SWTPC announces first dual minifloppy kit under $1,000

Now SWTPC offers complete best-buy computer system with $995 dual minifloppy, $500 video terminal/monitor, $395 4K computer.

$995 MF-68 Dual Minifloppy
You need dual drives to get full benefits from a minifloppy. So we waited to offer a floppy until we could give you a dependable dual system at the right price.

The MF-68 is a complete top-quality minifloppy for your SWTPC Computer. The kit has controller, chassis, cover, power supply, cables, assembly instructions, two highly reliable Shugart drives, and a diskette with the Floppy Disk Operating System (FDOS) and disk BASIC. (A floppy is no better than its operating system, and the MF-68 has one of the best available.) An optional $850 MF-6X kit expands the system to four drives.

$500 Terminal/Monitor
The CT-64 terminal kit offers these premium features: 64-character lines, upper/lower case letters, switchable control character printing, word highlighting, full cursor control, 110-1200 Baud serial interface, and many others. Separately the CT-64 is $325, the 12 MHz CT-VM monitor $175.

$395 4K 6800 Computer
The SWTPC 6800 comes complete with 4K memory, serial interface, power supply, chassis, famous Motorola MIKBUG® mini-operating system in read-only memory (ROM), and the most complete documentation with any computer kit. Our growing software library includes 4K and 8K BASIC (cassettes $4.95 and $9.95; paper tape $10.00 and $20.00). Extra memory, $100/4K or $250/8K.

Other SWTPC peripherals include $250 PR-40 Alphanumeric Line Printer (40 characters/line, 5 x 7 dot matrix, 75 line/minute speed, compatible with our 6800 computer and MITS/IMSAI); $79.50 AC-30 Cassette Interface System (writes/reads Kansas City standard tapes, controls two recorders, usable with other computers); and other peripherals now and to come.

Enclosed is:

___ $1,980 for the full system shown above (MF-68 Minifloppy, CT-64 Terminal with CT-VM Monitor).
___ $995 for the Dual Minifloppy
___ $325 for the CT-64 Terminal
___ $175 for the CT-VM Monitor
___ $395 for the 4K 6800 Computer

___ $250 for the PR-40 Line Printer
___ $79.50 for AC-30 Cassette Interface
___ Additional 4K memory boards at $100
___ Additional 8K memory boards at $250
___ Or BAC # ______ Exp. Date ______
___ Or MC # ______ Exp. Date ______

Name ______ Address ______
City ______ State ______ Zip ______
You can now have the industry’s finest microcomputer with that all-important disk drive.

YOU CAN GET THAT ALL-IMPORTANT SOFTWARE, TOO

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

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

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

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

CROMEMCO HAS THE SOFTWARE

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

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

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

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

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

ADVANCED CONTROLLER CARD

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

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

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

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

Z-2 USERS:

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

Cromemco

Specialists in computers and peripherals

2400 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400

Circle 353 on inquiry card.
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 IMSAIs, 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-0): $145. Assembled (Model BSW-0): $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**
- DIGITAL INTERFACE (OUR NEW TU-AKT). 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.
In the Queue

Foreground

26 RELOCATABILITY AND THE LONG BRANCH
   Programming Techniques—Bormann
64 AN APL INTERPRETER FOR MICROCOMPUTERS, Part 3
   Interpreter Design—Wimble
86 HOW TO IMPLEMENT SPACE WAR
   Software—Kruglinski
166 ANALYZE YOUR CAR’S GAS ECONOMY WITH YOUR COMPUTER
   Applications—Beuerschunb

Background

18 HOW TO WRITE AN APPLICATION PROGRAM
   Software—Jenkins
60 OTHELLO, A NEW ANCIENT GAME
   Software—Duda
70 AN 8080 SIMULATOR
   Software—Chung
114 FUNDAMENTALS OF SEQUENTIAL FILE PROCESSING
   Applications—Smith
130 C: A LANGUAGE FOR MICROPROCESSORS?
   Languages—Madden
140 SIMPLE APPROACHES TO COMPUTER MUSIC SYNTHESIS
   Hardware—Schneider
146 STRUCTURED PROGRAM DESIGN
   Software Design—Higgins
156 COMPUTER INFORMATION ARRANGEMENT
   Applications—Holladay
168 MASTERMIND
   Games—Milligan

Nucleus

4 In This BYTE
6 The Colorful Future of Personal Computing
11 Letters
22 About the Cover... and Some More of the Same
30 Languages Forum: Defining LIL, a Little Interpretive Language
50 Commodore’s New PET Computer
54 The NCC: A Dallas Delight
58 Technical Forum: More on Inexpensive Plotters
160 Book Reviews
172 BYTE’s Bits
174 BYTE’s Bugs
177 Clubs, Newsletters
184 Ask BYTE
190, 198 What’s New?
194 Classified Ads
208 BOMB
208 Reader Service
One whole subset of the personal computing world is provided by the users and manufacturers of programmable calculators. All the problems of creating applications software which users must solve on bigger machines are present, and often intensified by lack of scale, in these smallest of personal computers. William B Jenkins gives some useful information on the general process of creating an application program, and the specific problems of doing it on an SR-52 programmable calculator, in his article entitled How to Write an Application Program.

One of the conveniences of the 6800, 6502 and similar microprocessors is a relative branch method which allows one to construct position independent code which can be relocated by simply moving the programs involved. But these forms are typically limited to a 1 byte displacement, a limitation which Robert Borrmann shows how to overcome in the 6800 case by using appropriate stack manipulations and "long branch" subroutines. Read his article Relocatability and the Long Branch in this issue.

Looking for a different type of board game to play on your computer? How about the current game fad Othello (known as Reversi in England)? In Othello, a New Ancient Game Richard O Duda provides a short article with details for this game of skill and tactics.

This month, Mike Wimble concludes his 3 part series about an APL interpreter with An APL Interpreter for Microcomputers, Part 3: Mathematical Processing. With this segment, the functional design of interpreter is completed. Watch future issues for results of the Great APL Interpreter Contest inspired by Mike's article.

At first glance a simulator designed to run on the computer it is simulating may not seem very useful. Kin-man Chung feels differently for he wrote one. His article, An 8080 Simulator, describes one such program and gives ideas on how it can be put to good use.

Do you use cassettes as your principal mass storage medium? Then you will benefit from Wayne D Smith's discussion of Fundamentals of Sequential File Processing when it comes time to write software using such media.

Want to get involved in pitch generation for computer music synthesis? Thomas Schneider explains several approaches you might consider in his article, Simple Approaches to Computer Music Synthesis.

Using flowcharts to gather the logic for a program does not mesh with the current trend of structured programming. One technique that is directed towards the structured program approach is the use of Warnier-Orr diagrams. Use of these diagrams, as described by David Higgins in his article Structured Program Design, will result in accurate, well structured programs that will work correctly the first time they are executed.

The home computer has many uses besides number crunching and game playing. One of these uses, discussed by David Holladay in Computer Information Arrangement, is an information retrieval system. This type of system could be used to make your own dictionary type reference, help keep track of your files with cross reference, or simply make a personal version of the Schwann Catalog for your record collection.

Sensible automobile owners have long had the habit of recording mileage and gasoline filling figures at each visit to the service station. In this issue John P Bauernschub explains how to Analyze Your Car's Gas Economy with Your Computer in a short article presenting a complete BASIC program for this application.

Are you looking for a stimulating thought game to play with your computer? The game of Mastermind as described by W Lloyd Milligan in his article of that name will force you to think in a very logical manner if you want to have a chance at winning.
Announcing the West Coast's largest Personal Computing Show. April 28, 29, and 30, 1978 at California's brand new Long Beach Convention Center. This is a selling show with 180 booths (each draped, carpeted and with 500 watts of electricity). Three full days of conference sessions. There will be home brew exhibits, exhibitors lounge, inquiry badge system, computerized registration, a newsroom, and a full blown advertising and promotional campaign to bring you thousands of qualified buyers.

I wouldn't miss it for anything.

Send me the show kit:
Name:__________________________Title:__________________________
Company:__________________________
Address:__________________________State:__________________________
Zip Code:__________________________Phone:__________________________

Call: (714) 973-0880 Or mail this coupon: PERCOMP 78,
1833 E. 17 St., Suite 108, Santa Ana, CA 92701.
Sign early, corner booths are limited.

Circle 354 on inquiry card.
The Colorful Future of Personal Computing
(or What the World Needs Is a Good Mass Produced High Resolution Color Display . . .)

By Carl Helmers

This commentary on the possibilities of color imagery and display was inspired by a fantastic image processing system which was surely witnessed by many of the 36,000 people who thronged to the National Computer Conference in Dallas TX this past June. The system in question is produced by a company called Comtal, located in Pasadena CA. It is referenced variously as the Model 8000-S and “Vision One” in the literature I picked up at the site of the demonstration in the main exhibit area of the conference.

This Vision One system is not exactly a personal computing product. Its price tag in the $70,000 range makes it a candidate for laboratory or institutional use, but hardly a peripheral for the individual of ordinary means. Its characteristics include a built in LSI-11 computer with extensive software, direct refresh raster graphics hardware with 512 by 512 8 bit picture elements (ie: 262,144 bytes in its serial CCD refresh memory), and a hard surface disk drive. One of the prime practical applications of this system is its use by the Jet Propulsion Laboratory of Pasadena as the analysis and enhancement processor for the color photos returned from the Viking landers on the planet Mars. But the artistic and personal use attributes of such a color display are immense, as was demonstrated by the

Continued on page 42
The Computer for the Professional

Whether you are a manager, scientist, educator, lawyer, accountant or medical professional, the System 8813 will make you more productive in your profession. It can keep track of your receivables, project future sales, evaluate investment opportunities, or collect data in the laboratory. Use the System 8813 to develop reports, analyze and store lists and schedules, or to teach others about computers. It is easily used by novices and experts alike.

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

See it at your local computer store or contact us at 460 Ward Dr., Santa Barbara, CA 93111, (805) 967-0468.
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, $4750 in kit form, $5450 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.

The functional beauty of Sol Computer Systems is more than skin deep. A look inside reveals a simple elegance of design and sturdy construction.
A CMOS LOGIC PROBE

This is in response to Tom Kryst's letter in the July 1977 BYTE, page 148, expressing a need for a low cost CMOS logic probe. I was faced with a similar requirement while working on a software development system for the RCA 1802 CMOS microprocessor. Not being a person who believes in reinventing the wheel, I borrowed Kurt Christner's TTL design from the January 1977 BYTE, page 82, adding a CMOS inverting buffer at the front end, as shown below. Please note that a CD4069 inverter may not be substituted for the more expensive CD4049, as the current sinking capability of the latter device is necessary.

I hope this modification may prove useful to some of your readers.

Frank A Weissig
343 NW 8th St, Apt 3
Corvallis OR 97330

Within three days I received a very courteous and apologetic phone call from the company with an open invitation to call them collect in the event I experienced further difficulty. With this kind of response it would seem likely that they intend to back up their commitment to quality support as stated in their advertisement.

To a second party, my finished product may not appear to go beyond the Intercept Jr offered by InterSil, based on their IM6100 chip, which goes for around $450 with 1 K of programmable memory. This compared to the $1200 I have spent may be called to question when considering only the physical result. The difference, as I see it, is that I will have acquired an understanding of the PDP-8 computer that I would never

KUDOS FOR THE LD-14

I'd like to make a few comments about one of your advertisers, namely Logic Design Inc of Laramie WY. I purchased their LD 14 Tutorial Training Computer which was written up in January 1977 BYTE. Although I have not finished wire wrapping the unit, I am confident that I am getting my $1200's worth. One thing that has impressed me about the company is the quick response to a problem I was having one evening because of a gross error on my part. Some of the wires melted together as the result of tying VCC to ground. I became a little nervous that maybe I damaged something, so I wrote to complain about their failure to give any guarantees that their product is idiot proof.

Phil Winninghoff
815 Plata Rd
Arroyo Grande CA 93420

ASIMOV'S PERSONAL COMPUTER ANTICIPATION, CIRCA 1950

In his article, "Why aren't there any Altairs on Arcturus II?" (April 1977 BYTE), H Melton states that, "I can

SWTP 6800 OWNERS—WE HAVE A CASSETTE I/O FOR YOU!

The CIS-30+ allows you to record and playback data using an ordinary cassette recorder at 30, 60 or 120 Bytes/Sec! No Hassle! Your terminal connects to the CIS-30+ which plugs into either the Control (MP-C) or Serial (MP-S) Interface of your SWTP 6800 Computer. The CIS-30+ uses the self clocking 'Kansas City'/Biphase Standard. The CIS-30+ is the FASTEST, MOST RELIABLE CASSETTE I/O you can buy for your SWTP 6800 Computer.

PerCom has a Cassette I/O for your computer!
Call or Write for complete specifications

PerCom Data Co.
P.O. Box 40598 • Garland, Texas 75042 • (214) 276-1968

PerCom — 'peripherals for personal computing'
remember only one old story that used pocket calculators. . . .” As an avid SF fan, and as an even more avid calculator (HP) freak, I must beg to differ with Mr. Melton. In the Foundation Trilogy, by Isaac Asimov, one finds the following passage:

Seldon removed his calculator pad from the pouch at his belt. . . . Its grey, glossy finish was slightly worn by use. Seldon’s nimble fingers . . . played over the hard plastic that rimmed it. Red symbols glowed out from the grey.

How many of us have not performed the act just described? And the Foundation Trilogy was, I believe, written in the early 1950s. I must, however, agree with the general tone of the article: most of us (including SF writers) are just not able to predict the rate of technological progress.

W Gray Mansfield
5042 Guava Av
La Mesa CA 92041

PRINTING BARS ON SELECTRICS?

I see you’re printing optical code. Is it possible to get an IBM ball (or keys) which prints letters and code simultaneously? If yes, where are they available?

Rob Loring
Twin Oaks Community
Rt 4, Box 169
Louisiana VA 23093

Yes, but it’s expensive. Custom tooling is available from several sources for IBM style balls; thus, while one could design a bar code font, it would not be a project undertaken lightly.

LIGHTING THROUGH THE PAPER?

The idea of distributing software through PAPERBYTES is great. Why don’t you publish PAPERBYTES on one side of a page only, leaving the other side blank? Light could then be shone from underneath the paper to illuminate the coded material.

This technique would greatly simplify the optical system needed for the bar code reader. It would eliminate the light source and lens in the reader (wand) and reduce specular reflection problems.

I plan on trying this approach soon. Do you or your readers have any suggestions?

Andrew A Modia
108 Clemens Ct
Lansdale PA 19446

The main problem with the approach of backlighting is that it leaves one whole side of the paper blank, something which is less than optimal when it comes to publishing information in books and magazines on expensive paper.

PRINTING QUOTES ON AN HP9830 IN BASIC

As a user of Hewlett-Packard’s 9830 BASIC machine, I have discovered inc. of its (very few) shining features. The BASIC interpreter has a statement pair WRITE and FORMAT, which FORTRAN users will recognize. This is a rather little known fact, as HP buries it in an obscure portion of their manual.

In order to get quotes out of it, something like this would be written:

10 FORMAT B
20 WRITE (15, 10) 34, “HELLO”, 34

where 15 is the device code of the main printer. The output looks like this:

“HELLO”

What goes on is that B format will convert the decimal number given it (constant, variable, or whatever) and output it directly as ASCII; the quote mark is 34 in ASCII. This can also be used to print nonkeyboard characters; square brackets, reverse slashes, and so on.

The other way to do this requires the Advanced IO ROM, as HP sells their software as plug in (black or gray) boxes. The OUTPUT statement allows you to use a string as an output device. Example:

10 FORMAT B
20 OUTPUT (A$, 10) 34,
30 PRINT A$

with the result:

“"

So, all you users of the HP9830, now you can print quotes and other goodies. This almost makes up for the crudity of HP’s strings.

John Woods WB7EEJ
6541 126th Av SE
Belleview WA 98006

AS SALES EXPAND, SAFETY ASPECTS MUST BE IMPROVED

While experimenters look for the fastest, most powerful, least expensive microcomputer, they seldom look for the safest. From looking at ads for hobby computers, I learn that only OSI advertises their power supply as being listed by Underwriters Laboratories Inc. Some small computers have 117 VAC conductors that are exposed when the lid is off (as I learned one day by accident — no injury, just minor equipment damage); the same machines don’t have fuses in the unregulated low voltage high current lines. Just wait until you’ve welded a probe to a power trace to find out how much you need that item. It’s an industrial standard to avoid exposed high voltages whenever possible; and I see no reason to have an exposed, noisy

Continued on page 32
Welcome to ComputerLand. An incredible adventure into the world of personal computers. A one-of-a-kind shopping experience.

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

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

You can even plot your biorhythm.

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

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

Read on.

Genuine Service.

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

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

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

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

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

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

Right now!

Great Selection.

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

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

Plus a complete inventory of tools, books and accessories.

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

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

Exactly when you want it.

Be Our Guest.

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

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

The great computer store. RATED G.

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

ComputerLand

1922 Republic Avenue, San Leandro, CA 94577 (415) 895-9363
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. It's a complete, ready-to-use computer, not a kit. At $1298, it includes video graphics in 15 colors. It includes 8K bytes ROM and 4K bytes RAM—easily expandable to 48K bytes using 16K RAMs (see box). But you don't even need to know a RAM from a ROM to use and enjoy Apple II. For example, it's the first personal computer with a fast version of BASIC permanently stored in ROM. That means you can begin writing your own programs the first evening, even if you've had no previous computer experience.

The familiar typewriter-style keyboard makes it easy to enter your instructions. And your programs can be stored on—and retrieved from—a 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.

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 integer 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.

from any one of the following authorized dealers:

ALABAMA
The Computer Center
303 B. Pecor Place
Birmingham

ARIZONA
Wichita
415 N. Broadway

CALIFORNIA
ComputerLand
20863 Stevens Creek Boulevard, B-3-C
Cupertino, California 95014
(408) 996-1010

FLORIDA
Team Electronics
1154 E. Oakland Park Blvd.
F. Lauderdale

GEORGIA
Team Electronics
301 N. Fulton Drive
Atlanta

ILLINOIS
Team Electronics
304 W. Second Street
Springfield

KANSAS (continued)
Team Electronics
1221 Main Avenue

MINNESOTA (continued)
Team Electronics
1233 S. Robert Street
West St. Paul

NEW YORK
Computeland
3800 West 41st Street

SOUTH DAKOTA (continued)
Team Electronics
6439 Westheimer

TEXAS
Computer City
3004 E. Broadway

WASHINGTON
Home Computer Center
3312 Old Mill Road

WISCONSIN
Team Electronics
3028 Hunsinger Lane

Wyoming
6101 W. Broadway

If you would like to be an Apple dealer, call Gene Carter, Director of Dealer Marketing, 408-996-1010.
MOST OF TODAY'S PERSONAL COMPUTERS are microcomputers. Compared to our minicomputer they are slow and expensive.

Our mini "the MINI 12™" has been around for three years. We've been steadily selling it to industrial customers who have proven its reliability. Only recently have we paid much attention to the personal computing market. We've discovered that our computer is simply better, and most interestingly, LESS EXPENSIVE.

NONE OF THE MICROCOMPUTERS can offer you the range of software available to the MINI 12. Compatibility with the most popular minicomputer, DEC's PDP-8E, gives you a vast library of programs. You will also be pleasantly surprised when you experience the speed and sophistication of our EXTENDED BASIC compiler, or FORTRAN, ALGOL and FOCAL.

Unlike today's microcomputers, our Mini 12 is fully software supported by an advanced operating system, including...file handling text editor, symbolic assembler, loaders, ODT (octal debugging technique), batch processor, and system utilities. The MINI 12 comes with 8K of memory and is easily expandable to 32K. Because the MINI 12 processes 12 bit words (instead of 8 bit bytes) you need far less memory. For most applications...using our optional MINI STORE digital cassette system with the BASIC compiler, you will never need add another single byte!

Because we want you to have the pleasure of using your minicomputer right away, our MINI 12 comes fully assembled, debugged and ready to plug in. It includes a full front panel, controller for eight digital tape drives, and built-in serial interface plus other technical niceties like a binary loader and monitor bootstrap in ROM memory.

Frankly, when compared with microcomputers, we're surprised at how good a deal we are. Write... or better yet, give us a call @303-794-1634 and let us help you get it all together!

TLF Post Office Box 2298, Littleton, Colorado 80161, Telephone 303-794-1634

PDP-8E and Focal are Trademarks of Digital Equipment Corporation

Circle 358 on inquiry card.
How to Write an Application Program

The Texas Instruments SR-52 is a very impressive machine for its size. Several different libraries of programs are provided that include engineering, finance, and games applications. The game programs do, of course, provide the easiest way to impress your friends with the power of the SR-52. This is no different from using game programs with any computer system. For personal use, though, games can quickly lose their glamour. When this happens, it is time to learn how to program.

As the owner of a small business, there are at least two tasks which consume a fair amount of my time. One is payroll and the other is invoice preparation. Of the two, invoice preparation is the easiest program to start with, although both programs offer good opportunities to learn many of the intricacies of programming. These techniques also provide a practical way of learning general programming methods. The result is the development of skills that include analyzing the problem and designing a logical way (flowchart) of identifying the best method for solving the problem. Next comes the conversion of the flowchart to the program language that your machine uses. To do this, let's start with the problem of developing a program to help prepare invoices.

Invoice Program

The first step is to think out what you would do to manually prepare an invoice. There are several ways of doing invoices. We are manufacturers of retail products and sell various quantities of many different items. Our invoices show the suggested list price for each item. After the items are totalled, a discount is calculated and subtracted from the total. The result is the net amount that the customer pays. There are other costs, such as shipping charges, which are also added. Figure 1 shows a flowchart for this first step.

Making a flowchart for this first step may seem to be unnecessary. What it does is give you a graphic description of the steps involved. From this, some of the next steps become more obvious. For example, in an invoice, we multiply one item by another and save the total. This basic step is repeated for as many items as required. This type of repetition is one of the things a computer does best. In the same way, it is the very thing that bores a human fastest.

The end steps require adding the individual totals and getting a grand total. The remaining steps are straightforward arithmetic.

The initial conclusion (one which we sort of knew from the start) is that this problem lends itself to solution by computer. We will add one more set of calculations so that the final program can handle a second class of merchandise that does not have any discount applied. This addition is similar to the steps shown in figure 1, and we will include it in our next reiteration of the design.

SR-52 Features

What we have done so far has been independent of the system that we will be using. At this point, we have to learn a little about the specific hardware.

The SR-52 allows programs to be input using the calculator keyboard. The individual keys could be considered as a set of...
op codes for a high level language interpreter, with each key performing a set of fairly complex operations. The SR-52 provides several categories of key functions. It has the conventional arithmetic functions: +, -, x, ÷; and a set of memory functions that allow you to manipulate the 20 available memory areas. These functions include STORE, RCL (recall), SUM, PROD (product), and EXC (exchange). Trigonometric functions, Sin, Cos, Tan and other standard functions such as x^2, y^x, and 1/x are also included. An important set of keys are those that allow the SR-52 program to make decisions. These can perform tests on program data in order to determine what the next program step might be. The SR-52 is shown in photo 1.

The SR-52 has a maximum of 224 program steps. To program the machine, you simply take the list of key sequences and push the buttons in that sequence until you are done. This is a little oversimplified, but it really isn't much more than that. If your program has been properly designed, the SR-52 will be able to do all of the finger crunching work when you put in the data. When you are satisfied with the program, you can save it on a magnetic card for future use.

Zeroing In

What we have done up to this point has given us a logical approach to the program solution of our problem, and an overall notion of how the SR-52 can do the job for us. Now we have to get a little closer to the SR-52 to know exactly how to translate our flowchart to key sequences.

Our program should be easy for a non-technical person to use. This means that entering the data should use very few steps. The program should do most of the work until the problem (invoice) is complete. To help accomplish this, the program will be written to use the optional SR-52 printer as a recorded output. The printer, shown in photo 2, in addition to providing a hard copy output, actually makes program development easier. By printing desired results as they occur we can avoid using too many of our precious 224 program steps to store and then later recall results. In some cases, programs that would be too long, more than 224 steps, without the printer can be handled on the SR-52. Additionally, since the printer can be operated under program control, some limited data formatting can be accomplished. Last of all, the printer allows the user to verify the accuracy of the data input.

Invoice Program Development

In the invoice problem we are working on, we can see from the initial flowchart that we are going to use several basic SR-52 functions. These are data storing, recalling, multiplying, subtracting, adding and printing.

We will now expand our initial flowchart

Photo 2: Printer attachment for SR-52.
to include the actual steps we will ask the SR-52 to do for us. In this expanded flowchart we have added the second category of merchandise we want to price. One other design consideration that we want to include is the manner in which we will have the user input the data. The SR-52 has a set of five buttons that are labeled A, B, C, D and E. These same buttons can be set for a second use which adds five more functions labeled A', B', C', D' and E'. These buttons are called "user defined keys." Depending on the way you write the program, a given user button will do whatever you want. For our use, we will want to use these buttons to define to the program what our data means. Table 1 shows the definition we have given to these keys for this program. Looking at the work the SR-52 does when a user defined key is pushed gives you an idea of the power of the machine.

To get the invoice data into the program, the user keys in the numbers on the keyboard and presses the appropriate user defined key.

Now, we can expand our initial flowchart to what we see in figure 2. The numbers in circles adjacent to various boxes are the order in which we will write the program sequence. We will refer to these numbers in the following discussion of the new flowchart.

1. This is some of the housekeeping that any program should have. It assures you that the initial condition of the calculator or computer is what you expect it to be. Here we want to CLR (clear), CMs (clear data memories), FIX 2 (fix the decimal point of the computations to two places) and HLT (halt). We want the machine to stop and wait for us to input data.

2. In this step we are going to define key C to mean discount. We are going to STO (store) it in data memory location 03, print what we entered, advance the paper one line and HLT again.

3. Key D is defined to be the shipping amount of the order; it is stored in data memory 04. We then halt. We could have printed this value, but I chose not to because it will be used at the end of the program where, if it is the wrong amount, not much harm is done.

4. Key A is the quantity of merchan-

Continued on page 152
Look To The North Star HORIZON Computer.

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

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

WHAT ABOUT PERFORMANCE?
The Z80A processor operates at 4MHz — double the power of the 8080. And our 16K RAM board lets the Z80A execute at full speed. HORIZON can load or save a 40K byte disk program in less than 2 seconds. Each diskette can store 90K bytes.

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

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

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

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

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

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

North Star ★ Computers
2465 Fourth Street • Berkeley, California 94710 • (415) 549-0858

Circle 359 on inquiry card.
About the Cover

The artwork on our cover this month and the samples found on the following pages were created by Thomas A. Defanti and Guenther Tetz of the University of Illinois at Chicago Circle. The system used to create these works used a high level "Graphics Symbiosis System" (acronym: GRASS) grafted onto a PDP-11/45 computer with a Vector General 3DR display scope, a data tablet and 30 channels of analog input and output for interactive control of the programmed parameters of the display. Consider it in some sense the visual equivalent of a Moog synthesizer's audio functions. A more complete description is found in a paper entitled "The Digital Component of the Circle Graphics Habitat" which was published on pages 195 to 203 of the proceedings of the 1976 National Computer Conference (a monstrously thick book available from AFIPS, the American Federation of Information Processing Societies, 210 Summit Av, Montvale NJ 07645).

Artists Defanti and Tetz sent in several slides of their work as part of the art contest we ran last year in the September 1976 issue of BYTE. They report that work is progressing on a microprocessor based version...
of GRASS, work which may result in some heady progress in low cost graphics art using computers.

Around the BYTE offices, we tend to think of the cover as the “aquamarine-print” of the design of a house for the inhabitants of Arcturus IV. Whether this is true or not is open to some question.

The other photographs also came to us untitled, so we list them simply as photos 1 thru 8. Why? Well, it occurred to us that a bit of informal fun (and perhaps a bit revealing as well) could be had by leaving it to readers to suggest appropriate titles. Watch the letters columns of future issues to see what comes of this request for titles.
A position independent relocatable program is a program that can be moved to any convenient place within your memory space, and executed without any changes. One of the very nice things about the Motorola 6800 instruction set is the relative addressing mode used in its branch (BRA) and branch to subroutine (BSR) instructions; instructions using this addressing mode are inherently relocatable without patching.

To illustrate relative addressing consider the 6800 instruction: 20 35 (hexadecimal). Hexadecimal 20 is the code for the branch (BRA) instruction, so 20 35 means jump to the address that is hexadecimal 35 bytes beyond the instruction which follows this branch instruction. Thus, if a program initially occupies hexadecimal locations 0080 to 0200, and includes at location 0100 the above branch instruction, this causes the processor upon encountering it to jump to location 0137. If the program is now copied from its original location 0080 to 0200 into a new location 1580 to 1700, the branch instruction (now at location 1600) will correctly cause the processor, upon encountering it, to jump to location 1637.

Unfortunately, the 6800's branch instructions allow only one byte to be used as the relative displacement. This allows a maximum branching range of only hexadecimal +7F (+127 decimal) to hexadecimal -80 (-128 decimal) bytes. What do you do if you have need for longer-range branching, as is the case in larger programs? If you use the jump (JMP) or jump to subroutine (JSR) instructions, which include absolute address references, the program is no longer relocatable without modification. It loses the position independence feature which makes a generalized program read only memory possible.

One solution to the problem of writing large relocatable programs is provided by the long branch (LONGBR) and long branch to subroutine (LONGBS) subroutines described here. Although the listings show starting addresses of 278E and 276B respectively, both routines are completely independent of memory address space location. They can be used in two ways.

The first way is to incorporate both routines into your own system monitor, which presumably occupies a fixed location in your memory space. This is the way I use them. Then, to execute a long branch within a program you are developing anywhere in memory, you simply execute a jump to subroutine LONGBR, and follow the JSR LONGBR instruction by a 2 byte adder which indicates how many bytes ahead or
<table>
<thead>
<tr>
<th>Hexadecimal Address</th>
<th>Hexadecimal Code</th>
<th>Label</th>
<th>Op</th>
<th>Operand</th>
<th>Commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td>2768</td>
<td>DF 20</td>
<td>LONGBS</td>
<td>STX</td>
<td>XSTOR</td>
<td>[save X for later];</td>
</tr>
<tr>
<td>276D</td>
<td>30</td>
<td>TSX</td>
<td></td>
<td></td>
<td>X := S + 1;</td>
</tr>
<tr>
<td>276E</td>
<td>0F</td>
<td>SEI</td>
<td></td>
<td></td>
<td>[turn off maskable (IRQ) interrupts];</td>
</tr>
<tr>
<td>276F</td>
<td>AE 00</td>
<td>LDS</td>
<td>0,X</td>
<td>S := @X [would be return address];</td>
<td></td>
</tr>
<tr>
<td>2771</td>
<td>09</td>
<td>DEX</td>
<td></td>
<td></td>
<td>X := X - 2 [point to location of WBRA copy];</td>
</tr>
<tr>
<td>2772</td>
<td>09</td>
<td>DEX</td>
<td></td>
<td></td>
<td>[patch the stack];</td>
</tr>
<tr>
<td>2773</td>
<td>AF 00</td>
<td>STS</td>
<td>0,X</td>
<td></td>
<td>[copy WBRA];</td>
</tr>
<tr>
<td>2776</td>
<td>31</td>
<td>INS</td>
<td></td>
<td>S := S + 2 [calculate actual return address];</td>
<td></td>
</tr>
<tr>
<td>2777</td>
<td>AF 02</td>
<td>STS</td>
<td>2,X</td>
<td></td>
<td>[patch the stack];</td>
</tr>
<tr>
<td>2779</td>
<td>35</td>
<td>TXS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2781</td>
<td>36</td>
<td>APUSH</td>
<td>PSHA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2782</td>
<td>31</td>
<td>INS</td>
<td></td>
<td>S := S + 2 [point stack]</td>
<td></td>
</tr>
<tr>
<td>2785</td>
<td>32</td>
<td>PULA</td>
<td></td>
<td>A := WBRAL;</td>
<td></td>
</tr>
<tr>
<td>2786</td>
<td>EE 00</td>
<td>LDX</td>
<td>0,X</td>
<td>X := WBRAB [point to OFFSET];</td>
<td></td>
</tr>
<tr>
<td>2789</td>
<td>AB 01</td>
<td>ADDA</td>
<td>1,X</td>
<td>LBA := WBRAL + OFFSET [cal-]</td>
<td></td>
</tr>
<tr>
<td>278A</td>
<td>36</td>
<td>PSHA</td>
<td></td>
<td></td>
<td>culate relative branch</td>
</tr>
<tr>
<td>278B</td>
<td>34</td>
<td>DES</td>
<td></td>
<td></td>
<td>target and insert in stack region;</td>
</tr>
<tr>
<td>278C</td>
<td>32</td>
<td>PULA</td>
<td></td>
<td>A := WBRAL;</td>
<td></td>
</tr>
<tr>
<td>278D</td>
<td>A9 00</td>
<td>ADCA</td>
<td>0,X</td>
<td>[complete the sum];</td>
<td></td>
</tr>
<tr>
<td>278E</td>
<td>36</td>
<td>PSHA</td>
<td></td>
<td></td>
<td>[finish patching stack];</td>
</tr>
<tr>
<td>278F</td>
<td>34</td>
<td>DES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2790</td>
<td>32</td>
<td>PULA</td>
<td></td>
<td>[restore old A];</td>
<td></td>
</tr>
<tr>
<td>2793</td>
<td>DE 20</td>
<td>LDX</td>
<td>XSTOR</td>
<td>[restore old X];</td>
<td></td>
</tr>
<tr>
<td>2794</td>
<td>0E</td>
<td>CLI</td>
<td></td>
<td></td>
<td>[reasonable interrupts];</td>
</tr>
<tr>
<td>2797</td>
<td>39</td>
<td>RTS</td>
<td></td>
<td>LBA := offset of the long branch to the subroutine or long branch operation;</td>
<td></td>
</tr>
<tr>
<td>2798</td>
<td>DF 20</td>
<td>LONGBR</td>
<td>STX</td>
<td>XSTOR</td>
<td>[save X for later];</td>
</tr>
<tr>
<td>2799</td>
<td>30</td>
<td>TSX</td>
<td></td>
<td></td>
<td>[set up for]</td>
</tr>
<tr>
<td>2801</td>
<td>DF</td>
<td>SEI</td>
<td></td>
<td></td>
<td>branch without</td>
</tr>
</tbody>
</table>

The following symbols are used for the values in the stack area:

- **WBRA** = "would be branch address," a 16-bit value with high order byte WBRAH and low order byte WBRAL. This is the pointer to the relative branch OFFSET which follows the JSR to LONGBR or LONGBS.
- **OLD** = old contents of stack, unchanged by these routines.
- **RA** = "return address," a 16-bit value with high order byte RAH and low order byte RAL, computed by these routines.
- **LBA** = "long branch address," a 16-bit value with high order byte LBAH and low order byte LBAL, the computed target of the long branch to subroutine or long branch operation.
- **OLDA** = old value of the A accumulator, saved on the stack during computations.

