheme does, however, provide double error detection. Whenever there is a double error, that is, whenever exactly two bits in the message are dropped, we can detect this fact. This is true even when both errors are in the same column. Of course, in that case, there will be no way to tell what column the errors are in. All of the column parities will be right, because an odd number of bits with two changes in it remains an odd number of bits. The only way we can tell that some-
thing is wrong, in this case, is that exactly two of the row parities are wrong. The same thing is true in reverse, if the two dropped bits are in the same row, but not in the same column.

In fact, this scheme will not only detect all double errors, it will detect almost all multiple bit errors. If there are more than two dropped bits, as long as these are randomly distributed (and not caused by a scratch on a tape, for example), then the probability is overwhelming that all of them, or at least most of them, will be in different columns, and thus there will be quite a number of column parities that will be wrong. We cannot refer to the given scheme as a multiple error detection scheme, in general, because there are some cases in which errors go completely undetected. For example, consider the four bits we treated earlier, namely bits (15, 2), (17, 1), (15, 1), and (17, 2). Suppose that these four bits are all dropped, and that no other bits are dropped. Now we have an undetectable error: All our parities, including those in the fifteenth and seventeenth columns and in the first and second rows, will be right. But this is so infrequent an occurrence that it may, for all practical purposes, be ignored.

Standard parity checking schemes of this kind can be improved upon in two ways: The first is by increasing their efficiency; the second is by increasing the number of errors that can be corrected.

Suppose that there are no errors at all in a given 256 bits.
1 to 255. The efficiency of the scheme is not too badly affected; we now have eight check bits for every 255 data bits, rather than eight for every 256.

The second problem with the scheme is that we have not considered the possibility that one of the check bits might be wrong. Suppose that the last check bit is wrong, and that all the other bits are right. Then using the scheme above, we would obtain the 8 bit pattern 00000001, and this would tell us that it was bit number 1 of the original 256 that was wrong. In general, an error in any of the bits numbered 1, 2, 4, 8, 16, 32, 64 or 128 can be confused with an error in one of the check bits. The solution, however, is again very simple: We just leave these bits out. We are now transmitting only 255-8 = 247 bits of data, with eight check bits, and again the efficiency is not too badly affected. If the 8 bit pattern we obtain contains seven 0 bits, and only one 1 bit, then we know that it is a check bit that is wrong.

The third problem, however, is more serious. Suppose that more than one bit out of the 256 was dropped. Note that in our scheme, if there is an error of any kind, we determine one particular bit position to change. In other words, we always assume that if there is an error, it is a single error. If there is a multiple error we are always going to do the wrong thing. This is in contrast with the two-dimensional parity checking scheme, in which we almost always know that something is wrong, no matter how many bits get dropped. The only solution to this problem is a partial one: We can forget about correcting errors and use this scheme to detect errors only; and if we do this, all double errors will be detected. In other words, this scheme can be used for single error correction or double error detection, but not both.

A remarkable property of our scheme is that the eight check bits can all be generated simultaneously. We take the exclusive OR of all the binary integers 00000001 thru 11111111 (or 1 thru 255 in decimal) — leaving out 1, 2, 4, 8, 16, 32, 64 and 128, as noted above — which correspond to one bits in the message. That is, if the j-th bit in the message is a one bit, then the integer j, written in binary, is exclusive ORed with all other integers i with the same property. The resulting 8 bit quantity consists of precisely the eight check bits we need. For 1 ≤ k ≤ 8, the k-th bit of this quantity is the exclusive OR of as many one bits as there are positions i in the message, such that bit i is a one bit and the k-th bit of the integer i is also a one bit, together with a number of zero bits, which do not affect the exclusive OR. An algorithm for performing this process is as follows:

1. Initialize so as to point to the first bit of the message.
2. Set R1 = 3. (R1 will contain the index i as above.)
3. Set R2 = 0. (R2 will contain the eight check bits.)
4. Set R3 = 4. (R3 will be 4, 8, 16, 32, etc., as above.)
5. If the current bit in the message is a zero bit, skip the next step (that is, go to step 7).
6. Set R2 equal to the exclusive OR of R2 and R1.
7. Point to the next bit of the message.
8. Set R1 = R1 + 1.
9. If R1 ≠ R3 then go to step 5.
10. Set R3 = R3 + R3.
11. If R3 ≠ 512 then go to step 7.

At this point, if we are writing a message, we append the eight check bits in R2 on to the end of the message. If we are reading a message, we read the next eight bits and form the exclusive OR of these bits with R2. If the result is zero, the message is without error. If it is 1, 2, 4, 8, 16, 32, 64 or 128, then one of the check bits is wrong. If it is anything else — call it j — then the j-th bit is wrong, and must be changed (to a zero if it is a one, or vice versa).

It should also be clear that there is a scheme like this for any number m of check bits. We have here taken m = 8, and the number of data bits is 2m−m−1 = 256−8−1 = 247; but we could have taken m = 4, for example, obtaining four check bits for each 11 bits of data. This provides another approach to the problem of multiple errors in 256 bits; we can require only that there be no multiple errors in 11 bits (say), at the cost of a certain loss of efficiency.

Is it possible to pick up more than one dropped bit at a time? That is, can we devise a scheme that is capable of double error correction? Yes, we can; we can even provide n-tuple error correction, for any (fixed) positive integer n. Schemes for doing this, however, are quite complex, and their complexity increases with the number of errors to be corrected. There is a whole subfield of electrical engineering called the theory of error correcting codes, which concerns itself with schemes of this kind. It is remarkable that error correcting codes involve one of the few known practical applications of the theory of Galois fields. (Every mathematician knows the tragic story of Galois, a French math student back in the Age of Dueling who got involved in a challenge to a duel, and, knowing his opponent was a far better duelist than he, spent his last night on earth.

Suppose that more than one bit out of the 256 was dropped.
A fundamental concept in all error correcting codes is "Hamming distance." Consider two code words \( C_1 \) and \( C_2 \); each code word consists of a message (data bits) together with the check bits for that message. Take the exclusive OR of \( C_1 \) and \( C_2 \); the number of one bits in the result is called the Hamming distance between \( C_1 \) and \( C_2 \). If the Hamming distance is 1, this means that \( C_1 \) and \( C_2 \) are the same, except for one bit position at which they are different. This in turn means that if it was that particular bit which was dropped, then \( C_1 \) will get mistaken for \( C_2 \), or vice versa.

On the other hand, suppose that a particular code has the property that for every pair of code words \( C_1 \) and \( C_2 \), the Hamming distance is 3 or more. Now suppose that \( C_1 \) is a code word and \( C_2 \) is not. Suppose that when a message is transmitted, then, due to some bit being dropped, it is \( C_2 \) that is received when it should have been \( C_1 \). That is, the distance between \( C_1 \) and \( C_2 \) is 1 (since only one bit was dropped). In this case the error can always be corrected. That is, of all possible code words, we can always tell that \( C_1 \) is the one we wanted. To prove this, suppose that there were another code word, \( C_3 \), that is actually the one we wanted. Then the distance between \( C_3 \) and \( C_2 \) would be 1 (since we are assuming that only one bit is dropped), and we already know that the distance between \( C_1 \) and \( C_2 \) is 1. But in this case the distance between the two code words \( C_1 \) and \( C_3 \) cannot be greater than 2, and this contradicts our assumption that two code words must have a distance between them of 3 or more.

In general, if the minimum Hamming distance between any two code words is 3, the code is a single error correcting code (although the actual correcting of the errors might, in some cases, be an elaborate and inefficient process). We can extend this immediately and say that, if the minimum distance is \( d = 2e+1 \), the code is an \( e \)-tuple error correcting code (usually referred to as an e-error correcting code), for any integer \( e \). This code will not detect any more than \( e \) errors, unless we sacrifice some error correction capability. If \( x \) and \( y \) are integers with \( x > y \) and \( x + y + 1 = d \), then we can use a code with minimum distance \( d \), as above, to correct any \( y \) errors and simultaneously detect any \( x \) errors. As a special case of this, if \( y = 0 \), we can detect any \( d-1 \) errors without any error correction capability at all.
Memories are made of this.


Now, on only one *fully static* card, the Digital Group has squeezed in a whopping 32K of memory. Which, with a little quick addition, means a full 64K architecture now requires only 2 boards instead of 8. That's a 4-to-1 space reduction ... and leaves one extra memory slot on the Digital Group's standard motherboard still available for future products.

All this and one low price, too.

It just may be the best news of all. Our full static, assembled and tested 32K memory board is only $995. Now that's worth remembering. It's substantially less than our equivalent assembled 8K board prices. (Please note: We're initially offering this 32K board assembled only, but kit versions will soon be available, too—at even lower prices.)

Here's what you get.

**Specifications:**
- 32K on single card
- Speed—450ns. All of our current CPUs will operate at full-rated speed.
- Decoding—Lower or upper 32K bank
- Power—+5V only @ 4A
- Card size—12” x 5” (excluding connector fingers)

**Features:**
- May be intermixed on Digital Group systems with our 8K memory cards
- All data and address lines are buffered
- *Fully* static memories—EMM 4801 (450ns) or equivalent

**Price:**
32K board complete, assembled and tested $995.00

For all the memorable details, just fill out the coupon below. (Then all you have to remember is to mail it in.)

---

**the digital group**

P.O. Box 6528
Denver, Colorado 80206 (303) 777-7133

☐ I promise to mail this in, so add me to your mailing list!

Name

Address

City/State/Zip

☐ Remember me? I'm already on your mailing list, but I need the memory spec sheet desperately.

Circle 21 on inquiry card.
Figure 1: Conceptually, it's easy to expand the effective memory space of a processor by adding a latched output port which provides extra high order bits of addressing. The result is a "paged" memory technique which can address a much larger memory address space. In this example, four bits added to 16 bits yields 20 address bits, giving a microprocessor an effective addressing range of 1048576 locations. However, there are disadvantages of such a technique in this simple conception: Each 64 K region is completely isolated from its neighbors, since the 10 port technique requires software switching between regions, which cannot in general be accomplished within one or a small number of instructions.

Give Your Micro a Megabyte

What is it that makes a microcomputer "micro?" There are many factors involved, but one of the most crucial is the small memory available in most hobby computers. I had scarcely gotten my Altair up and running with 8 K of memory before I wanted to write programs that exceeded its storage capabilities. Paging programs in and out from dual cassettes became a major nuisance, so I gathered my pennies and bought another 4 K memory board. In less than a month I needed even more storage. It is a fact of programming life that one must trade storage space for execution speed in programs. I tried to write a program in BASIC to play Mastermind on my Altair. Mastermind has 1296 possible moves, each move composed of four numbers. Storing this optimally would require about 5 K bytes; the BASIC code took nearly 6 K bytes, leaving me almost no space for my program. I tried to write the program to regenerate the moves each turn. This saved the space, but it took over ten minutes of computation to respond. Few human players can put up with such a wait. Don't think that this lack of memory afflicts only small machines. I work on several large computers, each of which has over 100 K bytes of storage. Programs have been written on each system which strain the memory limits. The unwritten law has it that "programs expand to fill all available space." We programming experimenters don't have much control over the structure of our processor architectures, but we do build up our own memory systems. In all systems that I have worked on, the memory was the single most expensive component. Peripherals rank second in most systems. This is the area in which we can upgrade our machines most readily from the "micro" class. This article will describe, in a general way, two methods for expanding the storage capability of a computer. These methods are widely used in big machines, but are rarely built into small machines. This need not be the case.

Present day 8 bit microprocessors typically have 16 bit address buses. This implies that a maximum of 64 K bytes can be directly addressed. Many minicomputers also have 16 bit buses, but often address up to a megabyte. How do they accomplish this apparent bit of magic? Obviously, they must come up with some more address bits somewhere. Minicomputers do it by using special registers called mapping registers. These registers contain the extra address bits. If the mapping register contents are fixed, then only 64 K can be addressed. But by dynamically changing the bits of (for example) a
4 bit mapping register, the processor can effectively address an entire megabyte. In minicomputers, the mapping function is usually performed automatically, when needed, by operating system software. Each task loaded in the system has access to 64 K, as determined by its mapping register. Homebrewed microprocessors can do this, too, as well as commercial systems with appropriate modifications. We need to add a register for mapping. We can't add the needed bits to the processor itself, but we can use peripheral registers as we like. Just treat the mapping register as an output device. Figure 1 shows how an 8 bit microprocessor with 16 address bits can generate an effective 20 bit address space using a 4 bit output port as a mapping register. A multitasking operating system could be written to switch up to 16 tasks, each of which has access to 64 K. With such systems, 64 K segments of data could be kept in partitions (64 K space defined by the map register), separate from the programs in which they are used. The opportunities for innovative system design are vast. As mentioned, the more memory a system has, the more complex its programs become.

"Wait a minute!" you are asking. "We were talking about microcomputers with typically 4 to 16 K. What good is extending the address space without the memory? Who could afford to build a megabyte of storage?"

Good questions. The answer is the second trick used by big systems ..., it is called "virtual memory." This is quite a sleight of hand technique. Basically, it amounts to this: If you can't have memory at the address, move the address to the memory you have. Nobody expects a system to actually have a megabyte of on line storage. That is too expensive for all but the biggest machines. Disk (or tape) is also a storage medium. It is slower, but a heck of a lot cheaper for large volumes of data. Suppose that some external storage device holds a lot of data blocked into 4 K chunks. Whenever the processor asks for data, the chunk that contains that data is read into a 4 K memory from which the processor gets it. This is the essence of virtual memory: a small but fast memory which is being filled by a large but slow external store. To the program it appears that its entire address space is filled with active memory; it just has to wait sometimes. Since most programs move more or less sequentially through memory, many memory accesses are often made to the same chunk of memory. These accesses are as fast as the real memory can be. The system can have several chunks active at the same time. This, again, is the
A trade-off of memory space for execution speed.

The rest of this article outlines a virtual memory system for a typical microprocessor. It will work with most popular processors. The one exception is Processor A which will halt the processor at any point in an instruction upon an external signal. This is absolutely necessary to halt the processor until the desired chunk (or more properly "page") of memory has been loaded. The 8080, 6502, Z-80, TMS9900 and other processors have this feature. The 6800, unfortunately, does not.

We will assume a page size of 4 K bytes. This means that the low order 12 bits of the address bus will address the memory page. We will assume a single 4 K page of real memory for simplicity. Two processors need access to this memory. The main processor uses it, but there must also be another processor which can access the memory page to move the data into and out of it. Thus, both the address and data buses must have switches to permit either processor access to the memory. Now, we need to know if a given address lies in the current memory page. The paging processor uses an output port as a page address register. This register is compared with the upper bits of the address from the main processor. A circuit for such a page select logic is shown in figure 2. It compares the upper four bits of the address bus against the contents of a 4 bit page register. It uses 7485 4 bit magnitude comparators to test whether the address is in the page identified by the page register contents. The output goes high when the address falls outside the page. (The 7485 is cascadable, so this circuit could be extended with extra bits to generate a virtual memory space of a full megabyte. This would require a megabyte of peripheral storage, or about four floppy disks.)

Processors are cheap these days. One may as well use a micro to control the external storage. It will be idle most of the time, when the main processor is happy with the present memory page. When the page select logic indicates that a change of page is necessary, the paging processor goes into action. The main processor is made to wait. The memory buses are switched to the control of the paging processor. The paging processor reads the upper four (or more) bits of the main processor address. If the present page of memory has been written into, then the new contents of the page must be restored onto the external store. The new page must be read in from external storage and placed into the memory page. The page written latch is reset, since we will want to know if the main processor changes the contents of the page. Finally, the page address register is updated. This releases the main processor to continue processing as though nothing had happened. It never noticed that the paging processor took over. The mechanism of storage used by the paging processor is a detail of the design of the virtual memory system. It could use hard disks, floppy disks, 3M drives, digital or even audio cassettes. (A very ingenious commercial unit using a microprocessor and interpreter for the APL language runs such a virtual memory system on cassettes. Watching this little beast run its cassettes around while running a program is quite impressive. It features a 256 K byte effective address space, limited by the storage on a cassette. [See the MCM/
A simple flowchart of such a paging processor program is shown in figure 3. The main program is simply initialization followed by a wait loop. The reset function of the main processor also starts the paging processor. There is a small difficulty here: The memory page must be initialized by the paging processor before the main processor can use it. Some means must be found to keep the main processor busy while the memory page is being initialized. A oneshot triggered by the reset line (or by the paging processor), which holds the main processor in the wait state for sufficient time for the memory to be set up, will work. An output bit from the paging processor can also be used. The initial page would probably be the operating system. This brings up another interesting point. Since the entire content of memory is stored on an external, and nonvolatile, medium, one can think of the entire memory space as a bit like ROM. A page can be write protected by not letting the paging processor restore that page from real memory. Such memory protection can be implemented by a ROM which is addressed by the page address lines from the main processor and which resets the page written latch. Thus, the page processor doesn't think that the page has been changed, and doesn't write on the external copy. Software in the paging processor can implement program control of "write protection."

Another point is that if the main processor has ROM in its address space, one doesn't want the paging processor to waste time trying to page into that address area. A similar use of a ROM in the paging processor can inhibit paging of certain pages in which the main processor ROM is located.

With a virtual memory system, and perhaps memory mapping, there is often no need to do IO with large data blocks. Everything is, effectively, in memory address space and IO is often hidden by the paging process. This can greatly simplify programs that use lots of data.

Virtual memory systems are not especially cheap. Using a floppy disk, a system such as I describe here would probably cost around $2000 to build. A single floppy stores about 256 K bytes; so with mapping, this system would allow a quarter megabyte address space. Think of wiring together two thousand 2102s! Think of the power supply that would require! Besides, 2102 chips cost much more than a dollar each. The virtual memory space could be doubled by adding another floppy drive. It should take quite a while for you to exceed these storage limits. When you do, the system readily expands. You're not likely to byte off more than your micro can chew.●

Figure 3: What to do when a page fault occurs. This is a simplified flow diagram of a virtual memory system control process, including initialization and response to a page fault interrupt. It is assumed to be running in a microprocessor dedicated to memory management.
An Introduction to Numbers

The concept of fixed point numbers, scaled numbers and floating point numbers originated in the scientific computer environment at a time when computer was generally considered to be either a scientific computer or a business computer rather than the general purpose computer of today. Business computers used fixed point numbers but designers felt no need to describe them as such because that was the only kind of number used. Some business computers used decimal arithmetic rather than binary arithmetic and allowed a variable amount of decimal digits for various variables and values of programs. Some business computers referred to their decimal digits as characters and regarded every character whether or not was a decimal digit having a decimal digit value. But here we're concerned with the wider concepts of number representation in a general purpose computer.

In the general purpose machine we regard all numerical values to be binary numbers in some sense. The point in fixed point and floating point is not a decimal point but is a binary point. A binary point in a binary number plays the same role as the decimal point in a decimal number. The binary number 101 (meaning 101.) has the decimal value of 5. The binary number 10,1 has the decimal value of 2.5. The binary digit to the left of the binary point has the place value, positional value, of one; the binary digit next further left has the place value of two, then four, then eight, etc. The first binary digit to the right of the binary point has the place value of one half, then one fourth, one eighth, etc. Binary 1010.1010 is decimal 10.625. Conversion is seen by adding the digit's place values 10.625 = 8 + 2 + 0.5 + 0.125.

The binary value in a register or memory location contains only binary digits as a succession of binary zeroes and ones. It has nothing in it that is explicitly a binary point. It is the responsibility of the programmer to decide the assumed position of the binary point. If the binary point is assumed to be to the right of the least significant binary digit, the value is an ordinary integer. Such a value is often called a "fixed point" number as in FORTRAN or PL/I. The binary point can be assumed to lie anywhere within the word or anywhere outside of the word. When the binary point is assumed to be fixed at any place other than at the right of the least significant bit (LSB) it is commonly called a scaled value, or scaled fixed point value. Any other fixed placement is a scaled binary number.
Logic Probe 1 is a compact, enormously versatile design, test and troubleshooting tool for all types of digital applications. By simply connecting the clip leads to the circuit's power supply, setting a switch to the proper logic family and touching the probe tip to the node under test, you get an instant picture of circuit conditions.

LP-1's unique circuitry—which combines the functions of level detector, pulse detector, pulse stretcher and memory—makes one-shot, low-rep-rate, narrow pulses—nearly impossible to see, even with a fast scope—easily detectable and visible. HI LED indicates logic "1", LO LED, logic "0", and all pulse transitions—positive and negative as narrow as 50 nanoseconds—are stretched to 1/2 second and displayed on the PULSE LED.

By setting the PULSE/MEMORY switch to MEMORY, single-shot events as well as low-rep-rate events can be stored indefinitely.

While high-frequency (5-10MHz) signals cause the "pulse" LED to blink at a 3Hz rate, there is an additional indication with unsymmetrical pulses: with duty cycles of less than 30%, the LO LED will light, while duty cycles of over 70% will light the HI LED.

In all modes, high input impedance (100K) virtually eliminates loading problems, and impedance is constant for all states. LP-1 also features over-voltage and reverse-polarity protection. Housed in a rugged, high-impact plastic case with strain-relieved power cables, it's built to provide reliable day-in, day-out service for years to come.

**CSC'S MULTI-FAMILY LOGIC PROBE 1.**

**AT $44.95, IT DIGS UP A LOT OF INFORMATION WITHOUT BURYING YOUR BUDGET.**

**PULSE LED**—Lets you know what's going on—and off. Indicates positive and negative pulse and level transitions. LP-1 stretches pulses as narrow as 50 nanoseconds to full 1/2 sec. (3Hz pulse rate)

**PULSE/MEMORY Switch**—PULSE position detects and stretches pulses as narrow as 50 nanoseconds to full 1/2 sec. Switch to MEMORY and it stores single shot and low-rep-rate events indefinitely. HI/LO LED's remain active.

**HI/LO LED's**—Display level (HI-logic "1", LO-logic "0") of signal activity at node under test.

Non-corrosive nickel-plated probe tip and clip leads—For reliable contacts and maximum life.

Rugged high impact plastic case—Built to take it...in the lab or in the field.

Protected—Features built-in reverse polarity and over-voltage protection; strain-relieved power cable.

Contemporary Specialties Corporation
EASY DOES IT

44 Kendall St., Box 1942 New Haven, CT 06509
WFX 710-468-1227

West Coast Office: Box 7809, San Francisco, CA 94119
TWX 910-372-7092

See your CSC dealer or call 203-624-3103 (East Coast) or 415-421-8872 (West Coast) 9 AM to 5 PM local time. Major credit cards accepted. Add $2.50 for shipping and handling in the U.S. and Canada on direct orders of $50.00 or less; $3.00 for orders over $50.00. On all foreign orders add 15% to cover shipping and handling.

Circle 22 on inquiry card.
The values which can be represented as an integer in one 8 bit byte are from zero to 255 when all values are considered to be unsigned, and therefore positive. In signed two’s complement notation the fixed point values in one byte are the integers from -128 to +127. In either case there are 256 different values possible, of which all are integers with no fractions permitted.

Scaled fixed point binary is not often used except as a part of a floating point word. When scaled numbers were widely used in earlier computers, the binary point was frequently placed, or assumed to be placed, in the center of a long computer word of 36 bits, 48 bits or 60 bits in length. Another common placement used was at the left of the most significant bit (MSB) but to the right of the sign bit of a two’s complement number. In scaled values of this type the values are always less than one and greater than minus one. Many modern large computers have instructions to facilitate operations on these fractional scaled values.

<table>
<thead>
<tr>
<th>Sign Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Assumed Binary Point</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Scaled fixed point values will not figure strongly in our futures, except in those rare cases where speed or other application dependent criteria require optimization.

For signed and scaled fractional two’s complement numbers, the largest binary value is 0.1111111 and the smallest binary value would be 1.0000000 where the digit to the left of the binary point is the sign bit; however, if we are to limit the range to fractional values, we must throw out the 1.0000000 state and treat 1.0000001 as the most negative value. This leaves us with 255 states ranging from +127/128 to -(127/128) or performing the division, +.9921875 to -.9921875.

Extending Precision

The fact that a particular computer uses bytes for its memory storage and registers does not mean that a datum must be one byte. A unit of data can be any number of bits regardless of the computer word length. It can be 19 bits on a 13 bit word machine if you program such a construction. This would be unusual but it certainly is possible. It is customary to define the numerical data so that it will use 1,2,3 or some other whole number of bytes or words. Dividing data on memory address boundaries cases and simplifies programming.

Scaled values were useful enough for many purposes but their use was troublesome to the programmer. If a value got too large, there was danger of overflow. If a value got too small, there was a loss of significance caused by too many leading zeroes and the danger of the value becoming zero. Special scaling factors had to be used from time to time to keep the problem in hand, and of course the effects of the scaling factors had to be removed when the computations were completed. The invention of floating point numbers cured most of the scaling difficulties.

Enter Floating Points

A floating point number has two parts for each value. One part is a fraction which is a scaled fixed point number as described above. The fractional part has many names. It may be called the fraction, mantissa or coefficient. The typical floating point number has a fraction whose absolute value is always less than one. The minimum fractional value is determined by the base of the other part of the floating point number. This other part of a floating point number has been called the exponent, power or characteristic depending on whose description you read. It is the exponent or power to be applied to some base (also called the radix) that forms a scaling factor. The value of the number is the fractional part multiplied by the base raised to the power of the exponent’s value.

For many years the base for the exponent part of a floating point number was almost invariably two. For a base radix of two a nominal minimum absolute value for the fractional part is one half. We set up our hardware or software to force the most significant bit of the mantissa to be one, in order to “normalize” our numbers. Similarly, for each possible base we constrain the fractional value: For a base of four the minimum fraction value is one fourth. For a radix of eight the minimum fraction is one eighth and for 16 it is one sixteenth. IBM System 360/370 uses a radix of 16 for the exponent base in floating point numbers and so do various other computers and systems. Whatever the radix for the exponent, whenever the fraction is greater than or equal to the minimum value for that base, the floating point number is said to be normalized. (In order words, it is the “normal” or “best” form.)

There is very little standardization among floating point numbers. The radix for the exponent’s base may be two or it may be ten
Peripheral Vision is a brand-new company that's dedicated to selling reasonably priced peripherals for various manufacturers' CPUs.

We think you'll flip over our first product.

It's a full-size floppy disk for the Altair-Imsai plug-in compatible S-100 BUS. And it's available for as low as $750.

Here are the features:

- 1 interface card supports 4 drives
- Stores over 300,000 bytes per floppy
- Bootstrap EPROM included—no more toggling or paper tape
- Completely S-100 plug-in compatible
- Interface cabling included
- Drive is from Innovex (the originator of the floppy concept)—assembled and tested
- Interface card design is licensed from Dr. Kenneth Welles and the Digital Group
- Disk operating system with file management system included on floppy
- Cabinet and power supply optional

Prices:

<table>
<thead>
<tr>
<th></th>
<th>Kit</th>
<th>Assm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface card kit and assembled and tested drive</td>
<td>$750</td>
<td>$850</td>
</tr>
<tr>
<td>Power supply—+24V at 2A</td>
<td>45</td>
<td>65</td>
</tr>
<tr>
<td>Cabinet—Optima, blue</td>
<td>—</td>
<td>85</td>
</tr>
</tbody>
</table>

Now, a little more about our company.

Peripheral Vision may be brand-new, but we have some old-fashioned ideas about how to run our business.

We know there are serious incompatibilities among the different manufacturers' peripherals and CPU's. We want to get them together. And, we want to bring significant new products to market—products consisting of everything from adaptation instructions/kits for hardware and software to major new products.

It's a tall order, but we feel we're up to the task. Peripheral Vision has already obtained a license from The Digital Group to adapt versions of some of their products to the S-100 BUS. And we're working on getting more from other companies.

Most important to our customers, Peripheral Vision is committed to helping you get along with your computer. We'll do all we can to make it easy.

Write us now for all the information on our company, our philosophy and our exciting line of products. And be prepared to flip over all of it.

Send me the works, and I just might flip over it!

Name ____________________________
Address __________________________
City/State/Zip ______________________

Circle 23 on inquiry card.
or even 16. Within a floating point word the bits for the exponent can be either before or after the fraction. The number of bits for the exponents vary widely in different systems as do those for the fractions. There are various schemes for showing the signs for the exponents and the signs for the fractions.

The dynamic range of values allowed for a floating point number is determined primarily by the radix for the exponent and the number of bits in the exponent. The IBM System 360 and 370 hardware uses an 8 bit exponent of which one bit is used to take care of the sign for the exponent (it is not a truly a sign bit but the overall effect is much the same) and one bit handles the sign for the fraction which leaves six bits to determine the value of the exponent. The largest value in six bits is 63 which, when applied to a base of 16, produces 16^63 or approximately 7X10^75. The smallest positive exponent is equivalent to 16^-64 or approximately 9X10^-78. These exponents must be multiplied by some fraction between 0.0625 and 1 to get the final value of a floating point word. The exponent for a Univac 1100 series computer is nine bits which loses two bits for the signs. The remaining seven bits would seem better than the remaining six for the System 370 except that the radix of the exponent is two. The largest exponent is 2^127 or approximately 1.7X10^308 and thus much smaller than that for the IBM version. The Univac double precision floating point word allows numbers to 2^1024 or approximately 1.8X10^308 which is pretty impressive.

The precision, or accuracy limit, of a floating point number is determined by the number of bits in the fraction part. Just in case you’re mistakenly tempted to regard accuracy and precision as being the same, let us learn to distinguish between them. Precision relates to the ability to differentiate between value representations that are nearly the same. In terms of decimal values the precision can be to eight significant digits but if the accuracy is less than the precision then some or all of these eight digits are nonsense. Accuracy cannot exceed precision. Precision can be defined as the maximum possible relative accuracy. You cannot easily ascribe a precision to a 1 bit field because there is only one nonzero value possible.

In spite of slight conceptual error it is often convenient to regard the precision as the representational error caused by the variation of one in the least significant bit position for a field width in bits that does not include leading zero bits. Using this method for two bits we can stipulate four values, so we can say the precision is about 25%.

From this point let us decide to give precision as the precision in bits in the bit field that does not include leading zero bits. Then, for each such precision, we can compute an error that will have nothing at all to do with accuracy beyond placing a limit on the accuracy. This is equivalent to saying that a method or procedure which produces a value can be totally wrong but this does not reduce the precision in the value as it is represented. We can claim the square root of 4 to be 1.389567. This square root is quite precise but not very accurate. When one thinks of errors it is usually errors caused by all inaccuracies rather than just those errors caused by precision of expression: However, for the remainder of this article I will take a narrow view of errors and assume they are all caused by precision only. The nominal maximum relative errors for different field widths, measured in bits, are summarized in table 1.

Our approximate rule for the maximum error in a 2 bit value gave us 25% when the true value was about 20%. As the field width in the number of bits increases the approximate rule improves and is close enough in any case. It is important to remember that the effective field width does not include leading zeroes.

The floating point fraction on the Univac 1108 (Univac calls it the mantissa) is 27 bits wide and the most significant bit is

---

Table 1: This is a summary of the nominal maximum relative error that occurs for different field widths. The field widths are measured in bits. The maximum error column is calculated as being the maximum error or change in value that occurs if the least significant bit is lost or changed. The decimal digits column indicates the number of decimal digits which are unaffected by relative error in the representation.

<table>
<thead>
<tr>
<th>Bits</th>
<th>Number of Values</th>
<th>Maximum Error</th>
<th>Decimal Digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>.250</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>.125</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>.063</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>.031</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>.016</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>8E-3</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>4E-3</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>512</td>
<td>2E-3</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1k</td>
<td>1E-3</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>2k</td>
<td>5E-4</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>4k</td>
<td>2E-4</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>8k</td>
<td>1E-4</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>16k</td>
<td>6E-5</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>32k</td>
<td>3E-5</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>64k</td>
<td>2E-5</td>
<td>4</td>
</tr>
<tr>
<td>17</td>
<td>128</td>
<td>1E-5</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>256</td>
<td>4E-6</td>
<td>5</td>
</tr>
<tr>
<td>19</td>
<td>512</td>
<td>2E-6</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>1M</td>
<td>1E-6</td>
<td>6</td>
</tr>
<tr>
<td>21</td>
<td>2M</td>
<td>5E-7</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>4M</td>
<td>2E-7</td>
<td>6</td>
</tr>
<tr>
<td>23</td>
<td>8M</td>
<td>1E-7</td>
<td>6</td>
</tr>
<tr>
<td>24</td>
<td>17M</td>
<td>6E-8</td>
<td>7</td>
</tr>
<tr>
<td>25</td>
<td>34M</td>
<td>3E-8</td>
<td>7</td>
</tr>
<tr>
<td>26</td>
<td>67M</td>
<td>1E-8</td>
<td>7</td>
</tr>
<tr>
<td>27</td>
<td>134M</td>
<td>7E-9</td>
<td>8</td>
</tr>
<tr>
<td>28</td>
<td>268M</td>
<td>4E-9</td>
<td>8</td>
</tr>
<tr>
<td>29</td>
<td>537M</td>
<td>2E-9</td>
<td>8</td>
</tr>
<tr>
<td>30</td>
<td>1G</td>
<td>9E-10</td>
<td>9</td>
</tr>
<tr>
<td>31</td>
<td>2G</td>
<td>5E-10</td>
<td>9</td>
</tr>
<tr>
<td>32</td>
<td>4G</td>
<td>2E-10</td>
<td>9</td>
</tr>
</tbody>
</table>
always set for positive values. Thus the precision, expressed as the maximum representational error, is about one part in $10^8$ and is equivalent to seven or eight significant decimal digits. The single precision floating point fraction on the IBM System 370 computers is allowed a width of 24 bits but because the exponent radix is base 16, the normalized fraction can have from none to three leading zeroes. The precision therefore varies from an error of one part in $10^8$ for 24 bits to one part in $10^{17}$ for 21 bits. The equivalent decimal precision is about six or seven significant decimal digits. The IBM fraction is not quite as good as Univac's but the dynamic range allowed by the floating point exponent is greater. Both are greatly inferior to the CDC machines with their 12 bit exponent and 48 bit mantissa in a word of 60 bits.