The index (X) register and stack (S) values pointing to the stack area are noted above the content boxes in this representation.
behind you want to branch. For example, if you want to long branch from location 0100 to 0345, you would need the following code:

```
0100 BD 27 8E  JSR  LONGBR
0103 02 42  FDB $0242
```

The target address is computed by adding the offset (hexadecimal 0242) to the address of the byte following the JSR instruction (hexadecimal 0103):

```
0242
0103
0345
```

To execute a long branch to subroutine the procedure is similar except that you jump to subroutine LONGBS instead of LONGBR. For example, if your program is at hexadecimal location 0240 and you want to branch to a subroutine that is located at 0050, you would need the following code:

```
0240 BD 27 6B  JSR  LONGBS
0243 FE 0D  FDB $FE0D
```

where as usual the target address 0050 is the sum of the offset (FE0D) and the address of the byte following the JSR instruction (0243). Upon encountering the JSR LONGBS instruction at 0240, the processor would go to LONGBS for some massaging of data on the stack, and from there would go to your subroutine at 0050. Upon entering the subroutine at 0050, it will have the same values of accumulators A and B, and index register and stack pointer, as it had when encountering the JSR LONGBS instruction at 0240. The condition flags would in general be different, however. Upon returning from your subroutine, execution would resume with the instruction at location 0245.

When relocating a program using JSR LONGBR or JSR LONGBS instructions you would, of course, leave such instructions unchanged (assuming that your monitor incorporating the LONGBR and LONGBS routines was not itself being relocated). If the location of each long branch or long branch to subroutine call is being changed by the same amount as the target addresses, the program will work the same in the new location, just as was the case with ordinary relative branches.

The second way to use the LONGBR and LONGBS routines is to build them into the long program you are writing. While LONGBR and LONGBS are themselves relocatable, there remains the problem of branching to these subroutines from all locations within the program that will require long branching. This branching can be accomplished by installing "stepping stones" throughout your program which allows any location within your program to branch from "stone" to "stone," and thereby finally get to these subroutines.

To illustrate, suppose a program occupies memory locations 0200-04CF, including LONGBS at 0200 and LONGBR at 0223. Then the stepping stones might be:

```
0200 7E 20 80  LBS1  BRA  LONGBS
0200 7E 20 A1  LBR1  BRA  LONGBR
020C 20 80  LBS2  BRA  LBS1
020E 20 80  LBR2  BRA  LBR1
0214 20 80  LBS3  BRA  LBS2
0216 20 80  LBR3  BRA  LBR2
021C 20 80  LBS4  BRA  LBS3
021E 20 80  LBR4  BRA  LBR3
0224 20 80  LBS5  BRA  LBS4
0226 20 80  LBR5  BRA  LBR4
```

To execute a long branch from anyplace within the program, simply execute a branch to subroutine (BSR) to the previous LBRn stepping stone. To execute a long branch to subroutine, simply execute a BSR to the previous LBSn stepping stone. These branches are then completely relocatable. (Of course this technique of stepping stones can be used directly just as well, without LONGBS or LONGBR; but one chain of stones has to be devoted to each branch target. If more than two subroutines [or branch targets] must be used, less code will be required if LONGBS [or LONGBR] is used.)

**How It Works**

The operation of the LONGBS subroutine can be understood by a study of the program listing and comments shown in figure 1. The entries at the extreme right of the listing show the contents of the stack
region of memory at the conclusion of each step, which changes the stack contents or
the pointers (index register X or stack
pointer S) used to keep track of position
within the stack.

To help understand the operation of the
program let us consider an example in which
the processor encounters the following in-
struction sequence:

```
0736 BD 27 6B JSR LONGBS
0739 FC 1A FDB $FC1A
073B 86 02 LDA A #$02
```

In executing the JSR instruction at 0736,
the processor places the address 0739 (which
would ordinarily be the return address) onto
the stack, and jumps to 276B. However,
0739 as used here is not the actual return
address, since it is the location of the 2 byte
offset which will be used to form the actual
target address. Thus, 0739 is more prop-
erly called the "would-be" return address
(W8RA). Upon entering the subroutine
LONGBS, then, the stack looks as follows:

```
WBRAh WBRAL RAh RAL OLD
(07) (39) (07) (38)
```

where SP with the arrow denotes the posi-
tion in the stack being pointed to by the
stack pointer, WBRAh is the high order byte
of WBRa, WBRAL is the low order byte of
WBRa, and OLD denotes stack contents
before executing the JSR LONGBS instruc-
tion; these "old" contents of the stack will
not be disturbed.

The first line of subroutine LONGBS
stores the values of IX away for later retri-
val. I happened to use low core location
0020 to 0021 for this purpose, but
any available programmable memory loca-
tion can be used instead. (You may prefer to
use a spare 2 byte location in the progran-
mable memory devoted to MIKBUG in
many systems.) The next six lines of code
transfer to IX the burden of keeping track of
our stack location, while the stack pointer
itself is used to copy the would-be return
address into two additional stack locations.
Since we are fooling with the stack
pointer, it is necessary to prevent interrupts
from occurring at this time. The SEI (set
interrupt mask) locks out the maskable
interrupts until further notice. We now have
the following stack picture:

```
IX
WBRAh WBRAL WBRAl WBRAl OLD
(07) (39) (07) (39)
```

The next three lines (2775-2777) correct the
would-be return address to the actual return
address 073B:

```
 IX
WBRAh WBRAL RAh RAL OLD
(07) (39) (07) (38)
```

In line 2779 the stack pointer is returned to
its usual role of keeping track of position in
the stack. Since accumulator A is to be used
for some data manipulation in subsequent
instructions, its original value (the value it
had when JSR LONGBS was executed) is
stashed away on the stack for later retrieval.
After executing this instruction at PSHA the
stack appearance is:

```
SP
WBRAh WBRAL RAh RAL OLD
(07) (39) (07) (38)
```

The next ten lines (2778 to 2787) cause the
would-be return address WBRA to be in-
creased by the value of the 2 byte adder; the
result is the target address, denoted by
LBaH (long branch address high order byte)
and LBaL (long branch address low order byte).

The appearance of the stack now is:

```
SP
ORIGA LBaH LBaL RAh RAL OLD
(03) (53) (07) (3B)
```

The original values of accumulator A and of
index register IX are now retrieved, inter-
rupts are reenabled, and the processor jumps
to 0353 by executing the RTS (return from
subroutine instruction). Notice that be-
cause of the manipulation of the stack
during subroutine LONGBS, the processor does not
return to the program section which called
it; instead it jumps to the target address
(0353 in this case), because the effect of
executing the RTS instruction is to place
into the program counter the 2 byte address
pulled from the top of the stack.

When the subroutine which begins at
0353 has been finished, the processor exec-
utes the RTS instruction which ends it, and
this returns the processor to the actual
return address 073B.

The operation of the long branch routine
LONGBR is similar, except that the RAH
and RAL bytes are not needed or wanted.
Thus, the initial part of LONGBS which
duplicated the value of WBRA on the stack
is bypassed.
Defining LIL, a Little Interpretive Language

It is with great interest that I read the discussion by Donald J. Stavely in the April 1977 Technical Forum. Mr. Stavely has articulated a concept in which I have been greatly interested for the past several months, and in which I have invested a certain amount of work.

LIL, Little Interpretive Language, is a pseudocompiling language whose compiled code could be run on any microcomputer which has a LIL loader and a LIL interpreter. LIL is still in a state of flux, but may be spoken of in general terms. It is to consist of three separate programs: the compiler, the loader and the interpreter.

The compiler is to accept character keyboard input and build three structures: the symbol table (to be discarded at the completion of compilation) which contains the names of objects, the reference table which contains descriptors of objects and the memory addresses at which those objects may be found, and the compiled code which contains 2-byte descriptor addresses and 1-byte offsets to a table of function routines used by the interpreter.

The loader is to load reference tables and code into the proper areas of memory for processing by the interpreter, and is to modify all initial linkages established by the compiler so as to agree with the new positions of reference tables and code.

The interpreter is to interpret only addresses of descriptors and offsets to function routines. The interpreter is to allocate memory for data storage on a software stack; the hardware stack is to be used for data manipulation such as arithmetic functions.

The linkage for the source statement:

```
COUNT +1 INTO COUNT;
```

can be represented as in listing 1.

Control bytes serve the useful purpose of defining which of the following units of code are address pairs and which are singular offsets. The control byte in the previous example links to the code as follows:

```
1 1 0 0 1 0 X X (X = don't care)
```

COUNT 1 "+" "INTO" COUNT ";"
THE EXTENSYS RM64 MEMORY BOARD provides the most cost effective system memory found in the industry. The RM64 provides this because of our low cost per byte when compared to our competition plus the increased reliability of a single board over multiple boards containing less memory. The board is S-100 bus compatible making it usable in over a dozen different microcomputer systems including ALTAIR and IMSAI. The RM64 is available in three configurations: 32K, 48K or 64K bytes of memory all on ONE board. The board is completely assembled, checked out and burned in for at least 50 hours prior to shipment. This complete testing procedure allows Extensys to provide 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 byte 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;

Write protection in 16K blocks; and
- Fully socketed for 64K, allowing 32K and 48K versions to be upgraded at a later date.

Delivery of the RM64 is 15 to 30 DAYS upon receipt of order. Prices for the RM64 include shipping and handling prepaid in the continental United States. EXTENSYS Corporation is also announcing several other new highly cost effective products. These include a total floppy disk system based around File I/O board and a multiprocessor operating system. The other product, which interfaces with the RM64 memory board to create a megabyte or more of memory and adds full DMA capability to the File I/O board, is a Board Select/DMA board. Both of these products are S-100 compatible.

Contact your local computer store or order directly from EXTENSYS

592 Weddell Drive
Sunnyvale, California 94086
(408) 734-1525

Please place my order for the following:

<table>
<thead>
<tr>
<th>CNT</th>
<th>DESCRIPTION</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RM64-32K byte board</td>
<td>$895 ea.</td>
</tr>
<tr>
<td></td>
<td>RM64-48K byte board</td>
<td>$1195 ea.</td>
</tr>
<tr>
<td></td>
<td>RM64-64K byte board</td>
<td>$1495 ea.</td>
</tr>
</tbody>
</table>

Subtotal: $X.XX

California residents add 8% tax

Shipping and handling prepaid in continental United States

NAME: ____________________________
ADDRESS: ____________________________
CITY: ____________________________ STATE: ______ ZIP: ______
SIGNATURE: ____________________________
PHONE (INCLUDE AREA CODE):

Please check method of payment:
- Check Enclosed
- BankAmericard No. ________
- Master Charge No. ________
-Expiration date: ________

Circle 360 on inquiry card.
Continued from page 12

and hazardous 117 VAC power trace running long distances on a power board. The big boys have their equipment tested and listed by UL. It's a necessity if they expect commercial sales. If personal computer manufacturers want the kind of mass market you've been predicting in BYTE, they will have to follow suit. This stuff moves in interstate commerce, and if vendors sell a shock or fire hazard to some naive user, they can expect to be spanked by the Feds, something that tends to be a corporate hurt, and will damage this fledgling part of the computer industry. All the manufacturers who haven't, should clean up their acts before the Feds force something distasteful on them.

William R Hambleton 946 Evans Rd Nashville TN 37204

A NEW SUBSCRIBER COMMENTS

How long have you been publishing BYTE? Are any back issues available, or reprints of articles in back issues? Do you publish an index to previous issues? Could you recommend a book, or books, that are readily available, or list their sources on microcomputer soft and hardware, particularly software (from machine language level to high level programming languages)?

Daniel Owen Jenkins VIII Box 201 RD/F Clinton Grv Weare NH 03281

We have been publishing since September 1975. No back issues are available; however, The Best of BYTE Volume 1 at $12.95 contains most of the editorial materials from issues 1 to 12. An index is available for Volume 1; send a self-addressed stamped envelope.

FASTER MULTIPLY?

ENTRY: A=multiplier; E=multiplicand

ENTRY: A, E, C unchanged; D =0; HL=product

PRODUCTS NEEDED

The following are some things we think are needed in this field:

1. A real time clock-calendar board for the S-100 bus with an IO port and recharging circuit which would display time, date and day of week on video monitor or other peripheral. Calendar would be at least 200 years. Would accept input to start clock or reset it, such as a 60 Hz line signal, or a time signal from WWV, or a manual signal. Could be programmed (EPROM) to exhibit holidays and special dates or times, or emit a signal to initiate some action at a preset time and date. Would be designed to be highly resistant to transients in power supply. Would have 12 hour AM-PM or 24 hour switch selectable option, and a local standard time or daylight savings time or Greenwich time switch selection feature.


3. A generalized converter board for the S-100 bus, under software control, which could enable the user to readily recode keyboards, printers, and video displays with more than a few keystrokes. Could be used as an ASCII-EBCDIC converter, or could convert a keyboard from a standard QWERTY keyboard into a Dvorak Standard keyboard with a single command, or convert an ASCII into an APL or other special character keyboard.

4. A conversion to standardize the bus for the coming 9900 family of 16 bit processors before different buses proliferate.

5. A directory or clearinghouse of resources in this field, so that persons wishing to contact others doing or able to do things of a certain kind could do so.

Jon D Roland Micro Mart 1015 Navarro San Antonio TX 78205

COMMENTS ON SELECTRIC IO INTERFACING

As one who has designed an interface for both input and output between a Selectric 731 and my Z-80 system, I was interested to read Dan Fylstra's article “Interfacing the IBM Selectric Keyboard Printer,” June 1977 BYTE, page 46. Dan is to be complimented on a very good technical description, especially when compared with the naive and often just plain inaccurate information which has been published recently on this subject. There are several supplementary points which may be of interest to BYTE readers:

a) The office Selectric is a light duty mechanism in the same sense in which the term is used for the Teletype Model 33. It won't fail long time but the office Selectric is less rugged mechanically than the "heavy duty" mechanisms designed for use as computer IO devices. In my opinion, little reason exists to attempt to convert an office Selectric for computer use since a mechanism designed specifically for that purpose is available at a comparable price.

b) The heavy duty Selectric IO mechanism was marketed by IBM in two basic forms. The Selectric IO mechanism itself (without any switches, wiring harness or covers) was made available to original equipment manufacturers (OEMs) to be incorporated into non-IBM terminals such as the Dura, Tel, Anderson Jacobson, Dalel, etc. The designation of these mechanisms was 745 (15 inch carriage). They are often marked SER (meaning special engineering request) to denote modifications made by IBM at the manufacturing firm's request. These OEMs installed their own magnets, switches and interface electronics to achieve their own individual products. Many of these are now available on the surplus market. I have also seen some of these IO mechanisms which have been partially outfitted with IBM magnets for use as output printers only. Where OEM mechanisms have been used in a non-IBM product, IBM will service the Selectric mechanism only and not the magnets, switches or electronics.

c) The keyboard form of the heavy-duty Selectric IO mechanism is the one which IBM outfitted for use as an electronically driven IO device by installing its
MPI Flexible Disk Series B51 is a new generation of 5½” flexible disk products employing a ‘band driven’ head positioner providing very fast, highly accurate access eliminating ‘hysteresis.’ A new generation of floppy's is born featuring:

**Data Integrity** Using the ‘band drive’ concept for precise head positioning, better error rates are realized. Decreased mechanical wear guarantees diskette interchangeability for the life of the device.

**Reliability** The Series B51 uses 30% fewer components than competitive devices and together with the low power Schottky TTL inherently provides you the user 10,000 hours MTBF and no preventative maintenance.

**Low Cost Storage** The Series B51 is very small. Its dimensions are 3.25 x 5.55 x 8.0 inches and weighs only 3 lbs. Using double density encoding techniques the device can store up to 250K Bytes (unformatted). Single quantity price is $320.

The S100 Bus Controller Available to the hobbyist is the R100D intelligent double density controller compatible in architecture to the S100 bus and the Series B51 or its larger family member the Series 882. It features board microprocessor, DMA, simple macro commands, multiple sector transfer, formatting, up to 4 drive control, and requires no overhead software. Single quantity price is $299.

Also available is a complete system featuring single or dual drives, power supply, cabinet and a S100 Controller with a disk operating system. Single Drive System $699. Dual Drive System $999.

**MPI is your Company** Follow the ‘star’ and you will get the best price/performance buy in the market. For further information or dealer/distributor packages, call or write Keith Ullal, *Micro Peripherals, Inc.*, 8724 Woodley Avenue, Sepulveda, California, 91343 (213) 894-4076.

Bank of Americard MasterCharge Personal Check Accepted
own magnets and switches. These are the Models 73, 731 and 735 (hereafter “73X”) which are out of production in this country; I’m told that they are still produced in Europe. The 73Xs were used as the 360 console device, in MTSs (with some wire modifications) and were available to those who wanted a complete unit (less interface electronics) which IBM would service. Current maintenance agreement rates for 73Xs are about $150 a year. The 73Xs came in three flavors: BCD, Correspondence and MTST.

2. In considering which of the three kinds of Selectrics to use with a particular piece of equipment, the first factor to consider is the nature of the output. Feedback contacts are placed on the Selectric mechanism without this feature, and the position of these contacts determines which one of the three Selectric IO typewriters (IBM Form #241-5737-0, July 1973).

3. There is a great deal of confusion concerning the existence of a "print" magnet. Some OEMs have in the worst-case timing delays must be introduced below 10 characters per second because worst case timing delays must be introduced after each character or machine operation; and wear on the mechanism increases in proportion to the volume of output. On these points, IBM comments:

### Feedback contacts are timed to permit initiation of a “next cycle” prior to the end of the “closed loop” operation, and wear on the mechanism increases in proportion to the volume of output. On these points, IBM comments:
Texas Instruments makes MOS EPROMs even more affordable.

TMS 2708 now $21.50*  
The industry standard.

TMS 27L08 now $26.15*  
The low power 8K.

TMS 2716 now $36.90*  
The 2708 times two.

Remember a few months ago when EPROMs were expensive and hard to get? Due to TI's leadership they have become available microcomputer building blocks.

Prices have dropped dramatically; availability is excellent. Credit TI's high-yield, high-volume production.

TI's highly cost effective EPROMs feature a rugged, high-integrity ceramic package with sturdy gold-plated pins to withstand the repeated handling and insertions associated with reprogramming. And a gold-alloy-sealed lid for superior hermeticity.

TI offers a choice of three production EPROMs—all from stock.

- TMS 2708. The industry standard 8K EPROM. Fully TTL compatible.
- TMS 27L08. The industry first low power 8K EPROM fully compatible with the 2708. But less than one-half the power dissipation and 10% power supply tolerance.
- TMS 2716. A 2708 times two. Twice the memory (16K) in the same space. An economical plug-in upgrade for 2708s. And TI's 16K 2716 uses less power than a single 2708.

To order the affordable EPROMs, call your nearest authorized TI distributor listed to the left.
application. Even though you may de-bounce the transmit contacts, I would recommend waiting the necessary 3 to 5 ms until the bounce from the making of C1 disappears before latching the state of the transmit contacts.

I am currently considering whether there is sufficient interest in a complete, plug-in-and-go I/O interface to the 73X to warrant production of my design which meets all IBM specifications. I would offer a completely assembled and tested interrupt driven card for the S-100 bus with on card PROM for driver software, cable with connectors, and power supply for either the 24 V or 48 V magnets. The price would have to be $399 for the card (including software in PROM) and about $100 for the 50 wire cable and power supply. I would appreciate receiving indications of interest from readers who would find such an interface attractive.

Beardsley Rum II
3045 Ordway St NW
Washington DC 20008

CLOSED LOOP SELECTRIC
MORE COMMENTS

This letter is in reference to Mr Fylstra's article on interfacing to Selectrics, which appeared in your June 1977 issue, page 46. Although the methods he is suggesting are simple, they have several serious drawbacks.

First, he does not use the keyboard lock mechanism during the time that the Selectric is under computer control. This is important to prevent accidental key-board inputs which may very easily cause damage to the machine.

Also he suggests using a 5 V supply driving TTL for interfacing the contacts. IBM specifies a minimum voltage and current of 10 V at 10 mA for reliable operation of these contacts.

Lastly the method of driving he suggests is a form of open loop mode. Since the Selectric is not a synchronous device, output rates are only nominal. The importance of operating the machine in a closed loop mode cannot be overstressed. This is the only method recommended by IBM. Using the closed loop mode not only minimizes wear and tear on the Selectric but also provides output at the maximum rate of the machine.

I speak from experience, since our firm is in the business of manufacturing various forms of Interfaces to Selectric mechanisms.

John Schwartz
IBEX
1010 Morse Av, Suite 5
Sunnyvale CA 94086

APL CHARACTER GENERATORS ARE AVAILABLE

I have noticed various letters in your letters column from APL enthusiasts. Our firm produces a CRT terminal with an APL character font and keyboard (see June 1977 BYTE, page 24). The APL character generator is a small piggyback board which can directly replace a 2513 and plug into its socket. (Bit 7 is wired through normally unused pin 23 of the 2513.) On the keyboard we use decals mounting on the front keycap face. These decals are white on clear, of excellent quality and durability. We would be pleased to make these items available directly or through distribution. Interested persons may write me directly.

M C Volker
President, Volker-Craig Ltd
Waterloo Ontario N2J 321
CANADA

A NOTE FROM A EUROPEAN READER

I had my computer shipped air freight, and when it arrived, I found a very simple 50 Hz modification for my Southwest Technical Products CT1024. I cut the 60 Hz reference line. I assume this does bad things to a phase locked loop somewhere (my documentation isn't here either) but after it warms up it seems to work fine.

Bruce Turrie
Riverside Tower Esmorelitaan 3
POB 3, Apt 68, B-2050 Antwerpen
BELGIUM

USING ADCs FOR TEST INSTRUMENTS?

I imagine this could have a great deal of interest to your readers. If you cannot develop a solution to this problem maybe your readers can.

I am working on a research project for Scott Community College. I am working on a hardware and software scheme that will make a color TV into an oscilloscope. I am using the TV Dazzler and the D+7A interfaces along with the IMSAI 8080. I want to do the rest of the task through software. I am having problems designing the software needed in conjunction with these interfaces. Any help that can be provided will be greatly appreciated. It need only cover audio frequencies. It need have no extended sensitivity or multicolored display.

John Orvis
2125 Olympia Dr
Bettendorf IA 52722

A good idea for an article!
SELF-MODIFYING CODE

The letter "A Critique of Self-Modifying Code" in the June 1977 issue of BYTE will serve only, I predict, to discourage some readers from sending material to BYTE for publication. The BARC routines, applied within the context for which they were intended, are certainly worthwhile.

Phil Rynes
6824 Roberts Dr
Woodridge IL 60515

NEED VA APPROVED COURSES?

Some time ago one of your readers wrote that he would be interested in taking a VA approved course in which the student would build a microcomputer. This subject has also been of interest to me since I am presently taking a course in digital electronics at The National Radio Institute in Washington DC. NRI is one of the leading VA approved correspondence schools in the country.

I wrote the president of NRI, J F Thompson, concerning the subject of microcomputers. The attached letter is the warm response I received to my inquiry. I thought some of your readers would be interested.

Robert N Smith
1617 Grunther Av
Rockville MD 20851

McGraw-Hill Continuing Education Center
330 W 42nd Avenue N W
Washington C 2005
Telephone 202/234 1600

Pres and

January 5, 1977

Mr. Robert N. Smith
1617 Grunther Avenue
Rockville, MD 20851

Dear Mr. Smith:

I received your letter of December 19, 1976, and appreciate your interest in whether NRI is considering a course in Microprocessing with microcomputer kits. The answer simply is "yes." We believe that a technology program on microprocessors will help the computer industry bridge the shortage of service technicians.

Our thoughts at this time are to expand the Digital Computer Electronics course to include microprocessing and kit hardware. The microprocessor benchmarking kit will contain power supplies, a clock oscillator, a programmable read only memory, lab indicators, and control switches. When used in the course, it will serve as a teaching tool to reinforce the material in the lessons. It will provide real programming experience on a commercial microprocessor. Further, the design calls for its use with a number of peripheral devices. The specific devices, like CRT, video displays, TV, cassette recorder, disk, and others have not yet been determined.

We are confident that NRI will offer microprocessing courses that have high quality, a reasonable price, and be approved for veterans training. It is too premature now to provide you more specific information.

Again, thank you for taking the time to write me. Good luck in your studies for 1977.

Sincerely,

J.F. Roth
J. F. Thompson

I/O Board
Complete Kit $150.00
Assembled $240.00

4K PROM/RAM Board
Kit (less memory) $60.00
Assembled (less memory) $80.00

Take a look at these board features

I/O Board Model IO4
• 2 serial I/O ports (2 in and 2 out) with full handshaking
• 55 to 300 baud rate selection for both input and output
• 2 parallel (2 in and 2 out) I/O ports
• 20/60 mA current loop (with optical isolators) or EIA serial interfaces
• Independent DIP switch address selection for both serial and parallel ports
• Interrupt capability

4K PROM/RAM Board Model M89
• Fully buffered address and data lines
• Jump on reset circuitry allows execution of programs at any 1K boundary by hitting reset or may be modified to jump on power-up
• Jump address and enable, board address, memory protect, and wait states (0-2) all DIP switch selectable located conveniently at top of board
• RAMs and PROMs may be mixed in any 256-byte increments
• Monitor program available

All Cybercom "blue boards" are made of hi-grade glass-epoxy with gold contacts and include TI low profile sockets.

Contact your local computer store today for more information on the new I/O and PROM/RAM boards, or write direct to:

A division of Solid State Music
2102A Walsh Avenue
Santa Clara, CA 95050
(408) 248-2707

We're the blue boards.
Ohio Scientific advances the state-of-the-art of small computers.

- Challenger II with our ultra-fast 8K BASIC in ROM. Now you can own a computer with full BASIC and plenty of user workspace for as little as $298.00. And the BASIC is there the instant you turn the machine on!

- Challenger III is the remarkable computer which has 6502A, 6800, and Z-80 processors. This computer system allows you to run all software published in the small computer journals, yet, it costs only about 10% more than comparable single processor computers.

- Challenger Single and Dual Drive Floppy Disks. These full size floppy disks are available in kit form or assembled at about the same prices as our competitors' mini floppies. Yet, they store three times as much data as the minies.

- Ohio Scientific's new 9 digit precision business BASIC is only slightly slower than our ultra-fast 8K BASIC. Still faster and more powerful than anyone else's 6 digit precision BASIC.

- Our incredible new 74 million byte disk drive. That's right, 74 million bytes is available for as little as $6,000.00 complete with interface for any Ohio Scientific computer. This new disk is quite possibly the world's highest performance data storage device. It features an unbelievable 34 millisecond average access time and an ultra-fast data transfer rate.

- Now is the time for you to dump your 1974 design vintage S-100 computer and move up to the state-of-the-art!

For more specifics, send $1.00 for our new Fall Catalog.

OHIO SCIENTIFIC

11679 Hayden
Hiram, Ohio 44234
If you're just getting into personal computing and are buying your first machine, you're probably confused by the myriad of companies and products available.

However, there is one simple guideline you should follow when choosing your first computer. Be sure that it is capable of giving you full floating-point BASIC the instant you turn it on. Machines with full 8K BASIC in ROM cost as little as $298.00. Why should you settle for anything less?

**Model 500**

The Model 500 is a fully populated 8 x 10 P.C. Board with 8K BASIC in ROM, 4K RAM, serial port and Ohio Scientific Bus compatibility for instant expansion. All you need is a small power supply (+5 at 2 amps and −9 at 500 MA) and an ASCII terminal to be up and running in BASIC. And all for only $298.00.

**Challenger IIIP**

The Challenger IIIP from Ohio Scientific is the ideal personal computer complete with BASIC in ROM and plenty of RAM (4K) for programs in BASIC.

Complete with an audio cassette interface, the Challenger IIIP uses a full computer keyboard, not a calculator keyboard.

In addition, the Challenger IIIP comes complete with a full 64 character-wide video display, not a 40 character display. The user simply connects a video monitor or home TV set via an RF converter (not supplied) and optionally, a cassette recorder for program storage.

The Challenger IIIP comes complete with a 4 slot backplane and case for only $598.00. Fully Assembled.

**Super Kit**

The Super Kit is a 3 board set with a 500 board (like the Model 500) without the serial interface. The ROMs are configured for use with the included, fully assembled 440 video board to provide a full BASIC computer and terminal.

The Super Kit also includes a fully assembled 8 slot backplane board which gives you 6 open slots for expansion.

To be up and running in BASIC simply plug the boards together, supply power (+5 at 3 amps and −9 at 600 MA), add an ASCII parallel keyboard plus a video monitor or TV set via an RF converter (not supplied).

Total price for the “kit” $398.00.
Any serious application of a computer demands a Floppy disk or hard disk because a disk allows the computer to access programs and data almost instantly instead of the seconds or minutes required with cassette systems. In real-world application of computers, such as small business accounting, a cassette based computer simply takes too long to do the job.

Ohio Scientific offers a full line of disk based computers utilizing full size floppy disks with 250,000 bytes of formatted user work space per disk. That's 3 to 4 times the work space of mini-floppies.

**Challenger II**

Challenger II is available with a single or dual floppy disk and a minimum of 16K of RAM instead of ROM BASIC. The disk BASIC is automatically loaded into the computer so there is no need for ROMs.

Ohio Scientific's powerful disk operating systems allow the computer to function like a big system with features like random access, sequential, and index sequential files in BASIC and I/O distributors which support multiple terminals and industry-standard line printers.