It is not customary to use a signed value in the exponent part of floating point numbers. The more usual arrangement is to bias the exponent by adding a constant. The exponent range for the IBM floating word is from $16^{-64}$ to $16^{63}$. The 16 is not shown but is assumed. IBM adds 64 to the exponents so that the floating exponent part for the value $16^{-64}$ is zero and for $16^{63}$ is 127. A floating value of one is equivalently $16^{1/16}$ which yields binary $01000001000000000000000000000000$ as a single, 32 bit value. In hexadecimal this is $41100000$. The exponent part in hexadecimal is 41. Table 2 shows a summary of the binary and hexadecimal digit placement as used by IBM.

Negative floating values are typically formed one of two ways. The word as a whole is simply arithmetically inverted or else only the sign bit is inverted. Either of these methods is satisfactory and neither changes the dynamic range of the exponent nor the precision of the fraction.

Many IBM users go to double precision floating point because the precision equivalent to 6 decimal digits is not sufficient for their needs. In this case the exponent, and the dynamic range, is not changed but the fraction width is increased to 56 bits which is equivalent to about 17 decimal digits. Double precision on the Univac allows 12 bits for the exponent, versus the single precision 9 bits, and 60 bits for the fraction, versus the single precision 27 bits. Double precision on the CDC machines is almost ridiculous — the exponent is essentially repeated in the second word which allows 96 bits for the fractional part which is equivalent to more than 30 decimal digits!

If I were writing a complete programming system for an 8 bit byte machine, I would not use a 4 byte copy of the IBM floating word but would use two bytes for the exponent with an exponent base radix of two rather than 16 and four bytes for the fraction part. A 32 bit fraction is equivalent to about nine significant decimal digits and the exponent range would be ridiculously large. Or maybe one should use the IBM method but adding two bytes to the fraction part. My point here is that I personally do not like a word with only six decimal digits of precision.

<table>
<thead>
<tr>
<th>hexadecinal</th>
<th>4</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary</td>
<td>01000001</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
</tr>
</tbody>
</table>

**Table 2:** The breakdown of a sample number of a 32 bit word into binary and hexadecimal digit groups.
SYNCHRO-SOUND Enterprises
Your Small Computer Department Store

One-stop shopping for Hardware and Software

Compare our PRICES, PRODUCTS, QUALITY, DELIVERY, SERVICE...we carry everything you need in small computers, terminals, printers, monitors, keyboards, accessories. We'll meet or beat competitive prices on everything we sell!

NEW LEAR SIEGLER ADM-3A TERMINAL

- Full addressable cursor.
- Character Generation — 5 x 7 dot matrix.
- Display Format — Standard: 1920 characters, displayed in 24 lines of 80 characters per line.
- Character Set — Standard: 64 ASCII characters, displayed as upper case, plus punctuation and control.
- Communications Rates — 75 to 19,200 baud (switch selectable).
- Computer Interfaces — EIA standard RS232C and 20mA current-loop (switch selectable).
- Data Entry — New data enters on bottom line of screen; line feed causes upward scrolling of entire display with top-of-page overflow. Automatic new line switch selectable, end-of-line audible tone.

ADM-3A Kit $839.95
ADM-3A Assembled 1079.95
Lower Case Option 89.00

IMSAI 8080 MICROCOMPUTER

Powerful • Easy to use • Low cost

With 22 Slot Mother Board $619.95
With Z-80 CPU 849.95

OKIDATA Model 110 Line Printer

- 110 CPS dot matrix

Friction Feed $1099.00
Tractor Feed 1229.00
RS 232C Serial Interface 250.00

SPECIAL BUYS

HAZELTINE Modular One Video Terminal $1639.00
SORAC IQ120 Video Terminal Kit 959.00
DEC LA180 Printer 2769.00
COMPUCOLOR 8001 Color Graphics Computer 2595.00
ICOM Microfloppy Disc & Controller 989.00
NORTH STAR Microfloppy Disc & Controller 599.00
OKIDATA Model 22 Printer 2095.00
OMNITEC 401C 300 Baud Modem 300.00
2708 EPROMS 29.95
OAE Papertape Reader 69.95

SOL 20 In Stock For Immediate Delivery
The Model SS-1 Communications Terminal is a non-contact capacitive keyboard with a guaranteed life of over 100,000,000 operations.

- MOS/LSI Encoder with high reliability and low power consumption.
- Key Roll Over which eliminates operator error and increases throughput.
- Hysteresis for tear proof operation.
- Solid State performance at mechanical switch prices.
- Tactile Feedback at the operate point.
- ASR-33 Array with four mode encoding.

Keyboard Kit ........ $99.95
Enclosure (with numeric cutout) ............ 49.95
Numeric Pad ........ 34.95
Complete Kit (with pad) 179.95
Assembled Unit (with pad) ........ 229.95

We carry a full line of the following: TDL, OAE, Processor Tech., Hayden, Tarbell, IMSAI, Cromemco, Compucolor, Icom, Lear Siegler, Okidata, DEC, Javelin, Teletype, North Star, Heuristics, Peripheral Vision, TRW. Same day delivery and shipping on most items. Full, modern repair facilities on premises for complete servicing of everything we sell.

SOFTWARE
Attention ALTAIR DISK Owners

Now available—an Accounts Receivable System for the small user. This system is expressly designed to be run on an 8080 based microcomputer using an Altair floppy disk with Altair Basic, Video Terminal and Printer. Many features normally found only on large scale computers are included. Provided are such functions as: Adding New Customer Accounts, Deleting Dormant Accounts, Transaction Processing, and Report Generation. The Monthly Statements include both current and aged balances. Other reports generated are the Aged-Accounts Receivable and Delinquency Notices.

This easily usable system requires very little operator training. All communication is in an interactive mode with the program constantly prompting the appropriate responses. Operator errors are easily corrected and accidental database deletions are prevented by requiring additional confirmation.

This Accounts Receivable System, although supplied ready for use, may readily be modified to include special installation-dependent functions. All major programs are written in high level language making program additions and alterations a simple task. This program module is designed to serve as the basis for a complete accounting package. Provisions are included for linking Billing Inventory Control and General Ledger modules which will be available in the near future.

DECreter II

- 132 column printing • 30 CPS
- Full keyboard • Tractor Feed

$1769.95

SYNCHRO-SOUND Enterprises
The Computer People
193-25 Jamaica Avenue, Jamaica,
New York 11423 • 212/488-7067

Hours 9-4 daily and Saturday
Visit our new showroom . . .
working units on display
Dept. B BankAmericard • Master Charge

Please send me further information on: (list items)

Name ____________________________
Address __________________________
City __________________ State ______ Zip ________

Price List _________

Circle 25 on inquiry card.
I am an enthusiastic reader of your magazine. Since I own an SR-52 with PC-100 printer attachment, I have read "The Buried Gold in the SR-52" and "Desk Top Wonders" [December 1976 BYTE, pages 30 and 92] with great interest.

But I think that the register organization offers a much more fantastic ability: By the use of memory functions and program storage registers, a program can alter itself!

To find out the correspondence between codes and program register contents, we execute the following sequence using the SR-52's 2 digit keycodes:

*EE 12 STO 70 *leet *list

and we get:

display = 3.14159265412
printout = 00020 00101 00200 00354 00426 00559 00641 00731

The following program should prove very useful to SR-52 enthusiasts:

000 0 001 0
004 0 005 0
008 0 009 *LBL 010 A
012 0 013 0
016 *LBL 017 B 018 EE 019 5
020 8 021 + 022 RCL 023 7
024 1 025 = 026 STO 027 7
028 0 029 *rtn 030 0 031 0

To see how it works, perform the following sequence:

* π STO 06 5 STO 97
Input 0.7094336
Press B, display = 5.607904336 60
Press A, display = 3.141592654 00

The A function operates exactly the same way as if we had keyed in the following steps (in the learn mode):

*IND RCL 97

To replace those four steps by GTO 123 (whose codes are 41 01 02 03), we would merely input .3020141 and press B.

This unique feature opens a new dimension in programming.

For SR-52 games hunters, I offer the following version of NIM which allows up to nine rows of 1023 pawns each. (This program should be run with the PC-100 printer.)
I hope that such programs will encourage Texas Instruments to create a European SR-52 Users Club. Many people are for sure waiting for personal computer system dealers in Europe.

Example of Play

<table>
<thead>
<tr>
<th>Player</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
</tr>
<tr>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
</tr>
<tr>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
</tr>
<tr>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
</tr>
<tr>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
</tr>
<tr>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
</tr>
<tr>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
</tr>
<tr>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
</tr>
<tr>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
<td><strong>SR-52</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Player</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
</tr>
<tr>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
</tr>
<tr>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
</tr>
<tr>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
</tr>
<tr>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
</tr>
<tr>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
</tr>
<tr>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
</tr>
<tr>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
</tr>
<tr>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
<td><strong>Player</strong></td>
</tr>
</tbody>
</table>

|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|

Play Instructions
1. Load in the play card both A and B sides
2. Initialize: *fix D *CMs *reset
3. Enter the starting position: For each non-empty row, input its number, n, (1 ≤ n ≤ 9) and press B; then input the number of pawns, x_n, (0 ≤ x_n ≤ 1023) and press A.
4. To play against the SR-52, input the number, n, of the row you have altered and press B; then input the number of pawns that remain in this row and press C.
5. To begin a new game go to line 2 above.

For all of you non-afficionados of the Computer Art...
A Machine Code Relocator for the 8080

Many microcomputer hobbyists pass through an early stage in the development of their systems where machine language is the only mode available for programming. My system configuration at that point included an IMSAI mainframe and 4 K memory, a VDM-1 video driver and monitor, and a Tarbell cassette interface. The big problem when programming something like LIFE for the VDM in machine language occurs when it becomes necessary to insert a routine (call it routine y) between locations x and x+1 in memory. I found myself with two miserable options open:

1. Replace three or so bytes (the exact number depending on where the new instruction begins, on byte 3, 4, or 5) with a patch to routine y, placed somewhere else in memory, then restore the lost bytes at the end of routine y and jump back to location x+4, or so. This is not exactly ideal.
2. Manually relocate the program from location x+1 on down until the end of the program (unless you left plenty of NOPs to handle just such a situation — then, when you’re all finished, you’re left with a bunch of useless NOPs) to make room for routine y. This involves writing the whole thing out on paper first then toggling it in. Once again, far from optimal.

My solution to this dilemma was to write a machine language program which performs the second type of relocation above quickly and painlessly when given just 11 bytes of relocation data as shown in table 1.

The program, which I call the “relocator,” performs its duty in two phases. Phase I simply takes the block whose first address is a, and whose last address is b, then transfers it as is to the area beginning with location x+1. Phase II takes the label number argument and moves this block to the memory location after that label number is found.

Leor Zolman has been writing programs for 4 years, including applications such as medical graphics for Cedars of Lebanon Hospital in his hometown of Los Angeles. He is currently working on a 2 K resident monitor for the 8080, and characterizes himself as a BASIC games enthusiast. Mr Zolman is 17 years old and a freshman at MIT.

<table>
<thead>
<tr>
<th>Label</th>
<th>Number of Bytes</th>
<th>Address</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>2</td>
<td>0DDD, 0DDE</td>
<td>First address of block to be relocated</td>
</tr>
<tr>
<td>b</td>
<td>2</td>
<td>0DDF, 0DE0</td>
<td>Last address of block to be relocated</td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>0DE1, 0DE2</td>
<td>Destination address (ie: where the byte at location a should go)</td>
</tr>
<tr>
<td>d</td>
<td>2</td>
<td>0DE3, 0DE4</td>
<td>First address to have references fixed</td>
</tr>
<tr>
<td>e</td>
<td>2</td>
<td>0DE5, 0DE6</td>
<td>Last address to have references fixed</td>
</tr>
<tr>
<td>f</td>
<td>1</td>
<td>0DE7</td>
<td>Function select: 00 = fix references only 01 = move block and fix references 02 = move block only</td>
</tr>
</tbody>
</table>

Table 1: These six key pieces of information must be entered into the locations shown (beginning at hexadecimal 0DDD for listing 1) before relocation can be performed.
address c. That's it for phase 1. If the function select byte is equal to 02, then the relocator quits here and goes into an infinite loop which shows up as a known pattern in the lights of an Altair or IMSAI front panel. This is an indicator of the end of execution.

Phase 2 is the tricky part. All addresses (meaning all 2 byte operands of 3 byte instructions) which fall between items a and b above (inclusive) must have been added to the distance (number of bytes) which the block was moved. This value is easily computed by the expression c-a. For example, if a ten byte block was moved from starting location 10 to decimal starting location 75, then the value 65 must be added to any 2 byte operand in the program having a value from 10 to 20 inclusive.

The purpose of items d and e in the relocation data is to tell the relocator where it must look for such operands (references), so that they may be altered or "fixed" accordingly. Since the references may lie anywhere within the program (not necessarily just within the relocated block), d should be set equal to the very first location of the program, and e should be set to the last. The exception to this, which occurs when a data block lies in the way, will be covered later. [To avoid the exception, the "good" programming practice of keeping data separate from executable code is highly recommended . . . CH] Phase 2 may be executed without phase 1 if the function select is set to 0.

The beauty of this program becomes apparent when a friend having 32K in his machine hands you a tape of a program he wrote residing somewhere up in the sky (or so it seems to you, with only 4K) and you would like to run it. All you have to do in this case is:

1. Load in the relocator
2. Load in your friend's program in any free memory space
3. Feed the relocater this relocation data:
   a. the original first address of the program
   b. the original last address of the program
   c. the location where you loaded it (first location)
   d. same as c
   e. c + length of the program (this equals c + b - a + 1)
   f. 00 (function select to fix references only)
4. Examine the start of the relocater and hit run. When the LEDs stop flickering (about 1 second for each 1 K between

**Listing 1**: A relocation program for use on 8080a systems. The program takes any existing program and transfers it to any other location desired in memory with a minimum of fuss. Throughout this program, endless loops are used to signal the end of major operations. These loops will cause the data LEDs to output a constant light pattern on the front panel (line 0D37 for example). If desired, the program can be modified so that it jumps to any other desired routine at these points, or returns to your monitor. All numbers followed by an H are considered to be hexadecimal in this program (eg: 02H in line 0D35). Interestingly enough, while the program resides at starting location 0D00, it is perfectly capable of relocating itself!
Listing 1, continued:

<table>
<thead>
<tr>
<th>Address</th>
<th>Hexadecimal Code</th>
<th>Label</th>
<th>Op</th>
<th>Operand</th>
<th>Commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>06</td>
<td>CB</td>
<td>D</td>
<td>B,12H</td>
<td>move b to left, go back to CHEK2; or skip; move to next byte, etc.</td>
</tr>
<tr>
<td>67</td>
<td>01</td>
<td>CB</td>
<td>D,</td>
<td>TABLE2</td>
<td>start phase 2, select destination, move from c to a, etc.</td>
</tr>
<tr>
<td>6A</td>
<td>01</td>
<td>CB</td>
<td>D</td>
<td>CHEK2</td>
<td>move a to top, go to CB, move to destination, then byte n-1 would be moved, then n-2, etc., up to byte 1. Therefore, relocating forward into higher memory always works, but relocating backward into lower memory fails if the difference between the source and destination addresses is not greater than the block length. In such a case, say moving a block at b to location a, an intermediate relocation must be done from b to c (where lc-al &gt; length of block), and then a second relocation must be performed from c to a. Note: I could have had the relocator check the direction of the move</td>
</tr>
<tr>
<td>71</td>
<td>01</td>
<td>CB</td>
<td>D</td>
<td>D</td>
<td>start phase 1, load from destination, move to source, then byte n would be moved, then n-2, etc., up to byte 1. Therefore, relocating forward into higher memory always works, but relocating backward into lower memory fails if the difference between the source and destination addresses is not greater than the block length. In such a case, say moving a block at b to location a, an intermediate relocation must be done from b to c (where lc-al &gt; length of block), and then a second relocation must be performed from c to a. Note: I could have had the relocator check the direction of the move</td>
</tr>
<tr>
<td>7A</td>
<td>01</td>
<td>CB</td>
<td>D</td>
<td>E</td>
<td>start phase 1, load from destination, move to source, then byte n would be moved, then n-2, etc., up to byte 1. Therefore, relocating forward into higher memory always works, but relocating backward into lower memory fails if the difference between the source and destination addresses is not greater than the block length. In such a case, say moving a block at b to location a, an intermediate relocation must be done from b to c (where lc-al &gt; length of block), and then a second relocation must be performed from c to a. Note: I could have had the relocator check the direction of the move</td>
</tr>
</tbody>
</table>

Relocating to a Lower Memory Location

The block move (phase 1) is done tail first. For a block of length n, then, byte n of the source would be transferred to byte n of the destination, then byte n-1 would be moved, then n-2, etc., up to byte 1. Therefore, relocating forward into higher memory always works, but relocating backward into lower memory fails if the difference between the source and destination addresses is not greater than the block length. In such a case, say moving a block at b to location a, an intermediate relocation must be done from b to c (where lc-al > length of block), and then a second relocation must be performed from c to a. Note: I could have had the relocator check the direction of the move |

- d and e hit stop. You should see C3 in the data LEDs of an Altair or IMSAI. |

Figure 1: A special case in program relocation. The program block a thru b is to be moved to location c, but the integrity of the data block from f thru g must be preserved. The technique in this case is to change program references in two passes: up to, but not including, the data block, and then after the block. Locating all data at the end of programs will of course prevent this type of problem. |

94
and proceed either tail or head first accordingly, but the program is long enough to cause toggling headaches as it is.

Problems with the LXI Instruction

LXI instructions in which the operand happens to equal an address in the block between a and b but doesn't actually refer to an address (LXI H, 0000 is a good example) present a problem. If you are in the process of writing the program try to use LHLD instead (or, if worse comes to worse, LHLD with MOV B, H and MOV C, L for LXI B, xxxx) and set aside two bytes of storage at the end of the program for the data.

If the program was not written by you, or if you really want to use LXI, then the only alternative is to go through your listing (if you have one!) and change the locations in question. But this problem never caused me much trouble.

Data Block Problems

Avoid relocating data whenever possible. If relocation is necessary do not try passing through it with phase 2. Here is an example of how to handle data areas: In a program residing from location d to location e, a block starting at a and ending at b is to be moved to location c. As a complication, a data block resides from location f to location g. See figure 1 for a memory map of this situation. The relocation should be performed in two runs of the relocator as follows:

Run 1. Give relocation data of a, b, c, d, f-1, 1. This moves the block and fixes all references up to location f, the start of the data area.

Run 2. Give data of a, b, c, g+1, e, 0. This simply fixes up the remaining references after skipping the data area.

[Here again the practice of keeping data separate from executable code should alleviate this problem...CH/]

In the listing provided the relocator resides at locations 0000 to 0DFF, but it is self-relocatable, of course. The six key items of relocation data must be set up at locations 0DDD to 0D7 as shown in table 1, with the low byte coming first numerically in each case.

I sincerely hope that this program repays your understanding and toggling efforts with a vast reduction in relocation frustration.

The value of this program becomes apparent when a friend having 32 K on his machine hands you a tape of a program he wrote residing somewhere up in the sky (or so it seems to you, with only 4 K), and you would like to run it.
BASICally BASIC

(an informal introduction to BASIC)

The BASIC programming language was developed in the early 1960s at Dartmouth College as a conversational, problem solving language. It has wide applications in scientific, business, and educational environments since it can be used to solve both simple and complex mathematical problems from the user’s terminal. Due to the small number of clearly understandable commands and programming statements required, BASIC is one of the simplest languages to learn and use.

Like most programming languages, BASIC can be divided into two sections: elementary statements for simple programs and more advanced techniques and statements for complicated problems. As an introduction to BASIC, this article deals primarily with the elementary functions and statements recognized by BASIC as it is most frequently implemented, with program examples included to illustrate their applications. Several operating commands and a few of the more advanced features found in some BASIC systems will also be mentioned.

The specific features available on each computer system will differ slightly, forming many different dialects of the BASIC language. When writing programs to be compatible with different systems, it’s general practice to use only elementary statements common to most forms of BASIC. The BASIC programming manual or equivalent documentation should always be consulted to verify the commands and statements actually available and the conventions used for a particular system.

BASIC Operating Commands

After gaining access to BASIC in whatever manner required by a particular system, BASIC will normally respond with a message indicating it is waiting for a command. The various operating commands depend on the BASIC implementation used. Several of the commands usually available are:

**OLD:** Load a previously saved program. BASIC may request the old program (or file) name, depending on the system.

**NEW:** Enter (or write) a new program from the console. BASIC may request a program (or file) name depending on the system.

**LIST:** Print the current program on the console. Some systems may allow printing individual lines or groups of lines.

**RUN:** Compile and execute the current program. Some systems may allow starting at a particular line.

**SAVE, RESAVE, or REPLACE**
Save the current program.

**UNSAVE, PURGE, or SCRATCH**
Delete the current program.

**BYE, GOODBYE, or SYSTEM**
Exit BASIC.

In addition, some systems may accept abbreviated commands or offer various editing commands to possibly delete lines, resequence line numbers, etc. Most BASIC interpreters will usually type READY when waiting for an operating command. New lines are inserted by simply typing a line number followed by the desired instruction terminated with a carriage return. Line numbers and individual instructions will be discussed later.

Fundamental Programming Concepts

What does it take to represent a program in a high level language such as BASIC? As
Satisfy your appetite for computer knowledge with Sams cookbooks

Send for the cookbooks and manuals described. Increase your knowledge of minicomputers, microprocessors, computer technology, related computer circuits and peripheral equipment. Be satisfied or your money back.

How To Buy & Use Minicomputers & Microcomputers
By William Barden, Jr. This manual gives you the basics of minicomputers and microcomputers. Explains their hardware and software, the peripheral devices available and various programming languages and techniques. Allows you to decide which system is best for your needs. 240 pages; softbound. No. 21351 $9.95

CMOS Cookbook
By Don Lancaster
Your complete guide to the understanding and use of Complementary Metal-Oxide-Silicon integrated circuits. Gives usage rules; power supply design examples; applications; information on breadboards, testing, tools, and interface. Detailed coverage of logic and more. 416 pages; softbound. No. 21398 $9.95

Computer Dictionary and Handbook
By Charles J. Sippi & Charles P. Sippi
At your finger tips you have more than 22,000 definitions, acronyms, and abbreviations dealing with the field of data processing. Also 13 appendices cover a myriad of computer related subjects. 784 pages; hardbound. No. 20850 $19.50

TTL Cookbook
By Donald E. Lancaster
You’ll discover what Transistor-Transistor Logic is, how it works and how to use it. Discusses practical digital applications. You’ll learn to build TTL systems that entertain, test and train. 356 pages; softbound No. 21035 $9.95

Computer Dictionary and Handbook
By Charles J. Sippi & Charles P. Sippi
At your finger tips you have more than 22,000 definitions, acronyms, and abbreviations dealing with the field of data processing. Also 13 appendices cover a myriad of computer related subjects. 784 pages; hardbound. No. 20850 $19.50

TTL Cookbook
By Donald E. Lancaster
You’ll discover what Transistor-Transistor Logic is, how it works and how to use it. Discusses practical digital applications. You’ll learn to build TTL systems that entertain, test and train. 356 pages; softbound No. 21035 $9.95

TTL Cookbook
By Donald E. Lancaster
You’ll discover what Transistor-Transistor Logic is, how it works and how to use it. Discusses practical digital applications. You’ll learn to build TTL systems that entertain, test and train. 356 pages; softbound No. 21035 $9.95

RTL Cookbook
By Donald E. Lancaster
You will learn the how and why of Resistor-Transistor Logic. Obtain useful design information and many digital applications. 240 pages; softbound. No. 20715 $5.75

Microcomputer Primer
By Mitchell Waite and Michael Pardee
Written for the beginner in the computer field. All the basic concepts and characteristics of microcomputers are explored. The easy to understand language prepares you for further study. 224 pages; softbound. No. 21404 $7.95

Send your order Today!

How To Buy & Use Minicomputers & Microcomputers
By William Barden, Jr. This manual gives you the basics of minicomputers and microcomputers. Explains their hardware and software, the peripheral devices available and various programming languages and techniques. Allows you to decide which system is best for your needs. 240 pages; softbound. No. 21351 $9.95

Computer Dictionary and Handbook
By Charles J. Sippi & Charles P. Sippi
At your finger tips you have more than 22,000 definitions, acronyms, and abbreviations dealing with the field of data processing. Also 13 appendices cover a myriad of computer related subjects. 784 pages; hardbound. No. 20850 $19.50

TTL Cookbook
By Donald E. Lancaster
You’ll discover what Transistor-Transistor Logic is, how it works and how to use it. Discusses practical digital applications. You’ll learn to build TTL systems that entertain, test and train. 356 pages; softbound No. 21035 $9.95

TTL Cookbook
By Donald E. Lancaster
You’ll discover what Transistor-Transistor Logic is, how it works and how to use it. Discusses practical digital applications. You’ll learn to build TTL systems that entertain, test and train. 356 pages; softbound No. 21035 $9.95

TTL Cookbook
By Donald E. Lancaster
You’ll discover what Transistor-Transistor Logic is, how it works and how to use it. Discusses practical digital applications. You’ll learn to build TTL systems that entertain, test and train. 356 pages; softbound No. 21035 $9.95

RTL Cookbook
By Donald E. Lancaster
You will learn the how and why of Resistor-Transistor Logic. Obtain useful design information and many digital applications. 240 pages; softbound. No. 20715 $5.75

RTL Cookbook
By Donald E. Lancaster
You will learn the how and why of Resistor-Transistor Logic. Obtain useful design information and many digital applications. 240 pages; softbound. No. 20715 $5.75

NAME
ADDRESS
CITY
STATE
ZIP

Send books and/or wall charts checked below, $ enclosed. I understand that, if not completely satisfied, I may return my order within 10 days for a full refund.

21351 21398 21035 21080 21399 20850 21313 21351 21035 21168 20850 21404 21715

Canadian prices slightly higher.

Howard W. Sams & Co., Inc.
4300 West 62nd Street
Indianapolis, Indiana 46206

Circle 26 on inquiry card.
Table 1: Arithmetic Operators. A typical BASIC interpreter incorporates an arithmetic expression parsing algorithm which recognizes the usual add, subtract, multiply, 
divide and exponentiate operations. Precedence is described in the text, and parentheses are used for explicit ordering of operations.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>A+B</td>
<td>Add A and B</td>
</tr>
<tr>
<td>+</td>
<td>+A</td>
<td>A is positive</td>
</tr>
<tr>
<td>+</td>
<td>A-B</td>
<td>Subtract B from A</td>
</tr>
<tr>
<td>-</td>
<td>-A</td>
<td>A is negative</td>
</tr>
<tr>
<td>*</td>
<td>A*B</td>
<td>Multiply A times B</td>
</tr>
<tr>
<td>/</td>
<td>A/B</td>
<td>Divide A by B</td>
</tr>
<tr>
<td>**</td>
<td>A**B</td>
<td>A to the power B</td>
</tr>
</tbody>
</table>

with any computer language, BASIC uses certain notations for numbers, variables and operations.

Numbers

The computer can compute the value of expressions like 12/7 or SQR(9); but while they use numbers, they are not considered numbers and may not be used in lists of constant data. Numbers are self defining values which are generally expressed in decimal form and may be positive or negative (ie: 3 or -2.79. Most BASIC languages will also allow scientific notation for numbers, where the letter “E” is used to signify 10 raised to a power. A number expressed in this form, such as a*Eb, would indicate “a” times 10 raised to the power “b”.

With scientific notation, a number must always be present on both sides of the letter “E”, and may be either positive or negative. There is usually a maximum number of digits allowed with either number notation depending on the particular computer and BASIC implementation.

Variables

Simple numeric variables are generally specified by any single letter or any letter followed by a single digit. Thus, E7 would be interpreted as a variable along with A, B, C2, and X0. Certain computer systems may allow more advanced variable labeling. But if you expect compatibility with other BASIC systems, it is best to use this restricted, simpler form.

Numeric variables represent a number with a value. The value is assigned to the variable by either LET or READ statements. An assigned value will not change until the next LET or READ statement which explicitly changes the variable. In the typical BASIC interpreter, all numeric variables are initially set equal to zero before a program is run. It’s only necessary to assign an initial value to a variable when a value other than zero is desired.

Arithmetic Operators

Arithmetic computations in a language like BASIC are performed by evaluating single line formulas similar to those used in standard mathematical calculations. Any of the arithmetic operators listed in table 1 may be used to write a desired formula. In computations utilizing multiple operators, the order of precedence is determined by a very conventional set of rules:

1. A formula inside parentheses is evaluated before the parenthesized quantity is used in a computation.
2. Normally, an operator cannot follow another operator. The operators + and - can, however, follow the operators *, /, ** or **. In this case, the + or - takes precedence over the leading operator to indicate the sign of a quantity.
3. Without parentheses in a formula, ** and ** take precedence over * and /, which take precedence over + and -.
4. With only * and / operators and no parentheses, BASIC performs the operations from left to right in the order they are read.
5. With only + and - operators and no parentheses, BASIC performs the operations from left to right in the order they are read.

The following examples will help illustrate these rules and how a BASIC interpreter typically executes various computations:

B+3*A**2 = A squared times 3, then added to B
B+(3*A)**2 = Product of 3 times A is squared, then added to B
(B+(3*A))**2 = Product of 3 times A is added to B and the sum is then squared
A/B/C = Quotient of A divided by B is then divided by C
A-B-C = C is subtracted from the result of B subtracted from A
A**2**3 = The result of A squared is then cubed
The most cost effective products for your microcomputer.

THE RM64 MEMORY BOARD provides the most cost effective means for system memory addition in the industry. The board is S-100 bus compatible making it usable in over a dozen different microcomputer systems including ALTAIR and IMSAI.

It comes in three configurations: 32K, 48K or 64K bytes and is completely assembled, checked out and burned in for at least 50 hours prior to shipment. This complete testing procedure enables EXTENSYS to provide you a one year warranty on parts, labor and materials (assuming no misuse of the board occurs).

On board hardware is provided for:
- Individual memory bank address selection in 8K increments;
- Complete dynamic refresh logic without loss of processing efficiency while programs are running;
- Board select logic which allows more than one 64K byte board per system;
- S-100 bus compatibility including on-board voltage regulator;
- Memory overlap which allows memory sharing the same address space to coexist in the same system; and
- Write protection in 16K blocks

THE CGM77 COMPUTER GRAPHICS MACHINE is a fully tested system that provides the basis for developing the most sophisticated video graphics imaginable. The CGM77 board is S-100 bus compatible, making it usable on over a dozen different computer systems. The only additional components needed for a complete video graphics system are 8K bytes of RAM interfaced to the microcomputer system. The EXTENSYS RM64-32K RAM board is ideally suited for this.

Special control panels are available that provide universal control for interfacing with the graphics board. Output of the board is in the form of composite sync that can be used directly by a high resolution video monitor.

The heart of the system is the EXTENSYS GRAPHICS MACHINE. This is a sophisticated piece of software developed by EXTENSYS for the rapid implementation of video graphics. The GRAPHICS MACHINE permanently resides in PROM chips that are part of the board and therefore never need to be loaded. The GRAPHICS MACHINE software actually transforms the microprocessor into a much more powerful and sophisticated computer whose instruction set is ideally suited for the development of video graphics or video games.

Contact your local computer store or order directly from EXTENSYS.

Circle 27 on inquiry card.
Arithmetic Functions

In addition to the five arithmetic operations, a fully implemented BASIC interpreter can evaluate several standard mathematical functions. The elementary functions available in a typical BASIC are listed in table 2. Any valid formula may be substituted for X in all of the functions. More advanced functions are often available and will be discussed later. Various other mathematical functions may be available depending on the individual computer system used. Sometimes extended nonarithmetic functions are also made available.

Relational Symbols

When it is necessary to compare values, six mathematical symbols of relation are used in IF statements. Some systems may use the alternate written expressions, especially when the terminals do not have the standard symbols. Table 3 lists the standard symbols and alternate forms of relational operations.

Line Numbers

In any BASIC program, each statement must include a line number as mentioned earlier. The first line is usually numbered 10 and all following line numbers are generally in steps of 10. This allows insertion of extra statements between existing lines by using a line number between the two lines where the statement is to be added. New lines are correctly inserted by BASIC by using the line number specified with each statement. The only restriction on line numbers in a program is that each line number must be greater than the preceding line number.