Challenger II's with disks can have the following optional features:
- 16 to 192K of RAM memory
- Single or dual drive floppys
- Serial and/or video I/O ports
- Up to 4 independent users simultaneously
- Two standard line printer options
- Optional 74 Megabyte Hard disk
- Much more

Challenger II disk systems are very economical. For example a 16K Challenger II computer with serial interface, single drive floppy disk, BASIC and DOS costs only $1964.00 fully assembled.
Challenger III

Ohio Scientific proudly announces the ultimate in small computer systems, the Challenger III. This computer has a 3 processor cpu board equipped with a 6502A, 6800, and Z-80.

This system allows you to run virtually all software published in the small computer magazines!

The Challenger III is fully software and hardware compatible with Ohio Scientific products and can run virtually all software for the 6800, 8080 and Z-80 including Mikbug® dependent 6800 programs!

Incredible as this is, Challenger III costs only about 10% more than conventional single processor microcomputers. For example a 32K Challenger III with a serial interface and a dual drive floppy disk (500,000 bytes of storage) costs only $3481.00. Fully Assembled, complete with software. Terminal not included.

OHIO SCIENTIFIC
11679 Hayden • Hiram, Ohio 44234
Continued from page 6

"show off" sequences which were viewable at the NCC show. These included false color processing of digitized pictures, as well as dynamically changing kaleidoscopic abstract patterns.

Why Would Anyone Want . . . ?

The technological skeptics always pose a more or less unanswerable question of "why do such and so?". Some unnamed ancestor of western civilization probably received the same question in response to making the first wheeled carts, planting crops rather than gathering them wild, or tending and managing fires as the inspiration for the Prometheus myth. It is a repetitive pattern, uneasiness or fear when confronted with new ways of life.

I first ran into this form of question relative to the whole idea of personal computing when I seriously began exploring the possibility of building my own computer after the Intel 8008 first became available. I vividly remember a conversation with one of my professional associates at the time, which contained the question of why applied to personal computers: "Why would anyone ever want a personal computer?" Without answering the question, history has now shown that a lot of people do want such computers. At that time however, I attempted to weakly answer (my mind was made up, remember) with specific applications contrived on the spot as alleged justifications. I knew I wanted one, but I wasn't quite sure why . . .

Today, I would answer with a more general question and an equivalently general response. The more general question is "why would anyone ever want to use an arbitrarily chosen product of human technology (be it automobile, airplane, typewriter, computer,

---

Figure 2: A less direct display method which uses considerably less memory to generate a picture is the reprogrammable color character generator technique. In its most general form, each logical character definition contains a matrix of 4 bit color values for picture elements within the character. A display is formed by filling the 1024 character refresh table entries with character values pointing to the programmable character generator matrix. This form of display in the configuration shown requires only 9,216 bytes compared to the 32,768 of figure 1, a saving which is achieved at a price of considerable flexibility.
The first complete low-cost microcomputer system for home, business or education!

Radio Shack TRS-80

The TRS-80 is for people who want to use a computer now—without the delay, work and problems of building one. The system is fully wired, tested and U.L. listed—ready for you to plug in and use! Program it to handle your personal finances, small business accounting, teaching functions, kitchen computations, innumerable games — and use Radio Shack’s expanding line of prepared programs on cassettes. The 280-based system comes with 4K read/write memory and Radio Shack Level-I BASIC stored in read-only memory. Memory expandable to 62K bytes. Includes CPU, memory, keyboard, display, power supply, cassette data recorder, 300-page manual, 2-game cassette program. Designed and built in USA by Radio Shack. Only 599.95.

Clip and Mail Coupon Today!

Mail to: Radio Shack, Dept. TRS-80
205 N.W. 7th St., Ft. Worth, TX 76101

Send me more data on the TRS-80 microcomputer
- Description of applications, software and peripherals available through Radio Shack
- Owners’ newsletter
- Price list
- List of stocking stores and dealers

NAME ___________________________ APT. NO. ______
ADDRESS ____________________________________________
CITY ___________________ STATE ______ ZIP ______
Application Software!

You can buy software from anybody but ours works in your system.

We only sell one product, Quality.

We have been in business for over nine years building a reputation for providing a quality product at nominal prices — NOT what the traffic will bear. Our software is:

- **Versatile** — as most programs allow for multiple modes of operation.
- **Tutorial** — as each program is self prompting and leads you through the program (most have very detailed instructions contained right in their source code).
- **Comprehensive** — as an example our PSD program not only computes Power Spectral Densities but also includes FFT's, Inverse-transforms, Windowing, Sliding Windows, simultaneous FFT's variable data sizes, etc. and as a last word our software is:
- **Readable** — as all of our programs are reproduced full size for ease in reading.
- **Virtually Machine Independent** — these programs are written in a subset of Dartmouth Basic but are not oriented for any one particular system. Just in case your Basic might not use one of our functions we have included an appendix in Volume V which gives conversion algorithms for 19 different Bases; thats right, just look it up and make the substitution for your particular version.

Over 85% of our programs in the first five volumes will execute in most 8K Bases with 16K of free user RAM. If you only have 4K Basic, because of its lack of string functions only about 60% of our programs in Volumes I thru V would be usable, however they should execute in only 8K of user RAM.

All of our programs are available on machine readable media. For those that have specific needs, we can tailor any of our programs for you or we can write one to fit your specific needs.

**Scientific Research**

1712-B Farmington Court, Crofton, MD 21114

Phone orders call (800) 638-9194
Information and Maryland residents call (301)721-1148

Add $1.50 per volume handling, all domestic shipments sent U.P.S. except APO and P.O. Box which go parcel post. Foreign orders add $8.00/volume for air shipment and make payable in U.S. dollars only.

**Available at most computer stores**

Master Charge and Bank Americard accepted.

Our Software is copyrighted and may not be reproduced or sold.
piano, microwave oven, or television set)?'' I would now reply that individual use of the products of technology has the psychological and emotional justification of personal satisfaction, derived from being able to choose from a wider range of activities with such inventions than without.

The More Specific Whys of Color Displays

The advent of high resolution color displays for personal use is near, as I'll demonstrate later in this essay. I have some rough ideas as to the practical consequences of inexpensive color displays in a personal computing context. Any such comments must be viewed in the same light as my responses to the personal computing skeptic several years ago. I am already committed mentally to the concept of the color display with computing intelligence behind it as an artistic and expressive medium for individuals. My attempt to detail my commitment's consequences with specific uses is necessarily a naive first try at explaining the fascination of the concept, just as my earlier attempt to explain the fascination of personal computing by examples was a weak argument at best.

Versatility and Flexibility

Perhaps a major attraction of the color display is the generality it provides. There is a certain fascination with the realization of the general purpose generic form of any program or hardware conception.

Carrying this attitude into the realm of visual displays, I claim that the color display with relatively high resolution is inherently desirable because it represents the most general form of the concept of a two dimensional display. Every other form of a display is a subset of the general ability to paint arbitrary colors in arbitrary patterns on a color display screen. Any given pattern is a subset of the general set of all possible patterns one could make with "n" color levels and an "i by j" pattern of picture elements.

The consequences of this versatility and flexibility are the possibility of color emphasis and detailed representations in conventional applications plus a whole new set of possibilities in the area of visual arts.

In an article we printed in December 1976, Margot Critchfield of the University of Pittsburgh's Project Solo illustrated the
static artistic possibilities in several color renditions using a Cromemco TV-Dazzler and custom peripherals created by Thomas Dwyer and Leon Sweer. (The Dazzler is a first attempt at the personal color organ, a 64 by 64 element display with a limited range of eight colors and "off." The renditions which she illustrated are examples of the static representations of visual art which are possible merely using the color display as the equivalent of the traditional artist's media of oil or acrylic and canvas.

This use of the color display as a medium of static visual artwork is a necessary first step toward exploring the practical consequences of high resolution color display technology. With use of appropriate inexpensive mass storage devices, libraries of high resolution images can be conceived. (A full size floppy can hold about 500 K bytes of information with contemporary "double density" recording technology, or a total of about 16 unencoded pictures with a 256 by 256 grid of 4 bit color picture elements.)

In this mode of operation, the artistic user paints a picture on the display using a joystick or equivalent cursor control plus additional finger manipulated controls to select color, move patterns about with software, repeat patterns, etc. The key here is composition of a color image which is perceived statically. This is the mode of operation of the Project Solo Cybernetic Crayon mentioned earlier, and of the Cromemco Dazzler when it is used with a joystick.

Effective artistic use of this new form of visual imagery creation requires development of interactive software customized to the creator's tastes. The display by itself is not enough to make the facility complete, for it only becomes useful with the software equivalents of paint brushes and motion of the artist's arm. Distribution or copies of this form of the art can be done photographically, since the static image is what counts. In this sense, there is nothing startling or new about use of a color display and its computer backup for static images.

Art Forms Impossible Without Computer Controlled Imagery

Once the artist or experimenter (the two words are actually equivalent) learns to create a visual art work with computer aid the next step is to use the mechanism of the computer to produce effects which would previously have been difficult or impossible to achieve. Motion and change of images according to rules and techniques chosen by the artist are a part of the very act of composition in this new medium of dynamic visual art.

Here, we are talking about a dynamic and moving art form, the use of programming in a manner which can be directly perceived and understood by all viewers simply by observing it, just as music can be appreciated by anyone simply by listening. This is the true excitement of color imagery as an art form. The only visual antecedents are the use of film technology and choreography; but unlike much film and choreographic imagery, it is not constrained to images of human forms, since the display is general purpose and subject to various forms of abstractions and harmonies previously impossible with visual imagery. The key to this new art form is the time dependent algorithmic transformation of elements of the picture according to the artist's plan and implementation.

The time ordered nature of algorithmic visual arts which come from this source make a combination of the display imagery with music inevitable. If evolution of visual form with abstract or specific images is considered as a criterion, an artistic antecedent of this combination is found in the first experiments of Walt Disney in the form of the movie called Fantasia. (For those unfamiliar with that movie, it was a combination of classical music with cartoon technology which resulted in a feature length film.)

A later example of this combination of visual patterns with musical patterns is represented in the work of John Whitney over the past decade or so, using computers with photographic technology to make high resolution computer generated films which are synchronized to music. He gave an excellent taste of what can be done in his demonstration films and talks at the Personal Computing 76 show in 1976, and at the First West Coast Computer Fair this year. (He is continuing his work with equipment which has much in common with contemporary personal computing technology.)

A major tenet of John Whitney's concept of dynamic visual arts is the idea of visual analogs to the harmonies and melodic evolutions in the musical forms. With the visual processors we can achieve with today's technology, it is possible for many more individuals to begin experimentation with dynamic progressions of forms of sensation which include both visual and aural components. The coming of the high resolution, yet not inordinately expensive, color graphic display opens up the wider use of this art form.
The simplest examples of the algorithmic visual arts have been seen at the various exhibitions. Nearly every computer store which carries the Cromemco TV Dazzler uses the color version of the game of Life as a customer attraction, one mode of algorithmic visual art. A similar display sometimes seen is the colorful forms of a kaleidoscope represented in that product's 64 by 64 color matrix on a TV screen. Another example of a simple, algorithmic art form is the Color Eater program which Apple Computer uses to illustrate the capabilities of its computer and its integral 40 by 40 patch color display generator. But better (ie: high resolution) color technology is sure to follow in the near future.

But Art Is Not All...

The programmable versatility of the color display concept is applicable to more than just artistic purposes. From the artistic point of view, the display's content, whether static or dynamically changing, is the object of the exercise. But using the display as a part of the information processing system which is the personal computer is attractive because colors can convey additional information.

Bordering on the concept of dynamic art is the concept of the color display oriented video game. Space War works just fine on a black and white display, for the realism of the simulation only demands points in a two dimensional projection of three space. But consider the possibility of cartoon style animation applied to simulation games. If the game involves a park like setting in which the simulation players move, use the color display to represent that setting, with programs appended for generation of players' movement in the setting. If the game involves simulating a plane landing, or an automobile race, use programs to generate the moving effects on the screen, and variations of the background information. Such suggestions involve significant software development and processor bandwidth when the degree of realism becomes high; but, given the color display and a given processor's capabilities, there can be considerable improvement in the types of displays used with games.

Why put up with a simple numeric time readout for your digital clock software? With a color display for the output device, a very realistic analog clock face could be displayed with moving hands and styling to the user's tastes. The 256 by 256 by 4 bit resolution capability suggested as a near term technological goal should be more than adequate for this task.
Then consider the use of color in highlighting data. For example, one purpose for which I’d like to use a high resolution color display in the future is highlighting the various melodic themes and chord structures in digital representations of music on the display. Here the color information is used to enhance the different segments of a complicated structure, a principle which has been applied at many points in industry and which is certainly just as useful for personal data displays.

**Design Speculations . . . Color Displays**

While it is unlikely that any personally affordable color display will soon match the facilities of the Comtal 8000-S which got me started on this subject, improved resolution color displays are a very real possibility for personal computing within the near future. By improved resolution in a personal computing context, I mean displays within the bandwidth range of a standard color television set adapted for direct video input. Taking into account memory prices, and the capabilities of the color television device, this means basically some form of raster scan color generator with a 256 by 256 picture element resolution and four bits of color level information per picture element. I see two methods in which this hardware can be implemented, with varying capabilities. The simplest brute force technique is to have the image generation equipment incorporate memory directly to refresh the display. For the 256 by 256 by 4 bit display this brute force technique requires 32,768 (32 K) bytes of memory. (At a 40 by 40 matrix resolution, the Apple II computer’s video generator’s color mode uses this brute force technique with a smaller amount of memory.) Figure 1 shows conceptually how each byte defines two picture elements’ state of color through hardware which accesses the 32 K byte memory region to generate the display. When the price of a typical 32 K byte memory board is standing at about the $800 range assembled and tested for at least one product currently advertised (using 16 K dynamic chips), dedicating such a memory to a color display peripheral for perhaps $1600-2000 end user price is not unreasonable.

The second method of design I see coming is inspired by the video generator methods used in many video arcade games, and in particular the video generator of the Noval 760 computer, with which I became acquainted on a visit to Gremlin Industries in late 1976. This is the method of a reprogrammable character generator, which results in considerable compression of the amount of memory required to completely cover the entire plane of a display. (The inherent disadvantage, a trade off of function for less memory than the brute force technique, is that the picture must be represented as a closely packed array of sub-elements defined by the different character definitions, meaning that it is possible to run out of character definition space if complex patterns are involved.) Using this second method of indirect refreshing through a programmable character generator extended to include color information, a configuration which might be suitable for color is illustrated in figure 2. The particular design of figure 2 (which is not the only configuration possible by any means) uses 256 character definitions of 32 bytes per character, where each character has an 8 by 8 array of 64 picture elements of four bits per element. The hardware of such an indirect refreshing method scans the 1024 byte (32 by 32) array of characters in the refresh memory, then looks up the corresponding character definition in the user programmable color character generator. This form of the color raster display still gives resolution to a 256 by 256 grid, but requires only 9216 bytes compared to the 32,768 bytes of the brute force technique. As a result, one might expect to find this form appearing in less expensive color oriented peripheral products for personal computers.

With memory prices dropping consistently, I expect both general forms of color peripheral units to be appearing in the near future. The details will differ from these conceptual sketches, but the idea of the color display is here. There is no technological reason why such a display cannot be built and marketed to the readers of this magazine within the next one or two years.

---

**Attention Authors:**

As a “how to” for reader construction, high resolution color displays and software drivers for them are certainly high priority items for future articles to be published in BYTE.
Have you been waiting for a hard copy printer that doesn't come with a high price tag or very limited performance and features?

Your wait is over. The new Integral Impact is precisely what you've been looking for. It's a micropriced, max-performance, full-size, full feature, impact printer that's ready to plug directly into your mini or micro-computer system. No special hardware or software interfacing required. Just connect to any RS 232 or current loop serial data source, and you're in business.

The Integral Impact is a complete professional printer that's fully assembled, tested and ready for use. No extra-cost options, nothing to assemble or wire, no special software to write ... and no need, either, for expensive special paper. It prints the standard ASCII set of 64 characters using a 5 x 7 dot matrix at speeds to 120 characters per second ... on full-size, 8½-inch paper in fan-fold or roll form.

You can even print multiple copies with up to 132 characters per line. With a long-life (10,000,000 character) ribbon that is automatically re-inked during printing, and a rugged print mechanism with few moving parts, the Integral Impact offers you outstanding service and reliability.

The Integral Impact has a microprocessor-based controller that features variable character pitch sizes, multiple line buffering and enhanced, double-width characters. Serial baud rates from 110 to 1200 bits per second are supported or, optionally, the printer may be used with a parallel interface. Operator controls for power, printer mode and off-line paper motion are conveniently located on the front panel. Internal switch settings determine serial baud rate, character pitch and number of characters per line.

The micro-price: $745.00 complete.

Where can you get it? Send the coupon below to Integral and we'll ship direct to you. Or have us send you full details by checking the appropriate box on the coupon.

Make big printer performance an integral part of your system ... with the Integral Impact

Now...big printer performance at a mini-printer price.

I want big printer performance now. Circle 371 on inquiry card.

Here's my check for $

Bill my VISA Mastercharge account

Account No. Expiration Date

M C Interbank No.

Signature

*Handling and shipping charge: $5.00. Mass. residents add 5% sales tax.
Commodore's New PET Computer

One of the most interesting new products at the NCC show in Dallas was Commodore's new PET computer. Beneath its futuristic cover, the unit features 14 K bytes of read only memory containing an 8 K BASIC package, 4 K operating system, 1 K machine language monitor and 1 K diagnostic routine.

The display features 64 graphics characters as well as the standard 64 character upper case ASCII set. This gives the effect of high resolution when displayed on the built-in 9 inch (22.86 cm) video monitor. The keyboard is encoded so that by shifting to upper case, the user has access to the 64 graphics characters.

The 8 K extended BASIC package was designed by Microsoft, the people who have created a number of interpreters for personal computers. It features strings, integers, multiple-dimensioned arrays, 10 digit precision floating point capability, and "peek" and "poke" commands.

The price of the PET is $595 complete with 4 K bytes of programmable memory. The $795 version features 8 K bytes of programmable memory. All IO connections (excluding the built-in tape drive, keyboard and video display) are made via an IEEE-488 bus.

The PET is an excellent example of the true appliance computer: a neat, self-contained graphics oriented package designed for the mass market as well as for the serious experimenter.

Photo 1: Commodore's new PET computer at the NCC show in Dallas.

Photo 2: A closeup of the PET's unusual touch-sensitive keyboard with 73 keys and 64 shifted graphics symbols in addition to ordinary upper case ASCII characters.

Photo 3: A side view of the PET highlighting its modernistic lines.
You want to record your message verbatim—word for word—whether it's bits, bytes or "Dear Folks" translated into word processor language.

Our objective in manufacturing recording media for the electronics industry—digital tape cassettes, floppy disks, mag cards, computer cartridges—is to give you the finest, the best, the most dependable, the most cost-effective.

That means rugged, long-lived, abrasion-resistant recording media with superior magnetic qualities. If we made tires, they'd be steel-belted radials.

We delivered our first digital grade certified tape cassettes back in the beginning, 1969. We made the first commercial 3740-compatible floppy disks that didn't bear IBM's name. And the first Flippy® reversible flexible disks with anyone's name on them. The first mini data cassette is ours. And we've got the newest miniature flexible disk, the MD 525.

Now, Verbatim media. It's a new formulation of ferric oxides, an advanced macromolecular binder system to adhere it to the tough polyester film, and a process control system that demands over 200 separate quality checks before the material is cut, packaged, and certified to be 100% error-free.

The final quality check? "Make it pretty!" Our production people tell us that magnetic recording media is one of the rare instances in manufacturing where aesthetic appearance translates directly into final product quality. It has to look beautiful to work beautifully.

We have the formulas, the machines, the technology to make high quality recording media. But it takes the best people in the industry to deliver Verbatim disks, cards, cartridges and cassettes. You'll find them at your favorite retail computer store.
Let's face it. Loading and soldering PC Boards is not much fun for the kit builder. Even more important, it's the place where most of the trouble gets introduced. The real fun and education comes in running and testing boards.

Now the Semikit with fully tested IC's.

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

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

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

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

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

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

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

Other Semi's are coming your way.

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

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

By Chris Morgan, Editor

Over 36,000 people filled the Dallas Convention Center on June 13 to 16 to attend the biggest National Computer Conference ever. Virtually every manufacturer in the computing field was there either to exhibit or to take notes. But what made this year's NCC different from past shows was the NCC sponsored Personal Computing Fair and Exposition held concurrently in the Convention Center.
A Dallas Delight

Photographs by Charles Fito

Manufacturers of personal computers and related products turned out in force to display their wares before large crowds. Hopeful hobbyists and experimenters exhibited their home computer projects at the Personal Computing Fair (devoted entirely to noncommercial individual and group owned projects), and vied to win prizes.

Both sections of the show featured extensive seminars and panel discussions about present and future uses of computers; personal computing club representatives met to discuss the possibility of forming a National Club Congress.

It was gratifying to see long term NCC veterans mingling with hackers from all over the country in a congenial atmosphere filled with microcomputers that played games and dazzled the eye with elegant color graphics. One group learned from the other, and ideas were exchanged at extremely high rates.

Seeing everything at the show was quite a challenge: some 300 manufacturers were spread out over five football field's worth of exhibits upstairs. And the main floor exhibits were big: Data General, for example, brought an entire working planetarium controlled by a Nova minicomputer and capable of holding 50 people inside. Many of the "booths" were two stories high.

Photo 4: Commodore's new PET computer, a popular item at the show. The unit features 14 K bytes of read only memory including an operating system, diagnostic routines and an 8 K BASIC interpreter by Microsoft. It sells for $595 including 4 K bytes of programmable memory. A lengthy waiting list is predicted for the unit.

Photo 5: Deborah Ashman demonstrating her robot joey, who moved back and forth to the tune, "If I Only Had a Brain."

Photo 6: Wes Stewart working at the cabinet desk he built for his SwTPC 6800 computer system. The program running is a golf simulation.
high; others took up areas of 30 by 60 feet or more.

There was much to see in the Personal Computing section downstairs, too: Commodore's new PET microcomputer, a variety of the latest video games programs (including a tank game for micros), new music programs, talking computers, software packages for FORTRAN, business programs, floppy disks, and so on.

Summarizing the entire show is almost impossible: we found it inspiring (and sometimes frustrating!) to look at some of the state of the art devices, such as high resolution video displays, on view in the main exhibit hall.

From the looks of this convention, the computer industry is in good shape, and we look forward to an even bigger and better show next year.

Photo 7: Allen Isaacson demonstrates one of the exhibits at the Personal Computing Fair: a video game converted into a color graphics terminal that uses a Teletype instead of a joystick. The hardware was designed by Robert and Richard Benjamin, father and son, respectively.

Photo 9: Photographer Floto reflects on Sperry Univac's shiny display.

Photo 8: Personal Computer Fair first prize winner Tom Aschenbrenner shown with the system he amassed over a five year period from surplus computer parts. A Texas Instruments 980A controls a system which stores and transmits messages for the benefit of Dallas area hams.
John Montagna, computer engineer (above left), lead this successful network team in generating election results speedily, efficiently and reliably using predominantly TDL hardware and software. Montagna created three programs to get the job done. The text for a SWAPPER program was written and assembled using the TDL TEXT EDITOR and Z80 RELOCATING MACRO ASSEMBLER. The SWAPPER text and all debugging was run through TDL’s ZAPPLE MONITOR. The relocatable object code was punched onto paper tape. A MAIN USERS program updated votes and controlled air display. An ALTERNATE USERS program got hard copy out and votes in. The latter two programs were written in BASIC. Montagna modified the ZAPPLE BASIC to permit timesharing between the two USERS programs.

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

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

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

John Montagna

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

I was intrigued by the proposal to make “the world’s least expensive plotter” at the end of Robert D Grappel’s interesting article (“Give Your Micro Some Muscles,” March 1977 BYTE, page 35). Eliminating the conventional XY carriage mechanism and using model aircraft servos looks like a good start.

May I put forward a somewhat different mechanical arrangement which avoids the two problems you describe in connection with your design, viz the weight of servo 2 sitting on the end of servo 1 arm, and the complexity of the trigonometrical equations requiring solution?

In figure 1 may be seen my alternative set-up in which the two servos are both fixed to ground and arranged concentrically. The inputs \( \theta \) and \( \Phi \), rotations about the common pivot \( O \), control the XY position of the pen. The moving links comprise a 4 bar linkage which has been dimensioned to approximately satisfy the following conditions:

- When \( OA \) is fixed the point \( P \) traces a straight line locus.
- When produced this line passes through point \( O \).
- The velocity ratio between the angular displacement of the link \( OC \) and the rectilinear displacement of the point \( P \) is constant.

The upshot of all this is that we now, in place of an XY plotter, have an \( R\theta \) plotter! The radial distance of \( P \) from the origin \( O \) is proportional to the angle between the links \( OA \) and \( OC \) \( (\theta + \Phi) \). The polar angle is simply proportional to the angular displacement \( \theta \) of link \( OA \). It is now a straightforward matter to express \( X \) and \( Y \) in terms of \( \theta \) and \( \Phi \).

The professional may quibble about geometric distortion, but the homebrew man will discover an acceptable working range. I now confess that I have not actually built such a plotter myself. I am hoping someone else will be good enough to pursue the idea for me!

By the way I do enjoy your publication; it is most readable.

The suggestion you make is very attractive, especially when you consider the niceness of having the servos on concentric shafts. It is not clear, however, what the optimal geometry is for such a plotter. In figures 2a, 2b and 2c we did a paper and pencil exercise to look at a particular geometry chosen by trial and error among about three different extremes. In this geometry, we used a symmetrical quadrilateral for which dimensions of \( X \) and \( 2X \) were used, with the extension arm to the drawing point \( P \) located at a distance \( X \) from point \( B \). (Subscript notation on the points is used to identify cases.)

In figure 2a, we’ve drawn this geometry in a case which looks very promising. For three different values of the angular parameter corresponding to \( \Phi \) in figure 1, the construction produces an apparently straight line with \( \theta \) held fixed at some value. However, in cases of figure 2b and 2c, with \( \theta \) changed in either direction from its value in figure 2a, there is an apparent...
deviation δ from a straight line path for point P thru several states of angle Φ. Still, the construction looks quite promising if the problem can be set up analytically and parameters optimized for the smallest deviations as the angles are changed.

Another point worth mentioning: the useful range of the angles in this type of construction is very small. In going from the example of figure 2b to figure 2c, the change in angle θ is approximately 45 degrees. This small angular change would require gearing to spread the useful control range of a model aircraft servo (180 degrees) over a smaller angular displacement. Note also that in our choice of parameters for this example, the locus of points P does not intersect the origin when extended as a straight line. If the constraint that distance PC equal distance OC is observed, this will guarantee that P intersects the origin. However, this criterion only guarantees that the locus of all points reachable by the moving point P will pass through the origin, and experiments with a compass and straightedge will verify that this locus is not necessarily a straight line through the origin, but is an arc of a curve. . .

---

**Boards DO Something**

**CL2400**
Real Time Clock

$98—Kit $135—Assembled

If your system needs to know what time it is, our CL2400 is the board for you. The present time in hours, minutes, and seconds is always available for input, and is continuously updated by the highly accurate 60 Hz power line frequency. Need periodic interrupts? The CL2400 can do that, too, at any of 6 rates. Reference manual with BASIC and assembly language software examples included.

**PC3200**
Power Control System

PC3232 $299—Kit $360—Assem.
PC3216 $189—Kit $240—Assem.
PC3202 $39.50—Kit $52—Assem.

If your system needs on/off control of lights, motors, appliances, etc., our PC3200 System components are for you. Control boards allow one I/O port to control 32 (PC3232) or 16 (PC3216) external Power Control Units, such as the PC3202 which controls 120 VAC loads to 400 Watts. Optically isolated, low voltage, current-limited control lines are standard in this growing product line.

---

Canada Systems, Inc.

P.O. Box 516
La Canada, CA 91011
(213) 790-7957

---

Circle 374 on inquiry card.
Othello, a New Ancient Game

Othello is a 2 person board game based on the 100 year old game of Reversi. It resembles GO, but is much faster paced and easier to learn. If you are not already an Othello buff, this program will both teach you the game and provide an entertaining addition to your games library.

The game is normally played on an 8 by 8 checkerboard with pieces that are black on one side and white on the other. The game begins with four pieces making a checkerboard pattern in the center of the board as shown in figure 1. When it is your turn, you try to convert a run of your opponent's pieces, vertical, horizontal or diagonal, to your color. You can do this if you place your piece so that each end of the run is bounded by pieces of your color. Part of the action comes from the fact that you must either make such a conversion or forfeit your turn. The game is over after 60 moves or when both players must forfeit their moves.

The program has two strategies it can deploy against you, one elementary and the other simple. If losing to a computer would be too great a blow to your self-esteem, request the elementary strategy. After your confidence is fairly secure, you can risk a no holds barred encounter. Eventually you will tire of your prowess, and will want to modify the program to improve its play. Hopefully, the comments in the program will make its operation transparent and will invite modifications.

A few words about modifications that may be required to play at all. A random tie breaking strategy is used to provide variety, but it is not essential. If your BASIC does not provide random numbers, you can delete lines 1310 and 1320 with no significant change in performance.

At the outset the program asks if you want it to pause before making its move. This pause is most helpful with a display terminal, since it keeps the output from flashing by too quickly. The output of the board is done by the last subroutine in the program. It requires the ability to display at least 11 lines of 19 characters, and you will probably want to modify the program if your TVT cannot handle the 26 lines of 32 characters needed to see two boards and some accompanying output simultaneously.

Of course, if you use a relatively slow hard copy printer with this program, it will take a long time to type out all the boards. The obvious solution is to suppress this printout except on request, and use an actual board to record the positions of pieces for your own use.

Good luck, and remember: Othello was so named because it is a game of dramatic reversals.

Figure 1: These two example boards show the Othello game board at the beginning and end of a typical game. The game is always initialized to the configuration in figure 1a. Figure 1b shows the board at the end of a typical game. This game was played to completion with 60 moves. The computer was soundly beaten having only 20 pieces to the human player's 44.
Listing 1: A BASIC listing for Othello.