Elementary Programming Statements

The elementary statements of the BASIC language are identified by the first word of the statement.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIN(X)</td>
<td>Sine of X</td>
</tr>
<tr>
<td>COS(X)</td>
<td>Cosine of X</td>
</tr>
<tr>
<td>TAN(X)</td>
<td>Tangent of X</td>
</tr>
<tr>
<td>COT(X)</td>
<td>Cotangent of X</td>
</tr>
<tr>
<td>ATN(X)</td>
<td>Arctangent of X</td>
</tr>
<tr>
<td>EXP(X)</td>
<td>e raised to the power X</td>
</tr>
<tr>
<td>LOG(X)</td>
<td>Natural logarithm of X</td>
</tr>
<tr>
<td>SQRT(X)</td>
<td>Square root of X</td>
</tr>
<tr>
<td>ABS(X)</td>
<td>Absolute value of X</td>
</tr>
</tbody>
</table>

Table 2: Typical BASIC Mathematical Functions. The usual implementation of a full BASIC interpreter provides this list of built-in mathematical functions. The precision and range of numbers depends upon the details of the implementation and its number representation internally.

LET Statement

The LET statement is used to assign a given number or the result of a calculation to a particular variable or group of variables. The general form of the LET statement is:

\[ \text{LET} \quad \text{variable} = \left\{ \begin{array}{l} \text{formula}^* \\
\end{array} \right. \]

*Any item shown in brackets will indicate a general form of an argument, such as:
\( \text{variable} = \) any variable line number = any line number

With some BASIC implementations, the word LET may be optional in the statement.

Examples of assigning a value to a single variable would be:

10 \ LET \ X=1
20 \ LET \ A=B*2/3

In many BASICS, several variables may be assigned the same value by a single LET statement, such as:

30 \ LET \ X=Y2=B=4

Also, a variable may appear on both sides of the equal sign since the entire calculation is completed using "old" values of data before the result is assigned to variables on the left hand side of the "=" sign. For example, the statement:

100 \ LET \ X=X+1

will take the old value of X added to one and assign the result as the new value of X. The LET statement is not an algebraic equality, but a command to perform the computations and assign the answer to the variables specified. It is an unfortunate fact of computer language life that most languages use the same "=" symbol for assignment (data transfer) and relational tests (data comparison).

READ and DATA Statements

These statements are used to enter information into the computer for a given program. The READ statement assigns values obtained from DATA statements to the variables listed in the READ statement. A DATA statement must be present in order to use a READ statement.

Before a program is run, the typical BASIC interpreter will take all DATA statements in the order they appear and store them in a large data block. Every time a READ statement is encountered, the next available numbers from the data block are assigned to the variables in the READ statement. If the data block runs out of data, the program will stop and an error message will be printed.

READ statements are normally located near the beginning of a program since data must be read in before working with it. DATA statements may be located anywhere in a program, as long as they occur in the
Use SpeechLab to directly control any S-100 Bus Computer such as Sol, IMSAI, Altair and so on. SpeechLab can teach you almost as much as the Bell Laboratories know about voice recognition, voice control and computer input.

SpeechLab digitizes and extracts data from speech waveform and applies pattern matching techniques to recognize the vocal input. Response is real time. The system features 64 bytes of storage per spoken word and can handle up to a 64 word vocabulary. And recognition after very little practice is 95 percent or better.

$299* assembled and tested

When we talk price everybody's skeptical. And why not? We give you a complete hardware/software system, a 275 page laboratory manual, 95 page hardware manual and high fidelity microphone.

The lab manual includes 35 graded experiments with over 100 tables and graphs. In fact, it's the only introductory volume on speech recognition currently available.

Software includes SpeechBasic Basic programming language in source and paper tape, assembly language speech recognition program in source and paper tape, hardware self-test program in source and paper tape.

SpeechBasic plot, correlation, recognition and advanced recognition programs are offered in source.

Hard to believe, you bet. True? A Los Angeles customer says, “I love your kit!!! I have 40 boards and 2 IMSAI's and your kit was the best documented of them all. I love the way you integrated the software and hardware together. I love your lab manual!”

We loved those comments. They tell the story better than we ever could. The LA customer did ask who the founders of the firm were so he could relate better. They're a couple of gifted young engineers who got tired of the big firm, big technology trip and decided to take a chance with a better idea.

You can't get better quality You can't get more performance

Sure, more complex, higher price equipment is available for about 50 times more money. It won't do much more than you can do with SpeechLab. And the quality and state-of-the-art engineering can't be any better. We use CMOS design for low power and ultimate reliability.

See SpeechLab at your nearby computer store

Selected computer stores have SpeechLab on display. Visit your nearest. If he doesn't have it, ask him to contact us or simply write us directly.
Table 3: Typical BASIC Relational Operators. These operators are used to specify comparisons between operands which might be numbers, variables or formulae. In certain BASIC implementations, substitutes are made when special characters are not available. For example, in a homebrew computer system using an older Baudot Teletype model, such substitutions would be a virtual requirement.

<table>
<thead>
<tr>
<th>Standard Symbol</th>
<th>Alternate Expression</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>=</td>
<td>A=B</td>
<td>A equals B</td>
</tr>
<tr>
<td>&lt;</td>
<td>LT</td>
<td>A&lt;B</td>
<td>A is less than B</td>
</tr>
<tr>
<td>&lt;=</td>
<td>LE</td>
<td>A\LE B</td>
<td>A is less than or equal to B</td>
</tr>
<tr>
<td>&gt;</td>
<td>GT</td>
<td>A&gt;B</td>
<td>A is greater than B</td>
</tr>
<tr>
<td>&gt;=</td>
<td>GE</td>
<td>A\GE B</td>
<td>A is greater than or equal to B</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>NE</td>
<td>A&lt;&gt;B</td>
<td>A does not equal B</td>
</tr>
</tbody>
</table>

Correct order. Many programmers find it convenient to place DATA statements just before the END statement at the end of the program. Each READ statement is of the form:

```
READ [list of variables]
```

While each DATA statement is of the form:

```
DATA [list of numbers]
```

Example:

```
100 READ X,Y,Z
200 DATA 1,2,3,4,5,6,7,8,9,0
```

The statements in this example would read the first three numbers of the data block (1,2, and 3) and assign them to X, Y, and Z. Always remember that only numbers, not formulas, may be used in DATA statements and 14/3 and SQR(7) are formulas.

**PRINT Statement**

PRINT statements are used for a variety of purposes in a typical program. While the command is called "PRINT" in many systems this means in fact "send data to the primary BASIC output device," such as a hard copy printer or electronic display device. The following example illustrates the format used to print the results of a computation or a variable value:

```
10 PRINT X, SQR(X)
```

This example would print or display the current value of X followed by the square root of X a few spaces to the right. To print a specified message, the message is enclosed in quotes and the following format is used:

```
20 PRINT "NO ANSWER"
```

This example would simply print the words "NO ANSWER" on a single line without the quotes. If desired, messages and values may be intermixed:

```
30 PRINT "THE SQUARE ROOT OF" X "IS" SQR(X)
```

This statement would type the following line if the value of X were nine:

```
THE SQUARE ROOT OF 9 IS 3
```

To skip a line or issue only a carriage return and line feed, the PRINT statement is used with no arguments:

```
40 PRINT
```

With many of the more advanced BASIC systems there are various print control characters and special statements which may be used to control the spacing of the data being printed. This control may vary between systems and is beyond the scope of this article. For the present, it's sufficient to understand the primary use of the PRINT statement and to know there are often methods available to print data in any desired format.

**GO TO Statement**

The GO TO statement is used to unconditionally jump to some specified statement in the program other than the next sequential statement. The line being jumped to may be anywhere in the program, before or after the current line. The general form of the GO TO statement is:

```
GO TO [line number]
```

Example:

```
100 GO TO 200
```

This example would always cause the program to skip to line 200 whenever reaching line 100.

**IF Statement**

The IF statement is used to conditionally jump from the sequential order of statements according to the truth of some relation. The program will skip to the designated line number only if the relation specified is true. The IF statement, sometimes referred to as a conditional GO TO statement, has the general form of:

```
IF [formula] [relation] [formula], THEN [line number]
```

Example:

```
120 IF X=0, THEN 200
130 IF SIN(X) <= 0.5, THEN 80
```

In this example, if the value of X is zero, the program will skip to line 200. If the sine of X is less than or equal to 0.5, the program will jump back to line 80. When both conditions are false, the program will go to the next line in order. The word "THEN" may be replaced with "GO TO" or the comma before "THEN" may be optional depending on the system.

**ON Statement**

Where the IF statement provides a two
THE LATEST IN TAPE SYSTEMS

MODEL CC-8 - $175.00

4800 BAUD CASSETTE RECORDER

An ASYNCHRONOUS NRZ type Recorder with remote motor start/stop. Error rate 10^-8 at 4800 BAUD. Can be used from 110 to 4800 BAUD into a UART or "Bit Banger PIA" - no clocking required. This is not an audio recorder. It takes RS232 or TTL signals from the terminal or computer and gives back the same signals. No audio interface is used. Motor start/stop is manual or through TTL or RS232 signals.

Tape speed is 3.2"/second nominal; 1.6"/sec. optional. 110 volt, 60 Hz, 5 watts, (220 Volts on special order). Can use high quality audio cassettes (Philips Type) or certified data cassettes. Can be used in remote locations from a 12 Volt battery.

Recommended for DATA LOGGING, WORD PROCESSING, COMPUTER PROGRAM RELOADING and DATA STORAGE. Especially recommended for 6800 systems, 6502 systems, 1800 systems and beginners with the 8080 systems. Manual control except for motor start/stop. 6800 or 8080 software for file or record searching available on request with order. Used by major computer manufacturers, Bell Telephone and U.S. Government for program reloading and field servicing. AVAILABILITY - Off the shelf.

OVERSEAS: Export Version 220 volt 50 Hz. Write factory or: Megatron-Datameg, 8011 Putzbrunn, Munchen, Germany; Nippon Automation 5-16-7 Shibay Minato-Ku, Tokyo, Japan; Hobbydata, FACK 20012, Malmo, Sweden; G. Ashbee, 172 Iffield Road, London SW 10-9ag: Trintronics, Ltd., 186 Queen Street W., Toronto, Ontario, Canada; EBASA, Enrique Barges 17, Barcelona 14, Spain; ARIES, 7, rue Saint Philippe du Roi, 75008 Paris; Microlem 20131, Milano, Italy; Eagle Electric, Capetown, S. Africa.

For U.P.S. delivery, add $3.00 Overseas and air shipments charges collect. N.J. Residents add 5% Sales Tax. WRITE or CALL for further information. Phone Orders on Master Charge and BankAmericard accepted.

Canadian Distributor:
Trintronics Limited
186 Queen Street West
Toronto, Canada M5V 1Z1
Tel: (416) 598-0262

MODEL 3M3A - $220.00

50 KILOBAUD CARTRIDGE RECORDER

This is a self clocking (1/1) high speed recorder. Loads BASIC in under 2.0 seconds. Recording is done at 19.2 Kilo baud. Playback at 50 Kilo baud. File or Record searching is done at 50 Kilo baud and loading is automatic. Worst case access time about 2 minutes for up to 2 megabytes on the 3M Data Cartridge.

Tape speed 10"/sec. on record, up to 30"/sec. on playback. Records one clock track and one data track on each pass (2 passes). Recording is NRZ unencoded from RS232 or TTL signals.

This recorder requires one Parallel port for motor control, and one serial port for data and clock. (Cannot be used with UART's or UART boards such as the 3P4). Used with USART's, ACIA's or other 1/1 clocking I/O devices under software control only. No manual controls. Software for 8080 and 6800 available. Power supply is built in, 110V, 60 Hz. 220 V, 50 Hz for export.

OVERSEAS: Export Version 220 volt 50 Hz. Write factory or: Megatron-Datameg, 8011 Putzbrunn, Munchen, Germany; Nippon Automation 5-16-7 Shibay Minato-Ku, Tokyo, Japan; Hobbydata, FACK 20012, Malmo, Sweden; G. Ashbee, 172 Iffield Road, London SW 10-9ag: Trintronics, Ltd., 186 Queen Street W., Toronto, Ontario, Canada; EBASA, Enrique Barges 17, Barcelona 14, Spain; ARIES, 7, rue Saint Philippe du Roi, 75008 Paris; Microlem 20131, Milano, Italy; Eagle Electric, Capetown, S. Africa.

For U.P.S. delivery, add $3.00 Overseas and air shipments charges collect. N.J. Residents add 5% Sales Tax. WRITE or CALL for further information. Phone Orders on Master Charge and BankAmericard accepted.

Canadian Distributor:
Trintronics Limited
186 Queen Street West
Toronto, Canada M5V 1Z1
Tel: (416) 598-0262

2SIO (R) CONTROLLER

$190.00 ($160.00 Kit)

PROVIDES MONITOR AND TAPE SOFTWARE in ROM TERMINAL and TAPE PORTS on SAME BOARD CONTROLS ONE or TWO TAPE UNITS (CC-8 or 3M3A)

This is a complete 8080, 8085, or 280 system controller. It provides the terminal I/O (RS232, 20 mA, or TTL) and the data cartridge I/O, plus the motor controlling parallel I/O latches. Two kilobytes of on board ROM provide turn on and go control of your Altair or Imad, NO MORE BOOTSTRAPPING. Loads and Dumps memory in hex on the terminal, formats tape cartridge files, has word processing and paper tape routines. Best of all, it has the search routines to locate files and records by means of six, five, and four letter strings. Just type in the file name and the recorder and software do the rest. Can be used in the BiSync (IBM), BiPhase (Phase encoded) or NRZ modes with suitable recorders and interfaces.

This is Revision 7 of this controller. This version features 2708 type EPROM's so that you can write your own software or relocate it as desired. One 2708 preprogrammed is supplied with the board. A socket is available for the second ROM allowing up to a full 2K of monitor programs.

Fits all $100 bus computers using 8080 or 280 MPU's. Requires 2 MHz clock from bus. Cannot be used; with audio cassettes without an interface. Cassette or cartridge inputs are RS232 level.

AVAILABILITY - Off the shelf.
way fork in a program, the ON statement (if it is implemented) allows a multiple path switch. The general form of the ON statement is:

```
ON [formula],
GO TO [list of line numbers]
```

Any formula may be used and the instruction may contain any number of line numbers as long as the statement fits on a single line.

The value of the formula is computed and truncated to an integer. If the result is one, the program skips to the line number occurring first in the list; if the result is two, it skips to the second line number in the list, etc. If the result of the formula computation is less than one or greater than the number of line numbers listed, an error message is printed.

Example:
```
20 ON X-1, GO TO 100, 200, 300
```
This example will cause one of the following:
- If X is 2, program will go to line 100
- If X is 3, program will go to line 200
- If X is 4, program will go to line 300
- If X is less than 2 or greater than 4, an error message will be printed.

The comma preceding "GO TO" may be omitted or the word "THEN" may be substituted for "GO TO" on some systems.

END Statement

Every program must have an END statement and it must be the statement with the highest line number in the program. The general form of the END statement is simply the word "END":
```
9999 END
```

STOP Statement

The STOP statement is equivalent to a GO TO statement, where the line skipped to is the END statement of the program. The statement is simply the word "STOP":
```
1000 STOP
```

REM Statement (Remark)

The REM statement provides a method of inserting remarks and comments in a program listing. Even though whatever follows REM is ignored, the line number may be used in a GO TO, IF, ON, or GOSUB statement. The general form of the REM statement is:
```
REM [any desired information]
```
Example:
```
1000 REM THIS ROUTINE AVERAGES 10 NUMBERS
```

Elementary Programming Examples

Now that the elementary statements and functions have been introduced, three sample programs are included which will help illustrate how statements are combined to accomplish various computations or comparisons.

Loops

The simplest way to execute a sequence of instructions a given number of times is by using a program loop. The block of instructions within the loop is executed repeatedly until a given condition is satisfied. In BASIC there are two statements used to specify a loop: The FOR statement is used at the beginning of the loop and the NEXT statement at the end. The general form of the FOR statement is:
```
FOR [numeric variable] = [formula1] TO [formula2] STEP [formula3]
```
Depending upon the BASIC implementation, the word "BY" may be substituted for "STEP" in the FOR statement.

The accompanying NEXT statement is of the form:
```
NEXT [numeric variable]
```
The variable in the NEXT statement must be the same variable as in the FOR statement for each loop.

Loops may be nested to a level dependent on the particular BASIC implementation system, but the loops must be legally nested and not crossed. Figure 1 shows some examples of legal and illegal nesting.

The value of the numeric variable starts at the value of formula1 and is changed by the
CURSOR CONTROL. Fore space, back space, up, down, new line, return, home, tab, PLUS ABSOLUTE CURSOR ADDRESSING.

TRANSMISSION MODES. Conversation (half and full Duplex) PLUS BLOCK MODE — transmit a page at a time.

FIELD PROTECTION. Any part of the display can be "protected" to prevent overtyping. Protected fields are displayed at reduced intensity.

EDITING. Clear screen, typeover, absolute cursor addressing, erase to end of page, erase to end of line, erase to end of field.

DISPLAY FORMAT. 24 lines by 80 characters (1,920 characters).

CHARACTER SET. 96 characters total. Upper and lower case ASCII.

KEYBOARD. 73 keys including numeric key pad.

REPEAT KEY. 15 cps repeat action.

DATA RATES. Thumbwheel selectable from 75 to 19,200 baud.

SCREEN. 12 inch rectangular CRT — P4 phosphor.
value of formula3 (either positive or negative) each time BASIC completes an execution of the loop. If the word "STEP" and formula3 are omitted, the step value defaults to positive one. It should be pointed out, however, that the control variable may be changed within the body of a program loop if desired. Exercise care when using non-integer step values since non-integer numbers cannot be represented exactly in binary and may be truncated or approximated within the computer. This may cause an undesired number of loops to be executed due to the unexpected results. [See W Douglas Maurer's "Software Bug of the Month #2" on page 81 of the July 1976 BYTE.] For this reason, every effort should be made to use an integer step value whenever possible.

Example:
5 LET F=1
10 FOR X=1 TO 10 STEP 1
20 LET F=F*X
30 PRINT X,F
40 NEXT X

This example program loop will be executed 10 times as X is incremented by one between the values of one and 10. During each pass of the loop, the numbers X and X-factorial will be printed. The FOR statement in this example could have been shortened to:

10 FOR X=1 TO 10
since the step value was one.

Subroutines

When a particular sequence of statements is to be used repeatedly in several different places within a program, they may be written as a subroutine. A subroutine is a completely self-contained program accessed by the main program or other subroutines using a subroutine calling convention.

In BASIC, subroutines are entered (or called) by using GOSUB statements anywhere in the main program (or other subroutines) of the general form:

GOSUB line number

The GOSUB statement is similar to a GO TO statement except the computer saves the location of the GOSUB statement before going to the subroutine. Statements are then executed sequentially until a RETURN statement is encountered. The RETURN statement directs the computer to return to the line following the GOSUB statement that called the subroutine. It consists simply of a line number and the word "RETURN":

1000 RETURN

and is usually the last line of a subroutine.

Subroutines may appear anywhere in the main program but should only be entered with a GOSUB statement and exited via a RETURN statement. A subroutine may actually contain several RETURN statements as long as one will be used. By using a GOSUB statement within one subroutine to call a second subroutine, subroutines may be nested. The level of subroutine nesting is sometimes limited depending on the particular BASIC implementation used.

Example:
10 LET L=4
20 PRINT "THIS PROGRAM PRINTS"
30 GOSUB 190
40 PRINT "THIS MESSAGE WITH"
50 GOSUB 190
60 PRINT "FOUR BLANK LINES"
70 GOSUB 190
80 PRINT "BETWEEN PRINTED LINES"
90 GOSUB 190
100 REM Then the numbers 1 to 10 are printed with the
110 REM number indicating the number of blank lines
120 REM following that line.
130 FOR L=1 TO 10
140 PRINT L
150 GOSUB 190
160 NEXT L
170 PRINT "DONE"
180 STOP
190 REM Subroutine to print "L" blank lines.
200 IF L>0, THEN 220
210 RETURN
220 FOR X=1 TO L
230 PRINT
240 NEXT X
250 RETURN
260 END

This example illustrates how subroutines are utilized and the use of multiple RETURN

<table>
<thead>
<tr>
<th>Sample Program =1:</th>
</tr>
</thead>
<tbody>
<tr>
<td>This program reads a group of numbers from a data block and prints the average of the numbers read. The number 9999 is used as a dummy item in the data block to indicate the end of the data block. The individual numbers of the data block are read until 9999 is found. Since all variables are initially set equal to zero until assigned another value, there is no need to include a statement to initialize S and C to zero (LET S=C=0).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Program</th>
<th>Explanation</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 REM print average of numbers</td>
<td>Data</td>
<td>End of program</td>
</tr>
<tr>
<td>20 Read X</td>
<td>Read a number X</td>
<td></td>
</tr>
<tr>
<td>30 IF X=9999, then 70</td>
<td>Check for end of data if X=9999</td>
<td></td>
</tr>
<tr>
<td>40 LET S=S+X</td>
<td>Add X to sum</td>
<td></td>
</tr>
<tr>
<td>60 LET C=C+1</td>
<td>Increment the data count C</td>
<td></td>
</tr>
<tr>
<td>60 Go to 20</td>
<td>Go back for more data</td>
<td></td>
</tr>
<tr>
<td>70 Print S/C &quot;is the average of&quot;</td>
<td>Compute and print average</td>
<td></td>
</tr>
<tr>
<td>C &quot;numbers&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80 Data 5,7,3,9,27,54,31,9999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90 End</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**top values, professionalism**

Computerland stores are exciting, enjoyable places to visit. From the striking decor to the fun and challenge of the Computerland Game room, you'll find Computerland stores a completely new shopping experience. You can count on Computerland stores for total professional support whether your needs are those of a computer hobbyist, education, science or business user. The skilled, management-trained staff offers knowledgeable service, expert maintenance, and software guidance.

**broadth of products**

At Computerland stores the emphasis is on quality products. Major brands like Cromemco, DEC, Diablo, IMSAI, Lear Siegler, Polymorphic Systems, TDL and Vector Graphic to name a few.

<table>
<thead>
<tr>
<th>Location</th>
<th>Address</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeHart Street</td>
<td>Morristown, NJ 07960</td>
<td>(201) 539-4077</td>
</tr>
<tr>
<td>DeHart Street</td>
<td>Morristown, NJ 07960</td>
<td>(201) 539-4077</td>
</tr>
<tr>
<td>6840 La Cienega Blvd.</td>
<td>Inglewood, CA 90302</td>
<td>(213) 776-6808</td>
</tr>
<tr>
<td>813 B Lyndon Lane</td>
<td>Louisville, KY 40222</td>
<td>(502) 425-6308</td>
</tr>
<tr>
<td>104 W. First Street</td>
<td>Tustin, CA 92680</td>
<td>(714) 544-0542</td>
</tr>
<tr>
<td>24001 Via Fabricante</td>
<td>Mission Viejo, CA 92675</td>
<td>(714) 770-0131</td>
</tr>
<tr>
<td>22634 Foothill Blvd.</td>
<td>Hayward, CA 94542</td>
<td>(415) 538-8600</td>
</tr>
<tr>
<td>4233 Convoy Street</td>
<td>San Diego, CA 92111</td>
<td></td>
</tr>
<tr>
<td>50 East Rand Road</td>
<td>Arlington Heights, IL 60004</td>
<td></td>
</tr>
<tr>
<td>225 Elmira Road</td>
<td>Ithaca, N.Y. 14850</td>
<td>(607) 277-4888</td>
</tr>
<tr>
<td>16065 Frederick Road (Rt 355)</td>
<td>Rockville, MD 20850</td>
<td>(301) 948-7676</td>
</tr>
<tr>
<td>1612 Niagara Falls Blvd.</td>
<td>Buffalo, NY 14150</td>
<td></td>
</tr>
<tr>
<td>Thousand Oaks, CA</td>
<td>opening soon:</td>
<td></td>
</tr>
<tr>
<td>Park Ridge, IL</td>
<td>Franchise Opportunities available — Contact: Ed Faber, President Computerland Corp (415) 895-9363.</td>
<td></td>
</tr>
</tbody>
</table>

Every store is completely stocked with tools, books and a broad range of accessories. **beginners welcomed**

Do you want to get started with microcomputers? Are you seeking expert guidance on computers, peripherals, software? The place to visit is your nearest Computerland.
the single letter denoting a list or table name may also be used as a simple variable without confusion. The same letter may not, however, be used to denote both a list and a table in the same program since a list is actually a single column table.

Example:

```
10 FOR I = 0 TO 2
20 FOR J = 0 TO 3
30 READ M(I,J)
40 NEXT J
50 NEXT I
60 DATA 1,2,3,4
70 DATA 5,6,7,8
80 DATA 9,10,11,12
```

This example routine shows how a 3 by 4 table may be entered into a program using elementary statements and how the table entries are specified.

Storage space is automatically reserved for any list or table with subscripts (typically) of 10 or less (the exact number depends on the implementation). For larger subscripts, space must be reserved by using a DIM statement. A DIM statement may appear on any line before the END statement, but it is normally placed at the beginning of the program. A DIM statement may also be used to reserve less space for a small list or table when a large program requires more program space.

The general forms of DIM statements for lists or tables are:

- `DIM [list name] ( maximum subscript )`
- `DIM [table name] ( maximum row subscript, maximum column subscript )`

Separate DIM statements are not needed for each list or table; several lists or tables may be specified in a single DIM statement by separating them with commas.

Example:

```
10 DIM V(15)
20 DIM M(20,20)
```

May also be written in a single DIM statement as:

```
10 DIM V(15),M(20,20)
```

This example reserves space for 16 items in vector V:

- `V(0), V(1), ... V(15)`

and 441 items in matrix M:

- `20 + 1 rows and 20 + 1 columns`

Advanced Statements and Functions

This section will briefly introduce the more advanced and specialized features of BASIC. These features may or may not be available in a particular BASIC implementation, whereas the elementary statements and functions already presented should always be found. Refer to the programming manual for your particular BASIC inter-
• ANY NUMBER OF FILES MAY BE OPEN (IN USE) AT ONE TIME
• THE NUMBER OF FILES AND SIZE OF FILE IS LIMITED ONLY BY THE SIZE OF THE DISC
• MERGING FILES REQUIRES NO EXTRA DISC SPACE
• NO WAITING FOR THE DISC TO RE-PACK
• LONGER DISC LIFE — MORE EVEN DISC WEAR

Our Basic Floppy Disc System (BFD-68) must, in all modesty, be called superb. It comes completely assembled with a disc controller that is plug compatible with the SWTPC 6800. In fact, all our products use the 6800 standard SS-50 (Smoke Signal 50) bus used by SWTPC. The cabinet and power supply are capable of handling up to 3 Shugart Mini-Floppy Drives. One drive is included in the price of the BFD-68 and others may be added easily at any time. Or you may save money by ordering the dual-drive BFD-68-2 or triple drive BFD-68-3 (picted). Price: BFD-68 $795, BFD-68-2 $1169, BFD-68-3 $1539, SA-400 Drive $390.

The BFD-68 includes our Disc Operating System Software. Our software provides for soft-sectorized disc format consisting of 128 bytes per sector, 18 sectors per track and 35 tracks per disc. The software provides direct commands to name and rename files, transfer memory to disc and disc to memory and to automatically jump to the starting location of any program loaded from disc to memory. The direct command names are: RUN, GET, GETHEX, CLOSE, SAVE, DELETE, APPEND, RENAME, COPY, LIST, FIND, LINK and PRINT. In addition, the Disc File Management subroutines are available to create files under your program control.

A bootstrap PROM is included on the controller board to initiate the Disc Operating System which loads into a 4K memory board located at 7000 or optionally at D000. Thus, you can be up and running from a cold start in just a few seconds.

SUPER SOFTWARE

Free patches are provided for SWTPC BASIC version 2.0 and Co-Resident Editor/Assembler. These patches allow the SAVE and LOAD commands to work with the disc or the cassette at your option.

SUPER EDITOR: Smoke Signal Broadcasting now has its own editor. It is a content oriented editor with string search and block move capability. Changes may be made by referring either to line number or string content or a combination of references. Naturally, it is designed for file transfers to and from the BFD-68. Price: SE-1 $29 on diskette.

ALL OUR PRODUCTS EXCEPT THE PS-1 ARE COMPLETELY ASSEMBLED.

SUPER ASSEMBLER: Inputs source code from file on the BFD-68 disc system and outputs object code to disc file. Assembly listings include alphabetized and tabulated symbol table. Price: SA-1 $29 on diskette.

Complete source listing included for both editor and assembler. Order both for $53 and save $5.

SMARTBUG — A CURE FOR MIKBUGITIS: A super smart Motorola-Mikbug replacement that preserves almost all Mikbug entry locations so your present programs will run without modification. Uses ACIA instead of PIA and includes many additional features including a software single-step trace command. A SMARTBUG listing is included and object code is provided on a 2708 free with each P-38 series board purchased. Source listing available separately for $19.50.

NEED A FULL SIZE FLOPPY? Our P-38-FF is a plug-in interface card to the ICOM Frugal Floppy™. It includes all the features of the P-38-I plus one 2708 EPROM containing the ICOM bootstrap software. Just plug the P-38-FF into your SWTPC 6800 and your ICOM into the P-38-FF and you're ready to use the Frugal Floppy and ICOM's 6800 software package. Price $299.

Our P-38 is an 8K EPROM board containing room for 8 2708's. Or, you may use it to hold up to 7 2708's plus your Motorola Mikbug or Minibug II ROM. The P-38 addressing is switch selectable to any 8K location. Price $179.

The P-38-I contains all the features of the P-38 plus an interface to the Oliver Paper Tape Reader and our EPROM Programmer. Price $225.

The PS-1 Power Supply Kit provides plus and minus 16 volts required for the P-38 series boards. Also, it allows a wiring modification to be made to the 8 volt supply that will increase its output by one volt. Price $24.95.

Our M-16-A is a 16K single power supply STATIC RAM memory system. The M-16-A is fully buffered and requires only half the power of a similar size system using low-power 2102's. With the M-16-A, you can expand your system to 48K and still have room left for one of our EPROM Boards. The M-16-A is switch selectable to any 4K starting address and hardware write protect is included. Quick delivery. Price $595.

BANKAMERICARD, VISA AND MASTER CHARGE WELCOME.
Sample Program #3:

This simple inventory program for three products illustrates the use of the ON statement by reading an item code number followed by an item count from the data block. The ON statement decodes the item code to determine what total to add the new count to. When done, the program prints all the current inventory totals.

<table>
<thead>
<tr>
<th>Program</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 REM Inventory</td>
<td>Program name</td>
</tr>
<tr>
<td>20 READ N,11,12,13</td>
<td>Read data count and original totals</td>
</tr>
<tr>
<td>30 READ I,C</td>
<td>Read code and count</td>
</tr>
<tr>
<td>40 IF I &gt; 3, then 130</td>
<td>Check for valid code</td>
</tr>
<tr>
<td>50 IF I &lt; 1, then 130</td>
<td>Decode item code</td>
</tr>
<tr>
<td>60 On I, go to 70,90,110</td>
<td>Add to item #I</td>
</tr>
<tr>
<td>70 Let I+1=C</td>
<td>Continue</td>
</tr>
<tr>
<td>80 Go to 150</td>
<td>Decrement data count</td>
</tr>
<tr>
<td>90 Let I+2=12+C</td>
<td>Exit program</td>
</tr>
<tr>
<td>100 Go to 150</td>
<td>Print error message</td>
</tr>
<tr>
<td>110 Let I+3=13+C</td>
<td>Finish data</td>
</tr>
<tr>
<td>120 Go to 150</td>
<td>Print totals</td>
</tr>
<tr>
<td>130 Print &quot;invalid item code&quot;</td>
<td>Blank line</td>
</tr>
<tr>
<td>140 Stop</td>
<td>Combined total</td>
</tr>
<tr>
<td>150 Let N=N-1</td>
<td>Data</td>
</tr>
<tr>
<td>160 IF N &lt;&gt; 0, then 30</td>
<td>Original totals</td>
</tr>
<tr>
<td>170 Print &quot;item #I &quot;</td>
<td></td>
</tr>
<tr>
<td>180 Print &quot;item #I &quot;</td>
<td></td>
</tr>
<tr>
<td>190 Print &quot;item #I &quot;</td>
<td></td>
</tr>
<tr>
<td>200 Print</td>
<td></td>
</tr>
<tr>
<td>210 Print &quot;total items =&quot;</td>
<td></td>
</tr>
<tr>
<td>220 Data 6,1020,1349,714</td>
<td></td>
</tr>
<tr>
<td>230 Data 1,12,17,3,27,2,11,3,212,1,136</td>
<td></td>
</tr>
<tr>
<td>240 End</td>
<td>Program end</td>
</tr>
</tbody>
</table>

Notice how this program checks for an illegal value before the ON statement. An invalid item code causes a message to be printed and the STOP statement terminates the program.