00010 REM OTHELLO
00020 REM PLAYS THE GAME "OTHELLO" WITH TWO STRATEGIES:
00030 REM
00040 REM 1. TAKE THE MAXIMUM NUMBER OF PIECES
00050 REM 2. ADD A BONUS FOR OUTSIDE POSITION
00060 REM BOARD IS THE ARRAY A, BOUNDED BY O'S (BLANKS)
00070 REM A = 0 for EMPTY SQUARE
00080 REM / B FOR BLACK SQUARE -- X (INTERIALLY -1)
00090 REM A = W FOR WHITE SQUARE -- O (INTERIALLY +1)
00100 REM I AND J ALWAYS USED FOR ROW/COLUMN INDICES
00110 REM I4 AND J4 STORE INCREMENTS TO THE 8 NEIGHBORS
00120 DIM A(9,9),I4(8),J4(8),C(8),S(2)
00130 REM INITIAL GREETING
00140 PRINT "GREETINGS FROM OTHELLO";
00150 PRINT "DO YOU WANT INSTRUCTIONS (Y OR N) ";
00160 INPUT X$ 
00170 IF X$ = "N" THEN 290
00170 IF X$ <> "Y" THEN 160
00180 PRINT
00200 PRINT "OTHELLO IS PLAYED ON AN 8 X 8 CHECKER BOARD."
00210 PRINT "ROWS NUMBERED 1 TO 8 AND COLUMNS A TO H, EXCEPT"
00220 PRINT "THE CENTER FOUR SQUARES, WHICH FORM THE"
00230 PRINT "PIECES IN THIS WAY IF IT IS AT ALL POSSIBLE."
00240 PRINT "IF IT IS NOT POSSIBLE, YOU FORBID YOUR TURN BY"
00250 PRINT "ENTERING 0 FOR YOUR (ROW,COL) MOVE.
00270 PRINT
00280 REM INITIALIZE
00290 REM "SHOULD I WAIT BEFORE MAKING MY MOVES (Y OR N) ";
00300 IF F2 = 0 THEN 560
00310 IF X$ <> "Y" THEN 410
00320 PRINT "OK. TYPING ANY CHARACTER WILL LET ME GO."
00330 IF X$ = "N" THEN 460
00350 IF X$ <> "Y" THEN 560
00360 INPUT X$ 
00370 IF X$ = "N" THEN 520
00370 IF X$ <> "Y" THEN 460
00390 INPUT X$ 
00400 IF X$ = "N" THEN 500
00400 IF X$ <> "Y" THEN 480
00420 S1 = 2
00430 S2 = 0
00440 INPUT S1 
00450 IF X$ = "N" THEN 500
00450 IF X$ <> "Y" THEN 480
00470 S2 = 0
00480 INPUT S2 
00490 IF X$ = "N" THEN 500
00490 IF X$ <> "Y" THEN 480
00510 FIX S1 ++
00520 B = -1
00530 M = +1
00540 M = 1
00550 M = 0
00560 M = 0
00570 FIX K = 2
00580 READ JN(K)
00590 NEXT K
00600 DATA 0,-1,-1,-1,1,1,1,1
00610 FOR K = 1 TO 8
00620 READ JN(K)
00630 NEXT K
00640 DATA 1,0,-1,1,0,-1,1,0
00650 FOR K = 1 TO 8
00660 READ JN(K)
00670 NEXT K
00680 DATA A,B,C,D,E,F,G,H
00690 REM SET UP A NEW GAME
00700 FOR I = 0 TO 8
00710 FOR J = 0 TO 9
00720 A( I,J) = 0
00730 NEXT J
00740 NEXT I
00750 A( 1,1) = W
00760 A( 1,5) = W
00770 A( 3,5) = W
00780 A( 5,5) = W
00790 C( 1) = 2
00800 M( 1) = 2
00810 N( 1) = 4
00820 Z = 0
00830 REM HUMAN'S CHOICES
00840 PRINT "DO YOU WANT TO HAVE X OR O ";
00850 C = W
00860 INPUT X$ 
00870 IF X$ = "X" THEN 920
00880 IF X$ <> "O" THEN 970
00890 C = B
00900 H = W
00910 H = B
00920 PRINT "DO YOU WANT TO GO FIRST (Y OR N) ";
00930 INPUT X$ 
00940 IF X$ = "N" THEN 1920
00950 IF X$ = "Y" THEN 930
00960 REM PRINT INITIAL BOARD
00970 GOSUB 3100
00980 GO TO 1690
00990 REM COMPUTER'S MOVE
01000 IF F2 = 0 THEN 1200
01010 INPUT X$ 
01020 B = -1
01030 J = 3
01040 T = C
01050 T2 = H
01060 REM SCAN FOR BLANK SQUARE
01070 FOR I = 1 TO 8
01080 FOR J = 1 TO 8
01090 IF A( I,J) = 0 THEN 1380
01100 REM FOUND A BLANK SQUARE
01110 REM DOES IT HAVE AN OPPONENT AS A NEIGHBOR?
01120 GOSUB 2620
01130 IF F1 = 0 THEN 1380
01140 REM FOUND AN OPPONENT AS A NEIGHBOR
01150 REM HOW MANY OF HIS PIECES CAN WE FLIP?
01160 REM (DON'T DO IT NOW)
01170 U = -1
01180 GOSUB 2620
01190 REM EXTRA POINTS FOR BOUNDARY POSITION
01200 IF S1 = 0 THEN 1380
01210 IF J - 1 < 0 THEN 1230
01220 S1 = S1 - S2
01230 IF J + 1 > 8 THEN 1260
01240 S1 = S1 - S2
01250 REM IS THIS BETTER THAN THE BEST FOUND SO FAR?
01260 IF S1 < S1 THEN 1380
01270 IF S1 > S1 THEN 1340
01280 REM A TIE; RANDOM DECISION
01290 REM THE NEXT TWO EXECUTABLE STATEMENTS CAN BE DELETED
01300 REM FOR A VERSION OF BASIC-WITHOUT-RANDOM NUMBERS
01310 R = RND(1)
01320 IF R > 0.5 THEN 1380
01330 REM YES
01340 B1 = S1
01350 J1 = J
01360 J1 = J
01370 END
Listing 7, continued:

02000 PRINT "SORRY, THAT DOESN'T WORK: TRY AGAIN";
02010 GOTO 1720

01390 NEXT I

02060 PRINT "OF MY PIECES"
02050 PRINT "OF YOUR PIECES"

01100 IF J = I THEN 1720

01310 IF 1 > A THEN 1720

01100 IF I < 0 THEN 1720

01700 PRINT "THAT GIVES YOU";
01710 PRINT X;
01720 IF X = 1 THEN 1710
01730 IF X = 0 THEN 1720

01100 PRINT "I CAN'T"

01700 PRINT "YOU ARE FORFEITING YOUR TURN (Y OR N)"

02470 INPUT X
02480 IF X = 1 THEN 1710
02490 IF X = 0 THEN 1720

01100 PRINT "THE SQUARE IS OCCUPIED; TRY AGAIN"

01300 GC TO 1710

01270 FOR J = 1 TO 1700

01260 IF C(J) = 0 THEN 1700

01100 NEXT J

01840 GOSUB 1620

01550 REM CHECK IF BLANK

01540 IF A(J) = 1 THEN "NO"

01530 IF A(J) = 0 THEN "YES"

01520 PRINT "HERE'S WHAT THAT SQUARE IS OCCUPIED; TRY AGAIN"

01490 GC TO 1700

01300 REM CHECK FOR LEGAL NEIGHBOR

01310 GOSUB 2070

01290 IF F1 = 1 THEN 1700

01280 PRINT "SUNNY, YOU ARE NOT NEXT TO ONE OF MY PIECES;"

01270 PRINT "TRY AGAIN"

01330 GC TO 1700

01250 REM CHECK IF LEGAL PIECE

01240 IF A(J) = 1 THEN 1700

01230 IF A(J) = 0 THEN 1720

01220 REM PRINT "SUNNY, YOU ARE NOT NEXT TO ONE OF MY PIECES;"

01210 REM EVERYTHING LEGAL; MAKE HUMAN'S MOVE

01200 REM PRINT "THAT GIVES YOU";

01190 PRINT X;

01180 PRINT "OF MY PIECES"

01170 X = 1

01160 GOSUB 2080

01150 IF S(J) = 0 THEN 2070

01140 PRINT "SORRY, THAT DOESN'T BLANK A ROW; TRY AGAIN"

01130 GOSUB 1710

01120 REM EVERYTHING LEGAL; MAKE HUMAN'S MOVE

01110 REM PRINT "THAT GIVES YOU";

01100 PRINT X;

01090 PRINT "OF MY PIECES"

01080 X = 0

01070 GOSUB 2080

01060 IF H(J) = 1 THEN 1700

01050 IF H(J) = 0 THEN 1720

01040 PRINT "HERE'S WHAT THAT SQUARE IS OCCUPIED; TRY AGAIN"

01030 GC TO 1700

01020 REM PRINT "END OF GAME WRAPUP"

01010 PRINT "GOODBYE";

01000 PRINT "YOU WON!"

00990 GOTO 2500

00980 IF C(J) = 0 THEN 2190

00970 IF C(J) = 1 THEN 2190

00960 IF C(J) = 1 THEN 2190

00950 IF C(J) = 1 THEN 2190

00940 IF C(J) = 1 THEN 2190

00930 IF C(J) = 1 THEN 2190

00920 IF C(J) = 1 THEN 2190

00910 IF C(J) = 1 THEN 2190

00900 IF C(J) = 1 THEN 2190

00890 IF C(J) = 1 THEN 2190

00880 IF C(J) = 1 THEN 2190

00870 IF C(J) = 1 THEN 2190

00860 IF C(J) = 1 THEN 2190

00850 IF C(J) = 1 THEN 2190

00840 IF C(J) = 1 THEN 2190

00830 REM PRINT "END OF GAME WRAPUP"

00820 PRINT "GOODBYE";

00810 PRINT "YOU WON!"
NOW! Every Business Can Own One

A secretary, accountant, and financial advisor all rolled into one for less than $1.50 per hour. Sound too good to be true? Let the ADMINISTRATOR do it for you.

Our "General Ledger Program" is a complete comprehensive business system designed to keep all of your company's records without the need for updating from other programs and there is no need to keep monthly tear sheets to be added together for the end of the year reports as our system will provide you with year end account totals for cash, accrual, hybrid and chart of accounts systems. This program generates over 30 major reports. Including: 941's, P/L's, Balance Sheets, and year end account totals for filing Federal Income Tax Schedule C's and/or 1120's plus a lease purchase plan and 24-hour field service in most areas. Hard to believe! For less than $250/month you can lease your very own, nothing else to purchase.

ADMINISTRATOR I includes a miniature micro computer; S-100 Buss, with over 65K of user RAM. No switches to set, Power-on operation, multiple I/O interfaces, line printer, Video Terminal, Double-Density Disk, Disk Extended Basic and applications Software diskettes complete with full documentation (includes General Ledger, Payroll, Word Processing, Medical A/R, A/P, Engineering, Statistics, more) includes "Help" and Tutorial Software.

ADMINISTRATOR I.......$8999.00*

Additional 600K disks optional.

Business Software Too!
Includes over $75,000.00 of business programs free and that's only part of what we're giving away. We built our reputation providing quality software at affordable prices. Now we are going to do the same with our Administrator I

Compare at $30,000 for other micros or $70,000 for mini's
*Add $100 for Express shipping and handling • No Purchase Orders — include 50% deposit with all COD orders.

SCIENTIFIC RESEARCH
1712-B Farmington Court • Crofton, MD 21114
(301) 721-1148
Dealer Inquiries Invited
An APL Interpreter for Microcomputers

Part 3: Mathematical Processing

Introduction

In part 2 of this 3 part series (see last month’s issue) I covered the expression evaluation portion of an APL interpreter. This month’s concluding installment deals with mathematical processing. Mathematical processing is a 2 part function: recognition followed by interpretation. Both of these topics are covered here, followed by a short summary.

The Great APL Interpreter Contest

As an incentive to those experimenters who would like to try writing their own APL interpreters based on this series of articles, BYTE announces the Great APL Interpreter Contest. We will award prizes for APL interpreters (suitable for publication with royalties to authors) based on Mike’s flowcharts (or independent of them if you prefer).

Contestants are free to write their interpreters for any microprocessor they choose. Entries will, however, be judged on their suitability for use on small systems with a minimum of 16 K bytes of memory, as well as on programming elegance and efficient use of space. All of these factors should therefore be kept in mind.

Entries should be addressed to BYTE, attn: The Great APL Interpreter Contest, 70 Main St, Peterborough NH 03458, and must be postmarked no later than midnight, February 28 1978. Entries must be in the form of a publication quality manuscript which describes the implementation of the interpreter and which includes a listing of source code and object code. Contestants should also submit machine readable source and object code in the form of paper tape or cassette.

The winners (if any) will receive $1000 plus normal author payments, should the entry be chosen for publication in book form or as an article in BYTE. We reserve the right to choose more than one winner under the same terms.

Judging will be done by the editors of this magazine. Those seriously interested in entering this contest should call Carl Helmers or Chris Morgan at BYTE, (603)924-7217. May the midnight oil burn prosperously for all.
Recognition of a Valid Operator

The recognition process involves testing the current syllable in table SP for an appropriate noun. If the noun is found, then CODE will be set to reflect the appropriate specified operation. Figure 29 shows this process clearly for the recognition of monadic operators. The recognition of dyadic operators (see figure 30) is a little more involved, but the principle is the same. The extra processing involved is merely to recognize the two possible special forms of a dyadic operator: the inner product and the outer product.

CALL MOP(B)
B set true if a monadic operator found.
B set false otherwise.

CALL DOP(B)
B set true if a dyadic operator found.
B set false otherwise.

Figure 29: Recognition of a monadic operator.

Figure 30: Recognition of a dyadic operator.
Interpretation of Monadic Operators

After a valid operator is recognized, it is interpreted. Interpretation involves unstacking pointers from table SVAL, which points to the operands; performing the operation to create a result in table D; and finally stacking a pointer to the result back in table SVAL. Figure 31 begins the description of monadic interpretation. Generally, a monadic operator generates a result having the same shape as its operand; in two cases, however, this is not true. The process of reduction produces a result with one less dimension than the operand. Also, the monadic operators iota, rho, and ravel may produce results having greater or fewer dimensions than their corresponding operands.

So, in figure 31, after saving a pointer to the argument for the monadic operator, a test is made for a reduction or mixed operator and the appropriate process is executed. The interpretation of nonspecial monadic operators is then straightforward. First, since the result will have the same shape as the argument, it is given the same descriptor as the argument. Next, the monadic operator is performed on each value in the argument (or on the single value for scalar argument) to produce each value in the result. Finally, a pointer to the result is stacked in table SVAL and index DA is updated to point to the next free space in table D.
GET THE BEST FROM... SEALS...

MOST ORDERS SHIPPED WITHIN 10 WORKING DAYS

ORDER FORM

KIT ASSEMBLED

KIT ASSEMBLED

4KROM

8KSC-Z

68KSC

8KSC

ASSEMBLY AND OPERATING MANUAL $4.00

NAME

ADDRESS

CITY

STATE

ZIP

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

EXPORT INFORMATION

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

Canadian and European community customers use our EXPORT DELIVERY SERVICE. Guaranteed delivery to your address. We take care of all charges for customs clearance, duties, federal taxes and insurance. Add exactly 35% to the prices shown (45% in Canadian dollars). Allow 5 extra days for delivery. Minimum export order $100.00. DEALER INQUIRIES WELCOME.

4KROM [READ ONLY MEMORY]

- Will accept up to 16 via 1702-A or 128 via 1702-B from leading to Completion Program.

- Programming Available at factory for $2.50. 16K/150W can accompanies by January per mutual terms.

- Each 1702-A has 4, 4KB block selection, 10 pin header, 10 pin header.

- We work with the minimum power supply series 5100 dual inverter.

- Switch selects wall power source to that even the未曾 1702-A can work in your system. 81111 dual inverters.

- Greatly enhanced on both sides of double board.

- Components designed for Completion of A.C. Board.

- Kit Price... $109.00  Assembled Price... $119.00

68KSC [STATIC MEMORY CARD FOR STATIC MEMORY]

- 619 Words of Static Memory

- Access Time - 800 nanoseconds (280 nanoseconds on request)

- Memory Chip 4167 OR 1627

- Battery Backup

- Address Selectable 8 bits, 16 bits, 32 bits

- Low Power

- All IC's with socket

- Serial interface on both sides of kit board

- Kit Price... $259.00  Assembled Price... $268.00

8KSC-Z [STATIC MEMORY CARD]

- All P&P COMPAQ 8K and 100 don't communicate

- Access Time - 220 nanoseconds

- Address Expandable up to 16 bits

- Memory Chip 2162C, OR 2163C, OR 2164C

- Battery Backup, 6 bits to 8 words

- Address Selectable 8 bits, 8 bits, 32 bits

- Power Supply: None

- Current Use: Less than 20W output 18

- All IC's area, control, and Data not used when not operating

- All IC's supplied with IC holder

- Order Price on both sides of double DSC Board

- Kit Price... $285.00  Assembled Price... $295.00

ORDER DELIVERY: FROM STOCK TO 10 DAYS

TELEX NO. 55-4744  TELEPHONE NO. 615/693-8655

SEALS ELECTRONICS, INC.
Interpretation of mixed mode monadic operators, as shown in figures 32 and 33, is a little more heuristic. For the iota operator the argument must be a nonnegative integer scalar and the result will be a vector. The operators rho and ravel take any form of argument and produce vector results. Again, after creation of the result in table D a pointer to the result is stacked in table SVAL and index DA is updated.
The new generation of Diskette Drives is here and under control.

PerSci has it—a family of diskette drives "design-years" ahead of competitive drives—now available in complete low cost subsystems for interface to 8080, 6800 and other major microprocessors.

The Highest Performance Diskette Drives:
PerSci diskette drives, both single and dual head units, offer a combination of performance features unique in the marketplace while still maintaining compatibility in existing systems:
- Voice coil positioning for access speeds seven times faster than competitive drives (76 tracks in 100 ms)
- A low power all DC system reduces cost and assures high reliability
- Automatic electric loading simplifies operation and protects media
- Small size permits 5 single drives or 4 dual drives to be mounted vertically in a 19 in. rack

The Most Powerful Diskette Drive Controller
The PerSci Model 1070 Diskette Drive Controller puts the advanced performance of PerSci drives to work in microprocessor based systems. An IBM format compatible, "intelligent" controller, the Model 1070 will handle from 1-4 drives with minimum demand on the host system.

In fact, with addition of a power supply and keyboard to the PerSci subsystem, the user can perform many floppy disk routines without additional hardware or software. Controller features include:
- 8080 microprocessor
- Formatter/control IC
- 4K disk operating system
- 1K data buffer
- 8080 or 6800 microprocessor interfaces
- RS-232 serial interface optional
- IBM data format

An Economical Diskette Drive Subsystem
A complete subsystem including a single diskette drive (Model 70), the Model 1070 controller with interface and a controller-to-disk-drive cable is available in single units or OEM quantities. For double capacity, a dual diskette drive (Model 277) subsystem is also available.

Don't settle for yesterday's diskette drive. Get the new generation under control from PerSci, Incorporated, 12210 Nebraska Ave. W. Los Angeles, CA 90025 (213) 820-3764

Peripherals a Generation Ahead.

Circle 361 on inquiry card.
An 8080 Simulator

A simulator can be very useful to the personal computing experimenter for a number of reasons. The idea of a simulator is to perfectly duplicate the operation of a certain computing system using the instruction set of another computer. For example, suppose you have an 8080 program which you want to run on your 6800 system rather than completely write a new program. Since the simulator acts functionally like an 8080 processor to interpret the 8080 object code bit patterns, its hardware registers and program counter, it is possible for the 6800 system to run the 8080 program. Another attraction of a simulator is that in software it is possible to achieve a number of debugging and checkout functions as a part of the simulator itself, since the machine being run is always under total software control. When a simulator for machine X is run on machine X, these attractions can justify the use of the simulator.

A simulator usually has two drawbacks however: memory requirements and speed reduction. The speed reduction drawback is usually the more objectionable feature, although a fairly complex simulator can easily take up 8 K to 16 K bytes of memory space, depending on the complexity of the computer system being simulated. It is not unusual that thousands of instructions are executed before the functions of a single instruction can be simulated. This means that the simulator runs at a speed thousands of times slower than the actual computer does, which may be a difficulty in some cases.

Writing a simulator is a straightforward process as its functions are more or less well-defined by the documentation of the computer being simulated. All it has to do is to recognize the machine instruction, carry out its function, and record its effects. Of course you will have to keep track of the

---

**Figure 1:** Basic flowchart for the simulator. Each of the square blocks constitutes a separate subroutine.
values of various registers, stack pointer, program counter, etc.

If, however, the computer being simulated is written on the same computer doing the simulation, eg: an 8080 simulator on an 8080 microcomputer or a Z-80 simulator on a Z-80, then a lot of work can be simplified. As a matter of fact, I have discovered a very neat trick for writing such a simulator that greatly reduces the program length and increases its speed. An 8080 simulator was written on my 8080 computer which uses about 350 bytes of memory. Using the simulator, a quite sophisticated debugging program was also written using another 1 K bytes of memory.

General Description

In a simulator the hardware registers, such as the program counter, stack pointer, index registers, conditional flags, etc, have to be simulated by memory locations. The instructions have to be recognized and executed, and the memory locations simulating the hardware registers have to be updated accordingly. Take as an example the INR A, increment accumulator, instruction in the 8080. The value 1 is added to the location simulating the accumulator, and the locations simulating conditional flags are updated according to whether the carry, sign bit, etc, are affected. This has to be done for all the instructions in the instruction set, using software. A major portion of the simulator program is actually “wasted” doing this kind of work. In my 8080 simulator, I still have to use memory locations to simulate hardware registers, but I do not simulate the instructions at all. How can this be done? The idea is actually quite simple. Instead of simulating the process of each instruction by elaborate software, I let the computer actually execute the instruction. Since the computer is simulating a computer whose language it speaks, this can always be done. Of course before we can do this, we have to fetch the instruction to be simulated into an instruction buffer, a section that is under simulator control. Under the correct operating environment, the simulator should remain in control right after executing the instruction in the instruction buffer. The various parts of the routine that simulate a single instruction cycle are illustrated in figure 1.

For those instructions that do not change the values of the program counter abruptly, there is no problem. After actually executing the instruction in the instruction buffer, the instruction next to it is the next instruction to be executed. The only problem which occurs is when there is a jump or call instruction. If this type instruction is actually executed in the instruction buffer, control would be forced to go to whatever location is specified in the instruction. The trick used is to change the call or jump operand of the instruction to a location under simulator control. The original address must be stored somewhere, of course; more about this later.

There are five types of instruction that

Listing 1: The 8080 assembler listing for the 8080 simulator.

; *** 8080 SIMULATOR ROUTINE ***
SIM:  LHL D PC ; Get the value of PC.
       MOV A,M ; Load the instruction into acc.
       CALL REC OGN ; Call the recognizer.
       LHL D NB YTE ; Get number of byte and
       XCHG ; load it into (D,E).
       LHLD PC ; Get the value of PC again.
       DAD D ; and add these 2 num to get the next PC.
       SHLD PC ; Store it back at PC.
; *** FETCH ***
MVI D,4 ; Reg D, which keeps track of # byte to fill the
         ; inst buff, is initially set to 4.
LXI B, LBUFF+2 ; (B,C) is set to the last byte of inst buff.
NXT:  DCX H ; (H,L) points to the last byte of the inst in user prog.
       MOV A,M ; Fetch the instruction from user program.
Listing 1, continued:

STAX B ; and store at the inst buffer.
DCX B ; starting from the last byte.
DRI D
DRE E ; Rem: E contains the number of bytes of the inst.
NXT: ; Repeat until all bytes of the inst are moved.
XRA A ; Set A=0, the NOP inst.
DRI D ; Decrement D.
JNZ NEXT ; If inst buffer is filled, quit.
STAX B ; else fill the rest of the inst buffer.
DCX B ; with NOP inst.
JMP NXT2 ; until all done.

*** PREPARE ***

PREPARE: LDA TYPE ; Now check type of inst for special preparation.
DRI A
JNZ NOT1 ; If it is not a type 1 inst, goto NOT1.
LHLD HL ; Type 1 - PCHL inst.
SHLD PC ; PC is loaded with the value of HL.
RET ; Return.

NOT1: DRI A
JNZ NOT2 ; If it is not a type 2 inst, goto NOT2.
LHLD IBUFF+1; ; The operand of the call inst in the inst buffer
SHLD JCAD ; it is temporarily store at JCAD
LXI H,CAPT ; and the address CAPT is put
SHLD IBUFF+1; into the instruction.
JMP LDSTK ; Goto LDSTK directly.

NOT2: DRI A
JNZ NOT3 ; If it is not a type 3 inst, goto NOT3.
LHLD IBUFF+1; ; Similar to type 2 inst.
SHLD JCAD ; Type 3 - JMP and Jcond inst.
LXI H,JPT ; except JPT is used instead of CAPT.
SHLD IBUFF+1; ;
JMP LDSTK

NOT3: DRI A
JNZ LDSTK ; If it is not a type 4 inst, goto LDSTK.
LHLD PC ; Type 4 - RST inst.
; The next 8 inst set PC to the inst buffer.
XCHG ; Set (D,E) to PC value.
LHLD USESTK ; Load address of user stack.
DCX H ;
MOV M,D ; Move high value of PC.
DCX H ;
MOV M,E ; Move low value of PC.
SHLD USESTK ; Update user stack pointer value.
LDA IBUFF+2 ; ACC now contains the RST inst.
ANI 70 ; The higher 2 and the lower 3 bits are stripped.
MOV L,A ; Set L = low address.
XRA A ; high address is zero.
MOV H,A ; Set H = high address.
SHLD PC ; store (H, L) at PC.
RET ; Return.

LDSTK: LXI H,0 ; Store the simulator stack pointer value.
DAD SP ;
SHLD STACK ; at location STACK,
LHLD USESTK ; and set the stack pointer to the value stored
SPHL ; at USESTK, which contains user stack pointer value.
DRI A
JNZ RESTORE ; If it is not a type 5 inst, goto RESTORE.
LXI H,REPT ; Type 5 - RET and Rcond inst.
; The address REPT is forced onto the stack.

*** RESTORE ***

RESTORE: LHLD BC ; Restore values of reg B and C from loc BC.
MOV C,L
MOV B,H
XCHG
LHLD DE ; Restore values of reg D and E from loc DE.
LHLD SW ; Restore values of acc and PSW from loc SW.
POP PSW ; Finally, restore values of reg H and L from loc HL.
LHLD HL ;

*** INSTRUCTION BUFFER ***

IBUFF: DB 0,0,0 ; The 3 bytes of inst in the inst buffer are executed.

*** SAVE ***

SAVE: SHLD HL ; Values of reg H and L are saved at HL.
PUSH PSW ; Values of acc and PSW are saved at SW.
PUSH H;

need special preparation and clean up work. These are:

1) PCHL,
2) CALL and call with conditional execution,
3) JMP and jump with conditional execution,
4) RST,
5) RET and return conditional.

Detailed Description

The size of the instruction buffer is set to hold one instruction. It is a 3 byte memory location for an 8080. If the instruction to be simulated is less than three bytes, the buffer is filled with NOP instructions. A 2 byte memory location, which we call PC, is used as a pointer into the user program to be simulated. It always points to the first byte of the next instruction to be simulated. Memory locations are used for storing the stack pointer, accumulator, program status word and the other six 8080 registers. Since the value of the actual 8080 stack pointer is shared by the simulator and the user program, variables STACK and USESTK are used to store the simulator and user program stack pointer values during alternate use.

Below is a detailed description of each of the modules in the simulator routine. The program is listed in listing 1, and should be read in conjunction with the text.

Recognizer

The recognizer routine determines the number of bytes used by the instruction and the type of instruction to be executed. This is done by the use of a table (TBL). Each entry in the table consists of four bytes, which we call b1, b2, b3 and b4. The algorithm used is (using a BASIC-like statement):

IF (((inst AND b1) XOR b2) =0)
THEN number-of-byte=b3,
type-of-instruction=b4

AND and XOR are the logical AND and exclusive OR operations to be carried out bit by bit. The logical operation for each of the entries in the table is checked until one is found to satisfy the logical condition. If no entry satisfies the logical condition, we assume the instruction is a type 6, a 1 byte instruction. This method has the advantage over a table of 256 entries in that it saves space and processing time. The box at the end of this article gives examples of how it works.

The simulator starts by loading the value of the instruction's address and calling the
ARTEC Introduces
The Expandable 32K Elephant

The 8K-32K Expandable Memory That Grows With Your System

Now, for the first time, you can have a reliable true static memory that will grow with your system. Start with the board and 8K memory. Then add on one, two or three 8K increments of memory up to 32K. 250 ns access time. The Artect 32K Expandable Memory allows you plenty of room for memory and all necessary support hardware.

For five years Artect craftsmanship and reliability has been proven in tough industrial use. Now, you too can enjoy breadboards and memories that will work time after time. Boards like the GP 100 and the wire wrap WW-100. Send for an Artect Board, your order will be sent the same day as received.

Board & 8K of memory—$290.00
9K add on kits—$255.00 ea.
Full 32K board—$1,055.00

GP-100—$20.00
Maximum design versatility along with standard address decoding and buffering for S—100 systems. Room for 32 uncommitted 16 pin IC's, 5 bus buffer & decoding chips, 1 DIP address select switch, a 5 volt regulator and more. High quality FR4 epoxy. All holes plated through. Reflowed solder circuitry.

WW-100—$20.00
A wire wrap breadboard, similar to the GP 100. Allows wirewrap of all sizes of sockets in any combination. An extra regulator position for multiple voltage applications. Contact finger pads arranged for easy pin insertion.

TO ORDER: Use your Mastercharge or BankAmericard. Or just send along a money order. Your order will get same day service.

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

Please send me:
☐ 32K ☐ GP-100 ☐ WW-100
☐ I've enclosed a money order.
☐ Bill my ☐ Mastercharge ☐ BankAmericard No.

Name ____________________________________________
Address ___________________________________________
City _______ State ______ Zip

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

10% discount for students & computer club members.

Circle 378 on inquiry card.
Listing 1, continued:

```
SHLD SW ; Values of reg D and E are saved at DE.
XCHG
SHLD DE ; Values of reg B and C are saved at BC.
MOV H,B
MOV L,C
SHLD BC
LOA TYPE ; Load TYPE into acc.
CPI 5 ; Is it a type 5 inst?
JNZ STSTK ; Skip the next inst if not.
POP H ; Clean the garbage from the stack.
STSTK: LXI H,0 ; Store the user stack pointer value
DAD SP
SHLD USESTK ; at location USESTK,
LHLD STACK ; and set the stack pointer back to the value
SPHL ; at STACK.
RET ; Return.

; CLEAN UP ***

CAPT: POP H ; For Type 2 inst, clean up the garbage on the stack
LHLD PC ; and load correct return address onto
PUSH H ; the stack.
JPT: LHLD JCAD ; For Type 2 and 3 inst, load correct execution address
SHLD PC ; previously stored at JCAD and store at PC.
JMP STSTK ; Resume processing at STSTK.
REPT: POP H ; For Type 5 inst, the correct return address is on the
; top of the stack.
SHLD PC ; which is popped and stored at PC.
JMP STSTK ; Resume processing at STSTK.

; THE RECOGNIZER SUBROUTINE ***

RECOGN: LXI H,TBL-2 ; Load the starting address of the table minus 2.
RECOGN: MOV C,A ; It is temporarily stored at C.
AGAIN: MOV A,C ; The inst is restored from reg C.
INX H
INX H ; (H,L) points to the 1st byte of entry in the table.
MOV B,M ; Get AND mask.
ANA B ; Perform AND operation.
JZ BYTE1 ; If result is 0; either NOP inst or end of table.
INX H ; Get XOR mask.
XRA B ; Perform XOR with B.
INX H ; If result is 0 then the inst is not recognized yet.
MOV A,M ; else get and
STA NBYTE ; store number of byte of the instruction.
INX H
MOV A,M ; Also get
STA TYPE ; and store type of instruction.
RET ; Return.

BYTE1: MVI A,1 ; This is a 1 byte inst.
STA NBYTE ; Store 1 into NBYTE.
MVI A,6 ; This is also a type 6 inst.
STA TYPE ; Store 6 into TYPE.
RET ; Return.

; VARIABLE AND DATA ***

NBYTE: DW ; Number of byte of the instruction.
TYPE: DB ; Type of instruction.
PC: DW ; Program counter for the user program.
STACK: DW ; Simulator stack pointer.
USESTK: DW ; User program stack pointer.
SW: DW ; User program accumulator and status word.
DC: DW ; User program D and E registers.
DE: DW ; User program D and E registers.
HL: DW ; User program H and L registers.
JCAD: DW ; Temporary storage for Type 2 and 3 inst.

; The table used in the recognizer consisted of entries of 4 bytes each.
; Byte 1: AND mask    Byte 2: XOR mask
; Byte 3: Number of inst bytes    Byte 4: Type of inst.
TBL: DB 377,351,1,1 ; PCHL inst
DB 377,318,3,2 ; CALL inst
DB 307,304,3,2 ; Ccond inst
DB 307,303,3,3 ; JMP inst
DB 307,302,3,3 ; Cond inst
DB 307,301,1,4 ; RST inst
DB 377,311,1,5 ; RET inst
DB 307,300,1,5 ; Rcond inst
DB 317,001,3,6 ; LXI inst
DB 347,042,3,6 ; STA and LDA inst
DB 367,323,2,6 ; IN and OUT inst
DB 307,006,2,6 ; MVI inst
DB 307,306,2,6 ; 2 byte acc arith inst
DB 0 ; end of the table
```

recognizer. The recognizer should return the correct values for NBYTE (number of bytes) and TYPE (type of instruction). The value of PC is then updated tentatively, by adding NBYTE to PC. We say "tentatively" because it may be altered later if the instruction happens to be a type 1 to type 5.

Fetch

The instruction is then moved from the user program to the 3 byte locations in the instruction buffer. The loop NXT moves the instruction, while the loop NXT2 fills the instruction buffer with NOP instructions to take care of short instructions.

Preparation

The first five types of instructions have to be specially processed before the instruction in the instruction buffer is executed. Type 1 and type 4 instructions actually bypass the instruction buffer.

Type 1: PCHL

The content of PC is set to the value stored at HL.