The words NUMBER OF DAYS without quotes would be printed followed by a question mark. The number typed followed by a carriage return would be assigned to the variable D. The data entered via an INPUT statement is not saved with the program so it should only be used when entering small amounts of data or data that is unknown until the program is run.

**RESTORE Statement**

The RESTORE statement typically permits reading data from the DATA statements of a program more than once. Whenever a RESTORE statement is encountered, BASIC resets the data block pointer to the first entry in the data block. The next READ statement then starts reading the data block all over again.

**SIGN Function**

The sign function:

\[ \text{SGN}(X) \]

returns a value indicating the sign of the argument specified. The value one is assigned for any positive number, zero for zero, and -1 for any negative number.

Example:

\[ \text{SGN}(2.75) = 1 \]
\[ \text{SGN}(0) = 0 \]
\[ \text{SGN}(-0.25) = -1 \]

The sign function can be combined with an ON statement to give a three way branch depending on the sign of a number:

\[ 60 \text{ ON SGN}(X)+2, \]
\[ \text{GO TO 100, 200, 300} \]

This instruction will jump to 100 if \( X \) is negative, 200 if \( X \) equals zero, or to 300 if \( X \) is positive.

**INTEGER Function**

The integer function:

\[ \text{INT}(X) \]

returns the greatest integer of \( X \) that is less than or equal to \( X \).

Example:

\[ \text{INT}(2.98) = 2 \]
\[ \text{INT}(-2.05) = -3 \]
\[ \text{INT}(4) = 4 \]

The integer function may be used to round numbers to the nearest integer by adding 0.5 to the number:

\[ \text{INT}(X+0.5) \]

It may also be used to truncate a number to any specific number of decimal places (n):

\[ \text{INT}(X*10^n+0.5)/(10^n) \]

**Random Numbers**

Most forms of BASIC provide a means of generating a random number between zero and one. This function is generally used to simulate events that happen in a somewhat
Please send me the following:

- CP/M Documentation (Complete set of 6 manuals) for $25.00.
- CP/M Documentation and CP/M System Diskette for $70.00.
- Free CP/M brochure and price list.
- Information concerning high level languages and optional packages.

NOTE: IMS Associates and Digital Systems distribute CP/M with their hardware systems.

Bank AmericaCard No. Exp. date Master Charge No. 

Exp. date Check or M.O. enclosed.

California residents add 6% sales tax. Amount enclosed

Name Address

City State Zip

DIGITAL RESEARCH Post Office Box 579 • Pacific Grove, California 93950

Circle 111 on inquiry card.
MAT READ a,b,c... Read the specified matrices or vectors, dimensions previously specified.
MAT M = ZER Set all components of matrix M equal to zero.
MAT M = CON Set all components of matrix M equal to one.
MAT M = IDN Set up square matrix M as an identity matrix.
MAT PRINT a,b,c... Print the specified matrices or vectors.
MAT INPUT V Call for the input of a vector.
MAT B = A Set matrix B equal to matrix A.
MAT C = A+B Set matrix C equal to the sum of matrices A and B.
MAT C = A-B Set matrix C equal to the result of matrix B subtracted from matrix A.
MAT C = A*B Set matrix C equal to the product of matrix A multiplied by matrix B.
MAT C = TRN(A) Set matrix C equal to the transpose of matrix A.
MAT C = (k)*A Set matrix C equal to matrix A multiplied by the number k which must be in parentheses and may be given by a formula.
MAT C = INV(A) Set matrix C equal to the inverse of matrix A.

Special MAT Functions...

DET Equals determinant of a matrix after inversion.
NUM Equals number of components following a MAT INPUT.

60 LET X=INT(A*RND+B)
This instruction will generate an integer random number between A and B, where A is larger than B. Thus, to simulate a dice roll for a game, with integers between 1 and 6, use:

INT(6*RND+1)
To aid in debugging programs, the typical BASIC RAND function generates the same set of random numbers in the same order each time a program is run. In some implementations inserting a RANDOMIZE statement as the first statement in a program using random numbers, repeated runs of the program will produce different results.

User Defined Functions
In addition to the standard BASIC functions, some BASIC interpreters allow up to 26 additional functions to be defined with the DEF statement. The name of the defined function must be three letters, the first two always being "FN" (eg: FNA, FMB,..., FNZ) and the DEF statement may appear anywhere in the program. Each DEF statement defines a single function and it can contain any combination of other functions and/or variables besides those denoting the arguments of the function. Any variable that is not an argument of a function will use its current value in the program. For example, to repeatedly use the function:

\[ e^{x^2} + 3x + Z \]
define the function:

DEF FNE(X) = EXP(X**2)+3X+Z
and then call for different values of the function by:

FNE(2), FNE(A+B), etc

The current value of Z is used each time the function is called. Also, each defined function may have zero, one, two, or more numeric variables as arguments of the function.

Examples:
10 DEF FNA = 3.1.166*(R**2)
20 DEF FNB(X,Y) = (X+Y) / (X*Y)
30 DEF FNR(A,B,C,D) = FNB(A,B) + FNB(C,D)

Multiple line defined functions are constructed by dropping the equal sign and ending the function definition with a FNEND statement. The function name without arguments is used as a temporary variable to compute the function's value. Multiple line functions usually cannot be nested and there must not be a transfer from inside the function to outside its range, or vice versa. Also, GOSUB and RETURN statements are not typically allowed within a multiple line defined function.

random way such as a dice roll in a game. The general form of the random function reference is:

RND
with no argument needed. If a particular BASIC implementation does require an argument, the number one is typically used:

RND(1)
To generate a random single digit integer, the following instruction might be used:

40 LET X=INT(10*RND)
If it is desired to generate a random integer number between two limits, this instruction may be used:
DataSync
DS-100

Educated Terminal...

- Internal microprocessor control for high reliability.
- True word processing with character and line insert/delete.
- 24 lines by 80 characters on a full nine-inch screen.
- Special individual function keys control the block screen send, screen protect, forms mode, tabs, scroll up and down, print page, line centering mode, and other modes.
- A full cursor control key pad is included.
- All switches are Cherry gold cross point for extra long life.
- A full-function light pen attaches to the terminal for easy control of all displayed characters on screen.
- Two serial RS232 ports transmit up to 9600 baud.
- Tarbell format cassette interface reads and writes up to 4800 baud to and from the screen or your computer.
- All switches, including baud rate, parity stop, local-RS232 CRT controls, and on-off are accessible from the front.
- Weighs under 30 lbs. including strong ABS cabinet.
- Terminal can also be used as a full word processor or text editor.

$695
Kit

$795
Assembled

DataSync 16K RAM

Fast access, low power, dynamic RAM memory board with transparent memory refresh. S100 bus compatible, expandable to 64K Bytes.

$298
Assembled

DataSync
201 W. Mill / Santa Maria, CA 93454 / 805/963-8678
Example:
10 DEF FN M(X,Y)
20 LET FN M = X
30 IF Y = X, THEN 50
40 LET FN M = Y
50 FN END
This example returns the larger of the two argument values whenever the function is called.

**ASCII** (single character or mnemonic of nonprintable character)
Returns a decimal value for the ASCII code of the argument.

**CHR$** (numeric formula)
Opposite function to ASCII, argument is truncated to an integer which becomes the content of a one character string which is returned.

**INSTR** (numeric formula, string formula, string formula)
Searches for the second string within the first string. Search starts at character position specified by the numeric formula truncated to an integer or at the first character if omitted. Returns the position of the first character in the substring if found, 0 if not.

**LEFT$** (string formula, numeric formula)
Returns a substring of the string formula, starting from the left. Substring contains the number of characters specified by the numeric formula truncated to an integer.

**LEN** (string formula)
Returns the number of characters in its argument.

**MID$** (string formula, numeric formula, numeric formula)
Returns a substring of the string formula, starting at the character position specified by the first numeric formula truncated to an integer. The substring contains the number of characters specified by the second numeric formula truncated to an integer, or the substring continues to the end of the string if the second numeric formula is omitted.

**RIGHT$** (string formula, numeric formula)
Returns a substring of the string formula, starting from the right, containing the number of characters specified by the numeric formula truncated to an integer.

**SPACES** (numeric formula)
Returns a string of spaces, the length specified by the numeric formula truncated to an integer.

**STR$** (numeric formula)
Returns a string representation of its argument (the number is converted to a string).

**VAL** (string formula)
The opposite function of **STR$**. Returns the number represented by the string formula (the string is converted to a number).

In addition to these functions, the **CHANGE** statement is used to convert between strings and numeric data and vice versa as follows:

**CHANGE** [string formula] TO [numeric formula]
Changes the string specified to a numeric vector. The zero element of the vector will contain the number of characters in the string and the decimal ASCII code of the letters will be stored in the numeric vector. The statement:

**CHANGE** [numeric vector] TO [string variable]
changes the numeric vector to a string. The zero element of the vector must contain the number of characters in the desired string.

**Table 5**: A “typical” set of string manipulation functions implemented in a BASIC interpreter. This table lists functions found in the author’s experiences with BASIC on Digital Equipment Corporation computers; similar sets of string functions are often implemented by other “extended BASIC” interpreters.

**Vectors and Matrices**
Since operations on lists and tables occur frequently, a special set of matrix instructions and functions are usually available in BASIC. The list of typical MAT instructions and functions plus a brief description of their operation is found in Table 4. Many of the matrix statements require special conditions to exist in order to be legal or may allow operations on vectors as well as matrices. While every vector has a zero component and every matrix has a zero column and row, these are ignored by MAT instructions. Any numeric array referenced in a MAT statement other than a MAT INPUT, will be set up as a matrix unless specifically declared as a vector in a DIM statement.

**Strings**
So far, only operations dealing with numeric information have been discussed; but some forms of BASIC also process alphanumeric information in the form of strings. A string is a sequence of characters including letters, digits, spaces, or some other printing characters. In BASIC it cannot, however, contain a line terminator such as a line feed, carriage return, form feed, or vertical tab. Strings are normally enclosed in quotes (ex: "ANSWER") when used as string constants, but may be omitted in some instances.

Variables may be used for simple strings and string vectors but not for matrices. Any variable or vector name followed by a dollar sign ($) stands for a string:

A$, B2$, or C$[1]
String variables or constants are used just as numeric quantities in most BASIC statements. However, numeric and string data are kept in two separate data blocks and are utilized independently of each other. The RESTORE statement discussed earlier resets both data pointers to the beginning of the data blocks. A typical implementation trick to reset only the numeric data pointer is to use: RESTORE*, and use STORE$ to reset only the string data pointer.

Table 5 shows some of the possible functions for manipulating strings in BASIC in some implementations.

**Other Features**
Many BASIC software systems, especially large time sharing systems, sometimes allow special operations involving data files and special formatting of data. An example is the PEEK and POKE facility which is used in Attair BASIC for bit level manipulation. It is
always best to refer to the BASIC programming manual for the particular implementation being used as I have pointed out many times before. Always check what features are offered and how to use them, since they can vary drastically from system to system. There may also be special editing or operating commands available depending, again, on the system used.

In Closing...

Now that the general statements and functions have been discussed and some of the more advanced features have been introduced, the best way to actually learn the BASIC language is by hands-on experience. Find a system with BASIC, check the available BASIC features, then start with small, simple programs. In no time at all you should be able to program almost anything desired. BASIC is an extremely easy language to learn and use so don't let the term "HIGH LEVEL LANGUAGE" frighten you away. Look for future articles on programming techniques, games, and applications as BASIC becomes available for more and more microcomputer systems. BASIC programs should be very easy to reproduce and convert for various specific systems, including microcomputers.

What's New?

A New Microcomputer Series from CGRS

CGRS Microtech Inc, POB 368, Southampton PA 18966, has announced a new line of microcomputers called the 6000 series which are all Altair bus compatible and based on the Mos Technology 6502 microprocessor. The Microputer I is a 3 board tutorial system featuring 256 bytes of programmable memory for $229.95. The Microputer II includes 2 K bytes of programmable memory with room for up to 4 K bytes of erasable read only memory.

Interface any Selectric I/O Writer to S-100 computers with a TYPEAWAY Selectric I/O Writer Interface. TYPEAWAY is compatible with any model 731 or 735 I/O Writer, including versions with correspondence or BCD coding, 24 or 48 volt solonoids, and common anode or common cathode diodes. TYPEAWAY is a complete package of everything needed to interface any model 731 or 735 I/O writer to an S-100 computer, including:

* S-100 compatible PC board with solonoid drivers, input ports, and PROM memory.
* Complete cables and connectors
* Power supply for 24 or 48 volt systems.

Optional word processing software is also available.

Add professional, economical hard copy to your S-100 system with an IBM Selectric I/O Writer and a TYPEAWAY interface.

Prices

Kit ........................ $225
Assembled .................. 325
Documentation .............. 25

Call toll-free anytime to place credit card orders:
800/648 5311

MICROMATION INCORPORATED
524 UNION STREET
SAN FRANCISCO, CA. 94133
415/398-0289

Circle 35 on inquiry card.
Douglas Electronics has announced the availability of a complete line of breadboards designed to be compatible with Digital Equipment Corporation's PDP-11 and 8/A minicomputers. The glass epoxy boards, which feature electroplated solder circuits and gold connector tabs, come in a variety of configurations and options which include extender boards with sockets, wire wrap boards and breadboards for custom logic circuitry. Contact Douglas Electronics Inc., 718 Marina Blvd., San Leandro CA 94577.

Attention Individuals, Clubs and Computer Stores: An 8080 Octal Code Card Is Now Available

Tychon Inc has announced this new 8080 Octal Code Card. The code card is a slide rule like aid for programming and debugging 8080 software in octal notation. It contains all the mnemonics and their corresponding octal codes. The instructions are all color coded to indicate which flags are affected during execution. The pocket sized card measures 6.5 by 3 inches (16.5 by 8 cm) and it provides the instructions in a neat, logical format for quick reference.

The back side of the card is printed with an ASCII code chart for all 128 characters plus the 8080 status word and register pair codes.

Delivery of the 8080 Octal Code Card is immediate and the price is $2.95 postpaid. Quantity discounts start at ten units, and custom imprinting is also available. According to Tychon, a hexadecimal version of the card will be available in the near future.

For further information write Tychon Inc, POB 242, Blacksburg VA 24060.

"We may have the surplus electronics you need!"

Terminals/Keyboard
With Schematics $250.00

- CRT Terminals
- Peripherals
- Assemblies
- Components

- Tapes Drives $80.00
- Keyboards $40.00 to $80.00
(ASCII Encoded)

Equipment Cabinets $45.00 to $90.00
(19" With Fan)

Send for a free catalog or call
Bill Blaney, toll free 800 258-1036
in NH 603 386-3705

Come to our showroom

WORLDWIDE ELECTRONICS, INC.
10 Flagstone Drive, Hudson, New Hampshire 03051

Circle 36 on inquiry card.

A New Way to Do Slit-N-Wrap

Vector Electronic Company, 12460 Gladstone Av, Sylmar CA 91342, has just sent along this photo of a neat new addition to their line of Slit-N-Wrap tools, the P160-FT1 pistol griped 110 VAC powered wrapping tool. This $80 tool will accept the P180 Slit-N-Wrap tool bit, or a standard wire wrapping attachment in the form of the P160-2A wrapping bit. While Slit-N-Wrap can be done by hand, it becomes much more convenient when it is powered as with this tool. (Because the Slit-N-Wrap technique requires no stripping of wire, it tends to be somewhat faster than standard wire wrap as a matter of course,
since the tool is always held in one's hand and is never put down for the stripping operation as in wire wrap with red wire.

Circle 575 on inquiry card.

**IMSAI Brochure and Catalogue**

A beautiful four-color brochure outlining the IMSAI product family came our way recently. In the same package was the current price list of this family of computers and computer products, which may be inspected at many local computer stores. A full catalog is also available according to the brochure, at a price of $1. Write IMSAI Manufacturing Corporation, 14860 Wickes Blvd, San Leandro CA 94577.

Circle 576 on inquiry card.

Altair Bus Floppy

Synetic Design Company, POB 2627, Pomona CA 91765, has sent along this picture of their new "ready to use" disk system for Altair compatible 8080 based computers. The product is called the FDS-2, and it employs ICOM standard size floppy disk drives. The total package includes the ICOM executive system, text editor and assembler for the 8080, plus the custom "executive handler" provided by SDC to allow user-specific attributes of the executive. From one to four drives may be attached to the controller, with two drives shown in this photograph. Maximum on line capacity with four drives is just over 1 Mb. The recording format is IBM 3540 and 3740 compatible to allow transfer of data to and from other computer systems on floppy disk cartridges. Files may contain source data, program object data or user generated data. The drive may be seen at Computer Mart of New Jersey and other stores nationwide. Delivery is quoted as three weeks ARO for small orders; price was not quoted in the press release materials.

Circle 577 on inquiry card.

---

**ANSWERING**

**Z80 SEMINARS**

**AUGUST 22-26**

(week preceding the Atlantic City convention, PC'72)

**CHOOSE FROM ONE OF THE**

**FIVE DAY and/or 2 EVENING SEMINARS**

**BRING THE WHOLE FAMILY**

(Recreation Facilities Available)

**REASONABLE**

Individual/Family Rates for a limited number at the college on a first come, first served basis

**TRANSPORTATION**

available to the ATLANTIC CITY CONVENTION

**Software courses (DAY)**

- Programming in Basic for the 8080 owner.
  An introduction to the Basic language - beginning in writing application programs. Emphasis on TDF's RR and TDF Basic for Z80. Includes intro to TDF FORTRAN. $199
- Assembly Language Programming for the Z80 8080
  An introduction to assembly language programming. Covers Z80, 8080, instructions and then branches to the extra instructions available for the 780. Includes intro to TDF's Z-80 Monitor and Macro-Assembler and also covers applications in interfacing and control. $239
- Advanced Assembly Language Programming for the Z-80
  For someone who knows the Z80. Covers z80 monitor instructions and how to get the most out of them. Emphasis on one of TDF's Macro-Assembler Application to digital logic replacement and process control. $299
- FORTRAN IV
  An introduction to the FORTRAN language. Beginning in writing application programs. Emphasis on TDF's ANSI-standard FORTRAN IV for the Z80. $199
- Word Processing with a Text Editor and Formatter
  Introduction to the use of the Text Editor and Text Output Processor for solving data processing and printing problems. Emphasis on TDF's Text Editor and Text Output Processor. Familiarized with system requirements. $299

**Hardware courses (DAY)**

- **Introduction to Hobby Computing**
  A survey course dealing in an exploration of the hobby computing field and defining the basics needed to be understood in order to get started. $299
- **Digital Logic Circuits**
  Introduction to digital logic circuits covering the 7400 TTL and the CMOS series. Codes, registers, counters, memory, combinational logic, etc. $299

**Evening Workshops**

- **How to Set Up a Computer Store**
  Course benefits from owner-operated computer stores and microcomputer manufacturers. How to become a dealer. How to get a franchise. How to operate a business. How to set up a service facility. . . . and more. $99
- **Kit Building Lab**
  Instructions and guidance on kit building. Bring your kit SPECIAL: Anyone taking delivery on a Z80 product during the course must be ordered in advance and will receive this workshop FREE. Write the center. $49

**SPONSORED jointly by**

TECHNICAL DESIGN LABS, INC. (TDL)
Princeton, N.J.
and
TRENTON STATE COLLEGE
Trenton, N.J.

For registration forms and brochure write:
Z80 SEMINARS Office of Continuing Education
TRENTON STATE COLLEGE  TRENTON, NEW JERSEY 08625
Did you know that your computer can answer the questions of how far and which way for you in microseconds for macrodistances? At least my SWTPC 6800 can with its 8 K BASIC high level compiler language.

The usefulness of this program will be apparent to anyone who sails a boat or flies an airplane. By inputting longitude (degrees and minutes) and latitude of the starting point first, and the destination second, the program will print out the distance in statute miles, or nautical miles with a minor change, and the true bearing. Look up the variation of magnetic north in your area and add that correction for a magnetic compass heading; or better yet, add it into the program.

The program of listing 1 is offered as a model to start from and can be modified or elaborated on according to the wishes of the user. It is based on what is known to cartographers as a Lambert Conformal Conic Projection map. This type of map uses a fixed reference point in its projection, so the method is less accurate the farther north or south you go, into Canada or Alaska for example, from the reference point in the US. However, it will be quite accurate within a restricted area such as that covered by an FAA aeronautical regional or sectional chart.

The requirement for this program grew from my Civil Air Patrol activities when we search for downed aircraft (a true hardware crash) and send rescue teams out to find them. Usually we have an airport to start from and grids to search in by air. The downed aircraft is spotted visually or by reception of a signal from its emergency locator transmitter (ELT). If found, its coordinates are radioed back to the mission coordinator. He then dispatches the ground teams to the rescue and here is where the questions "how far?" and "which way?" are answered by the computer.

The routine uses the trigonometric solution of a right triangle to determine the distance (hypotenuse) and direction (arcsine

---

Figure 1: A series expansion of the ARCSIN function, as found (above) and as evaluated partially for use in the expression in listing 1 at lines 158 and 160 of the program.
of X). The sides of the triangle X5 and Y5 are the differences between the two sets of coordinates of longitude and latitude.

To explain the program a little, notice that the ARC SIN X is a special subroutine which had to be created since ARC SIN, ARC COS and ARC TAN functions are not included in SWTPC 8 K BASIC. This ARC SIN X function is performed by use of the transcendental formula shown in figure 1a which is reduced to the equation of figure 1b. The formula, translated into BASIC, is shown on lines 158 and 160 of listing 1.

Since the ARC SIN X formula above is less accurate from 45° to 90°, we switch and use 90° - COS of the complimentary angle of the right triangle and get the same resolution as when the angle is 0° to 45°. This is written in BASIC in lines 155, 163 and 168 to 175. Also, notice that in lines 58 and 59 a very small number (1.E9) was added to the two sides X5 and Y5 of the right triangle X5, Y5, D. This extremely small error eliminates the problem of dividing into zero at the 90°, 180°, 270° and 360° points of the compass when one or the other of these sides of the triangle are zero. SWTPC 8 K BASIC doesn’t like zeros in your math formulas.

In line 60 the hypotenuse of the triangle, D, is multiplied by 1.1516 to get statute miles. This conversion factor assumes that your calculations have been in British nautical miles which have 6080 feet to the mile. If your calculations have been in American nautical miles, with 6076.115 feet to the mile, the conversion factor would be 1.1508.

In lines 50 and 54 the numbers 44.89 and 74.82 were derived from the ratio of nautical British miles per degree of longitude versus latitude in the northern hemisphere at about 43° north latitude. In other parts of the globe, farther north or south, it would be advisable to recalculate to maintain accuracy. (The distance of one degree of longitude gets smaller and smaller as you go north or south from the equator.)

To correct for the magnetic variation in your area add the degrees of correction at lines 162 and 174. For example, if the aeronautical chart shows 3° west variation, then add +3 to R5's value.

One last comment: the latitude and longitude are entered as a decimal number. The digits to the left of the decimal are degrees and to the right, minutes; seconds are not used. The program sorts it all out and understands what you are inputting by using the INTEGER functions in lines 50 to 56.

Try it for aircraft flight planning or a sailing trip in large bodies of water, then check the accuracy by actual measurement on an aeronautical chart.

Listing 1: A BASIC program used to calculate range and bearing information from navigational coordinates. (This program was typeset from a hand typed listing of the original, which had been developed and debugged from a computer sans hard copy.) In the sample printout, user input is preceded by a question mark query.
NOW!!
Z-80 Power for the S-100 bus without getting rid of your CPU card.

$159.95

assembled

DUTRONICS® a leader in low cost, low power ram boards has just announced it's Z 80 - 80 piggy back card. This plug – in board enables you to use your existing IMSAI, ALTAIR CPU card and upgrade your system to a Z – 80.

The card design is such that all you do is pull out your 8080 and 8212 chips, plug in the Board to the 8080 socket itself and the ribbon cable to the 8212.

A system monitor, on paper tape, is included with the board as well as a 280 Manual and Theory of Operation Manual.

Dutronics will also supply all additional software at no cost, when it becomes available. The price is $159.95 (assembled) only. OFF THE SHELF.

for more information call or write to:

R.H.S. MARKETING
2233 El Camino Real
Palo Alto, California 94306
(415) 321-6639

DEALER INQUIRES INVITED

B of A & MASTERCHARGE ACCEPTED
16384 BYTES for $485.00
assembled (with sockets) : tested - burned-in - guaranteed

A new high in S100 bus memory cost effectiveness. Fully assembled (with sockets), tested, burned-in and guaranteed. 4Kx1 dynamic memory chips (the same ones used by the ton in IBM compatible memory systems) combined with self contained control logic, yield a memory system with:

- Low power consumption, total board 5 watts.
- Transparent refresh, which means the memory looks static to the outside world.
- No waiting. In fact, xrdy is not even connected to the memory.
- Full DMA capability
- Reliable, low level clock and control signals.

Dynabyte

Dynabyte® brings to the S-100 Bus a state of the art, industrial quality memory system. 16K on a single board for $485.00, Guaranteed for 1 year.

for more information call or write to:

R.H.S. MARKETING
2233 El Camino Real
Palo Alto, California 94306
(415) 321-6639

DEALER INQUIRES INVITED

B of A & MASTERCHARGE ACCEPTED
Add Cursor Control to Your TVT II

If you own a TVT II and would like a simple circuit to provide for operation of the cursor and erase functions, then consider the circuit shown in figure 1. This simple circuit will accept ASCII control characters and decode them in a manner suitable for controlling all of the cursor control and erase functions, and it even has a few outputs that can be used to provide user defined functions, such as enabling external devices or turning on bells and whistles.

In the version presented here several functions are provided, including home, erase to end of line, erase to end of frame, cursor up, cursor down, cursor left, cursor right, and bell. In addition there are five other unused codes that may be assigned any way you wish. Control J and M are unused, since the TVT II already decodes these as carriage return and line feed.

How it Works

IC1 decodes the incoming ASCII character, and selects one of its 16 outputs to go to logical zero whenever bits 5, 6 and 7 are all logical zero. The TVT II data lines are.

Figure 1: Diagram for the TVT II cursor controller. IC1 is a 4 to 16 line decoder that will select which control command is wanted. Transistor Q1 is in the circuit to make it compatible to the Southwest Technical Products' screen read and UART boards. Diodes D1, D2 and D3 are in the circuit to protect IC1 when it is connected in parallel to the output of a gate on the main TVT II board. IC2 is wired as a one shot to activate the Sonalert when a control G is pressed.

Brother Thomas McGahee
The Salesian Center
80 S Sixth St
Columbus OH 43215
Sct ee n by the other boards. The circuit will be compatible with SWTPC's screen read and UART boards. It provides logic to advance the cursor when told to do so by the other boards. IC2 is a one shot set up to handle the bell function. A control G will cause a low going pulse at its A1 input. It will then stretch the pulse for a second or so to drive the Sonalert, or whatever you wish to use as a bell function. There are several other outputs from IC1 that can be used to initiate other bells and whistles in a similar manner. The J numbers in figure 1 refer to the connector plugs on the main TVT II board.

Table 1 is a listing of the 16 outputs and their function assignments. An "X" in the table indicates a free function waiting to be implemented. A mnemonic explanation of the functions at the right of table 1 helps when using the keyboard. If for any reason you wish to change which control keys perform which functions, just wire to a different output pin on IC1.

If you are using a diode encoded keyboard, you can add cursor control and erase keys quite easily, if you already have a control key. All you do is connect each cursor or erase key as shown in figure 2. The diodes serve to isolate the keys. Whenever the key is depressed it will simultaneously activate both the control and alphabetical keys. You can add as many of these type keys as you need. While you are at it, you may wish to add a calculator type array of switches for numerical entry. You simply wire them in parallel with existing numerical keys; no diodes needed. This will considerably speed up the entry of numerical data.
WANTED: Human with at least 1 year C++ programming experience. Must have Object Pascal experience and be familiar with object oriented design. Send resume to: [insert contact information].

FOR SALE: Antenna: Wilson 360106, 30" collinear. Excellent condition. $100. Contact Mike Ward, [insert contact information].

WANTED: Tapes: BASCOM Plus, Professionals 4, Professionals 8, Professionals 16, Professionals 32. Please contact David Jones, [insert contact information].

FOR SALE: IBM 386SX, 16 MB RAM, 10 GB HDD, CD-ROM, Windows 95. $600. Contact John Smith, [insert contact information].

WANTED: Tapes: BASCOM Plus, Professionals 4, Professionals 8, Professionals 16, Professionals 32. Please contact David Jones, [insert contact information].

FOR SALE: PC clone: ORD-8600, Intel Pentium 233 MHz, 64 MB RAM, 2 GB HDD, Windows NT. $500. Contact John Smith, [insert contact information].

WANTED: Tapes: BASCOM Plus, Professionals 4, Professionals 8, Professionals 16, Professionals 32. Please contact David Jones, [insert contact information].

FOR SALE: Antenna: Wilson 360106, 30" collinear. Excellent condition. $100. Contact Mike Ward, [insert contact information].
The Intelligent Video Interface

The MERLIN Kit with manual is $269. MERLIN assembled & tested is $349. Order yours today. MC and BAC accepted.

MEI 2K x 8 Mask ROM
Extended Monitor & Editor functions plus graphics and cassette subroutines ($34.95).

Serial Expansion Kit provides 3 parallel input ports and 2 parallel output ports via KYBD & Serial I/O Connector (MSEK — $45).

8-Bit Programmable Mode Register
Program MERLIN for ASCII, graphics and special display formats.

Software Tarbell Compatible Cassette Interface
Interfaces via keyboard cable. Software is available in the MEI ROM (MCAS — $29).

MBI 2K x 8 Mask ROM
System Monitor and Editor plus scrolling and I/O drivers.

MBI ROM/RAM — $39.95
(Eliminates need for expensive Monitor Boards.)

If you are still not convinced that MERLIN is the best buy in Video Interfaces/ Monitors see one at your local computer store or write for more info.

MiniTerm Associates, Inc.
Box 268, Bedford, MA 01730 (617) 648-1200

Circle 38 on inquiry card.
Giving KIM Some Fancy Jewels

When my KIM-1 arrived I was excited, but a little dubious; after all, I had read all those microprocessor books and still didn’t know beans about programming. Well, MOS Technology’s excellent manuals took care of that and I’m on my way to knowing what I’m doing. When I say excellent, I mean if I learned it, anyone can!

But KIM has a fundamental problem! I like KIM, but I just can’t see myself sitting there with a naked board gathering dust and me dropping ashes all over it while I sweat out a program. Also with MOS Technology’s welcomed announcement of the KIM-2 and KIM-3 programmable memory boards and the coming KIM-4 mother board, it’s really time to start thinking about putting KIM in a nice enclosure. But in order to bury KIM in an enclosure, I had to somehow bring the displays and keyboard lines out.

MOS Technology was nice enough to bring the keyboard out to the application connector, so one problem was solved before I started. But what about the display? I didn’t want to just parallel the existing readouts on the board because that would mean that much extra current, and this might not make the drivers too happy. I came up with the following solution which is fairly simple, yet a bit delicate to start with. Delicacy and discretion with KIM are requirements of the task of unsoldering the original readouts.
turning these displays into remarkable jewels.

The MAN 727 7 segment readouts (U18 thru U23 of the MOS Technology documentation) are unsoldered with an iron of less than 35 W and a "solder sucker" or similar vacuum solder removal tool. This is the only way I've ever found to successfully unglue items on double sided boards. This will allow you to replace the displays with one 16 pin socket as shown in figure 1. The 16 pin socket will go in the 14 pin U18 slot (there is room for the extra pin below). Use of the unused pins of the old 14 pin display pattern, plus one extra socket pin of the 16 pin socket, allows all the necessary control signals to be routed to the external display position.

In order to mount the 16 pin socket, take a small piece of Vectorboard with 0.1 inch spacing and lay it over the spot for U18 and line it up with the existing holes; this will be your marking guide. Now using a scribe or small drill, mark the spots for pins 4, 5, 8, and 14 (which were originally unused at U18). Do not mark the spot for pin 9 since it would go through the circuit on the other side. Hold the Vectorboard firmly when marking and check for alignment with a 16 pin socket after marking it. Now using a #58 drill at the highest speed you can get, carefully drill the extra four holes. Take your 16 pin dip socket and dip off pin 9 on the back side; now it may be soldered in place with pin 1 to the upper left. (If you solder five 14 pin sockets in the other display patterns, you can plug the six displays you removed into the sockets and check to make sure KIM is still talking to you).

Now we have four extra pins at U18 and five anode leads to bring out on U18's modified socket (which is called the adaptor socket) in figure 1. Actually we have five extra pins since pin 6 on all the displays is the decimal point cathode and is not used. All we have to do now is bring the five other common anode leads from U19 thru U23 to our modified U18 socket as illustrated in figure 1. This may be done with very small (#26) wire. I picked up pin 3 of all the other display sockets which seemed to give the easiest routing, straight down the middle. The wiring diagram of figure 1 shows the connections. Now you have the original on board display with the option of plugging into the adaptor socket with a 16 pin DIP plug for a front panel display when you put KIM in a box.