Type 2: CALL and Conditional Call

A CALL or conditional call instruction, a CZ ADDR for example, is changed to a CZ CAPT instruction, where CAPT is the starting address of a routine that takes care of the preparation for CALL and conditional call instructions. The original address ADDR is stored at JCAD for further processing at CAPT. When the instruction in the instruction buffer is executed, and in our example, if the zero flag is not set, the next sequential instruction is executed. If however the zero flag is set, then it would branch to location CAPT instead of the original location ADDR. Either way we are in control.

Type 3: JMP and Conditional Jump

A JMP or conditional jump instruction is treated similarly to a CALL instruction, except that JPT is used instead of CAPT.

Type 4: RST

These instructions are simulated by first loading the value of PC into locations indicated by USESTK-1 and USESTK-2, and decreasing the value of USESTK by 2. This simulates the push stack operation. PC is then set to its correct value. The low PC value is decoded by using an AND operation. The high PC value is zero.

Before starting preparation work for
24 Channel LOGIC ANALYZER, complete with 2 cards and 3 sets of probes (only one set shown).

Features
- 24 channels with 256 samples each.
- Display of disassembled program flow.
- Dual mode operation — external mode analyses any external logic system. Internal mode monitors users data and address bus.
- Selectable trigger point anywhere in the 256 samples.
- 0-16 bit trigger word format or external qualifier.
- 10MHz sample rate (50ns min. pulse width)
- Synchronous clock sample with coincident or delayed clock mode.
- User defined reference memory.
- Displays and system control through keyboard entry.
- TTL Logic level compatible (15 pf and 15 µA typical input loading).
- Includes annotated source listing.

The DATALYZER

The Databyte Logic Analyzer (DATALYZER) is a convenient, flexible, high quality device. Efficient engineering has allowed a combination of features previously available in only the most expensive units.

Designed to plug easily into your S-100 Bus, the DATALYZER is a complete system — for only $495. Display of disassembled program flow is a standard feature, not an extra. And the low price includes 30 logic probes, so you can hook up immediately, without additional expense.

The DATALYZER is available in kit form ($495), and as a fully assembled device on two PCB’s ($595). Four-week delivery, a substantial warranty, and the Databyte, Inc. commitment to service make the DATALYZER a worthwhile investment. Begin debugging by sending the coupon now.

Displays in Hex Displays in Binary

| Please send me the 24 Channel LOGIC ANALYZER |
| Kit — (manual included) $495.00 (Wis, res, add 4%) |
| Assembled and Tested (manual included) $595.00 |
| Operators' manual only $7.50 |
| Delivery of all items in four weeks to: |
| Name ________________ | Address ____________________________ |
| City ___________________ | State _______ | Zip ______ |
| Telephone ________________ | |
| Payment Enclosed: ☐ Check ☐ Money Order |
| ☐ BankAmericard ☐ Master Charge Exp. Date __________ |
| Number __________________ | Signature __________________ |

Databyte, Inc.

P.O. Box 14
7433 Hubbard Avenue
Middleton, Wisconsin 53562
Tel: (608) 831-7666
return instructions, we have to save the simulator stack pointer value and set the stack pointer to the user program stack pointer value. There is no instruction in the 8080 processor that directly transfers the value of the stack pointer to registers. We have to use a DAD SP instruction by setting register pair HL to zero first.

Type 5: RET and Conditional Return

The return address (REPT) is first pushed on to the stack. The philosophy here is similar to the call instructions. If the conditional test for the instruction is satisfied, the program will return to REPT instead of the original returning point. The extra unwanted entries on the stack should be cleared. The address REPT is within the simulator and the program is still under simulator control.

Restoration

Since the value of the user stack pointer has already been restored, we only have to restore the values of the accumulator, program status word, and the other six registers from memory locations where they were saved. The only way to obtain the program status word is to push PSW into the stack and pop it up again, making use of the user program stack pointer temporarily. This means that the location USESTK must be properly initialized for the simulator to work. Once the stack pointer is set to a specific value in the user program, that value is used for our temporary work.

Instruction Buffer

By now the instruction buffer should contain the desired instruction, either the modified one or the original one. When this instruction is executed, we should go either to REPT, or to CAPT, or to JPPT, or to the instruction next to it.

Clean Up

If we ever come to the location CAPT, it means that the conditional test for the CALL or conditional call instruction (if any) has been successful and a call has been made. The top of the stack now contains garbage, and should be replaced with the correct return address, the value stored at PC. PC is then set to its correct value, the one that we have previously saved at JCAD.

The clean up work for the JMP and conditional jump instructions is similar except that no stack is involved.

If we come to the location REPT, it means that the conditional test for the RET or conditional return instruction (if any) has
been successful and a return has been made. The top of the stack now contains the true return address. It is then popped and stored at PC.

**Save**

The values of accumulator, program status word, registers and the stack pointer are saved in memory locations, reversing the process in RESTORE.

This completes our discussion for the simulation of a single instruction cycle.

**Conclusion**

This article has demonstrated how an 8080 simulator can be written on an 8080 computer. Similar simulations can be done on any other microcomputer, using the same basic technique. Writing such a simulator can be greatly simplified by making use of the existing host computer. The scheme I have shown is a subroutine that simulates an instruction cycle. It can be called repeatedly to simulate continuous execution. Various tests can be made each time before calling the subroutine, and execution can be suspended at any time if necessary. A combination simulator and debugging system can be easily designed and implemented to include the following features:

- A monitor routine that parses, recognizes the commands and branches to the appropriate routines.
- A single step routine that calls the simulator routine once and returns.
- A set break point routine that allows the user to enter break point addresses into the break table.
- A dump register and load register routine that displays and loads the values of various registers.
- An examine, deposit, examine previous, examine next routine that simulates the functions of front panel switches.
- A run routine that repeatedly calls the simulator routine. Before calling the routine it compares the value of PC (the program counter) with the entries in the break table, checks whether the value of PC is within the designated limits, and stops execution or gives warnings if appropriate.

Such a combination simulator and debug system can be very useful in developing and debugging programs written in assembler or machine language. Such a project may turn out to be too complex for beginners, but it is one that is worth undertaking as a challenge by more experienced programmers.

---

**COMPLETE FLOPPY DISK SYSTEM FOR YOUR ALTAIR/IMSAI $699**

That's right, complete.

The North Star MICRO-DISK SYSTEM™ uses the Shugart minifloppy™ disk drive. The controller is an S-100 compatible PC board with on-board PROM for bootstrap load. It can control up to three drives, either with or without interrupts. No DMA is required.

No system is complete without software: we provide the PROM bootstrap, a file-oriented disk operating system (2k bytes), and our powerful extended BASIC with sequential and random disk file accessing (10k bytes).

Each 5” diameter diskette has 90k data byte capacity. BASIC loads in less than 2 seconds. The drive itself can be mounted inside your computer, and use your existing power supply (1.9 amp at 5V and 1.6 amp at 12V max). Or, if you prefer, we offer a power supply ($39) and enclosure ($39). Sound unbelievable? See the North Star MICRO-DISK SYSTEM at your local computer store. For a high-performance BASIC computing system, all you need is an 8080 or Z80 computer, 16k of memory, a terminal, and the North Star MICRO-DISK SYSTEM. For additional performance, obtain up to a factor of ten increase in BASIC execution speed by also ordering the North Star hardware Floating Point Board (FPB-A). Use of the FPB-A also saves about 1k of memory by eliminating software arithmetic routines.

Included: North Star controller kit (highest quality PC board and components, sockets for all IC’s, and power regulation for one drive), SA-400 drive (assembled and tested), cabling and connectors, 2 diskettes (one containing file DOS and BASIC), complete hardware and software documentation, and U.S. shipping.

MICRO-DISK SYSTEM . . . $699
(ASSEMBLED) . . . . $799
ADDITIONAL DRIVES . . . $425 ea.
DISKETTES . . . . . . . . $4.50 ea.
FPB-A . . . . . . . . . . . $359
(ASSEMBLED) . . . . $499

To place order, send check, money order or BA or MC card with exp. date and signature. Uncertified checks require 6 weeks processing. Calif. residents add sales tax.

NORTH STAR COMPUTERS, INC.
2465 Fourth Street
Berkeley, CA 94710

Circle 359 on inquiry card.
Figure 31: Interpretation of the reduction operators.

(DIAUX+D(DIAUX)+1)+0

DIAUX+1 I 0
YES

REDUCTION OF A SCALAR

D(DA)--1
Z=D(AUX+X
X=D(AUX)+1

BUILD THE DESCRIPTOR

X>0
YES

D(2) X 0
YES

Z=Z+1
X=X+1

D(DA)=D(DA)*D(2)

ZI=D(DA)
ZI=D(DA)+D(DA)+1

ZI IS TOTAL NUMBER OF ELEMENTS IN RESULT
D(DA) IS NOW LENGTH DESCRIPTOR ADJUSTED TO PROPER LENGTH

X=AUX+2
Z=DA+2

TEST IF OPERAND IS A VECTOR

DIAUX+1 I 1
NO

RESULT IS A SCALAR

Z=DA+D(DIAUX)+1

NO

RESULT IS MATRIX OF ONE LESS DIMENSION

ZI=DZ X+1
Z=Z+1

COPY THE SIZE DESCRIPTORS, OMITTING THE LAST ENTRY

CODE=0 OF CODE
YES

IEL=0

FORM IDENTIFY VALUE

DIZ=SFLOAT
Z=Z+1
X=X+1

...
Interpretation of the Reduction Operators

Interpretation of the reduction monadic argument is one of the longer processes. The argument is any nonscalar value, and the result, which has one less dimension than the argument, is formed by applying the reducing operator iteratively to every element in the highest dimension.

Figure 34 illustrates the reduction process. First, the descriptor of the result is built. This descriptor will be the same as the argument descriptor except that the last dimension is omitted and, correspondingly, the length of the result may be smaller. Next, the identity value is determined by the type of reduction specified. If the last dimension of the input argument is zero, the reduction results in the identity value. Otherwise the elements of the result are computed by applying the monadic operator to the vector of elements along the last dimension of the input argument. Note that since reduction proceeds right to left across this vector, it actually results in the dyadic use of the monadic operator.

Interpretation of Dyadic Operators

Any routine designed for the interpretation of dyadic operators must, in contrast to monadic operators, be able to handle a myriad of possibilities. Not only are there the special cases of inner and outer products, but iota, reshape and concatenate operators must be specially handled. Furthermore, the nonspecial cases may have scalar arguments, nonscalar arguments of the same size and shape, or a scalar argument and a nonscalar argument. Each case requires some extra processing.

Figure 35 begins the description of dyadic operator interpretation. Two pointers are popped from stack SVAL. OPG points to the right argument of the dyadic operator and OPD points to the left argument. Next, the various processing method requirements are determined as described above. Inner and outer product interpreters as well as IOTA and CATEenate interpreters were not implemented in this version of the APL interpreter and will not be described here. The special processing then continues at one of the following blocks: RESHAPE, SCAL, SAME RANK and DIF RANK.
Calculating a Numerical Dyadic Result

Before proceeding further, look briefly at figure 36. Whenever a numerical dyadic result is needed, this subroutine is invoked. This saves the redundant coding which would result if the various special dyadic handlers each calculated their own results. Instead, the special handlers need only set up the required SFLOAT and VALUE arguments for processing by this subroutine. The result is a smaller, simpler and better structured program.

<table>
<thead>
<tr>
<th>Figure 36: Computation of a numerical dyadic result.</th>
</tr>
</thead>
</table>

| 52, ADDITION | SFLOAT + VALUE + SFLOAT |
| 53, SUBTRACTION | SFLOAT + VALUE - SFLOAT |
| 54, MULTIPLICATION | SFLOAT + VALUE • SFLOAT |
| 55, DIVISION | SFLOAT + VALUE / SFLOAT |
| 56, EXPONENTIATION | SFLOAT + VALUE # SFLOAT |
| 71, EQUALITY | FEQUAL |
| 72, INEQUALITY | FNOTEQUAL |
| 73, LESS THAN | FLESSTHAN |
| 74, LESS OR EQUAL | FLESEQ |
| 75, GREATER OR EQUAL | FGREATERTHAN |
| 76, GREATER THAN | FGREATER |
| 77, AND | AND_SUBR |
| 78, OR | OR_SUBR |
| 57, MAX | FMAX |
| 58, MIN | FMIN |
| 59, LOG | FLOG |

RETURN
• ANY NUMBER OF FILES MAY BE OPEN (IN USE) AT ONE TIME
• THE NUMBER OF FILES AND SIZE OF FILES 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 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 (pictured). Price: BFD-68 $795, BFD-68-2 $1139, BFD-68-3 $1479, SA-400 Drive $360.

The BFD-68 includes our Disc Operating System Software. Our software provides for a soft-sectored 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 transfer to and from the BFD-68. Price: SE-1 $29 on diskette or cassette.

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 or cassette.

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-FP 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-FP into your SWTPC 6800 and your ICOM into the P-38-FP 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 $229.

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. In 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. $529.

BANKAMERICARD, VISA AND MASTER CHARGE WELCOME.
Special Cases for Dyadic Operator Arguments

When both arguments of the dyadic operator are scalars, processing is simple (see figure 37). Similarly, for nonscalar arguments with the same size and shape, the scalar process is more or less applied iteratively (see figure 38). Where only one argument is a scalar, interpretation is a hybrid of figures 37 and 38. This is shown in figure 39. Here the scalar value is applied iteratively to the vector or array.

Figure 37: Dyadic interpretation for the case in which both arguments are scalars.

Figure 38: Dyadic interpretation for the case in which both arguments are nonscalar but of the same shape and size.

ALPHA Z-80 $495 ASSEMBLED

- 12 SLOT MOTHER BOARD
- 12 CONNECTORS (S-100 BUS)
- 17 AMP POWER SUPPLY
- HEAVY DUTY CABINET WITH FAN
- Z-80 CPU BOARD: All Sockets Included, Gold Contact Fingers, High Quality Glass Epoxy PC Board, Double-Sided, Plated Through Holes, Requires Only +8 VDC / 800 MA.

NOTE: 22 SLOTS/30 AMP POWER SUPPLY - $595.

ADS also sells IMSAI and NORTH STAR Assembled at Kit Prices + LEAR SIEGELER ADM 3A Assembled at $888.00. TERMS: Cash with Order. Prices include Freight. (N.C. Residents add 4% Sales Tax.)

ADS
Data Acquisition, Computation and Control
ROUTE 4 BOX 171A
BOONE, NORTH CAROLINA 28607

CIRCLE 382 ON INQUIRY CARD.
Figure 39: Dyadic interpretation for the case in which only one argument is a scalar.

DYADIC

AUX ← DA
Y ← 1

D(AUX) ← (D(Z+Y+1)
AUX ← AUX+1
Y ← Y+1

Y ≤ D(Z)?
YES

AUX ← AUX-1
Y ← 1
Y1 ← Z+2

NO

AUX NOW POINTS TO LAST ELEMENT IN ARRAY

Y ← Y = D(Y1)
Y1 ← Y1 + 1

P
YES

NO

Y1 ← 1

S

LP

TRUE

VALUE ← D(AUX)
SFLOAT ← D(Z+1)

FALSE

VALUE ← D(X+2)
SFLOAT ← D(AUX)

DYAD_COMK

COMPUTE RESULT

D(AUX) ← SFLOAT
AUX ← AUX-1
Y1 ← Y1 + 1

Y1 ≤ Y? YES S

NO

S

SVALUE ← DA
DA ← DA + D(Z)

RETURN

Fun and Games from BITS

101 BASIC Computer Games by David Ahl. A treasure store of games and simulations, 101 all written in BASIC. It contains both a complete listing and a sample run, along with a descriptive write-up. 256 pages, $7.50.

What to Do After You Hit Return or PCC's First Book of Computer Games. This is PCC's first book of computer games, a compendium which includes descriptions of numerous games, and listings of 37 selected BASIC games. New second edition, $8.

BITS Inc
70 Main St
Peterborough NH 03458

For ease in ordering, use the coupons on pages 175 and 176, writing in the name(s) of the book(s) you want.

Please note that processing may exceed 30 days in unusual cases.

Circle 383 on inquiry card.
Interpreting the Dyadic Reshape Operator

The last dyadic interpreter described is shown in Figure 40. This process interprets the dyadic reshape operator, but the general methodology used is of more importance. By examining how the various pointers are manipulated in this subroutine, as well as the previously described subroutines, the reader should be able to design any other dyadic subroutine. The procedure, as with most numerical processors, is to first check the legality of the arguments. Following this, the descriptor of the result is built. Finally the value elements of the result are computed and a pointer to the result is stacked in SVAL and index DA is updated.

Conclusion

Anyone who has worked with powerful high level interpreters knows that it is no mean task to describe them in so short a series of articles as this. Nevertheless there should be enough information here to allow the building of a powerful APL interpreter. Admittedly, there is room for improvement.
in some areas of design and implementation. Using the experience I gained from my first implementation, I have incorporated many desirable features into several microprocessor implementations which are currently being designed.

Anyone designing an APL interpreter based upon this series of articles and encountering problems should first consult Mr Robinet's article, "Architectural Design of a Directly-Executed APL Processor" (National Technical Information Service, US Dept of Commerce, 5285 Port Royal Rd, Springfield VA 22161), and, failing resolution, should contact me. Finally, although I have attempted to verify the accuracy of all tables, charts and methods used herein, it is still possible that errors or omissions remain.

During the time that these articles have been prepared, I have been actively working on an 6800-based APL interpreter. This work has led to changes and improvements to the flowcharts. Readers who would like to have the latest information are welcome to send a stamped, self-addressed envelope to Mike Wimble, 6026 Underwood Av SW, Cedar Rapids IA 52404.

BUILDING YOUR OWN COMPUTER?

WHY NOT DO IT THE OBJECTIVE DESIGN WAY?

STARTER SET INCLUDES FRONT AND REAR PANELS OF ANODIZED ALUMINUM PRE-PUNCHED AND DRILLED FOR MAXIMUM DESIGN VERSATILITY. COMPLETE PLANS AND SPECIFICATIONS INCLUDE INFORMATION ON NECESSARY PARTS AND SUPPLIERS, IDEAS FOR CUSTOMIZING, AND DETAILED INSTRUCTIONS FOR BUILDING A 12 OR 22 SLOT S-100 CARDFRAME AND POWER SUPPLY.

- AN INTERESTING AND INEXPENSIVE WAY TO GET INTO PERSONAL COMPUTING!
- A GREAT SCIENCE FAIR PROJECT!
- IDEAL FOR SCHOOLS AND UNIVERSITIES!

ASK ABOUT OUR INTELLIGENT FRONT PANEL OPTION

ORDER THE CRATE KIT FOR ONLY: $39.95

PLUS SHIPPING: U.S.—$3.00 CANADA—$4.00 FOREIGN—$10.00 ALL ITEMS SHIPPED AIR FREIGHT OR UPS BLUE LABEL SEND CHECK OR MONEY ORDER TO:

OBJECTIVE DESIGN, INC.
P.O. BOX 20325 TALLAHASSEE, FL 32304
FLORIDA RESIDENTS ADD 4% STATE SALES TAX.

THE VECTOR 1
Reset and Go
PROM/RAM BOARD
from
VECTOR GRAPHIC INC.

PROM: Space for 2K bytes, 1702A. Store bootstrap loaders and monitors.
RAM: 1K bytes, 2102LIPC, 450 ns, low power. NO NEED TO RELOCATE STACK WHEN ADDING MEMORY.
CIRCUITRY: Replaces memory write logic on ALTAIR™ and IMSAI front panels.
REGULATORS: Two regulators. No need for regulated power supply.
JUMP-ON-RESET: PROM program execution starts at any location in memory without interfering with programs in any other portion of memory.
S-100 BUS: +8 and -16 VDC; P/C BOARD SOLDER MASKED BOTH SIDES WITH PLATED THROUGH HOLES; ALL SOCKETS INCLUDED.
OPTIONAL Firmware: 512 byte monitor for use with Tarbell tape interface on 2, 1702A PROMs.

PROM/RAM KIT WITHOUT PROMS $ 89
+ OPTION A - SIO Rev. 1 or 3 P + S $129
+ OPTION B - 2 SIO (MITS) $129
+ OPTION C - 2 SIO 2 (IMSAI) $129
+ OPTION D - Poly Video Interface $159

California residents please add 6% tax.
IMMEDIATE DELIVERY FROM FACTORY OR YOUR LOCAL COMPUTER STORE

VECTOR GRAPHIC INC.
717 LAKEFIELD ROAD, SUITE F • WESTLAKE VILLAGE, CA 91361 • (805) 497-0733

Circle 384 on inquiry card.
You say you'd love to go and play "Pong," "Tank" or "Shoot 'em" down at the corner tavern but they won't let you in because you're not old enough? Or maybe you learned to play Space War back in college on a turbocharged PDP 911/S microcomputer and you don't think your poor microcomputer could keep up? Or perhaps you hate BASIC but can't think of a good reason to mess with 50 pages of assembly language. Even if you don't fit into any of the above categories, merely reading this article will sufficiently disturb your mental equilibrium to start you on your way to being a Space War freak, making all other problems fade into insignificance.

Yes, it's true, despite what false impressions you may have held. You can play Space War on your micro: real Space War with gravity and torpedoes and thrust and explosions and hyperspace, all right there in front of your eyes on the screen. No message slowly banging out on the Teletype "THE KLINGONS ARE APPROACHING," but your own spaceship orbiting the sun, able to change direction, accelerate and fire at the enemy. There's even a score to keep track of how many times you've been hit. "Hold it!" you say, after pouring over the listings. "This doesn't conform to ANSI 74 Space War Standards. Alpha Centauri isn't at coordinates 137, 245 like it's supposed to be." Relax, just change the program to suit your taste.

It should be apparent by now that you need more than just a "naked micro" to do the job, but what you do need is not too unreasonable. Table 1 lists the equipment I found necessary to implement a quite realistic Space War game which is the subject of this article.

If you don't like the part about using an oscilloscope with its small screen, don't knock it if you haven't tried it. You just have to sit close. Besides, you can move up to a bigger display later. A vector graphics system would be an improvement, allowing more odds and ends on the screen before flicker sets in. The ideal, however, is a raster scan TV monitor system with resolution close to 256 by 256; anything less is just too coarse. Thomas R. Buschbach's article in the November 1976 BYTE ["Add This Graphics Display to Your System," page 32]
Photo 2: Here the larger of the two ships is accelerating towards the lower right of the screen. This information is relayed visually by the trail that is seen in back of the spaceship. The small dot which is positioned at roughly equal distances from both ships is a torpedo which has just been fired by the smaller spaceship.

(or Using Your Oscilloscope as a Telescope)

should be helpful here. Bear in mind that if you go the refresh memory raster scan route you will have to think about erasing things, not a problem with the analog refresh approach.

Meanwhile, back to the software. To start with, BASIC is out because it's far too slow. What's in is 5.5 K bytes of programs and tables, all in assembly language. "What's the secret?" you say. "Why has Space War only been seen on expensive systems up to now?" The answer is a special added ingredient called TLU. TLU stands for Table LookUp and eliminates the need for all multiplications and divisions, making your program run ten times faster. Together with simple difference equations you can use table lookups to generate gravity for pretty

<table>
<thead>
<tr>
<th>TACTICAL MATERIEL NEEDED</th>
</tr>
</thead>
<tbody>
<tr>
<td>• An 8080 type microcomputer with 8 K of memory and a way to load programs. An Altair 8800 or equivalent will do fine. An assembler and text editor will be necessary if you want to modify the program conveniently.</td>
</tr>
<tr>
<td>• A general purpose DC oscilloscope with X input. Bandwidth isn't too important, but it must be DC. The bigger the screen the better. You can get one surplus for about $100, and they're handy for other things, too. A raster scan continuously refreshed point display can also be used with appropriate modifications of graphic generation and object motion routines.</td>
</tr>
<tr>
<td>• A 2 channel digital to analog converter. This device converts two bytes of digital data to two analog signals with a resolution of 1/256 (3.9x10^-3). It is the interface between the microcomputer and the oscilloscope, capable of displaying dots at any of 65536 locations on the screen in a 256 by 256 grid. The circuit is quite straightforward. The &quot;Beer Budget Graphics Interface&quot; described in November 1976 BYTE, page 26, will do fine. You could even skip the second latch on the Y strobe.</td>
</tr>
<tr>
<td>• Two or more hand held boxes, each with four push buttons. The two boxes together look like one memory byte or input port and are read by the program to determine what the player wants to do.</td>
</tr>
<tr>
<td>• Software functioning as described in this article.</td>
</tr>
</tbody>
</table>

Table 1: Basic facilities needed to implement Space War on a microprocessor.
Photo 3: The larger of the two spaceships has been hit and destroyed. It had fired a torpedo just before it was hit.

respectable orbits for your spaceships and torpedoes. Another thing, after subtracting 2 K for the gravity table you have 3.5 K of program to manage. This translates to 55 pages of assembly listing with macros expanded (more on macros later). To do this and not go crazy you have to have a system, a way to organize the code into modules for ease of understanding, modification and debugging. But first some preliminaries.

The Game of Space War

As the game starts you will see a sun in the center of the screen, two scores of zero at the bottom, and two spaceships at opposite corners. You and your opponent each have a ship. Each ship is uniquely identifiable and is moving slowly towards the sun. Photo 1 shows the screen at the beginning of the game as I implemented it.

Each player has a hand held box as shown schematically in figure 1. The functions of the buttons are as follows:

CCW: The player's ship rotates counterclockwise as long as this button is held down. Rotation increments are 45° and the rate is about one rotation every 5 seconds.

CW: Same as a CCW except rotation is clockwise.

FIRE: A torpedo is fired from the player's ship when this button is depressed. The torpedo always originates from the front of the spaceship and travels in the same direction the ship is pointing. Its initial velocity is constant relative to the ship's. Each ship at any one time has only one active torpedo at its disposal and thus can't fire a second until the first has hit a spaceship, hit the sun, hit the screen edge, or timed out.

Figure 1: The player control boxes consist of eight switches, four per player, which define data for one input port to the microprocessor. The pullup resistors boxed by a dashed line may be omitted, depending on details of your input port.
ACC: Acceleration is applied to the player’s ship as long as this button is held down. The acceleration is applied in the same direction the ship is pointed and is indicated visually by an exhaust trail at the rear of the ship.

The object of the game is to get your spaceship into a stable orbit around the sun and then shoot down the enemy. You have to, of course, watch out for the enemy shooting you down; and you have to be careful not to fall into the sun. If either of these two calamities should befall you, your ship will explode, your score will be incremented, points count against you, and a new spaceship will be delivered to the starting position.

You have a rather interesting option when the going gets tough. You can enter hyperspace by holding down both the CCW and CW buttons at the same time. Your ship disappears, then reappears at some random position with a random velocity. This happens not without risk; however, there is a definite probability, increasing as the game progresses, that your ship will explode when it comes back.

A word about orbits: both spaceships and torpedoes conform to Kepler’s laws, traveling in elliptical orbits around the central sun, with revolution periods averaging about 15 seconds. In general, the further away from the sun the ship is, the slower it moves. Thus, if a ship is in a long elliptical orbit the effect will be a little like “crack the whip”: the ship zips quickly around the back of the sun, then drifts slowly out into space, then back again. Torpedoes, usually moving faster than spaceships, are less affected by gravity and usually escape to the edge of the screen where they disappear. If they get close to the sun their trajectories may be bent as much as 90°. Due to limitations of the numerical method, the ships and torpedoes may not come closer than seven units (screen is 256 units wide) to the center of the screen. Thus the sun’s radius is defined to be seven units and anything coming that close is destroyed. A spaceship which starts anywhere on the screen with zero velocity will move radially inward to the sun and be zapped.

Photos 1 to 5 show some scenes from a typical Space War game. Some are snapshots and two are time exposures showing the motion of spaceships and torpedoes.

Software – General

Many of the techniques discussed here apply not only to Space War but also to other video games, display systems and real time applications. The only assumption made is that you know what an assembler

Photo 4: A time exposure showing the two ships in orbit and a near miss of a torpedo. The inner spaceship fired at the outer orbiting spaceship. Notice how the gravity effects of the sun curve the orbits of ships and torpedo.

Photo 5: A very long time exposure showing the maneuvering of the two spaceships. Notice that Kepler’s laws of motion are being followed, and the spaceship that is closer to the sun is orbiting faster than the spaceship that is farther away, since it has only traveled a portion of one revolution while the inner ship has completely circled the sun. These gravity effects encourage real time tactics and use of strategy in playing the game.
breaking the code up into modules you generally need only to edit and assemble a portion of the entire program to make a correction or enhancement. Linkage between modules is made through a few fixed locations in low memory (0 to 80 hexadecimal). There are two categories of modules: system and application. System modules exist to support a variety of application modules. The application module in this case is the Space War video game. The systems modules needed here include: graphics display processor, interrupt handlers and real time executive, system functions. The applications module is a collection of programs and data related specifically to the game. Let's look at each of these modules in detail, while referring to the block diagram in figure 2.

Display Processor

In order to have a video game you have to be able to put pictures on the screen. Still pictures are a good place to start; it's easy to make parts of them move later. With the digital to analog converter and XY oscilloscope hardware configuration of the "Beer Budget Graphics Interface" [by Peter Nelson, in November 1976 BYTE, page 26], it is necessary to paint the picture one dot at a time and also to continuously refresh any image put on the screen a minimum of 20 times a second. This suggests a loop consisting of a sequence of dot producing instructions executed over and over again. It would be very cumbersome, though, to have to write a new assembly language program every time you want a new picture. What about a single program which cycles through a table of XY coordinates? This is better, except it takes a lot of dots to draw a line or spaceship, and it's difficult to move something when you have to move all the dots together. A better solution is an interpretive "language" which you can use to describe pictures. A display file written in this language is interpreted by a special program called the display processor in order to provide the proper sequence of XY coordinates to paint the picture.

Understanding the display processor requires you to visualize the beam which defines a point on the face of the oscilloscope at position X,Y. A dot can be displayed at the beam position or at points relative to the beam position. This is very useful because you can define a spaceship, for instance, as a series of dots relative to the beam coordinates. To move the spaceship you merely move the starting beam position. Rather than define a series of dots individually, why not specify a vector
or row of dots? This can be done neatly in one byte where you specify both the length and direction of the vector. Let's say that you have defined your spaceship, but that you want to display it more than once on the screen. Just as in any other computer program you use a subroutine. The subroutine in this case contains a series of relative vector commands and is called from several places in the display program, just after beam positioning commands.