The front panel display may be wired as shown in figure 2. I used a modified clock display board and would have been just as well off wiring it by hand on Vectorboard. Also I used a 16 pin DIP male to male plug with another 16 pin socket mounted on the back of the display board, so the whole works can be unplugged if I want to go back to the bare KIM for testing or whatever. Just for a little added elegance and pizzazz, I replaced the two data display readouts with MAN 7G green LED displays for some classy emerald jewel effects.

This modification of KIM's display jewels is just the first step in getting KIM into a more workable medium for expansion and dressing her up a bit. I've also been working on the complete housing, but that's another story.

---

**Figure 2: A remote display board.** After removing the KIM-1 displays, a remote display board is constructed for mounting in a more convenient packaging arrangement. The wiring of the cable which plugs into the 16 pin socket installed at the U18 position of the KIM-1 board is listed in the diagram.

**SUMMARY OF INTERFACE CABLE**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Segment A Cathode</td>
</tr>
<tr>
<td>2</td>
<td>Segment F Cathode</td>
</tr>
<tr>
<td>3</td>
<td>U18 Common Anode</td>
</tr>
<tr>
<td>4</td>
<td>U20 Common Anode</td>
</tr>
<tr>
<td>5</td>
<td>U21 Common Anode</td>
</tr>
<tr>
<td>6</td>
<td>U22 Common Anode</td>
</tr>
<tr>
<td>7</td>
<td>Segment E Cathode</td>
</tr>
<tr>
<td>8</td>
<td>U23 Common Anode</td>
</tr>
<tr>
<td>9</td>
<td>No Connection (Do not drill on KIM)</td>
</tr>
<tr>
<td>10</td>
<td>Segment D Cathode</td>
</tr>
<tr>
<td>11</td>
<td>U18 Common Anode</td>
</tr>
<tr>
<td>12</td>
<td>Segment C Cathode</td>
</tr>
<tr>
<td>13</td>
<td>Segment G Cathode</td>
</tr>
<tr>
<td>14</td>
<td>U19 Common Anode</td>
</tr>
<tr>
<td>15</td>
<td>Segment B Cathode</td>
</tr>
<tr>
<td>16</td>
<td>U18 Common Anode</td>
</tr>
</tbody>
</table>
PAPERBYTES™
Forum

Yet Another Format?
Ian Robinson, 308 W 5th St, Nellisville
WI 54456, sent in an extended letter on an
alternative to the bar code format we pro-
posed in November and December, and have
begun using. He proposes a format which
looks like this:

```
  010101100010100  Binary data
```

However, there is one disadvantage to
this format: It requires two read sensors
since the data clock bars and the data bars
must both be read simultaneously. He argues
that this method can be hand drawn with
quite a variation in density, which may be
true, but the whole idea of the bar codes was
to automate the printing process for mass
produced software distributed in maga-
zines.

BAR CODES AND OTHER TOPICS

I enjoyed the April 1977 issue of
BYTE, which I thought was somewhat
improved in areas of my interest. In
particular, I enjoyed your article, "A
Software Controlled 1200 bps Audio
Tape Interface." For a while, I thought
that our interests might be diverging,
but now your articles are tending more
toward the hardware direction, with
one particular area of common interest
that intrigues me greatly.

The idea of published software in
bar code form could well be the greatest
communications method yet, as far
as hobby level exchanges go. Here
are some additional ideas for your
consideration.

An almost perfect version of the
optical reader described by Frederick
Merkowitz in his article on page 77 of
December 1977 BYTE is available from
Wilcox Enterprises, whose add also
appears on page 85 of this issue. This
item sells for a mere $15, and contains
a B & L Microscope equipped with an
EPI illuminator (through the lens) and
a 10x objective lens. The microscope
assembly is equipped with a photocell
and silt assembly, and is mounted on
a precision set of slides. The assembly
is scanned or driven by a motor (every
unit I have ordered has been damaged,
but the motor can be replaced easily,
by a semiskilled hacker) and gear
assembly.

An infrared emitter and sensor
combination, mounted in a TO-18
package, is available from Ultra Sensors
Inc at 2400 West 102nd St, Suite 313,
Minneapolis MN 55431. The price is
$4.50 in unit quantities (ask for the
Cyclops Evaluation Kit). This device is
easily made into a light pen style reader,
or any other configuration one might
desire. There seem to be no significant
problems with ambient light, or angle
of approach with this device, if the
self-contained lens is held in contact
with the paper while reading. This device
works best if it is powered by approxi-
mately 10 kHz pulses of 5 to 10 V mag-
nitude, through a 27 Ohm current
limiting resistor (or it can be driven
by a simple IJT transistor oscillator
operating at 10 kHz). Connect the
photocell to a LM 308 op amp through
a 100 pF capacitor followed by a 3.3 k
Ohm resistor to ground (ie: a differen-
tiator). If you rectify the output of
the op amp, you have the scanned
bar code signal!

A bar code printer (coded for the
standard bar codes, aka super market)
is available from:

Medical Record Systems
Division of Ames Color-File
Corporation
12 Park St
Somerville MA 02143

They ask a very high $6,500 (in hobby
terms) price for this critter, and the
electronics for bar code formulation and
interface are not of particular interest
in our case. Possibly the printer mech-
nanism alone can be purchased. With
such a machine connected to a key-
board and a universal (ie: programmable)
cassette tape interface, BYTE maga-
zine could reproduce almost anyone's
software recordings into one of the
bar code formats that you have been
discussing. This would make the dream
of software exchange a reality. (If BYTE
would like a formal quotation for
construction of such a system,
just ask, This is my business. A first
guess at the cost, with bar code printer,
would be approximately $13,000.)

I don't enjoy writing software,
and I find the manual entry of software
listings a very boring task. Published bar
codes could change this.

I have recently made several video
monitors by converting a Magnavox
12 inch, transformerless set for a total
cost of $87. Each go to a Magnavox
dealer, and purchase an MG 5056WA2
less the video, audio and tuner mod-
ules. These are plug-in and can be
removed in minutes right in the show
room (take a 1/4 inch nut driver with
you). The dealer can use these modules
to service quite a few Magnavox sets,
and the total rebate has been $53.55
to me. By sawing the fingers off an old
PC card, you can fabricate a connector
to fit into the video modules socket,
and make connection to pin 4 (video)
and pin 8 (ground). Delta Electronics's
L9710 isolation transformer at 56 then
mounts in the same screw holes that
mounted the tuners! If you think there
would be sufficient interest in this
project, I could put together a project
article on same.

The front cover picture of April 1977
BYTE is fantastic. How about poster
size reproduction of this? I need several
(to justify the present conditions on
several of my own benches)!

Sumner S Loomis
Route 1, Box 131 A
Prairie Point MS 39333

Thanks for your excellent comments
and suggestions, but I thought I used
software to generate and decode the
information at the interface...CHI

---

HIGH-SPEED CASSETTE INTERFACES (The Stringy Disk)

CRI-B 60 to 12,000 baud status
controls, 1K monitor with
cassette routines provided
on tape.

SUMMER SALE! Take 10% off: Good thru Aug. 31,

DẠI JEN Electronics
7214 Springleaf Ct.
Citrus Heights, Ca. 95610
(916) 723-1050

This one does it all!
Announcing: Universal I/O, moni-
tor in EPROM, RAM, RS-232, cur-
rent loop, 3 parallel ports, cas-

High-Speed, Kansas-City, Altair, IMSAI, Poly, Tarbell compatible

Kit $120, Assembled $165
Manual $3.50

DAJEN Electronics
7214 Springleaf Ct.
Citrus Heights, Ca. 95610
(916) 723-1050

UCRI Relay control of two re-
corders, latched input port
for keyboard, tape reader,
etc. File-search capabilities.

Kit $135, Assembled $175
Manual $4.50

Circle 39 on inquiry card.

128
ASCII Keyboard Kit from Radio Shack

Radio Shack has announced the new Archer ASCII Keyboard Encoder kit for $37.80. The price includes all electronics and parts needed to assemble the keyboard as shown. An external 5 VDC 500 mA power supply (not supplied) is necessary to power the unit; this may be purchased locally. The printed circuit board alone, including assembly and parts manual, is available for $14.95. See your local Radio Shack dealer for details on their latest catalog sheets.

Circle 634 on inquiry card.

A New Multiplexed Analog I/O Card for the Z-80

Signal Laboratories Inc. 202 N State College Blvd, Orange CA 92668, is making available a new analog I/O card. With this board, intended for use with the Zilog Z-80 development systems, the engineer or scientist can monitor up to 16 analog channels in real time.

NEW... 1977
MICROCOMPUTER CATALOG

Hundreds of microcomputer products and accessories from many manufacturers.

Our new MICRO 77 Catalog provides you with a single, easy to use source for comparing and selecting microcomputers—from simple low cost trainers, to advanced systems designed for business, scientific, and engineering applications.

Whether you choose a kit or fully assembled unit, this easy to use catalog can help take the mystery out of selecting the computer system that is best for your application.

Included in this catalog is a convenient glossary of terms for reference, and sections covering:

- MICROCOMPUTER SYSTEMS
- PERIPHERALS
- DEVELOPMENT AND TEST EQUIPMENT
- SOFTWARE
- BOOKS AND PROGRAMMED COURSES

Send $1.00 (refundable with first order) for your copy today!

microcom systems
865 3rd Street South, St. Petersburg, Fl 33701

PCC'S Reference Book of Personal and Home Computing

Ever try to find the addresses of some manufacturers of, say, tape cassette or floppy disk interfaces for micros? Frustrating, isn't it? Well PCC has done something about it. This book lists hundreds of companies and stores selling hardware, software, and services. Survey articles on software, hardware, kits, applications and the future for the experienced and the not-so-experienced user of micros. Also included in this edition are bibliographies for further reference, book reviews, and an index of the articles from the major hobbyist magazines. All for the low price of $4.95, plus 50 cents postage.

Available through:

BITS, Inc.
70 Main Street
Peterborough NH 03458

Master Charge and
BankAmericard Welcome.

In unusual cases, processing may exceed 30 days.

Circle 51 on inquiry card.
The channels are sequentially sampled and sent through a 12-bit analog to digital converter. Two multiplexed digital to analog outputs are also available on the board, as well as four discrete (1 bit) outputs and a DC to DC converter, enabling the board to be powered with only +5 VDC. One analog to digital conversion takes a maximum of 20 us. A programmable gain amplifier is also included offering gains of 1, 2, 4 or 8. An option is available which allows the board to monitor up to 32 analog input channels. Price for quantities under ten ranges from $395 to $910, depending on options. Available from stock.

Circle 645 on inquiry card.

A Sophisticated New Small System

On Line Shirtware

Fed up with those tiresome Paris originals? Eschew the tacky route at your next computer club meeting with Martha Herman’s new computer T-shirt designs. Try REAL TIME OPERATOR on a dark green background, or maybe GARBAGE IN—GARBAGE OUT on orange. Other titles include COMPUTER WIT and RANDOM ACCESS; there are ten in all.

Shirts are available from Martha Herman, 114 W 17th St, New York NY 10011, for $5 plus $.60 postage per shirt, State ch st size.

Circle 646 on inquiry card.

No bells or whistles...just performance, a warranty and a low price.

For $107.00 take your choice: the 4K RAM board or the alpha video board. For $137.00 the video graphic board can’t be matched. (6K RAM and much more on the way.) These are not kits, but completely assembled, burned-in, tested boards with a 1 year warranty. No soldering, no messing, no chance of mis-connections...just plug ‘em in and you’re ready to go.

The 4K RAM has the same features and speed as what you’re used to (500 nsec, no wait states) but with a couple of extras you might not expect. Like a mechanical write protect switch that gives you positive memory protection. And Visaddress®, an easily accessible switch on top of the board for easy to read address selection.

The alpha video board offers Visaddress address selection and displays 128 ASCII characters; both upper and lower case. A standard 16 line by 32 character format with character selectable video reverse and socketed ROM are standard features. (Character generator ROM available with standard graphics characters. Optional Greek or ASCII control characters available.)

And our video graphic board gives you an exceptionally easy approach to computer animated dynamic displays.

Our matrix approach allows direct addressing of each dot in an 128 by 96 array, simplifying dynamic movement. And unlike DMA displays, our on-board RAM allows your CPU card to do other work during display time and with no dynamic RAM problems. And like our other boards, it has Visaddress address selection.

Quality, assembled boards at less than kit prices. But what else would you expect from a company whose prime product is electronic test instrumentation and microprocessing components?

Order direct, by check, BankAmericard or Master Charge (Add $.50 shipping, credit customers give us all the card numbers, please and Ohio residents add 4½% sales tax) or contact us for more information. Kent-Moore Instrument Company, P.O. Box 507, Industrial Ave., Pioneer, Ohio 43554, (419) 737-2352.

Circle 40 on inquiry card.
DO IT
WITH 16 BITS

DO IT BETTER

With the language that is best suited for your application: Business programming in COBOL, Scientific and Engineering in FORTRAN IV, Educational in BASIC, and Systems Implementation in our MACRO ASSEMBLER.

With hardware that will grow with your application, and protect your software investment.

DO IT FASTER

With extensive support and utility programs to speed development of your applications software. These include a Diskette Operating System, Relocatable Linking Loader, Load Module Library Editor, Symbolic Debugger, and Text Editors.

With a 16 bit processor that includes hardware multiply and divide, real-time clock, and peripherals like single or dual diskette subsystems with integral DMA controller.

DO IT WITH A MicroNOVA®

BPI

If you are serious about computing, call or write today. BPI Inc., 2205 East Broadway, Suite 6, Tucson, Arizona 85719 (602) 326-6975

© NOVA is a registered trademark of Data General Corporation, Southington, Massachusetts

Circle 41 on inquiry card.
SOFTWARE

All programs include: Complete assembler source listing, sample output, hex dump, sorted symbol table, plus complete instructions and thorough documentation.

Text Editing System for 6800. The best! $23.50
NEW Masemonic Assembler System for 6800.
$26
NEW Stack Oriented Arithmetic Processor
(6800) $23.50
NEW 8080 Klingon Capture Space game. $6.50
Special Game Packages Each containing 6 programs:
8080 $19.95. PD65-1 $19.95.
6800 PD68-1 $16.50
Complete 6800 Software Pack 16 programs.
PD68-3 $35.50
Battleship for 6800, like the board game. SL68-22 $8.00
Space Voyage* TSC's 6800 Star Trek game.
SL68-5 $12.00
Micro BASIC Plus The best 6800 "Tiny BASIC"
SL68-19 $15.95
Diagnostics for 6800. Very Important! SL68-23 $10.00
Stock Market for 6800. Lots of fun! SL68-7 $4.25
Complete Catalog of all of our programs. $2.50
Program-of-the-Month-Club* Join the hundreds of hobbyists already enjoying this service. No obligation and no time valued cards to return. Discounts offered. One year for $2.00
To Order: Include 3% postage, $1.00 handling on orders under $10.00. Indiana residents add 4% sales tax. Check your dealer!

TSC TECHNICAL SYSTEMS CONSULTANTS
BOX 2574 W. LAFAYETTE, INDIANA 47906

16K STATIC RAM

$459 KIT

* USES 4K STATIC RAMS - NO REFRESH
* VERY LOW POWER - LESS THAN 1 AMP
* 250 FAST - 250ms BOARD ACCESS TIME
* PROVISION FOR BATTERY BACKUP
* LOW PROFILE SOCKETS FOR ALL CHIPS
* EACH 4K ADDRESSABLE TO ANY 4K SLOT
* SOFTWARE/HARDWARE MEMORY PROTECT FOR EACH 4K
* SPECIAL PAGING FEATURE ALLOWS UP TO 1 MEGABYTE ADDRESSABLE MEMORY
* INSTRUCTION MANUAL INCLUDED
* 30 DAY WARRANTY ON PARTS 90 DAY ON ASSEMBLED UNITS

AVAILABILITY: STOCK TO 90 DAYS

PRICES: ASSEMBLED $150, 16K KIT - $145, 12K KIT - $135, 8K KIT - $125, 4K KIT - $115, 4K EXPANSION KIT - $95

CONSTRUCTION MANUAL FOR EVALUATION $1.75

QUANTITY DISCOUNT 5% BOARDS

IDEAL FOR HOME & WORK USE - LOW POWER - HIGH DENSITY - LOW PRICES

CASHIER'S CHECK OR M.O. PREFERRED, PERSONAL CHECK DELAYS SHIPMENT 1 WEEK, C.O.D. CHARGES $5 PER BOARD

PURCHASE ORDERS ADD 5%. NET 30, NO DISCOUNT

DEALER & OEM INQUIRIES INVITED

OMNI SYSTEMS INC.
P.O. BOX 7536, UNIV. STATION
PROVO, UTAH 84602

READER SERVICE NO. 185

The user can thus switch at any time from editing to assembling to debugging. Quickrun allows the use of symbolic debugging. Being able to refer to memory locations by their symbolic names eliminates the need to consult program listings to find memory addresses. Also included in the system are memory protect features and a microcontroller. The latter allows the user to simulate up to 20 K bytes of memory.

Circle 647 on inquiry card.

A New Drafting Aid for Printed Circuit Layouts

The Machine and Control Division of Intertech, POB 128, Farmington ME 04938, has announced a new drafting template for $8.95 to be used in designing "tapeless" printed circuit board layouts using ink pens, or pencils. The template produces pin layouts of dual in line integrated circuits that are twice actual size. The template has a versatile pattern that includes all the popular sizes up to 50 pin packages.

Circle 635 on inquiry card.

Badge Reader Module Announced

RD Products Inc., 6132 Route 96, Victor NY 14564, have announced their Mark I badge reader for $100 in quantities of 1 to 25. The reader is designed to read 1D cards (such as credit cards) having magnetic stripes. A maximum of 12 decimal or hexadecimal digits can be read on a magnetic stripe having

Circle 43 on inquiry card.
Your microcomputer is only as good as its memory

Your microcomputer is... in a manner of speaking... capable of thinking. The amount of information you can store, combined with the access speed, determines the bottom line capability of your microcomputer.

The Midwest Scientific Instruments' FD-8 Floppy Disk Memory System will enable you to get the most out of your microcomputer... 6800 or 8080. We have the most complete 6800 disk operating system available today, which interfaces to any microcomputer via a single PIA chip.

When used with the MSI FDOS Operating System... included with purchase... the FD-8 gives your microcomputer a level of performance which can only be matched by larger computers. You can have program save and load with named, variable length, password protected files at three levels: object code files, source code files, and BASIC files.

It has disk routines which can be used easily by low level assembler language programs as well as high level BASIC programs.

The coresident assembler/editor saves and loads source files from disk, assembles and stores object code on disk, and creates assembly listings on a high speed printer.

You get a full ANSI standard BASIC interpreter* with program load and save, from disk as well as tape, with both random and sequential disk data files. BASIC also has multi-I/O port capability under software control.

And all you need do to access any of these functions is type the appropriate commands on your terminal.

The FD-8 is complete, including all power supplies, cabinet, cables, software, manuals, and documentation.

You can order the FD-8 as a kit for $1,150.00, or wired and tested for $1,395.00.

Visit one of our dealers for a demonstration, or send for our free Catalog of MSI products.

*MSI BASIC is an additional $65.00

MSI is a dealer for Southwest Technical Products. Mastercharge & BankAmericard Orders accepted.

Circle 44 on inquiry card.
A Tutorial Training Computer

The Datacom 1000 computer, controller and tutorial card is a new product available from Datacom Engineering, POB 406, Southampton PA 18086. This card is available in two models, the tutorial version and the fully populated version. The tutorial version is $185 assembled and tested, with power supply; the fully populated version is $345 assembled, tested and with power supply. In either case you get a 6502 processor, an instantly usable computer with expansion capabilities, and a unique system of touch sensitive input keypads, seen along the lower edge of the board depicted in this photo. A documentation package accompanies each processor board.

Circle 636 on inquiry card.

Attention LSI-11 People
Wanting Mass Storage

Charles River Data System Inc., 235 Bear Hill Rd, Waltham MA 02154, has announced its new FD11 Floppy Disk Add-On System for PDP-11 computers as well as LSI-11 microcomputers. The system was designed to be hardware, software and media compatible with the DEC RX11 products. It includes write protect switches, unit select switches, up to four drives with one unit load on the DEC Unibus, PROM self-diagnostic programs and bootstrap loader. Track to track access time is 6 ms. Single quantity price of the drive and DEC compatible controller is $2750.

Circle 638 on inquiry card.

Make the HAL Connection

- 8080A MPU
- 1K Monitor/Debug Software (2708) 2K PROM
- 2-3624 PROMs
- On board space for: 4K ROM (2708), 2K RAM (2102A-4)
- TTY Current Loop or RS-232C serial I/O (8251)

and solve your lab or OEM computer problems

1. Connect to the 24 lines of Parallel I/O and ASCII or Baudot Serial I/O.
2. Connect to processor bus to add memory expansion board/EPROM programmer, I/O devices, or our unique CRT Terminal Board.
3. Connect to our power supply or use your own.

Lab users and OEM's alike will find solutions to their computer problems with the MCEM-8080 computer. All essential computer system elements are incorporated in this fully assembled, tested single-board computer. Some unexpected features of the HAL MCEM-8080 are: hardware "front panel" which allows setting a breakpoint and manual control of the computer; 1K ROM Monitor/Debug Software (with user callable, Intel compatible I/O routines) which greatly simplifies program development; Parallel and Serial I/O on the board; and very reasonable prices.

Call or write for further information on the HAL Connection.

You'll be glad you did!

HAL COMMUNICATIONS CORP.
Box 365B 807 E. Green Street, Urbana, Ill. 61801
Telephone (217) 367-7373

For the benefit of the steadily growing ranks of pocket calculator game enthusiasts, Hewlett-Packard is offering the Games Pac I. Intended for use with the company's HP-67 and HP-97 programmable calculators, the package contains an owner's handbook (shown) and 20 preprogrammed program cards. Games include Space War, Golf, Slot Machine, Dices and a Bowling Scorekeeper. The price is $35. Contact Hewlett-Packard, 1501 Page Mill Rd, Palo Alto CA 94304.

Circle 648 on inquiry card.

A New Version of the PCM-12 Microcomputer

Pacific Cyber/Metrix Inc., 180 Thorup Ln, POB 215, San Ramon CA 94583, has announced a new version of their PCM-12 microcomputer called the PCM-12A. The new unit is a 12 bit machine based on the Intersil IM6100 microprocessor, and is designed to be compatible with PDP-8
Minidisk and Verbatim™ — New Products from ITC

Information Terminals Corporation, 323 Soquel Way, Sunnyvale CA 94086, announces a new proprietary ferric oxide binder system called Verbatim™. This new material will form the storage medium for the company's floppy disks, magnetic cards, computer cartridges, and digital cassettes, mini data cassettes and the MD 525 minidisk. The new binder is said to be more resilient and less abrasive than previous binders.

The minidisk is approximately half the size of the standard floppy disk and provides one third the storage capacity. Priced at $5.95 per disk, the minidisk is intended for use in the small systems market as well as in power typing and editing systems. When you use high speed digital media, this type of product will prove most useful.

The HP9831A desk top computer is a personal computing entry of Hewlett-Packard, which comes in at the high end of the present day spectrum of price, $7200. This machine speaks BASIC with an 8 K byte source program string area expandable to 32 K maximum in 8 K byte increments. It uses an on board tape drive with 90 ips (229 cm/sec) search and rewind speed, 22 ips (56 cm/sec) data transfer speed. The tape cartridge can hold approximately 250,000 bytes and is available to BASIC applications programs as a resource. The average access time is quoted as six seconds. Auxiliary ROM cartridges are also available for expansion of its capabilities. The main display peripheral is a 32 character LED alphanumeric device, and the keyboard is used as the primary interactive input. Options include printers, floppy disks and other goodies. For inquiries contact Hewlett-Packard at 1501 Page Mill Rd, Palo Alto CA 94304.

Circle 641 on inquiry card.

Bite Analysis, Anyone?

The latest issue of Complot's Application Newsletter describes a new technique being used at the University of Michigan's School of Dentistry. Researchers are using computer graphics

PAPERBYTES
Tiny Assembler 6800 — Microprocessor Self Assembler

Originally described in the April and May 1977 BYTE, PAPERBYTES is now offering Jack Emmerich's Tiny Assembler 6800. This book contains the complete Tiny Assembler source listing plus object code in cross assembly format (space restrictions prevented printing of this material in BYTE). A bar code version of Tiny Assembler is included for convenience, as well as reprints of Jack's two articles and additional user manual materials. Tiny Assembler will run on any machine with MIKBUG and 4K of memory starting at address 0000, and is an excellent tool for the interactive development of functional blocks for a large structured program. Add it to your 6800 system and you'll have a valuable programming aid which can free you from the drudgery of machine language. The best part is the price: only $7. Order yours today!

Design and Implementation of a Microprocessor Self Assembler

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
</tr>
<tr>
<td>City</td>
</tr>
<tr>
<td>PAPERBYTES Tiny Assembler 6800</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

☐ Check enclosed
☐ Bill MC # Exp. Date
☐ Bill BA # Exp. Date

Signature

In unusual cases, processing may exceed 30 days.

You may photocopy this page if you wish to leave your BYTE intact.

Send today to:
BYTE Interface Technical Services, Inc.
70 Main St
Peterborough NH 03458
Dealer inquiries invited
All orders must be prepaid.

No. 700

Circle 51 on inquiry card.
An X-ray of the skull's profile is first digitized into 177 key coordinate points to facilitate plotting. A typical plot is then produced in 30 seconds, which can be overlaid to show deviations of the skull and jaw due to age, corrective surgery or other factors. Researcher Dr. Geoffrey Walker looks upon the technique as a major tool which can improve the quality of corrective facial surgery and other dental techniques. For further information contact Complot Application News, One Houston Sq (at 8500 Cameron Rd), Austin TX 78753.

Circle 642 on inquiry card.

A New Economical 6800 Software Package

Users of 6800 based microcomputers will be interested in Inpro Micro Systems' new software package called MIKADOS, a combination debugging program, assembler and operating system which resides in 2.5 K bytes of programmable memory. The assembler generates object code on the same line as the user's mnemonic to provide an immediate program listing. The price, which includes user manual and hexadecimal object code listing, is $12.95. Contact Inpro Micro Systems, P.O. Box 7776, Van Nuys CA 91409.

Circle 643 on inquiry card.

Now available direct from manufacturer...

PS-40
48-column Thermal Printer

- 64 ASCII characters (upper case)
- TTL level parallel interface
- STROBE-BUSY handshaking
- Active low new line implementation
- Local line feed (continuous)
- 20 characters-per-second throughput
- PS-40 mechanism and Print Head $230
- With TTL parallel interface — kit $350
- With TTL parallel interface — assembled and tested $400
- PS-40 with versatile multifunction RS-232C interface (F-8 microprocessor) — assembled and tested $550
- Power supply $100
- Case $80
- Paper roll (5¾” x 164”) $3
- Shipping Charges $4

Delivery — Four weeks ARO, F.O. B. Addison, Texas

To Telpar, Inc.
4132A Billy Mitchell Road, P.O. Box 796
Addison, Texas 75001 Telephone (214) 233-6631

□ Enclosed is my check in the amount of $_________ covering purchase of __________.

□ Place my order on Master Charge, please. My MC number is __________; expiration date is __________.

Signature
Name (please print)
Address
City State Zip

Money back guarantee! Texas residents add 5% sales tax.

telpar, inc.

Circle 46 on inquiry card.
quantities, and the press release says delivery is from distributor stocks. The company is located at 323 Soquel Way, Sunnyvale CA 94086.

Circle 644 on inquiry card.

Not for the Casual Amateur, But . . .

This “Traverscan” absolute free cursor digitizer from the H. Dell Foster Company, POB 32581, San Antonio TX 78216, is an interesting idea which experimenters might find challenging to emulate. This is a real time graphic digitizer which measures, counts and displays the X and Y coordinate values of points, with six digits per axis read directly in English or metric units. Numerous operational features include electronic scaling, 2 axis scalable integrator (or digital planimeter) for automatic area computation, and of course, output interfaces through RS-232C serial lines so that the data can be captured by a computer. One uses a graphic digitizer to get information from drawings into a computer by tracing lines and tracing around regions. The complexity of this particular device strongly suggests it has some microprocessor intelligence built into it locally. H. Dell Foster Company is a subsidiary of the Keuffel and Esser Company.

Circle 649 on inquiry card.

Improving Your Memory

If you want to experiment with APL or other sophisticated data constructs requiring extensive memory space, (see the editorial in the November 1976 BYTE), you may be interested in the new 64 K byte programmable memory

---

**THE COMPUTER ROOM**

SMALL COMPUTER SYSTEMS • SOFTWARE • AMATEUR RADIO EQUIPMENT

3455 Southwest Temple, Salt Lake City, Utah 84115 Phone: 801-466-4311

“WE TAKE THE MYSTERY OUT OF THE MICRO”

One Of The Nations Largest Full-Service Computer Stores. Over 1600 Square Feet Of Sales And Service Facilities.

WHEN YOU WRITE FOR OUR CATALOG AND ENCLOSE $1 TO HELP DEFray THE COST OF HANDLING AND MAILING, HERE’S WHAT YOU GET:

1. A CERTIFICATE GOOD FOR $2 ON YOUR NEXT PURCHASE.

2. THE COMPUTER ROOM EASY TO UNDERSTAND CATALOG COVERING.

   IMSAI
   THE DIGITAL GROUP
   POLYMORPHIC SYSTEMS
   SOUTHWEST TECHNICAL PRODUCTS CORPORATION
   TECHNICAL DESIGN LABS ETC.

3. THE COMPUTER ROOM “EASY GUIDE” TO HELP YOU PICK THE RIGHT SYSTEM, PERIPHERALS, COMPONENTS, AND SOFTWARE FOR

   THE BEGINNER
   THE ADVANCED
   THE EXPERT
   THE SMALL BUSINESS

4. A CURRENT LISTING OF PRESENTLY AVAILABLE

   SOFTWARE PUBLICATIONS PERIPHERALS

5. INFORMATION ON REPAIR SERVICE, LOW COST CUSTOM PROGRAMMING AND OTHER SPECIAL SERVICES.

AT THE COMPUTER ROOM YOUR WRITTEN QUESTIONS ARE HAPPILY RECEIVED AND PROMPTLY ANSWERED

WE ALSO STOCK A COMPLETE LINE OF AMATEUR RADIO EQUIPMENT

BANKAMERICARD MASTERCHARGE

137 Circle 47 on inquiry card.
board recently announced by Extensys Corporation, 592 Weddell Dr, Suite 3, Sunnydale CA 94086. The 5 by 10 inch (12.7 by 25.4 cm) board has a hardware provision for bank switching to add over one million bytes. This Altair compatible board also allows memory address to be set in 8 K byte increments and provides hardware protection in 16 K byte increments. Power supply voltages required are 12 V 300 mA, 5 V 750 mA, and -5 V 1 mA (Power supplies are not included). Memory overlap protection is provided to prevent conflict with existing memories. The price of the 64 K byte board is $1,195. A 32 K byte board is available for $895; and a 48 K byte board for $1,195.

Circle 650 on inquiry card.

Voice Synthesizers, Anyone?

MICOM Systems, 9551 Irondale Av, Chatsworth CA 91311, is manufacturing a Voice Response System intended for minicomputers, which uses a microprocessor as a control element. The press release describes a system which can accept Touch-Tone encoded inputs and give digitally generated synthetic voice outputs. This system, which is definitely oriented toward commercial systems markets, has prices beginning at $500.

Circle 651 on inquiry card.

A Decimal Order of Magnitude Difference in Price

We received this picture of a new Model 100A Logic Analyzer, produced by Paratronics Inc, 150 Tait Av, Los Gatos CA 95030, and seen in some detail in Popular Electronics, February 1977. This unit is available in a complete kit for $198.50, along with a 100 page assembly and applications manual.
IN WIRE-WRAPPING OK HAS THE LINE.....

MODEL WD-30

WIRE DISPENSER

- 50 FT. ROLL OF 30 AWG.
  KYNAR WIRE-WRAPPING WIRE
- CUTS THE WIRE TO LENGTH
- STRIPS 1 INCH OF INSULATION

AVAILABLE IN FOUR COLORS

WD-30-B BLUE WIRE
WD-30-Y YELLOW WIRE
WD-30-W WHITE WIRE
WD-30-R RED WIRE

$345 EACH

MINIMUM BILLING $25.00
ADD SHIPPING CHARGE $1.00
NEW YORK STATE RESIDENTS ADD APPLICABLE TAX

PATENT PENDING

OK MACHINE & TOOL CORPORATION
3455 Conner St., Bronx, N.Y. 10475 (212) 994-6600 / Telex 125091

Circle 20 on inquiry card.

According to its makers it compares quite favorably with commercial laboratory equipment products costing $2750 (and up). The output display is on an ordinary oscilloscope, with blanking, and consists of a truth table of the last 16 states of a 1 byte word of data. Various other useful features make this an interesting test instrument which may prove useful to many of our readers.

Circle 652 on inquiry card.