You can pick up the details of the graphics language by studying the nine commands shown in figure 3. Figure 3 also shows the layouts of the graphics commands. In my assembler the mnemonics along with the arguments are put right into a program and the assembler produces the object code by means of macros. (A macro is a way of telling the assembler to substitute a group of instructions for a given symbol.) All the macros are defined at the start of the program with dummy arguments. The real arguments are sub-

<table>
<thead>
<tr>
<th>Hexadecimal Op Code</th>
<th>Layout</th>
<th>Assembler Macro Form</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0 0 0 0 0 0 0 0</td>
<td>MBEAM x,y</td>
<td>The beam is moved from its previous position to the new coordinates x,y where x and y are between 0 and 255. This command does not cause a dot to be displayed, but is used as a setup for following commands.</td>
</tr>
<tr>
<td>02</td>
<td>0 0 0 0 0 0 0 1</td>
<td>MDISP x,y</td>
<td>Same as MBEAM except a dot is displayed at x,y.</td>
</tr>
<tr>
<td>04</td>
<td>0 0 0 0 0 0 1 0</td>
<td>LVEC x,y</td>
<td>A &quot;long vector&quot; is drawn from the previous beam position to the new coordinates x,y. The new beam position is now x,y.</td>
</tr>
<tr>
<td>06</td>
<td>0 0 0 0 0 1 1 0</td>
<td>SVEC</td>
<td>The relative vector is called the short vector to distinguish it from the absolute long vector. The list is a sequence of length and direction pairs where the length is 0 to 7 dots and the direction is specified by a number from 0 to 7 where 0 is straight up, 1 is 45° in a counterclockwise direction, 2 is 90° clockwise, etc.</td>
</tr>
<tr>
<td>08</td>
<td>0 0 0 0 1 0 0 0</td>
<td>PARAM s,o</td>
<td>This command alters the effect of all following SVEC commands until the next PARAM statement. It is possible to change both scale (s) and orientation (o) of all following short vectors. If scale is set to 1, for example, all short vectors would be twice as long as they would have been if scale were 0. If orientation is set to 2 instead of 0, all short vectors appear rotated 90° clockwise. Thus a figure drawn entirely in short vectors can be enlarged and rotated by changing one command.</td>
</tr>
<tr>
<td>0A</td>
<td>0 0 0 1 0 0 0 1</td>
<td>JUMP addr</td>
<td>Interpreter control is transferred to the command at address addr.</td>
</tr>
<tr>
<td>0C</td>
<td>0 0 0 1 0 0 1 0</td>
<td>JUMPS addr</td>
<td>Interpreter control is transferred to the command at address addr and the address of the command following the JUMPS statement is saved. This is the graphics subroutine call.</td>
</tr>
<tr>
<td>0E</td>
<td>0 0 0 1 1 0 0 0</td>
<td>RETS</td>
<td>Return from the subroutine entered by JUMPS command. Subroutines may be nested.</td>
</tr>
<tr>
<td>10</td>
<td>0 0 0 1 0 0 0 0</td>
<td>EXEC</td>
<td>Control is transferred to the executive so that the proper applications programs are executed. When these are finished, control is returned to the graphics interpreter command following the EXEC command.</td>
</tr>
</tbody>
</table>

Figure 3: The nine graphic commands used by the display processor showing the op code, related mnemonic, memory allocation, and meaning for each command. These instructions and related programs can be used to implement a wide variety of animated video applications.
Entry | Byte | Data | Function
--- | --- | --- | ---
1 | 0 | 1 | look at last bit of TIME only
1 | 1 | 0 | execute on even TIME only
2 | 3 | 0 | object block 1 starting address
4 | 5 | ship fly program starting address
2 | 0 | 1 | look at last bit of TIME only
1 | 1 | 0 | execute on odd TIME only
2 | 3 | 0 | object block 2 starting address
4 | 5 | ship fly program starting address

Table 2: This table design will allow the updating of the positions of two spaceships on alternating cycles. Byte 1 determines whether the updating takes place on an odd or even cycle. If it is set to zero, as in entry 1, updating will occur on an even cycle. If byte 1 is set to 1, as in entry 2, updating will occur on an odd cycle.

Bytes (Hexadecimal Offsets) | Function
--- | ---
0,1 | X (n), present x coordinate with 16 bits, 0 at screen center.
2,3 | X (n-1), previous x coordinate.
4,5 | X acceleration.
6,7 | Y (n), present y coordinate.
8,9 | Y (n-1), previous y coordinate.
A,B | Y acceleration.
C,D | pointer to MBEAM instruction display file.

Table 3: A list of object parameters which are found at the head of an object block and used by the object move function. These coordinates are continuously updated as the program progresses.

Now that you can describe a picture with a concise list of graphics commands you should be able to see how easy it is to animate the picture. All it takes is some other program in the system to periodically change parts of the display file. For example, if the MBEAM command preceding a spaceship subroutine call is given a label through the assembler, then the X and Y coordinates can be updated, causing the ship to move. To destroy the ship, replace the spaceship subroutine address with the explosion subroutine address.

The dispatch table (table 2) in the applications module is a list of all the tasks. Each task has a 6 byte entry having the format:

- **byte 0:** Mask which is logically ANDed with TIME and compared with the value in byte 1.
- **byte 1:** Execution time. When this matches the masked value of TIME the task is executed.
- **bytes 2,3:** Address of the object block for this task. It is passed to the program in register pair BC.
- **bytes 4,5:** Address of the program to be called by the executive.

If, for example, you want to update the position of each of two spaceships on alternating cycles, you set up two entries as in table 2.

Most of the task schedules are set up just once when you assemble the applications program. However, it is possible to dynamically schedule one task from another. That scheduled task runs “right now” if both the mask and time bytes are set to zero. It runs “n cycles from now” if the mask is all 1 and the time is TIME+n. The dynamically scheduled task will, however, have to later deschedule itself by setting the mask to 0 and time to 1.

Now for a few words on how programs interface with object blocks. Normally a program references a number of fixed memory locations using direct addressing. This works fine, for example, if you have one program flying one spaceship. For two spaceships you could write two programs,
SAVE MONEY! As of January 1, 1978, we are raising our subscription rates to meet the increased costs of producing and mailing BYTE to our more than 110,000 monthly readers. The new domestic subscription rates, effective January 1, 1978:

<table>
<thead>
<tr>
<th>Current Rates</th>
<th>Rates Effective January 1, 1978</th>
</tr>
</thead>
<tbody>
<tr>
<td>One year U.S.</td>
<td>$12</td>
</tr>
<tr>
<td>Two years U.S.</td>
<td>$22</td>
</tr>
<tr>
<td>Three years U.S.</td>
<td>$32</td>
</tr>
<tr>
<td>One year U.S.</td>
<td>$15</td>
</tr>
<tr>
<td>Two years U.S.</td>
<td>$27</td>
</tr>
<tr>
<td>Three years U.S.</td>
<td>$39</td>
</tr>
</tbody>
</table>

If you already subscribe to BYTE and still have several months to go before expiration of your subscription, you can still take advantage of current rates for renewal (even up to three years). Use coupon below. When your present subscription expires, your renewal order commences.

If you are not yet a subscriber to BYTE, the leading magazine for the creative home computer experimenter, don’t delay... mail this coupon and start getting your own copies of this invaluable magazine, before new rates become effective.

USE THIS MONEY-SAVING COUPON TODAY

To new subscribers:
Read your first copy of BYTE, if it’s everything you expected, honor our invoice. If it isn’t, just write “CANCEL” across the invoice and mail it back. You won’t be billed, and the first issue is yours.

Allow 6 to 8 weeks for processing.

© Byte Publications, Inc. 1977
but this would be a pain in the asteroid. How about grouping all the variables together and defining a data layout common to all spaceships? The data includes coordinates, the address of the MBEAM instruction in the display file, orientation and about a million flag bits $fu bit of hyperbole . . .$. The program knows which object block it’s working with because it started out with the block start address in an index register, in this case the BC register pair of the 8080. All memory references then go through this index register. Don’t let the fact that the 8080 microprocessor doesn’t have real index registers bother you; you simply use assembler macros to invent your own indexed instructions as follows:

LOADX reg, offset: Loads data from object block relative location “offset” into register “reg.”
STORX reg, offset: As above but stores.
LDBLX offset: Loads data from object block relative locations “offset” and “offset” + 1 into registers L and H respectively.
SDBLX offset: As above but stores.

Listing 3 shows the program execution and interrupt handlers.

System Functions

System functions are general purpose subroutines common to a number of applications programs. These include object move, acceleration lookup, random number and a system call program which links with calling programs. To call one of the functions you invoke the macro SCALL n where n is 0 for move, 1 for acceleration and 2 for random number routine. This macro generates the code RST 7, DB n causing transfer of control through memory location 38 hexadecimal.

Looking at the first function, object move, you find a function fundamental to all video games. The function depends on a fixed layout of object coordinates at the head of the object block as shown in table 3. Calling object move updates these coordinates according to the following formulas:

$$X(n+1) = X(n) + X(n) - X(n-1) + X_{acc}$$

$$X(n) = X(n+1)$$

The same equations are used for Y.

The XY coordinates of the MBEAM command are updated with the new higher order values of X(n) and Y(n). It is important to note that the X and Y bytes in the MBEAM command are referenced to the lower left of the screen, but the 2 byte coordinates carried in the object block are referenced to the screen center. These equations of motion represent what the math jocks call second order dif-
ference equations. Note the absence of multiplications and divisions. You can consider the difference between $X(n)$ and $X(n-1)$ to represent the X velocity of the object, and the corresponding Y difference to be the Y velocity. If you set both accelerations to zero and initialize the other coordinates appropriately, your object moves in a straight line with a constant velocity. If Y acceleration is a negative constant, the object falls in a parabolic trajectory like a thrown stone.

The object move function is sufficient for most video games, but not for Space War. For orbiting objects the accelerations are neither zero nor constant. Each point on the screen has unique values of X and Y acceleration. The acceleration lookup function finds these values for you, using the $X(n)$ and $Y(n)$ coordinates, so that when you alternately call it and the object move function, your spaceship or torpedo zips neatly around its orbit.

First generation Space War systems would calculate the accelerations each time they moved their objects according to the following formulas:

$$X_{acc} = cX/R^3 \quad \text{and} \quad Y_{acc} = cY/R^3$$

where $c$ is a constant and $R = \sqrt{(X^2 + Y^2)}$. This requires a total of five multiplications, two divisions and one square root extraction for each update, clearly impossible for most of today’s microprocessors.

Acceleration lookup, the heart of the Space War system, uses a 1 K by 2 byte table to find these accelerations, taking advantage of symmetry.

The third function, random number, returns an 8 bit pseudorandom number in the accumulator. This number is derived from a common shift register feedback scheme and has a repetition period of 255.

Listings of all system functions are shown in listing 4.

Gravity Table

Table 4 shows the 1 K by 2 byte gravity table used by the acceleration lookup function. The values were calculated by a FORTRAN program running on a larger computer. The table entries represent the absolute value of X acceleration in one quadrant. Y accelerations are found simply by transposing the indices. Because halving the distance from the sun causes the acceleration to increase by a factor of 4, the table can be magnified to produce more accurate values closer in. By proper shifting of indices and output values, the same table can be made to cover index ranges 0 to 16, 16 to 32, 32 to 64 and 64 to 128.

Thus the maximum value of 999 applies only inside the sun.

Space War Applications Module

This module, occupying about 2 K bytes of user program memory, specifically defines the game of Space War. It is composed of constants, macro definitions, system linkages, the dispatch table, object blocks and programs. It interfaces with all the system modules described earlier. All programs execute once for each related object unless otherwise specified. Listing 5, the applications module, is divided into several sections which are described in a separate box entitled “A Guide to the Space War Applications Module.”

Installation of Space War in Your System

The following steps might make it easier for you to get Space War up and running on an 8080 system:

- Make sure that your graphics output and button input work the way you think they do. Write short test programs if necessary.
- Modify the display processor module so it communicates with your particular graphics interface. The SHLD XYOUT instructions of the assembly listings are the ones you will want to look at carefully.
- Modify the ship fly and rotate programs in the applications module so that they read your buttons properly.
- Modify the keyboard handler (executive module) to accept interrupts from your keyboard. If you don’t have keyboard interrupts you can periodically read your keyboard in the rotate program.
- Assemble all programs and load in the following order: display processor, executive, numerals, applications, system functions and gravity table.
- Temporarily eliminate the EXEC command in the display file, then start execution at hexadecimal location 100. This tests the display file and the display processor module, the correct result being a still picture which should make sense.
- Restore the EXEC command and deschedule all but the first task in the dispatch table, then start at 100 again. The system should remain in the still picture mode because not enough tasks have been enabled to support object motion.
- Enable tasks one at a time, thereby testing each. If the program bombs you will know exactly where to look.
- Fasten securely all loose objects in your computer room in anticipation of the large unruly crowds which will soon gather.
Conclusion

With Space War you will have come a step closer to making your computer The Ultimate Toy. In the process you will have learned some software fundamentals which will make you less afraid to put together some large systems of your own. Maybe you can think up some new and original video games which match or surpass the appeal of Space War. I'll be watching the pages of BYTE for the results.

A GUIDE TO THE SPACEWAR APPLICATIONS MODULE

Important Constants

All constants used in fine tuning the game are defined in this first section. These include the loading address, collision radii of the sun and torpedoes, acceleration constants and time-out values.

Macro Definitions

All macro definitions used in the module appear in this section. These include the graphics command macros and the load and store indexed macros.

System Linkages

Space War references two bytes in the system area: TIME, as described under the executive routine, and NUMS, the starting address of the list of graphics numeral addresses. The system modules, in turn, reference three points in the Space War module: the start of the display file, the dispatch table and the keyboard decode program.

Dispatch Table

The layout of this table is described in detail under the executive routine. The table is preceded by one byte indicating the number of entries, in this case 14. The ENTRY macro is used to define the entries.

Object Blocks

These include: ship 1, ship 2, torpedo 1, torpedo 2, score 1 and score 2.

Initialization Program

Executes: At start of game.
Function: Zeros score.
Schedules the ship start program to run immediately.
Deschedules itself.

Ship Start Program

Executes: When scheduled by initialization program.
A fixed time delay after a ship is destroyed.
Function: Puts spaceship subroutine call into display file.
Sets coordinates and orientation to starting values.
Deschedules itself.
Executes: On hyperspace return when ship is not set to destroy.
Function: Puts spaceship subroutine call into display file.
Clears hyperspace flag.
Deschedules itself.
Executes: On hyperspace return when ship is set to destroy.
Function: Puts explosion subroutine call into display file.

Ship Fly Program

Executes: On every clock cycle.
Function: Updates ship coordinates and acceleration values.
Adds acceleration and displays exhaust if the ACC button is held down.
Bounces ship off screen edge if necessary.
Replaces ship routine with explosion routine if:
ship is too close to any torpedo
ship is too close to the sun.
Releases torpedoes which get close enough to destroy spaceship.

 Torpedo Fly Program

Executes: On every clock cycle.
Function: Updates torpedo coordinates and acceleration values.
Releases torpedo if:
torpedo hits edge of screen
torpedo gets too close to the sun.

Torpedo Fire Routine

Executes: When scheduled by the rotation program.
Function: Claims one torpedo belonging to the firing ship if it is not claimed.
Substitutes MDISP for MB EAM instruction at torpedo location in display file.
Computes initial position and velocity of torpedo from spaceship's coordinates and orientation.

Score Program

Executes: Every 16 clock cycles.
Function: Reads 1 byte binary coded decimal score value and converts it into addresses of two numeral subroutines which are inserted into display file.

Buttons Program

Executes: If hyperspace is entered (CCW and CW buttons both held down).
Function: Blanks spaceship.
Schedules ship start program after a time delay.
Indicates hyperspace destroy on random probability.
Executes: If CW button is held down.
Function: Increments orientation value.
Executes: If CCW button is held down.
Function: Decrements orientation value.
Executes: If FIRE button is down.
Function: Schedules torpedo fire program.
Function during all executions: Includes orientation value in appropriate PARAM instruction in display file, releases any timed out torpedoes, decrements counters for other torpedoes.

Keyboard Decode Program
Executes: In response to interrupt from keyboard, called from execute module.
Function: Schedules initialization program twice when the C key is hit.

Listing 1: The display processor takes care of all the video display routines used for the game.

```
: DISPLAY PROCESSOR MODULE
: COPYRIGHT 1974 D. KUHLIGEK

: INITIALIZATION
STACK EQU $FF8 > STACK AUDR
CINTC EQU $FF9 > HIC CONTROL ADJRESS
OSTRT EQU $FFA > STACK ADDRESS
MINTC EQU $FFB > ECIC ID CONTROL ADJRR
TIME EQU $FFC > SYSTEM TIME
TOP EQU $FFD > DISPLAY FILE START ADDR
YOUT EQU $FFE > CRT OUTPUT AUDR
ORG $0000
ORG $0000
ORG $0000

INIT: LXi 
LAL 
SNDL INCPT > LIMIT INCREMENT POINTER
LSP > STACK > LIMIT STACK POINTER
MV1, $100
STA XINTC > TURN ON K1 INT
MV1, $100
STA CINTC > TURN ON TIC INTERRUPT
EIP
LHLD TOP > LIMIT POINTER
SNDL PHTR > LIMIT INSTRUCTION POINTER

: OPCODE DECODING
MLOOP: LHLD PNTR > ADDRESS OF OPCODE
MOV C, R > OPCODE
MV1, B, $1 > OPCODE IN BC
LXi N, JMTAB > BASE OF JUMP TABLE
DAD B > ADD OPCODE
MOV C, A, R
INX H
MOV L, C > ADDR OF ROUTINE IN HL
PCHL > JUMP TO IT

JUMP TABLE FOR OPCODE PROCESSING
JMTAB: DW $BEAM, 18
DW $NISP, 12
DW $LUEC, 14
DW $SCEC, 16
DW $PARAM, 18
DW $JUMP, 1A
DW $JUMP, 1C
DW $RETS, 1E
DW $EXEC, 18
DW $SYNC, 1E

: MOVE BEAM - DOF DISPLAY POINT
$BEAM: LHLD PHTR
INX M
MOV D, M > GET X COORD
INX M
MOV E, M > GET Y COORD
INX M
SNDL PHTR > RESTORE POINTER
JMP MLOOP > GET ANOTHER INSTRUCTION

: MOVE BEAM AND DISPLAY POINT
$DOF: LHLD PHTR
INX M
MOV D, M
INX M
MOV E, M
INX M
SNDL PHTR
XCHG
SNDL YOUT > WRITE TO CRT
XCHG
JMP MLOOP

: SET ORIENTATION AND SCALE
PARN: LHLD PHTR
INX M
MOV C, M
INX M
SNDL PHTR
MOV L, C > NEW ORIENT & SCALE IN HL
```

The Mini-Micro Designer – a complete microcomputer system for just $830.50!*

Here's the real thing – a microprocessor that takes you right to "real world" situations for about half the price of other systems.

With our hardware, you'll receive the most complete software package in the business. 700 pages of clear instruction, written by Rony, Larsen, Titus – famous for their BUGBOOKS™. Designed to show you how to get your MMD-1 up and working even if you have no prior knowledge of digital electronics.

With our MMD-1 and M/I board combination you'll get all of the interfacing hardware you need, without costly extras.

Here's what we pack in for $830.50:
- 2.5K RAM
- 1.5K PROM (special D-Bug, Monitor and Keyboard Interpreter)
- Audio Cassette Interface
- TTY Interface
- Built-in Keyboard for Control and Data Entry
- Direct Access to latched ports
- Built-in Breadboarding Capability
- Single Step Option
- Monitors for Address and Data Busses. And more.

Best of all, it's on the shelf at your computer store now. Write us for an info-packed brochure and the name of the dealer nearest you.

*Suggested resale price (U.S.A.).

Circle 386 on inquiry card.

E&L INSTRUMENTS, INC.
61 First Street, Derby, Conn. 06418
(203) 735-8774 Telex No. 96 3536
Listing 1, continued:

JUMP TO SUBROUTINE MVJ E, e

JUMP TO SUBROUTINE MVJ MLOOP

RETURN FROM JUMP : LILL U PNTR Mil

SHORT VECTOR MOVE JMP MLOOP

RETURN FROM SUBROUTINE NETS : POP H : 1:1ST POINT FROM STACK

SHORT VECTOR MOVE SVEC : LHLD

LONG VECTOR MODE MOV C, 0F8H

FLOOP : MOV A, 8 ; X I nCREr!ENT De e

NEXT : MOV M, M ; NEW ADDRESS IN RC

INCPT: OW 0 POl<ITER TO llCR EMENT IN TABLE

LXH H, TIME 
OCR 0 OB - 2 
JNZ NLOOP ; LOOP IF NOT OOllE

INCPT: OW 0 POl<ITER TO l llCR EMENT IN TABLE

SHL D XYOUT : 1111 J11AL 001 TO c• r DB - 2 
NLOOP: 	 MOV A, B : X I !ICREMEUT DB -2

SVEC : LHLD

LONG VECTOR MODE MOV C, 0F8H

FLOOP : MOV A, 8 ; X I nCREr!ENT De e

NEXT : MOV M, M ; NEW ADDRESS IN RC

INCPT: OW 0 POl<ITER TO llCR EMENT IN TABLE

LXH H, TIME 
OCR 0 OB - 2 
JNZ NLOOP ; LOOP IF NOT OOllE

INCPT: OW 0 POl<ITER TO l llCR EMENT IN TABLE

SHL D XYOUT : 1111 J11AL 001 TO c• r DB - 2 
NLOOP: 	 MOV A, B : X I !ICREMEUT DB -2
Peripheral Vision is a brand-new company that's dedicated to selling reasonably priced peripherals for various manufacturers' CPU's.

We think you'll flip over our first product.

It's a full-size floppy disk for the Altair-Imsaí 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>Price</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 387 on inquiry card.
Listing 2: Numeral display module contains the graphics dot code to draw the numerals 0 thru 9.

Listing 3: The interrupt handler and executive module covers the real time clock, keyboard scan routines, and dispatches object blocks on correct execution times.
A logical forward step in Microcomputer design

the Processor Terminal. A new design by TEI and look at what you get... a complete, self contained microcomputer system with display and mass storage, a full keyboard and plenty of slot space for additional boards. And that famous TEI CVT power supply that makes brownouts a thing of the past.

Display — A 15" high-resolution black and white video display with an optical filter face plate to reduce glare and improve type visibility... Keyboard — Full upper and lower case ASCII detached keyboard with 8 programmable special function keys. Keyboard status indicators show computer BUSY or READY. And a 16-key numeric cluster pad set up calculator style... Disk Drive & Controller — A Shugart SA-400 mini-floppy disk drive. Soft sectored with a capacity of about 90 KB. IBM compatible format. Controller will handle 3 drives... CPU — 8080 based with a flexible design that allows you to implement a start up "jump to" operation to any dip switch selected byte address you choose. Merely turn on power or press RESET and you are off and running. Excellent for power failure automatic restart... Memory — 16K of static RAM memory. Low power chips. Selectable address assignment and memory protect features... I/O — 3P+3S input/output board. 3 parallel ports and 3 serial ports with selectable baud rates of 75 to 19,200. RS-232C and TTL outputs...

Video — A video board provides the support for the video display functions... Mainframe — A 12 slot mainframe with a 17-amp CVT power supply, motherboard assembly, heavy duty aluminum cabinet, fan and washable filter. All edge connectors and card guides provided... Software — CP/M disk operating system and BASIC provided on disk.

the Processor Terminal (Model MCS-PT) fully assembled and tested is priced at $3495.00.

the Processor Terminal partially assembled (We build the cabinet, keyboard, monitor, power supply, disk drive and motherboard and you build the CPU, RAM, I/O, Video and Controller boards which we supply as kits). This partially assembled unit is priced at $2995.00.
Listing 4: The system function module takes care of all system calls, acceleration lookups, and random number generation.

\[
\begin{align*}
% & \\
& \text{SYSTEM FUNCTIONS MODULE} \\
& \text{COPYRIGHT 1976 D. KRUGLINSKI} \\
& \text{SYSTEM CALL, MOVE, ACCELERATION, RANDOM} \\
& \text{MSTR EQU } 0C00H \quad \text{LOAD ADDRESS} \\
& \text{MACROS} \\
& \text{COMPLEMENT ML} \\
& \text{MACRO MOV A, H} \\
& \text{CMOV MOV A, L} \\
& \text{MOV L, A} \\
& \text{ENDM} \\
& \text{LOAD HL INDEXED (BC:BASE)} \\
& \text{LDBX MACRO OFFSET PUSH D} \\
& \text{LXI H, OFFSET DAD B} \\
& \text{MOV E, M} \\
& \text{MOV D, M} \\
& \text{XCHG POP D} \\
& \text{ENDM} \\
& \text{LOAD REG INDEXED (BC:BASE)} \\
& \text{HL DEstroyed LOADX MACRO REG, OFFSET LXI H, OFFSET DAD B} \\
& \text{MOV A, H} \\
& \text{META REG, M} \\
& \text{ENDM} \\
& \text{ST ORK MACRO REG, OFFSET LXI H, OFFSET DAD B} \\
& \text{MOV A, H} \\
& \text{META REG, M} \\
& \text{ENDM} \\
& \text{ORG } 38H \quad \text{RST 7 ADDRESS} \\
& \text{MP SYSL} \\
& \text{ORG MSTR \quad LOAD ADDRESS} \\
& \text{SYSTEM CALL FUNCTION} \\
& \text{ENTER ON RST 7 FOLLOWED BY FUNCTION #} \\
& \text{DESTROYS HL ONLY} \\
& \text{DISPCL: POP H} \\
& \text{INX H} \\
& \text{PUSH H} \\
& \text{DCH H} \\
& \text{CALL # ADDRESS} \\
& \text{PUSH D} \\
& \text{SAVE DE} \\
& \text{MOV E, M} \\
& \text{MOV D, P} \\
& \text{CALL # IN DE} \\
& \text{LXI H, CALIB \quad CALL TABLE BASE} \\
& \text{DAD D} \\
& \text{DAD J} \\
& \text{MOV J, M} \\
& \text{MOV D, M} \\
& \text{XCHG ; ADDR IN HL} \\
& \text{POP D} \\
& \text{RESTORE DE} \\
& \text{PCNL ; JUMP TO SUBROUTINE} \\
& \text{CALTB: DW MOVE} \\
& \text{DW ACCEL} \\
& \text{DW HAND} \\
& \text{DW } 0 \\
& \text{CALL SCALL } \& \\
& \text{GENERAL PURPOSE MOVE FUNCTION} \\
& \text{ASSUMES FIRST OBJECT BLOCK LOCATIONS AS FOLLOWS:} \\
& \text{X (N)} \\
& \text{X(N+1)} \\
& \text{Y ACCELERATION} \\
& \text{Y(N)} \\
& \text{Y(N+1)} \\
& \text{Y Acceleration} \\
& \text{POINTER TO 'MBEAM 1' INSTR} \\
& \text{CALL SCALL } \& \\
& \text{COMPUTE ACC TABLE OFFSET FROM X Y (IN DC)} \\
& \text{GETAD: MOV A, H} \\
& \text{REC} \\
& \text{REC} \\
& \text{REC} \\
& \text{REC} \\
& \text{ROTATE RIGHT 4} \\
& \text{MOV A, E} \\
& \text{AWI 7H} \\
& \text{MOV D, E} \\
& \text{M.O. TABLE OFFSET} \\
& \text{MOV A, E} \\
& \text{AWI 0EH} \\
& \text{MOV A, H} \\
& \text{M.O. Y} \\
& \text{REC} \\
& \text{AWI 6EH} \\
& \text{REC} \\
& \text{MOV E, A} \\
& \text{M.O. Y} \\
& \text{MOV A, E} \\
& \text{L.O. TABLE OFFSET}
\end{align*}
\]
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 wave form 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.

Heuristics, Inc.
Box B, 900 N. San Antonio Rd.
Los Altos, CA 94022, Phone(415) 948-2542
☐ Send me SpeechLab. I enclose $299.
☐ Send me more information.
☐ California residents add sales tax.

Name _____________________________
Street ______________________________
City __________________________ State ______ Zip ________

In Canada:
Trinronics Ltd.
186 Queen St. W.
Toronto, Canada M5V 1Z1

Heuristics
INC.
Listing 4, continued:

Listing 5: The applications module, which is specified for a particular program, is described in detail on page 96.
First, come to Micropolis. Where you get a MetaFloppy™ instead of an ordinary 5¼" floppy disk system.

Then, forget all things average.

Because the Micropolis MetaFloppy delivers over four times the 70K (or so) bytes of storage you'd expect. A whopp­
ing 315K bytes per disk to be exact. All formatted — not raw. And all on a single disk side.

But that's not all. Because Micropolis delivers everything else you need, too. Power Supply. Interface cable. Extended BASIC software. Plus an S-100 controller for your Altair 8800, IMSAI 8080, COMPAL 80, or the like.

There's even a dual drive model for maximum convenience. And savings.

The Micropolis secret: We build every component we sell. In fact, we're the only totally integrated 5¼" floppy disk system builder around. And where there's no middleman, there's no middleman to pay.

So, before you swallow someone else's price/performance story, find out how much your buck really buys and how much faster it buys it — at Micropolis.

For a free brochure and price sheet, write: Micropolis Corporation, 7959 Deering Ave., Canoga Park, California 91304 or phone (213) 703-1121.

Circle 390 on inquiry card.
Listing 5, continued:

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MACRO TO STORE REG INDEXED (30+BASE)</td>
</tr>
<tr>
<td>2</td>
<td>DESTROYS HL</td>
</tr>
<tr>
<td>3</td>
<td>STORY MACRO, REG, OFFSET</td>
</tr>
<tr>
<td>4</td>
<td>LXI H, OFFSET</td>
</tr>
<tr>
<td>5</td>
<td>DAD B</td>
</tr>
<tr>
<td>6</td>
<td>MOV REG, M</td>
</tr>
<tr>
<td>7</td>
<td>BRM</td>
</tr>
<tr>
<td>1</td>
<td>MACRO TO LOAD 2 BYTES FROM OFFSET</td>
</tr>
<tr>
<td>1</td>
<td>INTO HL (BASE+30)</td>
</tr>
<tr>
<td>2</td>
<td>LXI H, OFFSET</td>
</tr>
<tr>
<td>3</td>
<td>DAD B</td>
</tr>
<tr>
<td>4</td>
<td>MOV E, K</td>
</tr>
<tr>
<td>5</td>
<td>JMP H</td>
</tr>
<tr>
<td>6</td>
<td>MOV A, V</td>
</tr>
<tr>
<td>7</td>
<td>POP D</td>
</tr>
<tr>
<td>8</td>
<td>BRM</td>
</tr>
<tr>
<td>1</td>
<td>MACRO TO TAKE ABSOLUTE VALUE OF A</td>
</tr>
<tr>
<td>2</td>
<td>HWS MACRO</td>
</tr>
<tr>
<td>1</td>
<td>LJMP</td>
</tr>
<tr>
<td>2</td>
<td>JP POS ; JUMP IF A &lt; POS</td>
</tr>
<tr>
<td>3</td>
<td>CMA ; COMPLEMENT A</td>
</tr>
<tr>
<td>4</td>
<td>PUSH</td>
</tr>
<tr>
<td>5</td>
<td>EIXH</td>
</tr>
<tr>
<td>6</td>
<td>W</td>
</tr>
</tbody>
</table>
THE AJ 841 I/O—A COMPLETELY REFURBISHED IBM SELECTRIC TERMINAL WITH BUILT IN ASCII INTERFACE—JUST $995

Features:

- ASCII code
- 14.9 characters per second printout
- Special introductory price—$995 (regularly $1195). 75% discount over original price of new unit.
- Choice of RS 232 Serial Interface or Parallel Interface (requires 3P + S)
- Order direct from factory
- 30 day warranty—parts and labor
- Nationwide service locations

AJ 841 WARRANTY AND SERVICE IS AVAILABLE IN THE FOLLOWING CITIES:

Los Angeles    Cincinnati
Philadelphia   Detroit
Hackensack     Dallas
Columbus       Houston
Cleveland      Atlanta
San Jose       Chicago
Boston         New York
Washington, D.C.

HOW TO ORDER AN AJ 841 I/O TERMINAL

1. Make cashier's check or money order payable to: ANDERSON JACOBSON, INC.
   Address your request to:
   Personal Computer Terminal
   ANDERSON JACOBSON, INC.
   521 Charcot Avenue
   San Jose, CA 95131

2. Upon written notification, pick up your terminal at the AJ service office located in one of the above cities. Allow six to eight weeks for delivery.

3. A final check of your unit will be made at the local AJ service office at time of pick up.

4. For warranty or repair service, return unit to designated service location.

CLIP AND MAIL WITH ORDER

SELECT EITHER:
☐ RS 232 Serial Interface
☐ Parallel Interface (requires 3P + S)

Number of units ________________ at $995. each $ ________________

Local Sales Tax $ ________________
Shipping and handling $35.(f) each (excluding San Jose) $ ________________

TOTAL $ ________________

NAME ________________________________
ADDRESS ________________________________
CITY __________________ STATE _______ ZIP _______
PHONE (_______) ____________________

Circle 391 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) 328-6975

© NOVA is a registered trademark of Data General Corporation, Southboro, Massachusetts
Listing 5, continued:

MOV M, A
LDBLX PEPRT ;SHIP ENTRY ADDR
MVI M,-1 ;MASK = -1
LOA TIME
AD1 INTVL
INK H
MOV M,-1 ;TIME+INTVL TO PL ST

; REPLACE SHIP WITH EXPLOSION
LXI D,EXPLD
LDBLX PELMT ;SUB CALL
MOV X,E
INK M
MOV X,M
RET

SWAPX: LOXLX XN ;SWAP X COORDS
XCHG
LDBLX XNM
SUBD LX XNM
FAEX
LDBLX XNM
RET

; SWAPY: LOXLX YN ;SWAP Y COORDS
XCHG
LDBLX YNM
LDBLX YNM
LDBLX YNM
RET

; ACCELERATION TABLE
XARR: DW 0
DW ACON
DW ACON
DW 2
DW -ACON
DW -ACON
DW -2
DW ACON

; ON HEART SCREEN EDGE
LDAOX A, X0+1 ;X Y POS OF TORPEDO
ASS 255 VALUE OF Y
CPI LIMIT ;SUBTRACT LIMIT
JF BLANK ;NEAR EDGE IF > 0

; ON Y NEAR EDGE
LDAOX A, Y0+7 ;X Y POS OF TORPEDO
ASX 255 VALUE OF X
CPI LIMIT
JF BLANK ;NOT NEAR EDGE

; TORPEDO HIT SCREEN
LDAUX A, XH+1
ASX
CPI XH
JLST BLANK ;PLAY COLLISION INST
LDAUX A, YH+1
ASX
CPI YH
JLST BLANK ;PLAY COLLISION INST

; TARGET SHIP FLY PROGRAM
; CALL: TARGET POINTS TO MOVE
; CALL 0 :MOVE TITRED
; TEST IF X NEAR SCREEN EDGE
LDAOX A, XH+1 ;X Y POS OF TORPEDO
ASS 255 VALUE OF Y
CPI LIMIT ;SUBTRACT LIMIT
JF BLANK ;NEAR EDGE IF > 0

; TEST IF Y NEAR EDGE
LDAOX A, YH+7 ;X Y POS OF TORPEDO
ASX 255 VALUE OF X
CPI LIMIT
JF BLANK ;NOT NEAR EDGE

; TORPEDO HIT SCREEN
LDAUX A, XH+1
ASX
CPI XH
JLST BLANK ;PLAY COLLISION INST
LDAUX A, YH+1
ASX
CPI YH
JLST BLANK ;PLAY COLLISION INST

; SCORING PROGRAM
; SCORE: LOAD X, A ;SCORE VALUE
; SCEK: SET ZH
; SCORE: LOAD D, A ;SCORE VALUE
; INK </ink>
Listing 5, continued:

; PROGRAM TO ROTATE SHIPS, INITIATE TORPEDO FIRE,
; DO HYPERSPACE PROCESSING & CHECK
; FOR SPENT TORPEDOS

; DON'T CHECK BUTTONS IF ALREADY IN HYPERSPACE
LOADX A, HFLG
CPI D, 0
JNZ SPNCCK: CHECK FOR SPENT BULLETS ANYWAY

; CHECK FOR BOTH CW AND CCW,
; INDICATING HYPERSPACE
LOADX A, HYPM: MASK FOR HYPERSPACE
MOVT M, E
IFA D, 0
TRAF E
JNZ CWCK: NO - CHECK FOR CW

; SLKNN SHIP
LD3 X A, NULL: NULL GRAPHICS SUB
LD3 X PCPN: PN: R NOT CALL
NOVT M, 0
AXA H
NOVT M, 0
NOW M, A

; SCHEDULE SHIP START AFTER HOLY
LD3 X PEPT: SHIP START ENTRY
MV1 M, 1
LDA TIME
MV1 A, HOLY
MV1 M, 0
MV1 M, A

; SEE IF WE NEED TO DESTROY SHIP LATER
MV1 D, 1
SCAL 2: RANDOM # IN A
CPI 0: 1 - ZERO
JP NOST: YES, DON'T DESTROY
MV1 D, 0: NO-DESTROY

; END OF HYPERSPACE PROCESSING
; CHECK FOR CW CLOCKWISE ROTATION
CWCK: LOADX A, CWM: MASK FOR CW
MV1 A, D
JZ CWCK: CHECK FOR CW

; CHECK FOR CCW (COUNTERCLOCKWISE) ROTATION
CCCK: LOADX A, CCWM: MASK FOR CCW
MV1 A, D
JZ CCCK: CHECK FOR CCW

; CHECK IF FIRE BUTTON ON
FINC1: LD3 X SAPNT: BUTTON ADDR
NOVT M, A
LOADX A, FIERN: FIRE MASK
MV1 M, 0
MV1 M, 0
NOVT M, 0
MV1 M, A

; CHECK FOR SPENT TORPEDOS
SPNCCK: LD3 X PNPNT: TORPEDO POINTERS
; FIND CLAIMED TORPEDO WITH TIMEOUT
; FOR THIS SHIP
NOVT M, H: HL -> 9C
NOVT M, C, L
LOADX A, RENT
CPI 0: TEST FOR CLAIMED
MV1 M, D
GRT A: DECREMENT TIMER
STORX A, HONT: RESTORE IN BLOCK
HMZ
NC: MV1 M, 0

; END

; KEYBOARD DECODE PROGRAM

; CTLC EDQ: 'C'
KBC6CH: NOVT M, C
CPI 0: SAVE CHAR
CPI 1: CTLC "C"T
RCU: RETURN IF NOT
LX: MV1 M, INIT1: SCHED INIT
MV1 M, 0
LV: MV1 M, INIT2: SCHED INIT
MV1 M, 0
REI

BEAUTIFUL
NEW VECTOR-PAK
CASES for micro-computer
circuitry, assembled. Constructed
of aluminum, finished in vinyl. Slide out covers for easy
access. Includes card guides, heavy chassis plate, perforated
bottom cover for cooler operation.
Card guides perpendicular to front panel, Model VP1, $128.30.
Card guides parallel to front panel, Model VP2, $134.30.

S100 MOTHER BOARD, $29.50. 11 positions ready for
connectors. Glass epoxy, plated thru holes, circuitry for
active termination, 12 tantalum capacitors and instructions.
PLUS revolutionary Silt-N-Wrap wiring tools, Micro-Vector-
bord® printed circuit kits, I.C. sockets, extenders.

Send for new catalog.

VECTOR ELECTRONIC COMPANY, Inc.
12460 Gladstone Avenue, Sylmar, CA 91342
phone (213) 365-9661, twx 910-496-1539

Circle 393 on inquiry card.
The new VALUE-STANDARD in personal computing systems! Heathkit computers give you the power and performance to go wherever your imagination and programming prowess take you. They’re designed to get you up and running fast, interface with I/O devices easily and quickly, accept additional memory and I/O devices, store and retrieve data with speed and accuracy, respond to your requests with lightning speed. They offer complete mass storage capabilities, power and reliability for any programming application, and they’re priced low enough to give you real VALUE for your computer dollar! We’ve told you they’re the ones you’ve been waiting for — here’s why!

These Heathkit computer products are “total system” designs with powerful system software already included in the purchase price. They’re the ones you need to get up and running fast. And they’re backed by superior documentation including easy-to-follow step-by-step assembly and operations manuals, and service support from the Heath Company, the world’s largest and most experienced manufacturer of electronic kits.

NEW H8 8-Bit Digital Computer. This 8-bit computer based on the famous 8080A microprocessor features a Heathkit exclusive “intelligent” front panel with octal data entry and control, 9-digit readout, a built-in bootstrap for one-button program loading, and a heavy-duty power supply with power enough for plenty of memory and interface expansion capability. It’s easier and faster to use than other personal computers and it’s priced low enough for any budget.

NEW H11 16-bit Digital Computer. The most sophisticated and versatile personal computer available today — brought to you by Heath Company and Digital Equipment Corporation, the world leader in minicomputer systems. Powerful features include DEC’s 16-bit LSI-11 CPU, 4096 x 16 read/write MOS memory expandable to 20K (32K potential), priority interrupt, DMA operation and more. PDP-11 systems software for fast and efficient operation is included!

NEW H9 Video Terminal. A full ASCII terminal featuring a bright 12” CRT, long and short-form display, full 80-character lines, all standard serial interfacing, plus a fully wired and tested control board. Has autoscrolling, full-page or line-erase modes, a transmit page function and a plot mode for simple curves and graphs.

NEW H10 Paper Tape Reader/Punch. Complete mass storage peripheral uses low-cost paper tape. Features solid-state reader with stepper motor drive, totally independent punch and reader and a copy mode for fast, easy tape duplication. Reads up to 50 characters per second, punches up to 10 characters per second.

Other Heathkit computer products include a cassette recorder/player and tape for mass storage, LA36 DEC Writer II keyboard printer terminal, serial and parallel interfaces, software, memory expansion and I/O cards, and a complete library of the latest computer books. The Heath User’s Group (HUG) provides a newsletter, software library and lots more to help you get the greatest potential from your Heathkit computer products. We’ve got everything you need to make Heath your personal computing headquarters, send for your FREE catalog today!
HEATHKIT CATALOG

Read all about our exciting computer systems and nearly 400 other fun-to-build, money-saving electronic products in kit form.

Prices are mail-order FOB, Benton Harbor, Michigan. Prices and specifications subject to change without notice. Heath Company, Dept. 334-340 Benton Harbor, Mich 49022

HEALTHKIT CATALOG

FREE!

Also Available: the famous LA36 DEC Writer II Keyboard Printer Terminal only $1495 (Assembled)!

H8
8-Bit Computer
$375 Kit

HII
16-Bit Computer
$1295 Kit

H9
Video Terminal
$530 Kit

H10
Paper Tape Reader/Punch
$350 Kit

FREE!

HEATHKIT CATALOG

Read all about our exciting computer systems and nearly 400 other fun-to-build, money-saving electronic products in kit form.

Prices are mail-order FOB, Benton Harbor, Michigan. Prices and specifications subject to change without notice. Heath Company, Dept. 334-340 Benton Harbor, MI 49022

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

Name:

Address:

City: State:

CP-126 Zip:

Heath Company, Dept. 334-340 Benton Harbor, Michigan 49022

Circle 394 on inquiry card.
The newly available floppy disk drive units have opened many new areas of applications for microprocessor users. In the area of high utilization data, and fast interactive programs, the capabilities of random access disk storage devices are unequaled. Unfortunately, the price of such units often puts them outside the budget of personal computer users. Even when disk units are available, the advisability of using these devices for storing low utilization programs or data may be doubtful. You don't really need a $1000 auxiliary storage device to maintain a Christmas card list which is used once a year.

The new, special design digital cassette recorders have also generated a great deal of interest. These units allow software control of all tape functions, and provide for tape position sensing. While the price of these units is about half that of a single disk drive, they are still somewhat expensive. Like disk units, they also require fairly sophisticated interface hardware for proper operation.

Audio cassette units, on the other hand, are cheap and readily available. These units are rapidly becoming the standard auxiliary storage medium for small computer systems. Audio cassette units do, however, have several shortcomings. Compared to the other devices, they are slow, and most standard audio units do not have provisions for sensing the relative position of the tape. Further, most such units use a single motor for all tape operations, and accomplish the fast forward and rewind functions through gearing between the motor and the capstans. The physical pressure required to engage these gear trains virtually prohibits software control of these functions.

There are applications in which these shortcomings can be minimized. These applications are those that involve the use of auxiliary file storage where the information stored is inherently sequential in nature. While audio cassettes cannot compete with either digital cassettes or disk files in either speed or flexibility, they can provide very satisfactory results in the processing of sequential files.

This situation closely parallels the case in the data processing industry, where sequential tape files still play a major role in auxiliary storage. The economic advantages of these units can often offset the shortcomings of this type of storage.

About the Author
Wayne D Smith is currently an assistant professor of computer science at Angelo State University, San Angelo TX. He received his PhD in Computer Science from the University of Illinois in 1976. Besides his professional and hobby interest in computers he also enjoys operating an amateur radio station and flying.
Figure 1: Relationship between files, records and data items in a sequential file.

Nature of Sequential Files

Figure 1 indicates the basic relationship between the components of a file oriented data processing system. In this type of system, information is arranged in sets or collections called files. The information within a file is usually related by the type of data contained in the file. For example, a small business might maintain separate files for employee records, customer accounts and stock inventory.

Files are then subdivided into smaller units called records. There is a single record for each entity within the file. In the case of the files mentioned above, there would be a record for each employee in the employee file, a record for each customer in the customer file, and a record for each stock item in the inventory file.

A record is simply a collection of related data items. In any file processing system, the user specifies the data items that constitute a record. For example, the records in a Christmas card list file would probably contain five data items. These items would be: name, address, city, state and zip code. These items are related within a single record in that they all pertain to the same individual. A collection of these records, whether on magnetic tape, magnetic disk, or in an address book, constitutes a file.

Files which are stored on magnetic tape are called sequential files because the records in such a file are stored one after the other on the recording medium. One of the major disadvantages of this storage type is the sequential arrangement of files. In order to obtain the information stored in the Nth record in such a file, the preceding N-1 records must be bypassed. Without tape position indicators, this means that the preceding N-1 records must be processed at normal tape read speed before the Nth record can be read.

A second, related problem concerns identifying individual records. Since many records may be read before the desired record is reached, some provision must be made for determining when the correct record has been found. In practice, this problem is usually overcome by providing each record in the file with a separate data item that can be used to identify that particular record. This identification must, of course, be unique for each record in the file.

Many types of records already contain data items which may be used for identification purposes. An employee file, for example, would probably contain a social security number in each record. This number is unique for each employee, and could be used for record identification. In other cases, an additional data item would have to be
added to the record solely for purposes of identification. In the Christmas card list mentioned earlier, a sixth data item would have to be added in order to provide for record identification. For practical reasons, alphabetic data items, such as names, are seldom used for record identification.

While the records within a file do not necessarily have to be arranged in any special order, the search for a specific record is greatly facilitated if the records are stored in some orderly fashion. Sequential tape records are usually arranged in numerical order based on the identification number. In fact, this numerical ordering of records is a crucial requirement if efficient sequential file processing is to be possible. All the algorithms involved in processing sequential files are predicated upon this ordering.

The actual order of the records is immaterial. The records may be arranged in either ascending or descending order, depending on the preference of the user. Since ascending usually seems more natural to people, this arrangement will be assumed in the discussion which follows.

The sequential ordering of records has a large impact upon the speed with which a file can be processed. Suppose, for example, it becomes necessary to look up the addresses from five records in the Christmas card file mentioned above. If the records are unordered, each search must begin at the first record and search until the desired record is found. After listing this record, the tape must be rewound, and the search for the next record initiated from the beginning of the file. With this strategy the average number of records which must be read in order to find a specific record is one half the number of records in the file. Finding five records would require, on the average, reading 2.5 times the number of records in the entire file. In the worst case, it could require reading almost five times the number of records in the file.

If, on the other hand, the file is arranged in numerical order, the process is greatly simplified. In order to take advantage of the sequential ordering of the file, the numbers of the records to be found are also entered in sequential order. When this procedure is followed, processing time can be greatly reduced. After the first record is read and processed, the second search can begin without rewinding the tape. Since the second record has a higher number than the first one, that record must follow the first record. This means that the maximum number of records which must be read in order to process all five addresses cannot exceed the number of records in the file. This represents a considerable savings over the unsorted case.

Note, however, that the number of records read does not vary greatly regardless of the number of addresses to be found. That is, it requires about the same tape read time to process one record (average: one half the file length) as it does to process all the records in the file (full file length). This leads to the obvious conclusion that the type of files which are best suited to sequential storage are those in which a high percentage of the records are processed during each computer run. The number of records processed in a single run, divided by the number of records in the file, is called the activity ratio. As this value approaches one, processing efficiency approaches its maximum. A Christmas card list in which all records are listed for printing address labels has an activity ratio of 1.0. In this instance, sequential files offer a very efficient type of auxiliary storage.

Updating

As outlined in the preceding section, the processing of a sequential file does not differ greatly from the processing of any other type of file. The major differences appear when it becomes necessary to update, or change, the information which is stored in a sequential file. Updating may become necessary for any of several reasons. In some cases, data items may become inaccurate, as when a member of the Christmas card list moves. In other cases, it may become necessary to add or delete entire records from the file, as when a new employee is hired or an old one retires.

Due to the nature of magnetic tape drives, it is virtually impossible to position the tape head over a specific data item within a record. In fact, it is almost as difficult to position the tape head at the beginning of a specific record, since the record must pass the tape head before it can be read and identified. The process of adding new records in the correct sequential position presents additional difficulties. Inserting such records will necessitate moving all subsequent records down in the file. Minor errors in tape head position can result in erroneous or unreadable data.

To overcome these, and other, difficulties, the technique of using two tape units is employed. Instead of trying to change the information in the old file, a completely new file is created. This new file contains all the data and records from the old file that remain valid, plus any corrections, additions
Why you should buy a digital multimeter from the leader in digital multimeters.

If you're shopping for your first multimeter, or moving up to digital from analog, there are a few things you should know.

First, look at more than price. You'll find, for instance, that the new Fluke 8020A DMM offers features you won't find on other DMMs at any price. And it's only $169.*

Second, quality pays. Fluke is recognized as the leading maker of multimeters (among other things) with a 30-year heritage of quality, excellence and value that pays off for you in the 8020A.

Third, don't under-buy. You may think that a precision 3½-digit digital multimeter is too much instrument for you right now. But considering our rapidly changing technology, you're going to need digital yesterday.

If you're just beginning, go digital.

Why not analog? Because the 8020A has 0.25% dc accuracy, and that's ten times better than most analog meters.

Also, the 8020A's digital performance means things like 26 ranges and seven functions. And the tougher your home projects get, the more you need the 8020A's full-range versatility and accuracy. The 8020A has it; analog meters don't.

If you're a pro.

You already know Fluke. And you probably own a benchtop-model multimeter.

Now consider the 8020A: smaller in size, but just as big in capability. Like 2000-count resolution and high-low power ohms. Autozero and autopolarity. And the 8020A is MOV-protected to 6000V against hidden transients, and has overload protection to 300V ac.

Nanosiemens?

Beginner or pro, you'll find the meter you now have can't measure nanosiemens. So what? With the 8020A conductance function, you can measure the equivalent of 10,000 megohms in nanosiemens. Like capacitor, circuit board and insulation leakage. And, you can check transistor gain with a simple, homemade adapter. Only with the 8020A, a 13-oz. heavyweight that goes where you go, with confidence.

What price to pay.

$169.*

Of course, you can pay more. Or less. In fact, you could pay almost as much for equally compact but more simplistic meters, and get far less versatility. And, the 8020A gives you the 'plus' of custom CMOS LSI chip design, and a minimum number of parts (47 in all). All parts and service available at more than 100 Fluke service centers, worldwide. Guaranteed, for a full year.

Rugged. Reliable. Inexpensive to own and to operate; a simple 9V battery assures continuous use for up to 200 hours.

Where to buy.

Call (800) 426-0361 toll free. Give us your chargecard number and we'll ship one to you the same day. Or, we'll tell you the location of the closest Fluke office or distributor for a personal hands-on feel for the best DMM value going.

*U.S. price only

Fluke 8020A DMM for Home Electronics Experts: $169

FLUKE

Circle 395 on inquiry card.
or deletions that may be required. This is accomplished by using the old file as input (read only mode), while a new file is created in output (write only mode). Records which are to remain unchanged are transferred directly from the old file to the new file, via computer memory. Records which are to be changed receive part of the data items from the old file, but have other items updated from another input device (keyboard) before they are transferred to the new file. Records which are to be deleted from the file are read from the old file, but are not written on the new file.

Completely new records receive all the data items directly from the keyboard input device. After all data items have been entered, the new record is written on the new file. Program control insures that new records are entered in the correct sequential position in the new file. To facilitate the process of adding new records in sequence, all new files are initially created with substantial gaps between the record identification numbers.

The technique of creating an entirely new file instead of trying to correct an old file has one major disadvantage. This disadvantage is, of course, the requirement for two separate tape units. There are, however, several advantages which far outweigh this single shortcoming. One significant advantage is that the exact positioning of the tape is not critical. Since the new records are written on blank tape, there is no problem with record alignment, and writing over old data cannot occur.

Another advantage is that each tape unit is operated in a single mode. Using audio cassettes, this means that one unit is always in the record mode, while the other is in the play mode. During the update process, the tapes operate in only this mode. The fast forward function is not required, and the rewind function is used only after all processing has been completed.

This single mode type of operation greatly simplifies the operation of the cassette units during the update. Since each cassette operates in only one mode, the function for each tape drive can be set up before processing begins. During processing, the tape units need only be turned on or off as records are read, processed and written. Since almost all cassette recorders have provisions for remote motor control, software can be utilized to activate the tape recorders as needed by commanding a suitable interface circuit such as that in figure 2.

One final advantage to the 2 tape system involves the protection of data. Since the old file is used only in the read mode, it will
NOW, DISKS TOO!

And they're Disk-tinctly Digital Group.

With the addition of our all-new Disk Subsystem, the Digital Group becomes the only manufacturer of microprocessors to offer you a real choice. Tape or disk. Or both. Whatever you want.

We spent a lot of time designing and testing our new Disk Subsystem, and it shows in the result. The Subsystem is highly flexible... gives you direct access to data... and completes your Digital Group System.

Here's what the Disk Subsystem consists of:

- **Disk Interface Card**—provides the electrical interface between the disk drives and the computer.
- **Disk Power Supply**—provides all power required for up to 2 drives.
- **Disk Cabinet and Cables**—cabinets provide housing for 1 to 2 drives plus the power supply; assembled cabling is provided for connection between all system components.
- **Disk Drives**—the Digital Group Disk Subsystem supports standard 8" floppy disk drives from a number of manufacturers. Drives can be mixed, and in addition, mini-floppy drives are available for the function of a disk at minimum cost.
- **Documentation and Software**—the Digital Group's standard documentation includes construction guides, maintenance manual, theory of operation and other useful information to help you get the fullest value from your new Disk Subsystem. And there's also software to provide initial support.

The price? Low. It's become a tradition with the Digital Group to offer you high-quality, state-of-the-art equipment at the lowest possible prices. Our new Disk Subsystem is no exception.

For only $745, you can get a standard 8" floppy drive and an interface card. A complete single-drive Disk Subsystem with interface, power supply, cabinet, and cables is as low as $995.

Now the Digital Group system is really complete. If you don't own it all yet, take another longing look. In addition to our CPU, keyboard and video monitor, you can add an impact printer and the cassette storage system. All in our beautifully coordinated custom cabinets. All unmistakably Digital Group.

The Digital Group now has disks. And a lot more. Why not find out about all of our exciting products? Simply call or write today for direct information. From the Digital Group.
remain intact even in the event that the updating program goes awry. If the user follows the practice of always maintaining the most recent version of the data, he or she will never be further than one update away from a current file.

**Hardware Requirements**

Assuming that a microprocessor already has a cassette tape interface, the implementation of a sequential file processing system requires very little hardware. As mentioned above, two cassette tape recorders are required, one for input, reading files, and one for output, writing files. The microprocessor tape input and output circuits are connected to the appropriate tape unit, as shown in figure 2. Even when a single circuit is used for both input and output, no problem arises, since only one cassette will actually be operating at any given time.

The cassettes can, of course, be operated manually if desired, but software control is easily implemented. Two latched bits of an output port are used to provide tape motor control. These output signals are used to activate the tape motors through opto-couplers (Radio Shack #276-1628, Texas Instruments TIL 111, or equivalent) and a single transistor.

The opto-couplers are used to prevent any polarity or voltage problems which might arise from direct interconnection of the microprocessor and tape motor circuits. In many cassettes, the positive terminal of the tape unit will be connected to the microprocessor ground terminal through the microphone and earphone cables. This arrangement would prohibit direct interconnection of the two circuits. Even when the electronics of a specific cassette would permit direct interconnection, the couplers are still worthwhile. Using the couplers will allow changing recorders at a later date without regard to the voltage or circuit ground polarity.

The only critical part of the circuit is the polarity of the transistor. The polarity of the remote jack should be checked before completing the wiring. Almost any NPN transistor will work, provided it can handle the current required by the tape motor. The current requirement for a specific tape motor is easily determined by connecting an ammeter across the remote jack with the tape in the play or record position. There is a fairly high current surge as the motor starts, but this is of short duration. A steady state current of 50 to 100 mA is about average.

The inverters and LEDs shown in figure 2 are optional. The indicator LEDs have proven to be very useful, since they are the only way to determine whether a tape unit is on or off (short of looking at the cassette). They can also provide a measure of amusement when a long tape program is running.

The operation of this interface system is quite straightforward. In order to process a tape file, the user first insures that the appropriate motor control is off. The cassette tape is then loaded and the tape function control is set for play or record as required. If a tape is to be recorded, it must be advanced past any leader on the cassette.

After the tape is loaded and the tape control is set, program execution can begin. The functions of reading or writing the tape are then under software control. If a tape record is to be read, the applicable motor control is turned on by writing a 1 in the appropriate bit of the output port. Normal tape input routines are then used to read the record. When the record has been read, the motor is turned off. For writing a record, the motor control is turned on, and normal tape dump routines are used to write the record on tape. A software delay is necessary in this case in order to allow the tape motor to come up to speed before recording begins. A second or two is usually sufficient.

**Buffer Storage and Label Records**

Processing sequential tapes requires that a certain amount of memory be allocated for storing the records as they are read or written. This area is called a buffer, and is used to store the record currently being processed. The amount of memory required is minimal, since there is normally only one record in memory at a time. In special cases, when new records are being added to an existing file, storage for two records will be required. In all other cases, a single record length buffer is sufficient. Records are read into the buffer and then processed. Records may be written from the buffer onto a new file or to an appropriate output device as desired. For a generalized tape handling system, enough memory should be allocated to provide for twice the length of the largest tape record that will be processed. Not all of this area will be used by every program, but it should be provided in order to keep the system as general as possible.

In a sequential file processing system, a number of cassette tapes are required for storing the records of various files. More than one file may be stored on a single cassette, if desired, but this will result in
New books...for everyone who's into their own computer. Programs. Fun. Games. Languages. Excitement.

SCELBAL, the new microcomputer language that's simpler than machine language.

SCientific Elementary BAsic Language for "8008"/"8080" systems. A complete, illustrated program book. Routines. Techniques. Source Listings. Flow Charts. And more. Took several years to develop. Now yours for many years to come. First time that intimate details of higher level language has been offered for such a low price. Only $49!

You get 5 Commands: SCR, LIST, RUN, SAVE, LOAD.
14 Statements: REM, LET, IF...THEN, GOTO, FOR with STEP, END, INPUT, PRINT, NEXT, GOSUB, RETURN and optional DIM. 7 Functions: INT, SGN, ABS, SQR, RND, CHR, TAB. And, it runs in 8K and more. Here's all the data needed to customize a high level language for your system...at a fraction of the cost!

Order your copy today! Get started advancing your system! $49 p.p.d.

SCELBAL SUPPLEMENTS...
SCELBAL has taken off so fast, two special supplements had to be printed. First, there's Extended Math Functions: SIN, COS, LOG (BASE E), EXP (BASE E) and ATN...only $5.00 p.p.d. The second supplement, String Handling Capabilities, includes the numeric functions LEN and ASC. It sells for only $10 p.p.d.

Cook up mouthwatering programs for your "8080" or "6800"!

Delectable "how to" facts, including descriptions of "8080" or "6800" instruction sets. How to manipulate stacks. Flow charts. Source listings. General purpose routines for multiple precision operation. Programming time delays for real time applications. Random number generators. And more. You even get floating point arithmetic routines! Input/output processing for basic I/O programming through interrupt processing. And so much more, we can't list it all here. SCELBI's Software Gourmet Guides and Cookbooks for "8080" or "6800". (Specify!)

Order your copies today! Start cookin'! Bon Appetite. $9.95 each p.p.d.

GALAXY Microcomputer Outer Space War Games...

Captain your own starship on an inter-galactic journey to adventure. Meet alien ships in realistic combat. Plan a painstaking journey filled with battles, refueling problems, weaponry, warp factors and more—all against your "6800" computer. It is a complete book, written in machine language for 4K memory. It's an ongoing, ever-changing interstellar adventure, including source listings, flow charts, routines and much more. Blast off today to high adventure in outer space!

Order GALAXY today! $9.95 each p.p.d.

Scelbi Books are available in many fine Computer Stores.

Scelbi Computer Consulting Inc.

Milford, CT 06460

Prices 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.

Circle 377 on inquiry card.
slower operation unless the files are very short.

Each file will contain two distinct types of records. All the records except the first are the data records that contain the actual file information. The first record in each file is a special type of record called the label, or header, record. This record should have exactly the same format on all files. The label record contains general information about the file itself. It does not contain data, and hence does not need an identification number. It is always the first physical record on the tape.

The purpose of the label record is to provide all the information which is required in order to process the file. Since this information changes from file to file, the obvious place in which to store it is in the file itself. Therefore, the first record in every tape file is dedicated to the label record.

The label record may be as simple or as elaborate as the user wishes to make it. At a minimum, however, it should contain:

- The file name or other identification.
- The date created (due to updating, several files of the same name may exist).
- The length of a data record.

In a more elaborate system, the label record could also contain:

- The number and type of data items.
- A description of all data items and their type (i.e., binary, BCD, ASCII, etc.).
- A security code (access can be denied without knowledge of this code).

All the remaining records on the tape are data records. The length and format of these records will vary from file to file depending on the amount and type of data in the file. The number of data records in a file will also vary.

Any program which processes tape files of this type must first read the label record. The information contained in the label record is then used to set variables that tailor the program for reading and processing the data records from that specific file. This procedure allows a single generalized program to process files of various types and lengths.

Utility Programs

In order to support the processing of sequential files, three general purpose utility programs are required. An optional fourth program may be provided if the user wishes a separate program for duplicating existing files. Running an update program without entering any changes will accomplish the same thing, however.

The first utility program is one which creates tape files from auxiliary input. This program performs many of the functions of a text editor. Data is entered via the keyboard, and is stored in the buffer until all the required data items have been entered. A sophisticated file creation program can be designed to query the user for data items one at a time until the record is complete. When all the data items have been entered, the record is written on the file. The process is repeated until the file is complete.

The file creation program must insure that the label record is written on the new file before any data is entered. As with the data records, the program can be designed so as to query the user for this information prior to writing the label record.

It may be desirable to provide the tape creation program with information about the number, length and type, i.e., character, binary or binary coded decimal (BCD) of data in each data item. Information about the type of data would be used primarily for error checking, but could also be used to allow data packing. Packing of numeric data would allow two hexadecimal or binary coded decimal digits to be recorded as a single byte. This would require additional program complexity in converting from ASCII input to a packed format. This complexity may be warranted in cases where the packed format would facilitate arithmetic manipulation of the data directly from the tape file. Conversion back to ASCII would be required when listing the file.

As the data records are created, the user must supply the record identification number for each record. The user must insure that these are entered in the proper numerical sequence. In order to provide for the addition of new records at a later date, an initial file should have records numbered in increments of about 100.

The second required utility program is one which makes a hard copy listing of the data stored in a file. The label record may or may not be printed, depending on user preference. The simplest type program would perform an unformatted dump of the data exactly as it is read from the tape. A more elaborate program could separate the various data items into different positions or lines on the page. This type of program would allow printing the data in the form of a customer statement, a check or an address label.

By far the most complex of the utility programs is the update program. This pro-
THE BEST DEAL WE HAVE EVER OFFERED!
ROYTRON 8-LEVEL READER/PUNCH BRAND NEW $295.00

This paper tape 1/0 system operates at 50 cps. The mechanisms share a common drive motor, but are electrically independent. 24 volt signals are required to interface the internal switches. An interfacing manual is included. Perfect for typewriting, program decoding, program entry and many other paper tape applications.

USE S-100 BOARDS WITH YOUR KIM-1

Now use low-cost and readily available S-100 compatible circuit boards with your KIM-1! The Kims from Forethought Products adapt the KIM Bus to the S-100 Bus so that you have access to hundreds of memory and interface boards.

The Kims accepts up to 8 S-100 edge connectors, provides on-board power regulation for the KIM-1 and contains the KIM to S-100 interface logic. One S-100 edge connector is included, the KIM to Kims connector set must be purchased separately.