Two New Dalen Cassette Interfaces

Dalen Electronics, 7214 Springleaf Ct, Citrus Heights CA 95610, has announced two new cassette interface boards compatible with the Altair bus. The first board is a universal cassette interface called the UCRI, available for $175 assembled or $135 in kit form. Data transmission rate is user selectable from 520 to 41,000 bps. The unit can be optionally equipped with tworeed relays allowing the unit to independently control two cassette recorders. The second interface board, the CRH, has a maximum data transmission rate of 6000 bps and is available for $165 assembled or $120 as a kit.

Circle 571 on inquiry card.

A Source of Neat Ideas

Persons interested in discovering the functional characteristics of a well thought out editing terminal design would do well to look at the new "2645A Display Station User's Manual," publication number 02645-90001, put out by Hewlett-Packard Data Terminals Division. (Inquiries should be addressed to Inquiries Manager, Hewlett-Packard Company, 1501 Page Mill Rd, Palo Alto CA 94304.) This pamphlet of approximately 40 pages in length is the user's manual for this microprocessor controlled data terminal with editing capabilities, which make it extremely adaptable to numerous practical tasks.

Circle 658 on inquiry card.

A New Tutorial Kit from Motorola

Motorola has announced a new 8 bit 6800 based microprocessor kit for tutorial purposes called the Educator II. The unit contains a 128 byte
programmable memory, a 512 byte read only memory and a TTL clock circuit.

An executive program residing in

the read only memory contains rou-
tines for servicing interrupts, program
transfer to and from cassettes, tape
searching programs, and a program to
test the finished kit. The executive
program uses 14 bytes of the program-
able memory as a scratchpad mem-
ory, leaving 114 bytes for user pro-
grams. An optional 128 bytes of pro-
gammable memory can be added
later.

The Educator II comes with an
aluminum case for $169.95, including
assembly and operating manuals. The
unit is available from Motorola dis-
tributors. For further information, dis-
tributors and computer stores should
contact Motorola HEP/MRO National
Manager, 705 W 32nd St, Tempe
AZ 85282.

THE
THE
Circle 657 on inquiry card.

A Processor Board with Expanded
Features

Morrow's MicroStuff, POB 6194,
Albany CA 94706, announces an Altair
bus compatible plug-in processor board
designed to expand the capabilities
of existing 8080a microcomputer sys-
tems. Available for $325 assembled or
$250 as a kit, the board offers two
special features. The first enables the user
to step through a program at a rate
variable from 1 to 65,000 steps per
minute. The second feature prevents
the 8080a processor from shutting
off after a HALT instruction. While
in this state, registers, memory and IO
locations can be examined and altered.
A built-in 12 pad keyboard and 10 digit
readout are included on the board.

Circle 654 on inquiry card.

A New 6800 Microcomputer
Evaluation Kit

A new microcomputer evaluation
kit for M6800 systems is now available
from the Integrated Circuits Division of
Motorola Inc. This kit will prove quite
useful to those individuals with a min-
imal budget and a keen interest in
computers.

The MEK6800D2 kit, when assem-
bled, is a fully functional microcomputer
system based on the MC6800 micro-
processing unit and its family of asso-

Timesharing Software Listing Available

Gregory Research Associates, 1900
Greymont St, Philadelphia PA 19116,
are making available a directory entitled
Remotely Accessible Conversational Pro-
grams and Data Bases. The price is $28,
which includes three bimonthly updates.
The listing is a guide to thousands of
programs for timeshare users in the
areas of business, engineering and
science. The programs listed are written
in a conversational style which requires
no special knowledge of computer
languages. Users pay only for those
programs used based on rates listed
in the directory. Programs are received
via standard telephone lines; some
require terminal data rates of 120
characters per second.

Circle 653 on inquiry card.

THE PAIA 8700 COMPUTER/CONTROLLER

An exceptional price on an applications oriented 6503 based micro-processor system
featuring:

- 1K bytes RAM locations (512 bytes supplied)
- 1K bytes ROM (256 byte monitor included)
- 2 - 8 bit input ports
- 2 - 8 bit output ports (1 latched, 1 buffered)
- 34 key touch operated keypad (used by monitor
to allow entry & execution of user programs
also user definable)
- 2 - latched seven segment displays (used by
monitor to display memory location & contents
easily user programmed)
- Optional cassette interface ($22.50) fits
entirely on the processor board.

THE IDEAL, LOW COST SOLUTION TO IMPLEMENTING ALL THOSE WILD COMPUTER BASED CONTROL SYSTEMS YOU'VE BEEN DREAMING OF!

PAIA software currently available or under
development includes: Music synthesizer
interface; Home applications package including:
multi-zone fire/burglar alarm, real time clock,
energy saving heat/air-conditioning control,
computer generated "door-bell"; Model railroad
controller and more...........................

8700 COMPUTER/CONTROLLER KIT ......... $148.95
(requires 5v, 1.2A; 12v, 150 ma.)
Shipped direct from PAIA (add $3, 00 postage)

Circle 49 on inquiry card.

Also available at FULL-LINE Computer stores.
cassette interface circuitry for the display module also (25.4 by 15.9 cm). The keyboard and associated memory and dual phase encoding.

The display consists of six 7 segment LED readouts that display four address digits and two data digits in hexadecimal format. The keyboard is hexadecimal with eight additional command keys.

Also included in the kit’s three-ring binder is an assembly manual that covers testing, schematics, “JBUG” monitor program listing and parts listing. An MC6800 Programming Reference Manual and the MC6800 Microcomputer System Design Data book are also part of this package. The kit may be used “as is” or expanded to a full 64 K system through addition of buffers (and minor modifications of on board memory decoding).

Cost of the kit is $235, and it is available from Motorola Distributors. The 5 V, 2 A power supply and cassette recorder required by this device are not included, and can be purchased locally.

Circle 655 on inquiry card.

A New Prototyping Board

A new general purpose printed circuit board for the experimenter is available from World Wide Systems Corporation, 8305 Private Ln, Annandale VA 22003, for $24.95 postpaid. The 100 pin board is Altair bus compatible and features two gold plated 36 pin IO strips. A portion of the board has been reserved for the installation of a voltage regulator and heat sink; the power bus, ground plane and pins are labelled. Board dimensions are 5.125 by 10 inches (13.02 by 25.4 cm). The board can be used with point to point solder connections, wire wrap or a combination of the two. With this tool the experimenter can try his hand at designing custom peripheral interfaces.

Circle 656 on inquiry card.

INTERNATIONAL DATA SYSTEMS, INC.

400 North Washington Street, Suite 200
Falls Church, Virginia 22046 USA
Telephone (703) 536-7373

<table>
<thead>
<tr>
<th>$100 Bus Cards (ALTAIR/IMSAI Compatible)</th>
<th>USES</th>
<th>KIT PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>88-SPM Clock Module</td>
<td>Your computer keeps time of day regardless of what program it is executing. Applications include event logging, date entry, ham radio, etc. Provision for battery backup is included.</td>
<td>$96.00</td>
</tr>
<tr>
<td>88-UFC Frequency Counter Module</td>
<td>Measure frequencies up to 800 MHz or period with 1/10 microsecond resolution. Computer can monitor four separate inputs under software control.</td>
<td>$179.00</td>
</tr>
<tr>
<td>88-MODEM Originate/Answer MODEM</td>
<td>Use your computer to call other computer systems such as large timesharing systems. Also allows other computer terminals to “dial-up” your computer. Auto-dialer is included so your computer can call other computers under software control. Operates at 110, 134.5, 150, 300, and 600 bps.</td>
<td>$245.00</td>
</tr>
</tbody>
</table>

GENERAL PURPOSE PERIPHERALS

| MCTK Morse Code Trainer/Keyer | Hard/Software package which allows your computer to teach Morse Code, key your transmitter, and send prestored messages. Uses “NEW CODE METHOD” for learning. | $29.00 |
| TSM Temperature Sensing Module | Use it to measure inside and/or outside temperature for computerized climate control systems, etc. | $24.00 |
| DAC8 Eight Bit Digital to Analog Converter | Requires one eight bit TTL level latched parallel output port. Use it to produce computer music or to drive voltage controlled devices. | $19.00 |

Terms: Payment with order. Shipment prepaid. Delivery is stock to 30 days. Write or call for detailed product brochures.

Circle 50 on inquiry card.
Novel 8 Bit Multiplication

Here is a 1 byte multiplication routine that was submitted by Christopher D. Glaeser for use on an Intel 8080 along with his idea on how to save memory space and execution time. Working on the premise that one of the main problems with many multiplication routines is that they must call another routine to perform a double byte shift, he uses the DAD H command of the 8080 to surmount this difficulty. The DAD H command performs the double byte shift by adding the HL register pair to itself. This saves having to call another routine to perform this function. The operands are loaded into the C and D registers prior to calling the subroutine. The resulting answer is placed in the BC register pair. All of the numbers in this listing are in octal.

<table>
<thead>
<tr>
<th>Address</th>
<th>Op</th>
<th>Operand Label</th>
<th>Mnemonic</th>
<th>Commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td>006000</td>
<td>325</td>
<td></td>
<td>PUSH D</td>
<td>save original values of registers on stack;</td>
</tr>
<tr>
<td>006001</td>
<td>345</td>
<td></td>
<td>PUSH H</td>
<td></td>
</tr>
<tr>
<td>006002</td>
<td>132</td>
<td></td>
<td>MOV E,D</td>
<td>E = first operand;</td>
</tr>
<tr>
<td>006003</td>
<td>026</td>
<td></td>
<td>MVI D,000</td>
<td>initialize registers;</td>
</tr>
<tr>
<td>006005</td>
<td>152</td>
<td></td>
<td>MOV L,D</td>
<td></td>
</tr>
<tr>
<td>006006</td>
<td>142</td>
<td></td>
<td>MOV H,D</td>
<td></td>
</tr>
<tr>
<td>006007</td>
<td>006</td>
<td></td>
<td>MVI B,010</td>
<td></td>
</tr>
<tr>
<td>006011</td>
<td>171</td>
<td></td>
<td>LOOP</td>
<td></td>
</tr>
<tr>
<td>006012</td>
<td>037</td>
<td></td>
<td>JNC SKIP</td>
<td>if CY = 0 go to SKIP;</td>
</tr>
<tr>
<td>006013</td>
<td>322</td>
<td></td>
<td>DAD D</td>
<td>else HL = HL + DE;</td>
</tr>
<tr>
<td>006016</td>
<td>031</td>
<td></td>
<td>JNG SKIP</td>
<td>exchange HL with DE;</td>
</tr>
<tr>
<td>006017</td>
<td>363</td>
<td></td>
<td>XCHG</td>
<td>HL = HL * 2;</td>
</tr>
<tr>
<td>006020</td>
<td>061</td>
<td></td>
<td>DAD H</td>
<td>XCHG exchange HL with DE;</td>
</tr>
<tr>
<td>006021</td>
<td>363</td>
<td></td>
<td>DCR B</td>
<td>B = B - 1;</td>
</tr>
<tr>
<td>006022</td>
<td>005</td>
<td></td>
<td>XCHG</td>
<td>if CY = 0 go to LOOP;</td>
</tr>
<tr>
<td>006023</td>
<td>302</td>
<td></td>
<td>DCR H</td>
<td>MOV E,H  else B = H; [high order 8 bits of answer];</td>
</tr>
<tr>
<td>006026</td>
<td>104</td>
<td></td>
<td>DCR C,L</td>
<td>MOV C,L  C = L; [low order 8 bits of answer];</td>
</tr>
<tr>
<td>006027</td>
<td>115</td>
<td></td>
<td>POP H</td>
<td>POP D    restore original value of registers;</td>
</tr>
<tr>
<td>006030</td>
<td>341</td>
<td></td>
<td></td>
<td>return to calling program;</td>
</tr>
<tr>
<td>006031</td>
<td>321</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>006032</td>
<td>311</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COMPUTER ENGINEERING

THE MIDNIGHT SPECIAL . . .
... a full color limited edition poster for your Casey Jones fantasies.

The poster is 16½ by 21½ inches (41.9 by 54.6 cm) with a white border. The colors are the same as the original by Robert Tinney, which graces our July cover, minus the BYTE logo. The price is $3.00, plus 50¢ postage. It is shipped unfolded, in a mailing tube.

ORDER NOW. SUPPLIES ARE LIMITED.

BITS Inc
70 Main Street
Peterborough NH 03458

Posters $3.00 each
plus $.50 mailing

☐ Bill BankAmericard
☐ Bill Master Charge
☐ Check Enclosed
☐ Enclosed

S

No.

Exp. Date

Name:

Address:

City __________________ State ______ Zip:

In unusual cases, processing may exceed 30 days.

Circle 51 on inquiry card.
The answer to those unavailable early issues of BYTE. Best of BYTE, Volume I, edited by Carl Helmers Jr and David Ahl. This 384 page book is packed with a majority of material from the first 12 issues. Included are 146 pages devoted to "Hardware" and how-to articles ranging from TV displays to joysticks to cassette interfaces, along with a section devoted to kit building which describes seven major kits. "Software and Applications" is the other side of the coin: on-line debuggers to games to a complete small business accounting system is included in this 125 page section. A section on "Theory" examines the how and why behind the circuits and programs. "Opinion" closes the book with a look ahead as to where this new hobby is heading. It is now available through BITS Inc for only $11.95 and 50 cents postage and handling.

Here is just a taste of some of the titles to be found in the Table of Contents, from the nearly 70 that appear there:

The Shadow, Buck Rogers, and the Home Computer — Gardner
The State of the Art — Helmers
Could a Computer Take Over? — Rush
A Systems Approach to a Personal Microprocessor — Suding
Programming for the Beginner — Herman
Magnetic Recording for Computers — Manly
Assembling an Altair 8800 — Zarrella
Build a 6800 System With This Kit — Kay
The Digital Equipment LSI-11 — Baker
Flip Flops Exposed — Browning
Keyboard Modification — Macomber
Build a Television Display — Gantt
Build a Fast Cassette Interface — Suding
Add a Kluge Harp to Your Computer — Helmers
Write Your Own Assembler — Fylstra
Design a On Line Debugger — Wier and Brown
Processing Algebraic Expressions — Maurer
Total Kitchen Information System — Lau
A Small Business Accounting System — Lehman
Books of Interest
Magazines

Circle 51 on inquiry card.
Gourmet Reading from BITS

Security Electronics by John E Cunningham. To catch a thief, apply liberal doses of ingenuity and a modicum of cleverness. Find out what's been tried in conventional alarm systems before you go off computerizing your home security system, though. New 2nd edition $5.95

Introduction to Biomedical Electronics by Edward J Bukstein. What's been done in robot doctors? Nothing so far. But in terms of electronic aids to physicians and practices of health researchers, consult this background review of the field of biomedical electronics $5.95

Scelbi "6800" Software Gourmet Guide & Cookbook

Scelbi "8080" Software Gourmet Guide & Cookbook, both by Robert Findley. Have you tried cooking up a program lately on your 6500 or 8080 processor? Have you needed a dash of ideas on how to add spice to a program? Then the Scelbi "6800" Software Gourmet Guide & Cookbook and the Scelbi "8080" Software Gourmet Guide & Cookbook may prove to be quite useful additions to your library.

Both books contain a wealth of information on the 6800 and 8080 instruction sets, plus general programming techniques applied to the 8080 and 6800, conversion routines, floating point routines, decimal arithmetic routines, and much more. Order yours today and get a taste of what your processor can do. Both books are priced at $9.95 each.

Practical Microcomputer Programming: The Intel 8080 by WJ 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, an architecture which appears in the MITS Altair, 8800, Processor Technology SOL, IMSAI 8080, Polymorphic POLY-88, and other popular microcomputer system products.

After several preliminary chapters, the authors get down to practical details with topics such as moving data, binary arithmetic operations, multiplication and division, use of the stack pointer, subroutines, arrays and tables, conversions, decimal arithmetic, various 10 options, real time clocks and interrupt driven processes, and debugging techniques. Most examples are given in symbolic assembly form, with occasional listings of assembled code using a Computer Automation software development system. This 306 page hardcover book is well worth its $21.95 price and should be in every 8080 or Z-80 user's library.

You may photocopy this page if you wish to leave your BYTE intact. In unusual cases, processing may exceed 30 days.
How to find a camera, a lid, and the variety of current approaches to making the computer more intelligent.

Projects in Sight, Sound, & Sensation by Mitchell Waite. Dedicated "to all space cowboys." Detailed theory and practice of seven fascinating amateur electronics projects, along with a complete and detailed appendix on how to make PC boards. The projects included in this book are: The Syntheshape, an art generator that can be used to generate beautiful patterns on the screen of an oscilloscope. An electronic music box that will play over 3000 possible melodies when the lid is lifted. A way to control muscle tension explained in chapter 4. A muscle-wave biofeedback monitor can be used to achieve deep relaxation. The laser-light show transfers light into fascinating patterns in a darkened room. Other projects include a Kirlian camera, a digital ESP machine, and neon-light randomizer. $5.25.

The Great International Math On Keys Book, from Texas Instruments. Do you need a numerical algorithm for calculating exchange rates in a hotel on a foreign trip? Or figuring those mysterious "points" when facing the sanctimonious charisma of a banker at mortgage time? This book is a compendium of simple explanations and step by step procedures for accomplishing numerical solutions to numerous commonly encountered situations in daily life. Each entry is characterized by a statement of the problem, including the elements of theory required, and the keystrokes needed to solve the problem on an algebraic entry calculator. This is an invaluable resourcebook of information for the person who is inclined to manipulate numbers. $6.95.

How to Solve Problems by Wayne A. Wickelgren. When confronted by a problem, many of us spend more time puzzling over how to solve it than in actually doing so. This book analyzes and systematizes the basic methods of solving mathematical problems. The methods are described in terms of a modern theory derived from research in computer simulation of thinking. Examples illustrating these methods include chess problems, logical puzzles, and railroad switching problems frequently encountered in science and engineering. Whether your interest in solving problems is professional, recreational, or both, you will find this a helpful book. $6.50 softcover.

Chess Skill in Man and Machine edited by Peter W Frey. This is a most fascinating book, concerning itself with the when, how, and why of computer chess. The when describes past ACM computer chess tournaments, with the details of more than a dozen games. The how consists of the basics of both human chess skill and computer chess theory. It includes a detailed description of the best computer chess program to date (Northwestern University's CHESS 4.5), an end game program called PEASANT, and of various search strategies and heuristic computer chess theory which should enable one to write his own chess program. The book ends with the why concerning the contributions, now and in the future, of computer chess to understanding artificial intelligence, human intelligence, and learning. The only difficulty for the hobbyist's computer chess program is the need for a large computer for the fast processing of search strategies and large core storage for the program and its results, $14.80 hardcover.
Continued from page 11

sets a very high goal for us. When your latest program crashes, a poster like this on the wall could give you the inspiration to hit the restart button and try again.

I find that I can justify my activities in small computing a lot easier when I can point to an illustration like this than through the idea of the “ultimate toy.” One picture is indeed worth a thousand bytes.

Robert J Retelle
2005 Whittaker Rd
Ypsilanti MI 48197

WHY NOT 8 TRACKS?

I am curious to know why 8 track tapes are not used for memories instead of cassettes. I would think that by using one channel for a clock track you could have memory locations via a clock track counter. Also by moving from track to track, access time would be saved, wouldn’t it? Please elaborate.

Kurt Kramer
8634 Lake Isle Dr
Tampa FL 33617

Sounds like a good idea. I once saw a surplus automated cartridge turret mechanism which had been used by a radio station. This mechanism, which its owner intended to refurbish, would be used for an on-line storage system of rather large capacity based on the wide tape cartridges similar to an 8 track stereo cartridge. Maybe a reader will pick up on this and try it out, reporting the results as an article.

RETROSPECTIVE: BYTE’S
PORNOGRAPHIC PHOTOS MARCH ON

Never thought that BYTE would go for pornography. But without the benefit of a G-string, you show on page 54 of the March 1977 issue a bare soldering iron. Soldering irons without G can certainly corrupt innocent computers. Since your local Sears or Radio Shack do not generally bother to stock properly grounded ones, beginners should be warned.

Sholom Kass
367 Baden Av
San Francisco CA 94080

The G-string mentioned in Sholom Kass’s letter is the ground wire of a 3 wire 110V interface for power. It is standard industrial practice to use such grounding, especially where delicate parts such as microprocessors, and other LSI or CMOS gates are being used. The static charge which a normal human being can build up in a dry room with rug and rubber soled shoes can drive a spark across a considerable gap, and is measured in many tens of thousands of volts.

SOME NEWS FROM DEUTSCHLAND

My involvement with micros began when one of your associates, Dan Fylstra, gave a talk at IFIP/IFAC in Paris last summer, in which he managed to mention the Motorola 6800, the MOS Tech 6502, and also gave an example of a program in PL/M. On coming back to Darmstadt, I found two groups of people at our facility who were interested, nearly all professional computer engineers. One of them, Asbjorn Smitt, had already worked on a 6800-based system which we use for simulating satellite telemetry data; another, Helmuth Wothmann, had his own business and had already built a 6800-based system. So our choice as a club was somewhat inevitable. Asbjorn had bought the EVK 100 and the SWTPC AC-30 cassette controller, and we very nearly went EMI, but then the prices sort of doubled, so we went with OSI instead after our chairman, Graham...
Else (whose main interest is biological monitoring), had convinced everyone we needed something which could be started off cheaply. (In the end Graham is at the Rolls-Royce end of the market, with a Digital Group Z-80.) I had bought a SCAMP, which to my astonishment worked first time despite some startup trouble which we attributed to the rather bulky socket for the CPU, and which eventually I hope to work up into a controller for the Selectric I'm typing this on. Our club is now about 40 strong, with considerable purchasing leverage since there are believed to be about 30 people actively buying components (25 OSI superboards, 16 Selectrics, which was all I could get in my Volvo the last time I went to England, and if you know anything about the quaint European customs, avoid Belgium), and of course our own BYTE archive. We are also in contact with John Barnes, a founder and member of the British Amateur Computer Constructors' Association, and having heard of the success he has had in using standard 8 track cassettes with short endless loops and saturation recording, I have ordered a few for experiments. Graham and Helmuth in the meantime are working on Phillips cassettes.

Things would probably have started more slowly had it not been for the impact of BYTE and the fact that Jack Davies, the guy running PACS, lives in Darmstadt with a perpetual font of goodies which are available for inspection and (usually) instant purchase. The only complaint we have about OSI is that they have never dished out the listing of the Superbug monitor; so this has now been decompiled using the Thomson Lister (running on the first of our club machines, built by Bob Dees of ICI), and another program running on our 4/72. (ICI, not Amdahl!) I guess our next move is to find out if Tom Pittman has an OSI version of Tiny BASIC yet.

I suppose the next move would be for us to get some sort of European convention going. It would be a nice if it could appeal through your pages for any hobbyist clubs in Europe who would be interested in attending such a thing, or indeed if anyone is prepared to offer a site which is fairly central. There have been two attempts to get Europe-wide hobby clubs formed, but in both cases the impetus has come from the hobby industry rather than the hobbyists, who tend to view (other people's) commercial bias with suspicion. If we forget the launches which are scheduled for this year ESA can probably be persuaded to host it if we hear from enough people, but obviously we are equally curious if anyone has a 72 character graphics cum VDU interface for 625 line TV receivers, or a meta-assembler capable of compiling code for more than one micro, and so forth. Incidentally, why use OMEGA when there are satellite systems, like LORAN-c? Ralph Burhans is busy at work on a LORAN-c interface... CH/

Good luck to BYTE, may it grow ever heavier, and perhaps sprout a Euro-

$699 builds

Equinox 100™

THE FRONTRUNNER

Equinox 100™ is the 8080 CPU/S-100 Bus computer kit that's years in front of Altair® and IMSAI in design, function and front-panel programming capability. Equinox 100™ is easier to build, easier to program, easier to expand in the future and completely debugged right now. After all, it's from Parasitic Engineering, the leading supplier of debugging kits for the Altair® 8800. Before you invest in any micro-processor kit, discover the new Equinox 100™. At $699, it's clearly The Frontrunner. Write for free specs to Parasitic Engineering, P.O. Box 6314, Albany, CA 94706.

The Frontrunner from Parasitic Engineering
*A trademark of MITS Inc.

Circle 53 on inquiry card.
The MERLIN Super Dense add-on kit provides maximum resolution at a minimum cost. In fact, MERLIN with Super Dense has more capabilities than any other 9-100 bus video interface at any price!

Once you've seen 'Super Dense' graphic resolution you'll know there is nothing to compare it to...short of spending over $600...and even then you won't have all of the capabilities of SUPER Dense.

Super Dense provides true bit-mapping. Each and every point on the screen is controlled directly by a bit in memory. (Requires 8K of system memory.)

ROM character-graphics looked good for a while; then came MERLIN's 160 by 100 bit mapping graphics; and now... 620 by 200 bit-mapping graphics!!

If you're looking for a graphic display, MERLIN with Super Dense is the best there is. And if you hadn't considered graphics or thought it was out of your price range, consider what you could do with 320 H by 200 V graphics and for only 39$ extra.

The Super Dense add-on kit to the popular MERLIN video interface is now available with off-the-shelf delivery.

M320-K, Super Dense Kit...39$  
M320-A, Super Dense Assem...55$  
See MERLIN ad on previous page.

For information fast, write direct, or see 'Super Dense' at your nearest computer store.

MC and BAC accepted.

Super Dense GRAPHICS

320 Horizontal by 200 Vertical

If your computer can't BOL (or PLOT) it probably has MERLIN deficiency allergy.

(Equation Plotted with BASIC)

I was impressed with "The Built-In Logic Tester," Kurt Christner's article in the January issue, page 82. He had an idea for a super logic probe, considering the cost.

But what about those amateurs among us who have need of a CMOS probe rather than TTL? I would like to challenge Kurt to come up with an equally fine project for us nontechnically inclined who could really use a good quality, low cost CMOS probe.

By the way, I'm brand new with the January issue and I like what I see. It's a great magazine.

Tom Kryst  
212 Princeton  
Alpena MI 49707

COMPUTER POSSIBILITIES IN SECONDARY SCHOOLS

Attached is something that appeared recently in Science News (see page 168, March 12 1977 issue) which is a semi-popular weekly publication. What I found particularly interesting was the section concerning this year's winners of the national Science Talent Search. Of the ten winners, notice that four of them used computers in some capacity. Thinking back on my science fair entries of 20 years ago, such a thing borders on the inconceivable. Until very recently computers and computer time were so expensive that the economics of the situation precluded any but the work which had a very high probability of research or financial return. Therefore such investigations as these by high school students were impossible to justify on the basis of what they might produce.

Perhaps one of the things that those of us involved with the microcomputer and personal computing revolution tend to forget is that to most of the academic and scientific world at large, computers are simply tools to an end in whatever discipline is being pursued. Just think of what the possible benefits are should virtually limitless, although perhaps slow, computing time become available to any interested individual, such as these high school students. A local school could put together quite a respectable system of moderate capability for, say, 10 K bucks, and then allow appropriate students to pursue their interests generally unhindered by the constraints of financial return. The leap ahead in science education (and possibly science related areas of liberal arts) is truly mind boggling!

Robert R Wier  
POB 9209  
College Station TX 77840

I HATE BASIC, A POISON KEY LETTER

I wish to express my appreciation for your publication. For me, it has helped open the door to a new and fascinating world.

I wish to pose a general question. Why is it that, with all too few exceptions, the thrust of what might be called "microcomputer applications for the nonprofessional" seems to be lost in an endless loop called "games written in BASIC"? And as a corollary, one might well ask the question, "Of what redeeming social value is the latest listing of Star Trek?"

The Pong-Trek-Toe mentality seems to be ubiquitous; lift any magazine cover and underneath, in the table of contents, will be found an article dealing with games written in BASIC.

This is most assuredly not an attack upon BYTE magazine. It most assuredly is an attack upon the puerility of imagination manifested by microcomputer owners.

For every ten serious articles appearing in BYTE which are devoted to the embarrassment of riches in hardware being placed within reach, there is perhaps one serious and informative article devoted to software concepts. The term "concepts" should be emphasized; an assembly language listing for a specific microprocessor is not the expression of a concept.

The lack of interest in software concepts coupled with an overwhelming interest in amusing trivia does not speak well for the present state of the amateur group, from which emanates a great deal of "being in the vanguard of a revolution." Most of this talk is vainglorious nonsense.

At the present time, the only groups doing things worthy of the adjective "revolutionary" are the microprocessor manufacturers and the microcomputer system designers. For the purchaser of such riches to sit around playing Star Trek in BASIC borders on the perverse; to ascribe to such apathetic behavior the word "revolutionary" is a caricature.

It seems that the amateur group has been handed BASIC on a silver platter and is content to view it as the programming language. This is unimaginative at best.

The reason for being of microcomputer BASIC is simply to provide a workable language so that people will buy microcomputers. Insofar as I know, no one has ever claimed that BASIC uses the resources of a system organized in 8 bit words to its maximum benefit; BASIC has appeared on the microcomputer scene simply by default. It works and it is easy to learn and use. It sells microcomputers.

If Star Trek is the desired goal of our collective amateur ambition, then BASIC is sufficient; we need look no further.

However, if we wish to exploit the limited resources of the microcomputer in specific ways, looking toward specific
nontivial goals, then new programming languages should be written. There surely exists a diversity of ideas as to what constitutes good microcomputer programming languages. Attempts should be made to implement these ideas. We cannot sit back and wait for commercial organizations to do all the development.

Most of all, we need software concepts. Memory management concepts. Data manipulation concepts. Algorithms for mathematical functions.

What is a heap? How does it work? What does its presence imply about the philosophy of a language which uses it?

If, for every page devoted to Star Trek listings and other compatible nontivial goals, then new programming organizations to do all the development.

BYTE cannot sit back and wait for commercial endeavors, BYTE published a page devoted to software concepts, then we would all soon be the richer.

However, BYTE magazine cannot prosper by serving up beefsteak if the readership is clamoring for canned pork and beans.

From time to time, in various magazines such as BYTE, there appear willed-eyed letters to the editor accusing the programming profession of clannishness, a lack of willingness to share ideas, secretiveness and a general depravity. Such letters state the fact that we, the amateurs, are pure of heart and clean of hand. Is this true, or is it rather that we are so numbed by sitting around playing with BASIC (implemented by professional programmers) that we really just don’t have much of any depth worth to share with the world and seek to hide the fact by freely sharing what little we do have. Perhaps purity of heart and cleanliness of hand grow as the inverse of the amount of real knowledge which is available for sharing.

Why the great emphasis upon BASIC?

One would think that most microcomputers are hard wired for it. Have we no alternative ideas to explore? Is the fixation upon BASIC a result of its inherent superiority as a programming language for microcomputers, or is it due to a lack of knowledge concerning possible alternatives? The set of “possible alternatives” is not the universe of FORTRAN. We do not need FORTRAN. We do not need miniaturized versions of languages originally designed for implementation upon much more sophisticated hardware. We need languages based upon the material at hand: 8 bit and 16 bit computers with relatively primitive instruction sets.

While the Micro-Soft venture into APL represents a noble undertaking, it nevertheless embodies the faulty reasoning that, “If a language implemented on big computers is a good thing, then its implementation on a microcomputer must be equally good.” In a 64K system, what percentage of memory is required to support APL? What is left over for the user?

The microcomputer needs software which is conceived of in terms of the microprocessor. We, the amateurs, need to get up off our hind legs and do some thinking about language concepts.

In the future, is the personal microcomputer to be regarded as an aid for learning, or is it to be a toy in front of which the owner sits, glassy-eyed, playing the latest game written (by someone else) in BASIC?

Jack Cluff
34-57 73rd St
Jackson Heights NY 11372

What Are Riches For, If Not to Enjoy and Prosper By?

Regarding the BYTE magazine content, I seem to hear comments from all directions concerning too much of this, too much of that, not enough of this. Some say too much hardware, some say too much software - so the conclusion can only be that the balance is a good approximation of what is needed.

No one ever seriously should think of FORTRAN or BASIC as the be-all and end-all of high level languages. We do have demonstrated interest and familiarity with the part of users with these languages due to their widespread past use. But, if you want to write an interpreter or compiler, starting with an existing language is a shortcut (quite independent of marketing considerations) that bypasses the need to generate an unambiguous grammar and precise semantics of a new language from scratch.

Of course, the ultimate is to use an already established language for which you have an interpreter or compiler. The fact that I can use my BASIC for an occasional recreation or game by no means prohibits my use of that language for such "unnatural" things as systems programming. I know of at least one 8080 assembler written in BASIC, running on an Altair 8800 floppy disk system at a local college.

Another point worth closing on: Most professionals are really amateurs who have cleverly arranged things so that they get paid for what they like to do. The spirit of involvement with the activity, doing it "right" by some standard, is what counts.

1001 TALES OF ARABIAN BYTES

Thank you for a very interesting magazine. I am a recent subscriber having just received issues January and February 1977, forwarded to me from Texas. I am unwilling to wait for boat mail. When subscriptions are available by air mail to Saudi Arabia I will enter a subscription immediately regardless of the mail rates.

I have a small system running and am interested in all articles about Altair bus oriented products and 8080 software.

John R Fogle Jr
Box 403, c/o Aramco
Abqaiq SAUDI ARABIA

See the subscription card for foreign air mail rates.

John L. Hilburn and Paul M. Julich — both of Louisiana State University

Ideal reference for anyone involved with the design, use, or maintenance of systems using microcomputers, including those who have not had a formal course in digital logic.


1976 368 pp. Cloth $16.50

MICROPROCESSOR SYSTEMS DESIGN
Edwin E. Klingman — Cybernetic Micro Systems, Palo Alto, California

This landmark book in microprocessor systems design covers the spectrum from gate-level hardware all the way to language translation and interpretation.


1977 480 pp. Cloth $17.50

For further information, or to order a copy of either of these vital texts, please write to: Ben M. Cott, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632.
APL
Character Sets

The letter from Roderick Montgomery concerning the availability of a special character set for APL started some thoughts on the subject.

Lacking the availability of a PROM type character generator, there are other ways to do the job. The methods that come to mind are building up a generator out of standard bipolar ROMs, or using ordinary programmable memory parts which would be loaded by the computer at APL initialization time. The latter idea is especially appealing to me, since it would allow special characters to be carried into the machine as part of the program. Although this would require some programming, it would not use up memory space because it would be read directly into the character generator.

Charles J Billwiller
2313B Sierra Madre Ct
Rancho Cordova CA 95670

The use of reprogrammable character generators for television displays is starting to occur in products. One of the first products seen in advertising with this feature is the ECD Corporation's computer product, which has just such a user memory definable character generator. This technique has also been used by several video game manufacturers to achieve high resolution regions for graphics on a TV raster.

Is This a Valid Hot Board Placement Procedure?

A reader has suggested the following procedure to allow one to plug in and remove a circuit board while system power is maintained. We'd appreciate comments from some of the hardware oriented readers in the audience about the safety of this algorithm:

Removal of board:
0. Halt processor to avoid software carnage.
1. Connect jumpers from main supplies to board terminals to retain power during removal.
2. Remove board with care about alignment.
3. Remove jumpers to power down the board.

Insertion of a board:
0. Power up the board with jumpers from main supplies.
1. Insert board carefully to ensure proper alignment of edge connector.
2. Remove jumpers.
3. Restart processor.

The use of a parallel supply connection to the board during removal and insertion operations guarantees that the integrated circuits will be powered up before any low impedance driving voltages reach the inputs of onboard gates. A gate can be blown if voltages are applied to its inputs prior to the supply voltages. The common ground of the supply also eliminates static voltage differences between the two boards.

I suppose everyone who writes assembly language programs which eventually work becomes convinced he or she is the World's Greatest Programmer. At least I do. Luckily, there are books like Scelbi's "6800" Software Gourmet Guide & Cook Book to remind us how big the world really is. Author Robert Findley and associates at Scelbi have collected here a number of programming tricks which will interest almost any 6800 user.

Following a description of the 6800 instruction set and some general techniques, the author discusses conversion, floating point and decimal arithmetic, input and output operations, search and sort routines. Want an edit program? Many of the pieces are there: memory clearing, transfer of a section of memory, ideas for search routines. Need to process interrupts? The IO chapter has a discussion of interrupt processing which will be useful to anyone using the Motorola MIKBUG operating system read only memory. A chapter on conversion routines contains software for ASCII to
That's right, if everybody who read this ad for our BASIC SOFTWARE VOLUME 1 album would order one, we'd be sold out! Includes lots 'n' lots of your favorite Basic programs such as LUNAR LANDER and BLASTOFF!, plus new ones like FOURIER, FIT and AMPLE ANNIE. Plays through your Tarbell, Kansas City cassette interface (we coded all three ways). Or make cassette copies. Only 6 bucks. Don't be a dummy, order today! Satisfaction guaranteed or money back.

$6 to: SOFTWARE RECORDS
PO BOX 8401-B
UNIVERSAL CITY, CA 91508

(Calendar residents please add 10% sales tax)

Baudot, as well as the more usual BCD to and from binary.

For me, the best was chapter 5, "Floating Point Routines." Although (as the author notes) there are places where the code could be shortened, it is not that bad: a complete add, subtract, multiply and divide package (with conversions to ASCII, and from ASCII to floating point) in about 1.5 K bytes of relocatable code. The entire package is assembled (the only assembly in the book) with a hexadecimal listing in an appendix.

One reason the code is so short is the extensive use made of the index register. The stack, on the other hand, is not used at all (except for the automatic usage implied by each subroutine linkage). Since 6800 indexed addressing is relatively slow, do not expect the floating point package to be fast. Yet, it's not bad: Floating point multiplication, for example, takes a little more than 800 cycles, not quite 2 ms on a SWTPC 6800 system.

One curious feature of the 4 byte floating point word format used here is that the words are stored upside down, that is, if the least significant byte of the mantissa is stored at word N, then the rest is stored at N+1 and N+2 (with the sign being the most significant bit of byte N+2). The two's complement power of two exponent is stored at N+3. This is the wrong order for the 6800 for the following reason: One (tricky) way to increment a 2 byte word is to transfer it to the index register and increment that. The author knows this trick (as shown on pages 3 to 5), but apparently does not know that the index register load instruction LDX transfers the 16 bit contents at memory location M so that the least significant byte of the mantissa is stored at memory location M+1 and the least significant half contains what's in M+2. Thus, the least significant byte needs to have a greater address. The division program on page 5 to 19, for instance, might be shortened and speeded up by using this trick; to do so, however, would require storing the mantissa bytes in the opposite order, and rewriting the program.

The book will be of most use to programmers who employ an assembler program, since most of the ideas are presented in symbolic source language form. (The major exception is the floating point package.) The book is well written and, considering the diversity of topics, well organized. As a source of ideas, it is inexpensive at $10.

Jack Bryant
Mathematics Dept
Texas A & M University
College Station TX 77843
Dr Dobb's Journal of Computer Calisthenics and Orthodontia. Published by People's Computer Company, Box 310, Menlo Park CA 94025. Single issue, $1.50; one year (10 issues), $10.

Published by the people who brought you the Computer Faire, this toothsome periodical serves up an amalgam of filling software listings and highly spiced commentary to its monthly readership. Dr Dobb's Journal (Dr Dobb is disguised as editor Jim Warren) serves as a clearinghouse for new software information and the publication of such royalty-free programs as BASIC interpreters for use by readers. Because it is a nonprofit journal, DDI accepts no advertising but does print new product information as a service to readers. Editor Warren has attempted to bridge the gap between computer professionals and amateurs, and does not eschew the occasional biting remark in doing so. In Robert Benchley's immortal phrase, the tooth, the whole tooth and nothing but the tooth about Dr Dobb's Journal of Computer Calisthenics and Orthodontia is that it can help make your computer an indented servant. That about caps off this review...CM

M6800 ADVANCED SYSTEMS SOFTWARE RT/68©

RT/68® is mask-programmed on a 6830 ROM that replaces the Mikbug* ROM in your SWTPC 6800, Motorola Evaluation Module, etc. It is a powerful real time, multiprogramming operating system with many versatile system functions. RT/68® can support up to 16 concurrent programs at 8 priority levels. MICROWARE has improved the Mikbug® functions, added four more (Dump, Exec, Sys, Bkpt) and made tape load and punch program-usable. RT/68® is designed so programs that use Mikbug® I/O don't require changes.

- POWERFUL REAL TIME EXECUTIVE
- 100% SOFTWARE AND HARDWARE COMPATIBLE
- IMPROVED CONSOLE MONITOR
- EXPANDED PROGRAM DEBUGGING FEATURES
- SUPPORTS AUDIO CASSETTE I/O
- COMPREHENSIVE MANUAL INCLUDES SOURCE LISTING
- IMMEDIATE DELIVERY

ORDER RT68MR .......................... $55.00 ppd.

THE MICROWARE CORPORATION
P.O. BOX 954 Des Moines, Iowa 50304
Write or use reader service card for free brochure.
Phone Orders (515) 279-9856
BankAmericard and Mastercharge give all info on card.
Mikbug® is a trademark of Motorola, Inc.
**Confused About Printers?**

**MPI HAS YOUR ANSWER!**

<table>
<thead>
<tr>
<th>TTY REPLACEMENT?</th>
<th>THE SSP-40</th>
<th>$575</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW COST BUSINESS SYSTEM?</td>
<td>THE MP-40</td>
<td>$425</td>
</tr>
<tr>
<td>MINIMUM COST FOR HOBBIISTS?</td>
<td>THE KP-40 KIT</td>
<td>$179</td>
</tr>
</tbody>
</table>

Send for free literature

Bom 23101
Salt Lake City, Utah 84122
(801) 566-0201

---

**Fundamentals and Applications of Digital Logic Circuits**

Sol Libes

Library and say, "This book gave me my money's worth," I would have to say that the book would be *Fundamentals and Applications of Digital Logic Circuits* by Sol Libes. Anyone who is just starting to learn about digital electronics or computers should make it a point to study this book. Mr. Libes has taken a multitude of related subjects and blended them into a text that is easy to understand and just as easy to follow. In addition to the text, the book provides review questions at the end of each chapter and problems related to the text for the reader to solve.

Rather than jumping right into digital logic, the author starts out with the principles of semiconductors to show, in detail, how they function. Then, after this short course on semiconductors in the first chapter, the second chapter familiarizes the reader with binary numbers and coding systems. This particular chapter also shows how to convert numbers from one system to another and touches on the Gray code. These first two chapters establish the foundation for a great deal of the information the reader will receive from the remainder of the book.

During the next four chapters the reader learns the basic fundamentals of logic gates, flip-flops, counters and registers, and arithmetic logic circuits. Throughout the book each fundamental is illustrated and discussed in great detail. Through these same well done illustrations the reader next learns about various pulse sources and clock systems used in computers. Further, the author goes on to explain such devices as read only memories, as well as input and output equipment for computers.

Continuing with this stage-by-stage progression, the author then enlightens readers about the circuitry and theory of digital to
analog conversion. This tenth chapter also deals with multiplexing and digitally controlled analog devices. Finally, in the last two chapters of his book, Mr. Libes shows the reader more of the applications aspect of digital logic circuits. These two chapters explain the different types of circuitry used in digital voltmeters, multimeters, calculators and computers (the latter being a Digital Equipment PDP-8/E).

Throughout the book the reader is taken in step-by-step fashion from the basics of transistor workings to circuit applications in working digital machines. As a result, this book is one from which the beginner will be able to learn the fundamentals and build on them afterwards.

Michael P. Reardon
17 Earl Ln
Rothsville, PA 17673

ACM Pacific '77 Conference

Small computers, from programmable handhelds through minicomputers and micro networks, are the subject of technical papers sought for this year's ACM PACIFIC '77 conference of the Association for Computing's Pacific Region chapters. The San Francisco Bay Area meeting will be held at San Jose's LeBaron Hotel July 28 to 29, 1977.

"Exploring the Small Computer" is the theme of the conference which is expected to range in coverage from personal computing through small business applications and from computer parts and peripherals through bullet proof software. Papers will be presented on new developments in software and hardware in these areas and on minilanguage processors, minicomputer design, microcomputer systems, packaging of software for sale, portable microcomputers and software engineering "in the small." Papers on trends in software and current application, on the future of minis, and on related small computer topics are also to be presented.

Peter Szego, Ampex Corporation, is General Chairman of this year's regional conference, which is jointly co-sponsored by the Association's Pacific Region, Peninsula and Golden Gate Chapters. Informal symposia, workshops and invited papers are expected to be scheduled for the meeting in addition to the technical paper sessions. Special conference feature will be an evening "hobby computer" session, to be arranged by Jim Warren, editor of the home computer users magazine, Dobb's Journal of Computer Calisthenics & Orthodontics.

BYTE's Bits

21 START-AT-HOME COMPUTER BUSINESSES

Michael P. Reardon
17 Earl Ln
Rothsville, PA 17673

ACS Pacific '77 Conference

Small computers, from programmable handhelds through mini and micro networks, are the subject of technical papers sought for this year's ACM PACIFIC '77 conference of the Association for Computing's Pacific Region chapters. The San Francisco Bay Area meeting will be held at San Jose's LeBaron Hotel July 28 to 29, 1977.

"Exploring the Small Computer" is the theme of the conference which is expected to range in coverage from personal computing through small business applications and from computer parts and peripherals through bullet proof software. Papers will be presented on new developments in software and hardware in these areas and on minilanguage processors, minicomputer design, microcomputer systems, packaging of software for sale, portable microcomputers and software engineering "in the small." Papers on trends in software and current application, on the future of minis, and on related small computer topics are also to be presented.

Peter Szego, Ampex Corporation, is General Chairman of this year's regional conference, which is jointly co-sponsored by the Association's Pacific Region, Peninsula and Golden Gate Chapters. Informal symposia, workshops and invited papers are expected to be scheduled for the meeting in addition to the technical paper sessions. Special conference feature will be an evening "hobby computer" session, to be arranged by Jim Warren, editor of the home computer users magazine, Dobb's Journal of Computer Calisthenics & Orthodontics.

DATASEARCH

T30 Waukegan Road • Suite 100
Deerfield, Illinois 60015

Rush copies of "The Shoestring Star At-Home Computer Business Handbook to me right away -
NAME/COMPANY
ADDRESS
CITY/STATE/ZIP
CHECK ENCLOSED [ ] BANKAMERICAN [ ] MASTERCARD

PROCAM

by XYBEK

An extraordinary 2k memory board for your Altair-bussed computer

- On-board 1702A PROM programmer
- 256 bytes of RAM plus space for 1792 bytes of read-only memory (seven 1702A EPROMs)
- Supplied with one 1702A, pre-programmed with stand-alone programming software – no sense switches are used
- Supplied with programming power supply
- PRAMMER's own on-board clock makes it compatible with almost any Altair-bussed system
- All read and write sequences are generated via an on-board micro-programmed state machine, thus eliminating all one shots
- Complete 1702A programming in 18 seconds
- Includes complete listings for PRAMSYS, an eleven-function development system.

COMPLETE KIT: $209
Assembled and tested: $289

Immediate (off-the-shelf) delivery
California residents please add sales tax.
COD, Master Charge, BankAmerican and Visa accepted.

XYBEK • P.O. Box 4925 • Stanford, CA 94305
Telephone: (408) 296-8188
Dr Welles’ Economy Floppy Disk Drivers:
Machine Readable Object Code

Last month, we published Kenneth Welles’ article on the driver software for his Economy Floppy Disk. This issue, we continue that article with the machine readable bar code representation of the object code for the programs. These programs were presented as complete symbolic assemblies of the programs on pages 92-96 as “listing 1” of Dr Welles’ article in June 1977 BYTE. The materials presented here are:

Figure 1: Bar code representation of the object code. This block of machine readable bar codes contains information in the following format in each line of bars, which we call a “frame”:

- Sync character, binary 10010110
- Frame checksum, 8 bit summation of all remaining bytes in the frame, ignoring carry out of the high order bit.
- Relative frame identification, 8 bit ascending integer enumerating frames in the block of bars printed here.
- Length of frame, 8 bit number, “n”, giving the number of data bytes to follow.
- Data of frame, a total of “n” bytes as follows:
  - High order byte of 2 byte data address field.
  - Low order byte of 2 byte data address field.

The bit level format of the data is as follows: Each bit is defined by a bar and its following space. If the bar width is equal to the following space width, the data is a 0 bit; if the bar width is three times the following space width, the data is a 1 bit.

And a Bar Code Bug...

Thanks to the efforts of Ken Budnick, we found one fairly serious error in the generation of the texts printed in the March and May BYTEs. Ken has implemented detailed scanning programs for some microprocessors and one antiquated second generation machine and has tested them with a bar code reader design by Fred Merkowitz. The bug, which Walter Banks has since corrected in the phototypesetter driver programs he used to create the texts, is as follows: the checksum field of the bar code frames printed in those two issues omitted the frame identification byte and the data length byte from the calculation. Thus, to verify the checksum actually generated in the object text of the Tiny Assembler and the Bar Code Contest string, only the data field should be used to calculate this sum. The present bar code text generated for Dr Welles’ floppy disk driver routines calculates the checksum properly, and future examples will reflect this fix as well.
BYTE's Bits

National Student Programming Contest

The First Annual National ACM/UPR Student Programming Championships Contest took place February 2, 1977 in connection with the Computer Science Conference in Atlanta, GA. The contest was sponsored jointly by the Committee on Student Chapters and Memberships and Upsilon Pi Epsilon (National Computer Science Honor Society). Teams participated from: Catawba College, Chattanooga State Technical Community College, Clemson University, Georgia Institute of Technology, Georgia State University, Louisiana Technical University, Manhattan College, Michigan State University, North Carolina State University, Purdue University, Taylor University, Texas A & M University, University of Georgia at Athens, University of Missouri at Rolla, University of New Mexico at Albuquerque, and the University of Wisconsin at Platteville. The contest was conducted over a seven hour period from 5 PM to midnight on February 2. The facilities were provided by the Computer Center at Georgia Institute of Technology and the department of Information and Computer Science. Machine time was provided by Control Data Corporation on the Georgia Tech Cyber 74 computer. The teams were given four problems to solve using ANSI FORTRAN. These problems consisted of a Conversion from Roman to Arabic Numerals, A Character Manipulation for Rearranging of Names and Titles, the Determination of Amicable Numbers, and the Automatic Scoring of the Game of Bowling. The winners were determined by penalty points and the elapsed time taken for each problem. Although many teams were close on several solutions, only four teams completed three of the problems.

The National Champion Team is Michigan State University, who also won their regional competition. Second place went to Purdue University, third place to the University of Missouri at Rolla, and the fourth place to Georgia Tech. Trophies and certificates were presented to the participants at an Awards Banquet held February 3 at a luncheon in the Marriott Hotel. The participating teams also received free registration to the Computer Science Conference and attended two nights of informal social sponsored by the Committee on Student Chapters and Memberships.

Plans have been made to make this an annual event in connection with the Computer Science Conference. In 1978 it will take place on February 22 at the Plaza Hotel in Detroit, MI. Regional contests will be scheduled for the fall of 1977 to qualify teams to compete in the National Contest. Teams and sponsors for these qualification regional contests are being solicited from all regions within ACM. If you are interested in participating or holding such a contest, please contact Dr. Richard Newman, Academic Computing Services, Southern Illinois University, Carbondale IL 62901, (618) 536-2323.

Attention Educators with a Message

The National Association of Computer Applications to Learning (NAUCAL) will hold its 1977 annual convention in Dearborn MI on November 2-5, 1977. The convention will focus on educational computing, simulations in education, instructional materials and teaching strategies. Sessions that describe and illustrate computer applications in learning will be given special consideration.

Individuals who would like to present or who would like to suggest others who could present may write to John S-Camp and Mary Smith, Conference Cochairmen, Wayne County Intermediate School District, 33500 Van Born Rd, Wayne MI 48185.

Repairing Torn Paper Tapes

Translucent paper tapes are an inexpensive, efficient way of permanently storing information for a computer. However, after pulling the tape through a hand held paper tape reader a dozen or so times the tape tends to get pretty ragged. Sooner or later the tape will tear and have to be repaired before it can be used again. I have found that transparent Mylar movie film splicing tape is perfect for the repair job. The splicing tape is clear, strong and will not crack or yellow with age. It is also quite thin and will not interfere with the passage of the paper tape through the reader. The tape sticks very well to paper but if necessary can be removed without causing any damage. Mylar movie film splicing tape is available at any camera store.

Richard E Hetherington
637D Pendleton Av
Chicopee MA 01020

Circle 65 on inquiry card.
Saturday night Banquet of Filet Mignon $17.50
Over 150 Exhibits and 60 hours of seminars
Conferences being planned on: Clubs; Newsletters; Energy Conservation; Medical Applications; and more. Seminars being planned on: all aspects of Computing; Music; Basic Programming; Games; Education; Various Microprocessor Evaluations; Synthesis; S-100 Bus; and more.

**Personal Computing College™**
To be held at the Shelburne Hotel.

**Wednesday-Thursday-Friday, August 24-25-26**
3 Day Microcomputer Interfacing Workshop by TYCHON Inc.

A hands on computer workshop for the participants where they will spend almost 50% of their time working on well documented interfacing and software experiments. Students deal with the microcomputer at the bus level, interfacing the computer using solderless breadboarding techniques and assembly language programs.

Presented by Jonathan A. Titus and Dr. Christopher A. Titus, Authors of the famous "BUGBOOKS."

For reservation information and complete brochure contact: TYCHON Inc. c/o Shortess-Rawson, Dept PC77, P.O. Box 2203, South Hackensack, N.J. 07606.

**Friday, August 26**
6 Hour seminar by Dr. Adam Osborne
"Microprocessors — Where they came from and where they are going. An analysis of all products on the market today."

Dr. Adam Osborne will present a 6 hour seminar covering topics such as: Bringing order out of chaos; All microprocessors are not equal — each serves one market better than the other; Identifying those markets best suited to each microprocessor; Real sales volume anticipated for 1977; Comparisons including: 8085 vs. 280, 8048 vs. F6, etc.

For reservation information and complete brochure contact:

PC '77 Weekend admission at the door will be $10.00. Register before August 10th and SAVE 20% and AVOID WAITING IN LINE! Admission includes exhibits and seminars for both days, August 27-28th.

Please rush advance registration tickets for August 27-28th @ $8.00 each.

Please rush Banquet tickets for Saturday, August 28th @ $17.50 each.

Send reduced Hotel rate information.
Make all checks payable to "PERSONAL COMPUTING 77" and mail to:
PC77, Route 1, Box 242, Mays Landing, N. J. 08330.

Name __________________________ Amount enclosed __________________________
Address __________________________ __________________________
City __________________________ State __________________________ Zip __________________________

---

159
Microcomputer Handbook
by
Charles J Sippi

We often hear people say, "I'd like to find out more about microcomputers. I'm not a technical type, but I feel I can handle a fairly rigorous approach as long as it's well written. What's available?" Charles J Sippi's Microcomputer Handbook is one answer to this question. The book covers the present state of computer technology very well, concentrating on both hardware and software. Lucid and complete glossaries are combined with a variety of illustrations. Topics covered include: Microcomputers: where they are, what they are doing, and what is next; kits; distributed intelligence; and why the new systems are easier to use. The book was written by a computer industry lecturer and consultant and is highly recommended for the intelligent layman as well as for professionals and experimenters. The glossaries alone are worth the price of the book--don't miss this one! This hardcover reference is only $19.95, plus 50 cents postage.

Send now to:
BITS Inc
70 Main Street
Peterborough NH 03458

Please allow six weeks for delivery.

BUILD THE CRATE!

THE S-100 CARDFRAME THAT YOU CAN BUILD FOR LESS THAN $150
COMPLETE PLANS, INFO ON TOOLS, PARTS, SOURCES, ETC.

ORDER THE CRATE BOOK ........................................ $19.95

ENCOUNTER!
FOR GAME LOVERS, REALTIME GAMEBOARD WARFARE ON A VDM.
NO TAKING TURNS, JUST ACTION, STRATEGY, AND LOGISTICS.

8080 CODE ON P-TAPE + SOURCE, RULES ............... $16.95

USE SYS 8/SOFT PKG 1?
AUTO-LINE ENTRY, STRING EDIT, AUTO TABS, OCTAL, NBS.
MULTI-SECTION ASSEMBLIES, MORE ADDED BY OUR PROGRAM.

SOURCE ............................................................... $14.95

ADD $5 FOR P-TAPE, $10 FOR TARBEll TAPE

OBJECTIVE DESIGN, INC.
P.O. BOX 20325, TALLAHASSEE, FL. 32304

Clubs and Newsletters

Conducted by
Peter Travisano

Arizona Computer Society
The Arizona Computer Society meets regularly on the second Tuesday of each month at De Vry Institute, 4702 N 24th St, Phoenix AZ 85063.

A New Club in Boston
With all the computer activity in the Boston area this should be welcome news. A new group called the Boston Computer Club is forming for people with both hobbyist and industrial interests. As of now the club provides access to a timesharing PDP-8 and an Altair 8800a. For further information contact Jonathan Rotenberg, 17 Chestnut St, Boston MA 02108.

DUMPS in Delaware
A group of hobbyists at the University of Delaware recently formed the Delaware Users of Microcomputer Systems or DUMPS. The club is informal and the membership has a wide range of interests. Meetings are held on the third Monday of the month at 7:30 at New Central School, Academy St, Newark DE 19711.

Long Island NY—LICUS
The Long Island computer group is a growing concern made up primarily of high school and college students. Meetings are usually held on second Tuesdays at Commach High School South. To find out more write LICUS, POB 322, East Northport NY 11731.

State College PA
The Nittany Amateur Radio Club of State College PA sponsored a one day microcomputer seminar, an introduction to interfacing, programming and applications with an emphasis on real world situations. The
PERSONAL COMPUTING EXPO COMES TO NEW YORK FOR BIG BUSINESS

It's a brand new show in the world's biggest economic center specifically for manufacturers and buyers who are into personal computing. For the first time, this booming field will have a New York Coliseum showcase in the major population center in the east. It is planned as the largest public show of its type in the world that will attract enthusiastic buyers from a multi-state area.

WHY NEW YORK?

New York is the economic nerve center of the world. It also is the world's communications focal point, the one place that will put personal computing in a significant spotlight. New York is surrounded in depth by people who work in the computer field, by computer learning centers, universities, personal computing clubs, and thousands of others whose lives are affected by computers.

From this vast potential, Personal Computing Expo will draw the hard-core hobbyist, the interested student, and, because of a highly-publicized program of introductory seminars, those who are attracted and fascinated by computing but have not had exposure to the ways and means of becoming personally involved.

SHOW MANAGEMENT

Personal Computing Expo is being produced by H.A. Bruno & Associates, Inc., a firm in the exposition and promotion fields since 1923. Highly skilled in the production and promotion of consumer and trade shows, the company currently promotes the American Energy Expo, the National Boat Show, Auto Expo/New York. Promotion assistance also is currently rendered to the National Computer Conference and the Triennial IFIPS Congress in Toronto.

The show producer has promoted successful shows in the New York Coliseum every year since the building opened in 1957. Staff personnel are thoroughly familiar with the building, its services, management and labor.

EXCITING SEMINARS FROM "BYTE" MAGAZINE

Personal Computing Expo is endorsed by "Byte" magazine, whose staff is developing an exciting series of seminars and lectures for the exposition.

Visitors to the show will be able to attend these meetings free of charge. They will hear from lecturers such as Louis E. Frenzel and Carl L. Holder. More importantly, visitors will be able to attend meetings aimed at their proficiency levels, from beginner through intermediate and advanced personal computing.

FOR DETAILED INFORMATION CONTACT:
RAF II ANUZZI, Show Manager
H.A. BRUNO & ASSOCIATES, INC.
78 E. 56th Street
New York, N.Y. 10022
(212) 753-4920

Endorsed by BYTE Magazine

OCTOBER 28, 29, 30, 1977

PCE PERSONAL COMPUTING EXPO • NEW YORK COLISEUM
The 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, 4 Extra Control Lines
- 37-page manual included
- Device Code Selectable by DIP-switch
- Capable ofGenerating Kansas City tapes also
- No modification required on audio cassette recorder
- Complete kit $120, Assembled $175, Manual $4

TARBELL ELECTRONICS
20620 S. Leapwood Ave., Suite P, Carson, Ca. 90746
(213) 538-4251
California residents please add 6% sales tax
*ALTAIR is a trademark/tradename of MITS, INC.

Goodyear Computer Club

The Goodyear Computer Club in Akron OH is off to an auspicious start: just a few months old and able to boast a membership of more than 150. Generally the members are associated with the Goodyear Aerospace Corporation or the Goodyear Tire and Rubber Company. Contact the Goodyear Computer Club c/o J F Derry, D-109 Plt 1, The Goodyear Tire and Rubber Co, Akron OH 44316.

Aloha Computer Club

Don Henson, president of the Aloha Computer Club of Millani Town HI, has made an interesting offer. He’s willing to give monthly seminars limited to six beginners. Sounds like the best possible way to share computer information and get more of...
the many, many interested people involved in personal computing. Aloha meets on the first Wednesday of the month at 7:30 at the Waikiki-Kapahulu Library, 400 Kapahulu Av. Seminars are held at various times throughout the month. New members are more than welcome. To learn more write 94-360 Hokuula St, Apt 187, Millani Town HI 96789 or call (808) 623-1781.

Central Florida Computer Club
A new hobbyist group has been formed in Orlando. Jim Walton is the contact person for the Central Florida Computer Club. He can be reached c/o Data Entry Engineering, 1810 N Orange Av, Orlando FL 32804, (305) 896-4322.

Electronotes – Newsletter of the Musical Engineering Group
Those people interested in both electronic music and instrument construction will find Electronotes most valuable. An in-depth presentation focusing on the ENS-76 is planned for the next several issues. Electronotes is published by B A Hutchins, 203 Snyder Hill Rd, Ithaca NY 14850. Routine orders should be sent c/o Electronotes, 213 Dryden Rd, Ithaca NY 14850.

Pittsburgh Area Computer Club
The place to share computer lore in Western Pennsylvania is the Pittsburgh Area Computer Club. Contact PACC at 400 Smithfield St, Pittsburgh PA 15222.
A New Orleans Computerfest

The Jefferson Amateur Radio Club and the Crescent City Computer Club would like to announce the New Orleans Hamfest/Computerfest which will be held at the Hilton Inn in Kenner LA (directly across from the New Orleans International Airport) September 24 and 25. This is the ARRL Delta Division Convention for 1977 and is the largest "ham" outing in the deep south.

This year's event will feature a banquet Saturday night with entertainment, two days of commercial exhibits, flea markets and forums. There will also be a hospitality room, ladies' events, FCC examinations and more.

This year's grand prize is a complete Drake "C-Line" ham station, and many door prizes will be awarded each day.

Information on tickets, room reservations, etc, will be furnished upon request by contacting the New Orleans Hamfest/Computerfest, POB 10111, Jefferson LA 70181.

Important Notice to Clubs

Hazeltine Corporation, Greenlawn NY 11740, has recently circulated a letter from Frank J Cirillo of their Industrial Products Division, announcing a special package price of $1400 for refurbished H-2000 model terminals with Hazeltine tape cassette drive. The letter is directed to computer clubs, so if your local club did not receive a copy, write or call Mr Cirillo at (516) 261-7000 and inquire about this excellent deal. This terminal is one of the world's largest selling video display terminals and has had application in universities, hospitals, business, finance and government. At a $1400 price you get a used commercial grade piece of equipment which will prove quite applicable to many home computer situations.

Would your club benefit from a write-up in BYTE's Clubs and Newsletters section? It's easy enough to arrange. Just drop a line to BYTE, 70 Main St, Peterborough NH 03458, c/o Clubs and Newsletters, with the information you'd like published. Naturally we can't cover every club every month but we do make a special effort to mention new clubs and those that correspond regularly.

Those people looking for a more complete listing of computer clubs should refer to January 1977 BYTE.
BASIC Timing Delay

When writing game programs it is often convenient to have a time delay to add some exciting real life feeling to the game. This subroutine, submitted by Gregory A Worth and written for a Motorola 6800, will cause delays between 1 and 255 seconds. Register B is loaded with the number of seconds you wish to delay before jumping to the subroutine. LOOP1 is the basic time delay loop. This loop takes 20 clock cycles and is executed 50000 times giving a basic timing unit of 1 second, assuming a 1 MHz processor clock is used. If your system is using a different clock speed, appropriate adjustment of either the number of repetitions of the loop or the amount of time consumed in one pass through the loop will have to be made. The outer loop, LOOP2, allows the varying time delays by repeating the 1 second loop the number of times specified in register B. All of the numbers in this listing are in hexadecimal unless otherwise specified.

<table>
<thead>
<tr>
<th>Address</th>
<th>Hex</th>
<th>Code</th>
<th>Labels</th>
<th>Op</th>
<th>Operand</th>
<th>Commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td>10200A</td>
<td>ERM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2102AL-FULL SPEED MEMORY (500ns)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neatly fits power consumption of even a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2102LA PRIME FROM NEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2.00 EACH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64 - $1.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Proko Paper Tape Reader

A fully TTL buffered optical tape reader for reading 8-level paper tape. It's compatible with nearly all parallel input ports and has both positive and negative strobe outputs. The power (-9V @ 20mA) is derived from your I/O board, just supply a light source (study lamp), grab the tape and pull.

PTR-8 Kit $54.00 Assembled $68.00

Now, add mass storage to your micro with the North Star floppy disk system. Comes with a controller with bootloader, and powerful extended basic. (Basic supports sequential and random access disk files.)

North Star Disk Kit: $1012

The S.D. Sales Co. Z-80 CPU Kit For 5-100 base $14900

For large orders please send money order or cashier's check to avoid delays in waiting for check to clear. Check or money order only, California residents add 6% tax. All orders postpaid in U.S. All devices tested prior to sale. Money back 30 day guarantee. Sorry we cannot accept returned ICs that have been soldered to $10 minimum order. Prices subject to change without notice.
THOUSANDS OF KITS & PERIPHERALS

ALL ASR33 is and MORE $785

COPE 1030 $895 = 70 lb SHIPPING

SEND $1 FOR OUR MICROCOMPUTER CATALOG

THE ONLY TRUE ONES IN THE WORLD ON FULL DETAILS ON OUR COMPLETE LINE OF KITS AND UNITS, REVIEWS OF OVER 100 BOOKS, LISTS OF KITS AND SUPPLIES AND ALL ABOUT WORKING MICROCOMPUTER - AN INTRODUCTION TO PERSONAL COMPUTING.

TO ORDER EQUIPMENT

1. EXACT CHECK FOR FULL PRICE PLUS SHIPMENT COSTS ($30 - $15 IF OVER $250) + 10% OFF

ON KITS + PERIPHERAL (S) OVER $250 10% OFF

SEND CHECK OR MONEY ORDER TO:

Maximun Discounts

KITS & ASSEMBLED UNITS

SALE UP TO 20% OFF KIT PRICE WHEN A PERIPHERAL IS PURCHASED AT THE SAME TIME ($300 MAXIMUM DISCOUNT)

PERIPHERAL(S) OVER $250 20% OFF KIT PRICE

PERIPHERAL(S) OVER $500 10% OFF KIT PRICE

PERIPHERAL(S) OVER $1000 5% OFF KIT PRICE

TRICO ELECTRONIC SUPPLIES

2695 W. LINCOLN AVE., ANAHEIM, CA 92801

PHONE 714/821-0234
SPECIAL OFFER
100 STEPS PROGRAMMABLE SCIENTIST
#4525
REG. $89.95
$29.99

- RPN logic with "built-in" hierarchy for increased accuracy and speed in calculating sequences involving arithmetic, trigonometric, logarithmic, power or exponential functions.
- A rollable 4-level stack lets you review or use intermediate solutions.
- Eight-digit plus 2-digit exponent LED display with full-floating decimal system.
- Scientific notation for increased mathematical capacity.
- Sine, cosine, tangent and inverse trigonometric functions.
- Common and natural logarithms and antilogarithms.
- Instant automatic calculation of powers and roots.
- Single-key square root calculations.
- Single-key Pi entry.
- Separate storage memory.
- Square, square root and reciprocal calculations.
- Change sign and register exchange keys.
- MOS/LSI solid-state circuitry.
- Includes 3 AA rechargeable NiCad batteries.

MANUFACTURED IN USA BY
BY NATIONAL SEMICONDUCTOR
ONE YEAR UNLIMITED WARRANTY
10 DAYS MONEY BACK GUARANTEE

ALSO AVAILABLE
MODEL #4520 With the same features as above but without programming capability @ $28.88. Accessories same price as above.

SAME DAY SERVICE IF PAID BY CASHIER'S CHECK, MONEY ORDER OR CHARGE CARD.

FROM CONCORD
THE FIRST FULL FEATURE
LSI DMM KIT
INTRODUCTORY PRICE:
$77.77

- AUTO RANGING
- AUTO POLARITY
- AUTO ZERO
- 3 Large Digits (1/2")
- Rechargeable

NI-CAD BATTERIES: $6.00 • AC CHARGER: $4.95 • ENCLOSURE: $12.95 • TEST LEADS: $1.95 • SHUNT KIT FOR 3 CURRENT RANGES: $4.75 • SOCKETS $2.50

ORDERING INFORMATION
SHIPPING AND HANDLING - $3.00 + 50¢ Insurance
California residents add 6% sales tax

ELECTRONICS WAREHOUSE Inc.
1603 AVIATION BLVD. Dept. B
REDONDO BEACH, CA. 90278
TEL. (213) 376-8005
WRITE FOR FREE CATALOG
You are invited to visit our store at the above address.
COMPUTERS: circle desired kits, then show this ad to your owner.

1. **Active Terminator Board** $25
   - Plugs into your S-100 bus to actively terminate lines
   - Minimize the crosstalk, noise, overshoot, and ringing you can find on improperly terminated busses

2. **MOTHERBOARD(S)**
   - 10 Slots: $85
   - 18 Slots: $118
   - Prices include edge connectors
   - Includes active terminations for unambiguous data transfer
   - Uses glass epoxy, plate-through, double sided board

3. **8K EconoRAM II™** $163.84
   - Low current drain, zero wait states (1 wait with Z-80)
   - Tri-state outputs mate with any type of bus
   - Configured as two separate 4K blocks (with fully independent protect and address decoding to increase flexibility)
   - Includes low power Schottky support ICs and full buffering

4. **4K EconoROM™** $265
   - NEW AND IMPROVED—better listings; SF and PSW now implemented for more flexibility.
   - Find out why Jay Bell, editor of Print-Out, said:
     "Without a doubt, this board is the best buy I have made for my Altair 8800 out of the $5000+ I've spent." (Print-Out, September 1976 issue)
   - Now you can have editor, assembler, and monitor routines that make the 8080 a much nicer chip to work with.

5. **CPU Power Supply** $45
   - 5 Volts @ 4 Amps with crowbar overvoltage protection
   - +12 Volts @ ½ Amp and -12 Volts @ ½ Amp
   - Adjustable bias supply, 5 to 10 Volts at 10 mA
   - Intended for use with small computer systems or as digital bench supply

**TERMS:** Please allow up to 5% for shipping; excess refunded. Californians add sales tax. Street address must be included for COD orders. Place BankAmericard® and Mastercharge® orders ($15 minimum) by calling our 24 hour order desk at (415) 562-0636.

**FREE FLYER:** Just give us the word and we'll send out a copy of our flyer, which describes our complete line of products in greater detail.

Circle 86 on inquiry card.
CONNECTIONS PRINTED CIRCUIT EDGE-CARD
Burr-Tec Connectors — FHS 64 to 679 P.C. Cards
15/30 PINS (Solid Eyellet) $1.96
15/16 PINS (Solid Eyellet) $2.24
22/44 PINS (Solid Eyellet) $2.85
50/100 PINS (Spacing/PINS (Solid Eyellet) $6.95
25 PINS SUBMINIATURE
D025P PLUG 3.25
D025S SOCKET 4.95

PARATRONICS

Logic Analyzer Kit

- Analyzes any type of digital system
- Checks data in excess of 6 million words per second
- Troubles tests TTL, CMOS, DTL, DIT, and 74 series unpackaged components
- Signs off TTL, CMOS, DTL, DTI, and 74 series unpackaged components
- Performs tests on TTL, CMOS, DTL, DTI, and 74 series unpackaged components
- Includes complete CMOS instruction manual

Special 149.95

BUGBROK

Continuing Educational Series

BUGBROK I — Basic concepts of TTL Logic — over 90 experiments $12.00 incl

BUGBROK II — Advanced concepts of TTL Logic — over 90 experiments $20.00 incl

BUGBROK III — Expert — 800 experiments — $30.00 incl

S85 TIMES SYSTEMS SUBSCRIBERS with EXPERIENCES — over 100 design techniques $6.50 each

MCM-W-DESIGNERS PRIMER AND HANDBOOK a complete CMOS instruction manual $24.95

SPECIAL 45.95

JOYSTICK

Price: $25.00

Includes: Joystick, joystick cable, controller, and user manual.

Special $19.95

3½ DIGIT DVM KIT

$29.95 ea.

ELEPHANT KITS

S2 A X 1

$29.95 ea.

PLUGBOARD

$6.50 ea.

1/16 VECTOR BOARD

$14.95 ea.

HEAT SINKS

195-50 Envelope Copper Heat Sink with Nickel Finish $2.25
195-60 Copper Heat Sink with Nickel Finish $2.75
60-75X Nickel Plated Aluminum Heat Sink $1.00

HEXADECIMAL ENCODER 19 KEY PAD

$10.95 each

63 KEY KEYBOARD

$19.95

OPEN TO ORDER IN PARALEL, KEYBOARD ENCODER

$19.95

CONTINENTAL SPECIALS

PHOTO BOARD 6 $15.95

Other CS Protoboards

PHOTO CLIPS

DESIGN MATES

Logic Monitor

SPECIALTY PROTOSTRIPS

MCM-C — Designer $6.50
MCM-E — Function Generator $6.50
MCM-C — Logic Monitor $11.95

SPECIAL $39.95

CIRCUIT EXHIBITION KITS

$19.95

DIGITAL STORAGE

DIGITAL CIRCUIT KIT — 3½ INCH DIGITS $17.95

DIGITAL CLOCK KIT $17.95

DIGITAL CIRCUIT KIT — 6 DIGIT $65.95

DIGITAL CIRCUIT KIT — 6 DIGIT $65.95

JE603 PROBE

$9.95 Per Kit

Circle 87 on inquiry card.
**computer display terminal**

This display terminal has an integral controller, B/W cathode ray tube and keyboard. The system has a serial I/O interface for communication and an I/O interface for a printer.

**DISPLAY (P/N 4802-1095-501) FEATURES:**
- 17" B/W CRT
- 41 lines of data
- 52 characters per line
- Characters are generated by a diode matrix "graphic" technique
- 21 special push-buttons wired for a program call up
- Brightness Control
- Self-contained power supply

**KEYBOARD (P/N 4802-1115-501) FEATURES:**
- Reed switch technology
- 54 data keys
- 28 special keys detachable with cable

**LOGIC UNIT (P/N 4802-1157-502) FEATURES:**
- 1024 by 6 bit core memory
- Printer I/O interface
- Communication I/O interface

**POWER:** 115V, 50/60 Hz, 500 Watts

**WEIGHT:** 210 lbs. (including logic unit, keyboard, display and cables.)

FOB LYNN MASS (you pay shipping)

Check with order please.

**SPECTRA FLAT TWIST**
50 conductor, 28 gauge, 7 strands/conductor made by Spectra. Two conductors are paired & twisted and the flat ribbon made up of 25 pairs to give total of 50 conductor. May be peeled off in pairs if desired. Made twisted to cut down on "cross talk." Ideal for sandwiching PC boards allowing flexibility and working on both sides of the boards. Cost originally $13.00/ft

<table>
<thead>
<tr>
<th>Footage</th>
<th>10'</th>
<th>50'</th>
<th>100'</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP-324-A</td>
<td>$1.00</td>
<td>$9.00</td>
<td></td>
</tr>
<tr>
<td>SP-234-A</td>
<td>$1.00</td>
<td>$9.00</td>
<td></td>
</tr>
<tr>
<td>SP-234-B</td>
<td>.90</td>
<td>.90</td>
<td></td>
</tr>
</tbody>
</table>

In tall TO-5 can

DPDT, 24 volts. Brand new. Cost $16.00 each

SP-134 $3.00 each 2/$5.00

**WIRE WRAP WIRE**

**TEFZEL blue #30**
Reg. price $13.28/100 ft. Our price 100 ft $2.00; 500 ft $7.50.

**WIRE WRAP WIRE**

**MULTI COLORED SPECTRA WIRE**

Footage 10' 50' 100'
8 Cond. #24 $2.50 9.00 15.00
12 " 22 3.00 11.00 18.00
14 " 22 3.50 13.00 21.00
29 " 22 7.50 28.00 45.00

Great savings as these are about 1/4 book prices. All fresh & new.

Precision 16 pin DIP network as shown. Each resistor 1K. For pull-up/pull-down interface networks. Value over $1.00 each; New, CTS or Beckman

SP-320 pack of 6 $1.00

Circle 18 on inquiry card.

FREE CATALOG SP-9 NOW READY
P.O. Box 62, E. Lynn, Massachusetts 01904
MICROCOMPUTING FOR HOME AND THE SMALL BUSINESSMAN

The professional publication bringing microcomputing technology to the hobbyist, small businessman, educator, engineer and student. Every issue edited to bring technology and people together in the simplest manner.

INTERFACE AGE is packed with

- HARDWARE ARTICLES — Product profiles, comparisons, applications, modifications and construction projects.
- SOFTWARE ARTICLES — Microcomputer development software, short software routines, application software, off-line software storage formats, software communication standards and access to the microcomputer software depository for all INTERFACE AGE readers.
- TUTORIALS — Fundamentals of micro processors, basics for microcomputing, professional to technical transition information, understanding software, elementary math for computing.
- NEW PRODUCTS — Manufacturer profiles and latest product releases.
- USER COMMUNICATION — UPDATE — devoted to club and organization announcements and activities. — LETTERS TO THE EDITOR — Reader forum for expression of opinions and feedback on articles and features.
- MICRO-MARKET/FIFO FLEA MARKET — Low cost/no cost advertising for the new marketeer/garage sale enthusiast.

If you need to know how to get started in microcomputing or need the valuable software once your system is completed then INTERFACE AGE is a must for you.

Don't Delay — Subscribe Today!

INTERFACE AGE MAGAZINE

P.O. Box 1234
Cerritos, CA. 90701

12 Monthly Issues: $10 U.S., $12 Can./Mex., $18 International

☐ CHECK  ☐ MONEY ORDER

Name_________________________ __________________________
Address__________________________________________________
City __________________________ State ______ Zip. ________
Country ___________________________________________________

MAKE CHECK PAYABLE TO: INTERFACE AGE MAGAZINE

Circle 89 on inquiry card.
NEW COMPUTER INTERFACE BOARD KIT

Our new computer kit allows you to interface serial TTL to RS 232 and RS 232 to TTL. There are four of these supplied with the kit, so you can run up to four devices on one TTL or four separate TTL to RS 232 devices.
Typical use: You can use your computer to run an RS 232 printer, video terminal and two other RS 232 devices at once, without constantly connecting and disconnecting your terminals.

**$49.00**

GENERAL PURPOSE COMPUTER POWER SUPPLY KIT

This power supply kit features a high frequency torroid transformer with switching transistors in order to save space and weight. 115V 50 cycle primary. The outputs with local regulators are 5V to 10A, in one amp increments, -5V at 1A, ±12V at 1A regulators supplied 6 340T-S supplied.

**$79.00**

UNIVERSAL 4K MEMORY BOARD KIT **$74.50**

This memory board may be used with the F8 and with minor modifications may be used with KIM-top. 32-2102-1 static RAM's, 16 address lines, 8 data lines in, 8 data lines out, all buffered. On-board decoding for any 4 of 64 pages, standard 44 pin, .156" buss.

FOR FAIRBUG 4K F8 BASIC ON PAPER TAPE **$25.00**

PRINTED CIRCUIT BOARD

- 4.1/2 x 6-1/2" SINGLE SIDED PRINTED CIRCUIT BOARD
- 3.1750 X 3.1750" MASKED TOOLS
- 7 WATT LD 106 LASER DIODE R, $5.95
- 9 PIN DIP SOCKETS, $1.75
- 105 PIN DIP SOCKETS, $2.50
- 140 PIN DIP SOCKETS, $3.50
- 170 PIN DIP SOCKETS, $4.50
- 200 PIN DIP SOCKETS, $5.50
- 250 PIN DIP SOCKETS, $6.50
- 300 PIN DIP SOCKETS, $7.50
- 350 PIN DIP SOCKETS, $8.50
- 400 PIN DIP SOCKETS, $9.50
- 450 PIN DIP SOCKETS, $10.50

TANTALUM CAPACITORS

- 1000N 0.1UF 16V, $10.00
- 1000N 0.2UF 16V, $10.00
- 100N 0.1UF 16V, $10.00
- 100N 0.2UF 16V, $10.00
- 10N 0.1UF 16V, $10.00
- 10N 0.2UF 16V, $10.00
- 1N 0.1UF 16V, $10.00
- 1N 0.2UF 16V, $10.00
- 10N 0.001UF 16V, $10.00
- 10N 0.002UF 16V, $10.00
- 10N 0.003UF 16V, $10.00
- 10N 0.004UF 16V, $10.00
- 10N 0.005UF 16V, $10.00
- 10N 0.006UF 16V, $10.00
- 10N 0.007UF 16V, $10.00
- 10N 0.008UF 16V, $10.00
- 10N 0.009UF 16V, $10.00
- 10N 0.01UF 16V, $10.00
- 10N 0.015UF 16V, $10.00
- 10N 0.02UF 16V, $10.00
- 10N 0.025UF 16V, $10.00
- 10N 0.03UF 16V, $10.00
- 10N 0.035UF 16V, $10.00
- 10N 0.04UF 16V, $10.00
- 10N 0.045UF 16V, $10.00
- 10N 0.05UF 16V, $10.00
- 10N 0.055UF 16V, $10.00
- 10N 0.06UF 16V, $10.00
- 10N 0.065UF 16V, $10.00
- 10N 0.07UF 16V, $10.00
- 10N 0.075UF 16V, $10.00
- 10N 0.08UF 16V, $10.00
- 10N 0.085UF 16V, $10.00
- 10N 0.09UF 16V, $10.00
- 10N 0.095UF 16V, $10.00
- 10N 0.1UF 16V, $10.00
- 10N 0.105UF 16V, $10.00
- 10N 0.11UF 16V, $10.00
- 10N 0.115UF 16V, $10.00
- 10N 0.12UF 16V, $10.00
- 10N 0.125UF 16V, $10.00
- 10N 0.13UF 16V, $10.00
- 10N 0.135UF 16V, $10.00
- 10N 0.14UF 16V, $10.00
- 10N 0.145UF 16V, $10.00
- 10N 0.15UF 16V, $10.00
- 10N 0.155UF 16V, $10.00
- 10N 0.16UF 16V, $10.00
- 10N 0.165UF 16V, $10.00
- 10N 0.17UF 16V, $10.00
- 10N 0.175UF 16V, $10.00
- 10N 0.18UF 16V, $10.00
- 10N 0.185UF 16V, $10.00
- 10N 0.19UF 16V, $10.00
- 10N 0.195UF 16V, $10.00
- 10N 0.2UF 16V, $10.00

F8 EVALUATION BOARD KIT WITH EXPANSION CAPABILITIES **$99.00**

A fantastic bargain for only with the following features:
- 20 ma or RS 232 interface
- 64K addressing range
- Program control timers
- 1k of on-board static memory
- Built in clock generator
- 64 Byte register
- Built-in priority interrupts
- Documentation
- Uses Fairbug F8U

SOLID STATE SALES
P.O. BOX 785
SOMERVILLE, MASS. 02143 TEL. (617) 347-4005
WE SHIP OVER 95% OF OUR ORDERS THE DAY WE RECEIVE THEM

Circle 90 on inquiry card.
PRESTRIpped WIRE WRAP WIRE

Highest quality 32 gauge, uncoated silver plated wire for wrapping. Stripped 1/4" on both ends. Indicated lengths are lengths of insulated portion. Packages 100 per sturdy plastic vessel or 1000 per poly bag. Compare our prices! Available in Black, Red, Yellow and Green. Staionary design.

Length: Price per 100
1" $1.48 (WV30VC-1)
5" $1.48 (WV30VC-1)
25" $1.48 (WV30VC-1)
100" $1.48 (WV30VC-1)

3 LEVEL WIRE WRAP

1-9 5-24 25-100

SKT-8080 8 pin
1402 8pin
1405 8pin
1800 8pin
2200 8pin
2402 8pin
4004 8pin

3000 8pin
4000 8pin
5000 8pin
6000 8pin

12" roll wire, same as above (30 or 14p... 16p • •••• • 3.01 1000 low profile)

1.64
1.76
1.87
1.99
2.11
2.23
2.36
2.50
2.64
2.76
2.89
3.02
3.15
3.28
3.41
3.54
3.67
3.80
3.93

4.55
4.64
4.74
4.83
4.93

100 MN MINICOMPUTER PC CONNECTORS

20X5 @ 1227 meeting, solder tail or warp terminals. By TI, PCC-1005 (soldier), $4.99 PCC-1005W, $4.94 PCC-1005V.

NEW BOOK

1977 VERSION OF NATIONAL'S CMOS

A greatly expanded new contains 500 pages of data on the 74C series as well as the popular 4000 series. Also has data and applications on the more complex CMOS products for which National has become famous.

CMOS DATA BOOK—7th Edition by National...$3.95 Spec.

PRESTRIpped WIRE WRAP WIRE

Highest quality 32 gauge, uncoated silver plated wire for wrapping. Stripped 1/4" on both ends. Indicated lengths are lengths of insulated portion. Packages 100 per sturdy plastic vessel or 1000 per poly bag. Compare our prices! Available in Black, Red, Yellow and Green. Staionary design.

Length: Price per 100
1" $1.48 (WV30VC-1)
5" $1.48 (WV30VC-1)
25" $1.48 (WV30VC-1)
100" $1.48 (WV30VC-1)

3 LEVEL WIRE WRAP

1-9 5-24 25-100

SKT-8080 8 pin
1402 8pin
1405 8pin
1800 8pin
2200 8pin
2402 8pin
4004 8pin

3000 8pin
4000 8pin
5000 8pin
6000 8pin

12" roll wire, same as above (30 or 14p... 16p • •••• • 3.01 1000 low profile)

1.64
1.76
1.87
1.99
2.11
2.23
2.36
2.50
2.64
2.76
2.89
3.02
3.15
3.28
3.41
3.54
3.67
3.80
3.93

4.55
4.64
4.74
4.83
4.93

100 MN MINICOMPUTER PC CONNECTORS

20X5 @ 1227 meeting, solder tail or warp terminals. By TI, PCC-1005 (soldier), $4.99 PCC-1005W, $4.94 PCC-1005V.

NEW BOOK

1977 VERSION OF NATIONAL'S CMOS

A greatly expanded new contains 500 pages of data on the 74C series as well as the popular 4000 series. Also has data and applications on the more complex CMOS products for which National has become famous.

CMOS DATA BOOK—7th Edition by National...$3.95 Spec.

PRESTRIpped WIRE WRAP WIRE

Highest quality 32 gauge, uncoated silver plated wire for wrapping. Stripped 1/4" on both ends. Indicated lengths are lengths of insulated portion. Packages 100 per sturdy plastic vessel or 1000 per poly bag. Compare our prices! Available in Black, Red, Yellow and Green. Staionary design.

Length: Price per 100
1" $1.48 (WV30VC-1)
5" $1.48 (WV30VC-1)
25" $1.48 (WV30VC-1)
100" $1.48 (WV30VC-1)

3 LEVEL WIRE WRAP

1-9 5-24 25-100

SKT-8080 8 pin
1402 8pin
1405 8pin
1800 8pin
2200 8pin
2402 8pin
4004 8pin

3000 8pin
4000 8pin
5000 8pin
6000 8pin

12" roll wire, same as above (30 or 14p... 16p • •••• • 3.01 1000 low profile)

1.64
1.76
1.87
1.99
2.11
2.23
2.36
2.50
2.64
2.76
2.89
3.02
3.15
3.28
3.41
3.54
3.67
3.80
3.93

4.55
4.64
4.74
4.83
4.93

100 MN MINICOMPUTER PC CONNECTORS

20X5 @ 1227 meeting, solder tail or warp terminals. By TI, PCC-1005 (soldier), $4.99 PCC-1005W, $4.94 PCC-1005V.
The April BOMB, Exploded...

There are risks associated with every action one can take, and occasionally the results fulfill the worst expectations. The April 1977 issue of BYTE bombarded content with respect to the inclusion of several (ie: four articles) on a fiction theme, and one intentionally outdated (as April Fool) microprocessor update on the 8008 (which became obsolete for new design with the introduction of the Intel 8080). The winner, by a large margin, was the article by Jack Emmerichs on the Tiny Assembler 6800, which came in the +2 σ point relative to the mean for 13 articles. Second place went to a semifiction story by Steve Garcia (countering the trend just cited) entitled "Having a Private Affair With Your Computer." Steve's article came in at the +1 σ point, just slightly ahead of a group of articles which clustered above the mean at approximately the same displacement. The remaining fiction articles (and numerous comments on the BOMB cards) clustered around the −1.5 σ point of the observed distribution. These statistics were calculated with a programmable calculator. The results should be taken with a grain of salt, since the standard deviation was 24% of the mean. Why did I put so many of a similar type of article in the April issue? It was, after all, the April Fool issue; and it provided a form of controlled experiment, by tweaking the content a little far in one direction while observing the results in the tally of BOMB cards. The experiment has been concluded for the time being... C Helmers.

<table>
<thead>
<tr>
<th>Inquiry No.</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>163</td>
</tr>
<tr>
<td>14</td>
<td>50</td>
</tr>
<tr>
<td>7</td>
<td>23</td>
</tr>
<tr>
<td>11</td>
<td>32</td>
</tr>
<tr>
<td>51</td>
<td>129, 135, 142-145, 160</td>
</tr>
<tr>
<td>71</td>
<td>163</td>
</tr>
<tr>
<td>41</td>
<td>131</td>
</tr>
<tr>
<td>72</td>
<td>163</td>
</tr>
<tr>
<td>73</td>
<td>163</td>
</tr>
<tr>
<td>56</td>
<td>151</td>
</tr>
<tr>
<td>4</td>
<td>6, 7</td>
</tr>
<tr>
<td>24</td>
<td>87</td>
</tr>
<tr>
<td>74</td>
<td>163</td>
</tr>
<tr>
<td>75</td>
<td>163</td>
</tr>
<tr>
<td>61</td>
<td>154</td>
</tr>
<tr>
<td>76</td>
<td>163</td>
</tr>
<tr>
<td>77</td>
<td>163</td>
</tr>
<tr>
<td>47</td>
<td>137</td>
</tr>
<tr>
<td>31</td>
<td>107</td>
</tr>
<tr>
<td>68</td>
<td>162</td>
</tr>
<tr>
<td>83</td>
<td>167</td>
</tr>
<tr>
<td>22</td>
<td>83</td>
</tr>
<tr>
<td>23</td>
<td>1, 2</td>
</tr>
<tr>
<td>39</td>
<td>128</td>
</tr>
<tr>
<td>63</td>
<td>165</td>
</tr>
<tr>
<td>34</td>
<td>133</td>
</tr>
<tr>
<td>21</td>
<td>77</td>
</tr>
<tr>
<td>33</td>
<td>111</td>
</tr>
<tr>
<td>92</td>
<td>111</td>
</tr>
<tr>
<td>78</td>
<td>164</td>
</tr>
<tr>
<td>95</td>
<td>153</td>
</tr>
<tr>
<td>85</td>
<td>169</td>
</tr>
<tr>
<td>29</td>
<td>164</td>
</tr>
<tr>
<td>27</td>
<td>99</td>
</tr>
</tbody>
</table>

The content of each query is not shown. The reader is directed to the BYTE Inquiry Service for further information.

---

**Reader Service**

<table>
<thead>
<tr>
<th>Inquiry No.</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>86</td>
<td>169</td>
</tr>
<tr>
<td>*</td>
<td>HAL Communications 134</td>
</tr>
<tr>
<td>18</td>
<td>Heath Company 60, 61, 62, 63</td>
</tr>
<tr>
<td>12</td>
<td>Heuristics 101</td>
</tr>
<tr>
<td>80</td>
<td>Hybrid Tech 164</td>
</tr>
<tr>
<td>8</td>
<td>iCOM 26, 27</td>
</tr>
<tr>
<td>3</td>
<td>IMSAI 5</td>
</tr>
<tr>
<td>89</td>
<td>Interface Age 173</td>
</tr>
<tr>
<td>50</td>
<td>International Data Systems 141</td>
</tr>
<tr>
<td>67</td>
<td>James 170, 171</td>
</tr>
<tr>
<td>40</td>
<td>Kent Moore 130</td>
</tr>
<tr>
<td>94</td>
<td>Logic Design Inc 164</td>
</tr>
<tr>
<td>98</td>
<td>Meshina 172</td>
</tr>
<tr>
<td>97</td>
<td>Microcom 129</td>
</tr>
<tr>
<td>95</td>
<td>Microcomp 164</td>
</tr>
<tr>
<td>58</td>
<td>Mini Computer Suppliers 152</td>
</tr>
<tr>
<td>35</td>
<td>Mieromation 115, 117</td>
</tr>
<tr>
<td>60</td>
<td>Microware 153</td>
</tr>
<tr>
<td>44</td>
<td>MSI 133</td>
</tr>
<tr>
<td>82</td>
<td>Mikos 166</td>
</tr>
<tr>
<td>38</td>
<td>MiniTerm 125, 148</td>
</tr>
<tr>
<td>93</td>
<td>MITS CIV</td>
</tr>
<tr>
<td>81</td>
<td>Morrow 165</td>
</tr>
<tr>
<td>6</td>
<td>Motorola 91</td>
</tr>
<tr>
<td>62</td>
<td>nni 154</td>
</tr>
<tr>
<td>55</td>
<td>Mullen 151</td>
</tr>
<tr>
<td>29</td>
<td>National Multiplex 103</td>
</tr>
<tr>
<td>16</td>
<td>North Star Computers 57</td>
</tr>
<tr>
<td>66</td>
<td>Objective Design 160</td>
</tr>
<tr>
<td>12</td>
<td>OSI 40</td>
</tr>
<tr>
<td>20</td>
<td>OK Tool 76, 139</td>
</tr>
<tr>
<td>65</td>
<td>Oliver Audio Engineering 158</td>
</tr>
<tr>
<td>43</td>
<td>Omni 132</td>
</tr>
<tr>
<td>49</td>
<td>PAIA 140</td>
</tr>
<tr>
<td>53</td>
<td>Parasitic Engineering 47, 147</td>
</tr>
</tbody>
</table>

---

**Algorithm Challenge**

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from this list. Add a 9 cent stamp to the card, then drop it in the mail.

* Correspond directly with company.
Micromind is an incredibly flexible, complete and expandable, hardware/software, general purpose computer system. You won't outgrow it.

Hardware includes an 80 key, software-definable keyboard, I/O interface board, 6500A-series microprocessor (powerful enough for advanced computing), a high-detail graphics and character display processor, power supply, rf modulator, and connections for up to 4 tape recorders plus TV or monitor. An interconnect bus permits 15 additional microprocessors, parallel processing and vastly increased computing power.

System software—including ECD's own notsoBASIC high level language, on advanced error-correcting tape cassettes—provides a word processing editor, a powerful assembler, a debugger, a file system, graphic routines, and peripheral handlers. We also include dynamic graphic games: Animated Spacewar and Life.

ECD's standard Micromind μM-65 supplies 8K bytes of memory. Additional 32K byte expansion boards and a mapping option give Micromind expandable access to 64 Megabytes. Utilizing software-controlled I/O channels, Micromind's advanced encoding techniques load data from ordinary tape recorders at 3200 bits per second.

Micromind comes to you ready-to-use, factory assembled and fully tested. Among microcomputers, it has the largest memory capacity and the fastest storage. You're looking at the work of the finest display processor on the market. You won't find a microcomputer with a more powerful CPU.

You won't find a computer with a more flexible keyboard. You won't find anything to touch it at $987.54.

So, quit the kluge scene and key into Micromind. You'll be a main frame performer, with all the comforts of home. We're not fooling...this is the cat's μ!

ECD CORP.
196 Broadway, Cambridge, Mass. 02139
(617) 661-4400

Actual unretouched photographs.

Circle 92 on inquiry card.
It Comes Naturally With The Altair™ 8800b

The Altair 8800b from MITS, the second generation design of the microcomputer that started it all. The mainframe that has the abilities everyone is demanding from microcomputers today:

Expand-ability:
The Altair 8800b power supply and one-piece, 18-slot motherboard allow efficient and easy expandability for memory and I/O options. All Altair PC boards are designed to give you maximum capability/lowest power usage possible per board. This means that for each slot used you get more features and require less power than with any of the "off-brand" Altair-bus-compatible boards.

Whether you buy an entire system up front or choose to expand gradually, it’s easy to get the configuration you need—plus the complete family of Altair peripheral equipment, including floppy disk, line printer, audio cassette record interface, A/D converter, PROM programmer, serial and parallel I/O boards.

choice of four different memory boards and many others

Reli-ability:
The unique design features of the Altair 8800b, which have set the standard for the microcomputer industry, make it the most reliable unit of its kind. The Altair 100-pin bus, the now-standard design used by many imitators, has been "standard" all along at MITS. The unique Front Panel Interface Board on the Altair 8800b isolates and filters front panel noise before it can be transmitted to the bus. The all-new CPU board utilizes the 8080A microprocessor, Intel 8224 clock generator and 8216 bus drivers.

Flex-ability:
Meeting the diversified demands of an ever-increasing microprocessor market requires flexibility—not just hardware flexibility but software flexibility as well. MITS software, including the innovative Altair BASIC language, allows the full potential of the Altair 8800b computer to be realized.

8K ALTAIR BASIC has facilities for variable length strings with LEFTS, RIGHTS, and MIDS functions, a concatenation operator, and VAL AND STRS functions to convert between strings and numbers.

Extended ALTAIR BASIC allows integer, single and double precision variables, automatic line numbering and renumbering, user-defined string functions, PRINT USING for formatted output and a powerful EDIT command for editing program files during or after entry. Extended statements and commands include IF THEN ELSE LIST and DELETE program lines, SWAP variables and Trace On and Off for debugging.

Disk ALTAIR BASIC has all the features of Extended BASIC with the additional capability to maintain sequential and random access disk files. Utilities are provided for formatting disks and printing directories.

In all versions of ALTAIR BASIC you get the ease and efficiency of BASIC for the solution of real world problems.

Package II, an assembly language development system for the Altair 8800b, includes system monitor, text editor, assembler and debugger.

Afford-ability:
Prices for the Altair 8800b start at $840.00 for a kit and $910.00 for an assembled unit (all documentation included). For a complete listing of prices on all Altair products and a free brochure, contact MITS, Inc. 2450 Alamo S.E. Albuquerque, N.M. 87106 (505) 243-7821.

NOTE: Altair is a trademark of MITS, Inc.