<table>
<thead>
<tr>
<th>Item</th>
<th>Kit Price</th>
<th>Assembled Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>KIMS Backplane,S-100 Adapter</td>
<td>$125</td>
<td>$165</td>
</tr>
<tr>
<td>KIMS Power Supply, UHrug+B/B10A+15B3A, 15B1A</td>
<td>$60</td>
<td>$75</td>
</tr>
<tr>
<td>KIM-1 Inc. Documentation</td>
<td>N/A</td>
<td>$245</td>
</tr>
<tr>
<td>KIM to KIM Connector Set</td>
<td>$8</td>
<td>$11</td>
</tr>
<tr>
<td>S-100 Edge Connector</td>
<td>$5</td>
<td>$7</td>
</tr>
<tr>
<td>Cassette Recorder for KIM-1</td>
<td>$55</td>
<td>$55</td>
</tr>
<tr>
<td>4K Selectable RAM Board</td>
<td>$119</td>
<td>$178</td>
</tr>
<tr>
<td>4K Selectable RAM Board</td>
<td>$136</td>
<td>$219</td>
</tr>
<tr>
<td>4K Selectable RAM Board</td>
<td>$140</td>
<td>$245</td>
</tr>
<tr>
<td>4K Selectable RAM Board</td>
<td>$291</td>
<td>$369</td>
</tr>
<tr>
<td>Poly Video Terminal Interface</td>
<td>$218</td>
<td>$280</td>
</tr>
<tr>
<td>Matrix 2400 Alpha-Numeric VMAM</td>
<td>$210</td>
<td>$255</td>
</tr>
<tr>
<td>Matrix 256-128 Video Graphics Board, Displays 256 x 256 array</td>
<td>$205</td>
<td>$295</td>
</tr>
<tr>
<td>My Tiny BASIC 2.5L, kit in a paper tube</td>
<td>$5</td>
<td>$8</td>
</tr>
<tr>
<td>Focal (a DEC trademark) includes flashing piles</td>
<td>$40</td>
<td>$40</td>
</tr>
<tr>
<td>8502 Assembler with Source Code comes with the MOS version</td>
<td>$50</td>
<td>$50</td>
</tr>
</tbody>
</table>

SPECIAL! 10% off S-100 Boards when ordered with Kims.

ECONOMICAL KEYSTATIONS $29.50

A super buy on these used keystations allows us to sell them at this low price. These attractive units contain a Hickok-style keyboard which is easily modified to output ASCII character codes (we provide documentation). The interior of the enclosure is large enough for a KIM-1 and power supply. We also have a limited supply of keystation components.

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>56 key keyboard</td>
<td>$10</td>
</tr>
<tr>
<td>Mounts on 11 5&quot; centers</td>
<td>N/A</td>
</tr>
<tr>
<td>Keystation Enclosure</td>
<td>19</td>
</tr>
<tr>
<td>Dimensions 16&quot; x 16&quot; x 4&quot;</td>
<td>N/A</td>
</tr>
<tr>
<td>Copy Stand and Holder</td>
<td>5</td>
</tr>
<tr>
<td>Mounts on top of enclosure</td>
<td>N/A</td>
</tr>
</tbody>
</table>

MORE BARGAINS

Burroughs Self Scan Display Panel | $325.50
Model 9000 16 or 18 character display | $325.50
Ultrasonic Leaking Relay for Remote Control | $19.95
Operate 110v appliances from across the room | $19.95

data General 8011 CRT Terminal | $115.00
Brand new with numeric keypad, 4800 baud max | $115.00
DEC PDP8/L1 with 4K core memory | $200.00
Take-up or floor models available as low as 795.00

MATROX VIDEO RAMS AND GRAPHICS

Tops in the industry, please write for more info. See our KIM Corner for S-100 compatible boards.

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTX 816 Module (Upper Case only)</td>
<td>$179.00</td>
</tr>
<tr>
<td>MTX 1632 Module (Upper and Lower Case)</td>
<td>$275.00</td>
</tr>
<tr>
<td>MTX 162 SL Module (With 2552&quot;2)</td>
<td>$232.00</td>
</tr>
<tr>
<td>MTX 2480 Circuit Board</td>
<td>$395.00</td>
</tr>
<tr>
<td>MTX 23A-2 Circuit Board, 256 x 256 array graphics (multiples provided colors)</td>
<td>$320.00</td>
</tr>
</tbody>
</table>

BOOKS

Programming Microprocessors including arithmetic with floating points | $5.95
Build Your Own Talking Robot | $5.95
Telephone Accessories You Can Build | $3.95
Master Handbook of 1981 Practical Electronics Circuits | $9.95
Basic BASIC | $7.95
Advanced BASIC | $6.95
101 BASIC Games: The new classic volume compiled by Cicuit Arts | $7.95
What to Do After You Build Games: From the People's Computer Company | $9.95
CMOS Cookbook | $9.95
Introduction to Microcomputers Volume I: Adam Osborne explains CMOS low-power logic | $9.95
Introduction to Microcomputers Volume II: Then he reviews the current state of micros | $12.95
8080 Programming for Logic Design | $7.95
6800 Programming for Logic Design | $7.95
Seven Common BASIC Programs: Full of many useful and complete programs | $7.95

<table>
<thead>
<tr>
<th>Qty</th>
<th>Item</th>
<th>Price</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total for goods</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>+4% Michigan Residents</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>+4% Shipping and Handling</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>+$1 if order's under $20</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total Amount</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COMPUMART INC. 1250 N. Main St. Ann Arbor, MI 48104 994-4445
A Division of Newman Computer Exchange, Inc.
Figure 3: Sequential file updating algorithm allows the copying of records, deleting of old records, inserting of new records, and modifications of old records. This is a very generally applicable method, usable whenever serial format records are being processed.

The crux of the update program lies in the comparison of the keyboard entered identification number and the identification number from the tape record just read. When the tape ID is less than the keyboard ID, this indicates that the tape record just read is not to be changed. This record is copied to the new file exactly as it existed on the old file.

When the tape ID is equal to the keyboard ID, the tape record just read is one which is to be either changed or deleted. A keyboard entry determines which action is to be taken. If the record is to be changed, the new data items are entered from the keyboard into the buffer area, directly over the data that was read from the old tape file. When the changes have been completed, the corrected buffer is written onto the new file.

If a record is to be deleted, the program simply ignores the current tape record, and reads the next record into the buffer. The deleted record, therefore, does not appear on the new file.

When the tape ID is greater than the keyboard ID, this means that the program was unable to find a record ID on the old file which matched the keyboard ID. Therefore, the keyboard ID represents either a new record, or a typing error. When a new record is to be added, this record is entered into the second buffer area in memory. The use of this second buffer is necessary in order to prevent overwriting the last tape record read, which is currently stored in the first buffer. After the new record is entered, it is written to the new file from the second buffer. The added record now appears in proper sequence on the new file, while the last record read from the old file has been preserved. Processing then continues with the input of another keyboard ID number.

Whenever sequential files are processed, there is always a minor problem concerned with determining when the end of a tape file has been reached. In the program shown in figure 3, the old programmer's trick of using a dummy record with an ID number of all 9s is used to indicate the end of usable data in the tape file. The same code is used to signal the end of data entry from the keyboard. When this technique is used, the user must remember to provide this dummy record at the time a new file is created.

Examples

There are many instances in which the sequential file processing system outlined...
Many books on microprocessors and their use are now on the market, and most of them have names that sound alike. But Osborne & Associates' books have dominated this market since 1975, when our first book appeared. With rave reviews from all over the world — with more than five hundred university text adoptions, our books are all best sellers. In fact, "An Introduction To Microcomputers: Volume I — Basic Concepts" now holds the world's record in sales volume for any textbook sold for a profit.

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

**"An Introduction To Microcomputers: Volume O — The Beginner’s Book"**  By Adam Osborne.
This is the book for the absolute beginner. Assuming that you know nothing about computers, math or science of any kind, this book explains what computers are all about — and it takes you to the point where you can read Volume I. 300 pages.
Book No.: 6001 $7.50

**"An Introduction To Microcomputers: Volume I — Basic Concepts"**  By Adam Osborne.
The world's best selling textbook. This book explains, clearly, concepts common to all microcomputers, yet specific to none. 350 pages.
Book No.: 2001 $7.50

**"An Introduction To Microcomputers: Volume II — Some Real Products"** (Revised June 1977)  By Adam Osborne, Susanna Jacobson and Jerry Kane.
This book describes every common microprocessor and all of their support devices. Information is new and clearly written. Only data sheets are copied from manufacturers. 1200 pages.
Book No.: 3001 $15.00

**"8080A and 8085 Assembly Language Programming"**  By Lance Leventhal.
For the microcomputer user, a series of books provide complete programs, written in BASIC. All these books are by Lon Poole and Mary Borchers.
"Some Common Basic Programs" 200 pages.
Book No.: 21002 $7.50
"Payroll With Cost Accounting — In Basic" 400 pages
Book No.: 22002 $12.50
"Accounts Payable and Accounts Receivable" 400 pages
Book No.: 23002 $12.50 [Available November 30, 1977]

**"General Ledger System"**
Book No.: 24002 $12.50 [Available December 31, 1977]

**"8080A and 8085 Assembly Language Programming"**  By Lance Leventhal.
This book is for the assembly language programmer or student; it explains assembly language programming for the 8080A and 8085 microcomputers. The book contains numerous examples. 400 pages.
Book No.: 31003 $7.50 [Available October 31, 1977]

**"8080 Programming For Logic Design"**  By Adam Osborne. 300 pages.
Book No.: 4001 $7.50

**"6800 Programming For Logic Design"**  By Adam Osborne. 300 pages.
Book No.: 5001 $7.50

**"8080A/8085 Assembly Language Programming"**  By Adam Osborne and Susanna Jacobson.
Book No.: 7001 $7.50 [Available November 30, 1977]
above could be used. An obvious example is the Christmas card list mentioned earlier. This file could be created and then stored for use each December. By using a fairly simple listing program, mailing labels could be generated directly from the file. Sometimes in January, the update program could be run to reflect any new addresses received with that year's cards. Persons from whom cards were received for the first time could be added to the file, and those from whom cards were not received could be deleted. The tape could then be stored until the following December. The January update is chosen as a point at which the activity ratio could be expected to be high. More frequent updates are possible, if needed.

An example which is not quite so obvious would be to use a sequential file as storage for a text editor which prepares input for an assembler. In this case, a record would consist of a single line of assembly language code. The data items would include: line number (record identification), label, operation, address and comments. Data items for location and machine code would be provided, but would be left blank by the file creation program.

After the file has been created, it is transferred to the input cassette for processing by the assembler program. The assembler then generates the location and machine code for each record in the file. As the machine code is generated, a new file is created on the output cassette. This file is identical to the input file except that the data items for location and machine code are filled in by the assembler. Optional features in the assembler could provide for loading the machine code for execution, or for creating a hard copy listing of the program as it is assembled.

If errors are detected during assembly or execution, the original file can be corrected through the use of the update program. Since each line represents a record, incorrect lines can be changed by changing the data items in that record. Extraneous lines can be deleted, and new lines can be added, in their proper positions, through the process of deleting or adding records to the file.

In using an assembler of this type, a 2 pass approach is required. This technique is required in order to prevent unresolved references at the time the machine code is being generated. Without prior knowledge of all label locations, forward reference branch instructions could not be assembled as they are read from the input tape. The 2 pass approach offers the additional advantage of dividing the assembler into two separate programs, only one of which must be in memory at any one time.

A particular advantage to using sequential storage with this type assembler is that it allows the assembly of very large programs. Due to the nature of sequential file processing, there is never more than one source language statement in the machine at any time. By processing the statements one at a time, more memory can be devoted to the assembler itself and its reference tables. In a machine with limited storage, this can result in a more elaborate assembler than would otherwise be possible.

A final example is of a more commercial nature, and involves the processing of an accounts receivable file for a small business. The basic file contains permanent customer information, such as name, address, account number, etc. It also contains data items for previous balance, current balance, payments, and a number of charges. When the file is originally created, these last data items are all set to zero.

During the first month, an update program is run, using this file as input. The file is updated by entering charges for that month. After all charges have been entered, the program computes the total charges and enters this value as the current balance in the new record. The record is then written on the new file. When all accounts have been updated, a listing program is executed, using the new file as input. This listing program, in effect, generates the first month's statements for all customers.

During the second and all succeeding months, a slightly different update program is used. This program first combines previous balance, payments and all old charges to form a new previous balance. Entries are then accepted for both payments and new charges. When all entries are completed, a new current balance is computed. The listing program is then used to generate customer statements for the new month.

Arguments Against Audio Cassettes

There are two arguments which are often advanced against the use of audio cassettes in the manner outlined in the preceding sections. The first argument is that leaving the tape drives engaged for long periods of time can flatten the capstans, which causes wow and flutter when tapes are read. In a system such as that proposed above, this would not present a problem. Even when large files are being processed, the motors would not be engaged for more than a few minutes at a time. In the case of file creation programs, where this time might extend to
an hour or so, the tape unit is constantly being turned on and off. The probability of the capstan stopping in exactly the same position after each tape operation is quite small. Therefore, even when the capstan remains engaged for long periods of time, the exact position of the capstan is changing every few minutes.

Additionally, the capstan must be disengaged in order to remove the tape at the end of the program run. Since the tapes would normally be removed after processing, this would seem to preclude leaving the capstan engaged in the same position for a prolonged duration. For those people who are absentminded, the file processing programs could include a reminder to disengage the capstans after processing is complete.

The second argument against audio cassettes is their somewhat less than 100% reliability. Occasionally, a record will be written onto a file which cannot be read back. Even with good equipment, this can sometimes happen. In the file processing system described above, this presents only a minor problem, since it will seldom affect more than one or two records. If a record is unreadable, all the other records remain usable. Therefore, the tape update program can be used to add the unreadable record to a new file. Since the unreadable record does not "seem" to be on the old file, the process is treated as the addition of a new record by the update program. Even if several such records must be added the problem is of minor proportions, and represents more of a nuisance than a catastrophe.

Summary

It is readily admitted that sequential files are not a substitute for random access files. There are, however, many applications in which the disadvantages of sequential files can be minimized. In these instances, a substantial amount of file handling can be performed with very satisfactory results. In cases where disk files are not available, sequential files may be the only means of providing facilities for processing large volumes of data. Even in systems where disk files are available, the judicious use of sequential files can assist in conserving the more expensive disk storage facilities.

The software required to support a sequential file system is considerably less complex than that required by a disk file processing system, and the hardware costs are minimal. All in all, sequential files can provide a great deal of data processing capability at about one tenth the cost of a single disk storage device.

As Computer Enterprises Enters Their Second Year:

We thank you for your patronage and hope you continue to save with us on the world's finest hardware and software. Call us right now at 315-637-6208 to get in on the super deals below!

Brand New Products — Low, Low Prices. Save!

<table>
<thead>
<tr>
<th>Product</th>
<th>List</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSAI VIO Model B, Video Kit</td>
<td>$275</td>
<td>$225</td>
</tr>
<tr>
<td>IMSAI 16K Dynamic RAM Kit</td>
<td>$449</td>
<td>$382</td>
</tr>
<tr>
<td>IMSAI 32K Dynamic RAM Kit</td>
<td>$749</td>
<td>$637</td>
</tr>
<tr>
<td>Cromemco 16K Dynamic RAM Kit</td>
<td>$495</td>
<td>$421</td>
</tr>
<tr>
<td>Vector Graphic VECTOR 1+</td>
<td>$659</td>
<td>$560</td>
</tr>
</tbody>
</table>

Low-Priced Items You May Have Missed:

<table>
<thead>
<tr>
<th>Product</th>
<th>List</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>North Star Micro Disk System</td>
<td>$699.00</td>
<td>$599</td>
</tr>
<tr>
<td>Cromemco Bytesaver Kit</td>
<td>$145.00</td>
<td>$131</td>
</tr>
<tr>
<td>Cromemco Dazzler Kit</td>
<td>$215.00</td>
<td>$193</td>
</tr>
<tr>
<td>Cromemco Z2 Kit</td>
<td>$595.00</td>
<td>$535</td>
</tr>
<tr>
<td>Seal's 250ns RAM Kit</td>
<td>$295.00</td>
<td>$240</td>
</tr>
<tr>
<td>Seal's 8080 Wire Wrap Board</td>
<td>$37.50</td>
<td>$32</td>
</tr>
<tr>
<td>Our Own Extension Board with Connector</td>
<td>$30.00</td>
<td>$12</td>
</tr>
<tr>
<td>IMSAI RAM 4A-4 Kit</td>
<td>$139.00</td>
<td>$115</td>
</tr>
<tr>
<td>RAM 4A-A Socket Set</td>
<td>$22.00</td>
<td>$10</td>
</tr>
<tr>
<td>Compucolor 8001</td>
<td>$585.00</td>
<td>$427</td>
</tr>
<tr>
<td>Polymorphic VTl/64</td>
<td>$210.00</td>
<td>$189</td>
</tr>
<tr>
<td>Vector Graphic 500ns RAM</td>
<td>$265.00</td>
<td>$225</td>
</tr>
<tr>
<td>8K RAM Assm.</td>
<td>$99.00</td>
<td>$80</td>
</tr>
<tr>
<td>Vector Graphic PROM/RAM Kit</td>
<td>$89.00</td>
<td>$80</td>
</tr>
<tr>
<td>TDL ZPU Kit</td>
<td>$299.00</td>
<td>$224</td>
</tr>
</tbody>
</table>

Shipping charges: $10 per CPU on larger units; $1.50 per kit. $2.00 min. per order.

Delivery is stock to 30 days on most items. Shipment is immediate for payment by cashier's check, money order or charge card. Allow 3 weeks for personal checks to clear. N.Y. State residents add appropr. sales tax. Availability, prices and specs may change without notice.
ENTERPRISES, INC.

One-stop shopping for Hardware and Software

Everything you need in small computer systems with special emphasis on TERMINALS! Look at these units...compare price, quality, delivery, service...and you'll see why you don't have to look anywhere else!

**LEAR SIEGLER ADM 3A TERMINAL**
- Full addressable cursor
- Display format—24 lines of 80 characters per line
- Communications rates—75 to 19,200 Baud
- Computer interfaces—EIA standard

ADM 1A TERMINAL
- Display Format—80 characters per line by 24 lines
- Full cursor control
- Edit operations: clear screen, clear unprotected character type-over.
  (Options are Character Insert/Delete, Line Insert/Delete, Erase to end of page, Erase Line/Field)

ADM A1 Assembled...$1398.00

ADM 2A TERMINAL
- 1920 character display (24 x 80)
- 16 function keys for 32 commands
- Separate keyboard—119 keys
- 10 key numeric pad
- Single key edit operations
- Page, field or line edit
- Security, protected fields and much more

ADM 2A Assembled...$1895.00

**OKIDATA MODEL 110 LINE PRINTER**
- 110 CPS dot matrix
- Friction Feed...$1149.00
- Tractor Feed...$1279.00
- RS 232C Serial Interface...$260.00

**OKIDATA MODEL 22 LINE PRINTER**
- 125 lines per minute
- 132-column print line
- Upper/lower case
- 8 different character sizes
- 12 IPS paper slew

Tractor Feed...$2249.00
RS 232C Serial Interface...$379.00

**DECWRITER II**
- 132 column printing
- 10-30 CPS
- Full keyboard
- Tractor feed

$1695.00
SYNCHRO·SOUND
ENTERPRISES, INC.

CENTRONICS 703 SERIAL PRINTER
- Low cost of ownership
- Bidirectional logic seeking printing
- Microprocessor electronics
- Excellent print quality
$2395.00

CENTRONICS 761 PRINTER
- 300 Baud serial transmission
- Bidirectional and incremental printing
- RS232, CCITT-V24, or current loop interface
- Baud selection (110/150/300)
KSR with Keyboard $1695.00
Receive only version 1595.00

IMSAAI 8080 MICROCOMPUTER
- Powerful
- Low cost
- Easy to use
With 22 Slot Mother Board $619.95
With Z-80 CPU $849.95

HAZELTINE 1500 VIDEO TERMINAL
- Reverse video
- 24 x 80 display
- Programmable brightness levels
- RS232 and current loop and much more
Assembled .... $1149.00 Kit also available

HAZELTINE MODULAR 1 INTELLIGENT TERMINAL
- 1920 character display
- 8 different video levels
- Full editing capability
- Removable keyboard and much more
Assembled .... $1659.00

FLOPPY DISK SYSTEM
- Fully IBM 3740 media and format compatible
- Full formatter and controller built-in
ICOM Model FD 3712 Dual Drive System $2795.00

SPECIAL BUYS
- Sorac IQ 120 Video Terminal Kit $959.00
- DEC LA 180 Printer 2769.00
- Compucolor 8001 Color Computer 2595.00
- ICOM Microfloppy System 989.00
- North Star Microfloppy Disk Kit 599.00
- 2708 Eproms 27.95
- Javelin 9" Video Monitor 159.95
- Livermore Modem Model 76 299.30
- Sol 20 with Solos Kit (limited supply) 999.00
- Micropolis Model 1053MOD 2 1799.00
- IMSAI AP44-44 Col. Printer Kit 329.00
- TDL Xitan Alpha 1 Computer Kit 699.00

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

SYNCHRO·SOUND ENTERPRISES, INC.
The Computer People
193-25 Jamaica Avenue, Jamaica, New York 11423
212/468-7067 TWX: 710-582-5866
Hours 9-4 daily Visit our new showroom and Saturday Working units on display
Dept. BB BankAmericard + Master Charge

Circle 401 on inquiry card.
C: A Language for Microprocessors?

J Gregory Madden
JGM Development Labs
POB 2345
West Lafayette IN 47906

I would first like to congratulate Peter Skye for his work on the high level language project! The interest created by his letters in recent BYTEs has helped to prod the emergence of a high level language. PL/I is a fairly popular language, and a good choice as an initial high level machine independent language. But what about other languages? Is there perhaps one which might be a better choice? Let's take a brief look . . .

Enter FORTRANBASICSNOBOLPASCALCOBOLRPGCACPALALGOLPLI.

FORTRAN is a prehistoric beast from the 1950s, and though it has the highest "literacy rate" of any language in BYTE's 1976 survey, it is rather limited in its capabilities. BASIC is also popular. There are plenty of BASIC interpreters around and its capabilities (or, rather, lack thereof) are even less sufferable than FORTRAN's. Perhaps a BASIC compiler combined with an interpreter would be very helpful to speed up debugged programs, but let's continue. PASCAL is a reasonable candidate. COBOL would be fun to implement and it has a fairly high literacy rate; however, it is awfully wordy. Source programs take up lots of space and for microcomputers it is not a very good choice. ALGOL might be reasonable. As for RPG, FORMAC and TRAC, I know little about these languages; however, I don't believe they have any major advantages over others.

Enter C

What about this strange creature called C which has been vaguely mentioned here and there? Where did it come from, where is it going, and what's it all about?

C is a high level language based on an earlier language called B. It was written by several brilliant people* at the Bell Telephone Laboratories at Murray Hill Nj sometime around 1974. It is designed to run under the Unix operating system (also designed and written by these people) on a Digital Equipment Corporation PDP-11/45 thru 11/70 series machine. It now also runs on IBM equipment and is being rewritten for Interdata machines. The C users' rumor mill also mentions something about PDP-10 versions in the future. My purpose in writing this article is to convey the flavor of C and suggest its utility as a mode of expression for personal computing.

The language C offers a user the following features:

- Control structures which permit a flow of control by using the language, rather than having to program around the language.
- Expressions that eliminate most temporary variables and trivial statements, resulting in shorter, cleaner code.
- Pointers and character variables to do nonnumeric problems simply.
- Structures to enable easy implementation of complicated data concepts.

What does all of this mean? Basically, programming in C is easy, fun, and probably results in the fewest lines of comprehensible code per program of any general purpose programming language.

*Ken Thompson and Denis Ritchie.
FREE YOURSELF FROM THE ONE BYTE WORLD. MOVE UP TO THE TWO BYTE TEXAS INSTRUMENTS TMS-9900 6-BIT MICROPROCESSOR — WITH OUR — “SUPER STARTER SYSTEM” — TEC-9900-SS. SHOWN ABOVE, FEATURES HARDWARE MULTIPLY AND DIVIDE, 69 MINI-COMPUTER INSTRUCTIONS, 7 ADDRESSING MODES, EXPANDABLE TO A FULL 65K BYTES; MONITOR, TMS 9900 CPU, RAM, P-ROM, E-PROM, PROGRAMMER ALL ON ONE P-C BOARD BASIC OPERATING SYSTEM AS LOW AS $299 UNASSEMBLED $399 ASSEMBLED AND TESTED EXPLICIT MANUAL INCLUDED OR AVAILABLE SEPARATELY AT $35, TO LEARN MORE ... JUST TEAR OFF A PIECE OF THIS AD, PIN TO YOUR LETTERHEAD & RETURN TO TECHNICO OR CALL OUR HOTLINE 1-800/638-2893 TO RECEIVE FREE INFO-PACKAGE. ---DESIGN & TECH SUPPORT BY ROSSE CORP.
Into the Briny Deep

What's C made of? Let's take a look at some of the basic constructs of the language to see what makes it tick.

Each section of the C program consists of what we will call a function, something like a FORTRAN function. Even the main program is a function with the name main. The format is as follows:

```c
main( ) {
    statements
}
```

Arguments may be passed to a function by placing them between the parentheses in the definition and references to the function. C uses a call by value technique, rather than call by address for function arguments. This means that if a function changes the value of an argument variable, it does so only for that particular function call. How is this done? All variables in a function are stored on the stack rather than being assigned fixed locations. This means that storage space for variables is only allocated while they are being used, and is freed when control returns from a function. There are of course methods for changing the value of variables in the calling function during execution of the called function, but these methods are explicit exceptions to the call by value rule. Any C function can be used as a subprogram as well as its possible use as a main program. The example used main as a name, but as in many languages, there is a freedom to pick and choose descriptive names for C functions.

Variables are the usual: integer, character, floating point and double precision floating point. C variables are declared as in FORTRAN by a statement of the form:

```c
type var1, var2, ...
```

Where `type` may be: int, char, float or double. Variables may be made to stick around (ie: storage is allocated permanently rather than temporarily using the stack) if the word static is included before `type` in the declaration. Then the value will not change between function calls unless explicitly modified. Normally, variables are dynamically allocated space on entry to a function.

All variables must be declared in C, and declarations must precede executable statements. Statements are ended with a semicolon (`;`) to indicate to the compiler what is the logical end of a statement.

IO in C

IO is done with basically two system supplied functions called getchar and putchar. Each gets or puts one character from standard input or output; getchar is normally used by equating a variable of type char to it:

```c
c = getchar( );
```

The variable `c` will have the ASCII value of the next character available from the input file; putchar is normally used by giving it the variable to print:

```c
putchar( c );
```

The ASCII value of `c` would be output to the output file. The argument to putchar could be a single ASCII character:

```c
putchar( 'a' );
```
in which case the character “a” will be output to the output file.

printf is another useful built-in function, similar to the FORTRAN WRITE and FORMAT statements. It actually calls putchar to do the IO. Its arguments are a string of characters in quotes, followed by the variables to be printed. In the string, format specifiers consist of a percent sign followed by an optional field size, followed by a letter indicating the format to output as: d for a decimal number, o for an octal number, h for a hex number, s for an ASCII string and c for a single character:

```c
printf( "The value of c is: %d", c );
```

If the new line is desired, it must be specified by putting a \n in the string at the appropriate point:

```c
printf( "The string is %s, and the value of a is %d.\n", string, a );
```

This inserts a carriage return and line feed when the \n is scanned during execution.

Special Characters

A note is in order before we go on about special characters in C. How does one represent a carriage return or a tab, or a line feed, etc? In general, anytime you want a special character, it can be generated by a backslash followed by a single nonnumeric character. New line is \n, tab is \t, end of file is \0 (equivalent to 0). Also any ASCII character can be formed with a backslash and a 3 digit number: \040 is a space, etc. This enables all characters to be available for relational tests and use in computation.
SOFTWARE = TSC

TEXT EDITING SYSTEM: This 6800 editor is unlike any other micro editor. As well as the usual features, it also includes: content oriented commands, local and global commands, block move and copy, append and overlay features, as well as very comprehensive string manipulators. 5K
SL68-24 MANUAL & SOURCE LISTING $23.50
CT68-7 OPTIONAL CASSETTE $6.95
PT68-6 OPTIONAL PAPER TAPE $8.00

TEXT PROCESSING SYSTEM: A great companion to the TSC editor. The processor will allow convenient paragraphing, right hand justification, paging, titling, and general text formatting. 4K
SL68-29 MANUAL & SOURCE LISTING $32.00
CT68-9 OPTIONAL CASSETTE $6.95

TSC 6800 ARITHMETIC ROUTINES
SOAP: A very fast, 4 byte binary, floating point package. Includes integer and conversion routines as well.
SL68-25 $10.00
FLOATING POINT PACKAGE: A BCD math package with 9 digits of precision.
SL68-4 $6.50

SCIENTIFIC FUNCTIONS: Requires SL68-4 and provides all scientific functions including SIN, TAN, LOG, LN, HYPSIN, and others.
SL68-20 $10.00

DIAGNOSTICS FOR 6800:
No system is complete without a set of diagnostic programs. Includes 5 memory tests, serial I/O tests, parallel I/O tests, plus others.
SL68-23 SOURCE LISTING $10.00

MICRO BASIC PLUS: The best small BASIC available for 6800. In just 3½ K, a complete interpreter including GOSUB, IF THEN, FOR NEXT, DIM, ON GOTO and GOSUB, READ DATA, plus the functions RND, SPC, TAB, EXP, and ABS. Five full digit integer math is supported.
SL68-19 MANUAL & SOURCE LISTING $15.95
CT68-5 OPTIONAL CASSETTE $6.95
PT68-5 OPTIONAL PAPER TAPE $6.00

TSC MNEMONIC ASSEMBLER: Another 6800 resident assembler? Yes! But this one is many times faster than others due to a very efficient symbol handler. All the standard motorola options included as well as pseudo ops. It is very modular making it quite easy to adapt to most systems. 5K
SL68-26 MANUAL & SOURCE LISTING $23.50
CT68-8 OPTIONAL CASSETTE $6.95
PT68-9 OPTIONAL PAPER TAPE $8.00

DISASSEMBLER FOR 6800: Now one that is reasonably priced and includes the source listing!
SL68-27 MANUAL & SOURCE LISTING $9.00
PT68-7 OPTIONAL PAPER TAPE $4.00

SPACE VOYAGE: A full blown Star Trek program written in 8080 and 6800 assembler language. Runs much faster than similar BASIC versions and requires about 1/5 of the memory! Each game is different as you try to save the federation using your phasers, photon torpedoes, and shields. Searching out the menacing Klingons is accomplished using the short and long range scanners as well as moving about by firing the warp engines. Beware of sudden attacks, space storms, supernovas, and other unexpected events. This game is very addicting! 4K
SL68-5 6800 SOURCE LISTING $12.00
CT68-1 OPTIONAL CASSETTE $6.95
SL80-9 8080 SOURCE LISTING $12.00
PT80-1 OPTIONAL PAPER TAPE $7.00

6800 MULTIPLE-USER SYSTEM!
Now you can have 4 simultaneous users, all running BASIC, and independently! Give your micro the power of a large mainframe. Applications include: a great educational tool, small business applications such as multi station inventory system, industrial computer power expanding the personal computer system, etc. The system is presently available only for the SWTPC 6800 computer system and will support cassettes, floppy discs, and a printer. For complete details of the multi-user system, write for our 4 page brochure.
MUB-68 $129.95

ALL SOFTWARE CONTAINS: Complete commented source listing, users manual with complete instructions, printed hex dump, sorted symbol table, and sample output.

PROGRAM-OF-THE-MONTH CLUB." $2.00 for a one year membership. No obligations!

HOW TO ORDER: All orders should include check or money order. Add 3% for postage and $1.00 for handling for orders under $10.00. Send 2SC for complete TSC software catalog.

TECHNICAL SYSTEMS CONSULTANTS, INC.
Box 2574 W. Lafayette, IN. 47906
The If Statement

The C if statement is of the format:

```
if (expression) statement
```

where `statement` may be any C statement, and `expression` may be any statement which has a value. The `expression` is true if its value is nonzero.

### C relational operators may be selected from the following list:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equal to</td>
</tr>
<tr>
<td>!=</td>
<td>not equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>logical and</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>logical not</td>
</tr>
</tbody>
</table>

Relational operators may be combined in any way, along with parentheses, to produce an expression.

The `statement` portion of the if may be any valid C statement, or combination of statements. The `statement` may be made arbitrarily complicated by enclosing a set of statements in braces, `{ }`. These braces are similar to the BEGIN and END statements of several other languages.

The if may be associated with an else statement, which may also have an arbitrarily complex statement following it. Multiple elses can be used to construct logic that branches one of several ways:

```
if (expression) statement
else if (expression) statement
else if (expression) statement
```

While Statement

The while statement sets up a loop, whose general format is:

```
while (expression) statement
```

The `expression` is evaluated and if its value is true (not zero) does the `statement` and then starts again. The `statement` may of course be arbitrarily complex as in the if statement. Note that the `statement` is executed zero times if `expression` initially evaluates to zero.

Arithmetic

Arithmetic statements use the usual operators: +, -, *, and / along with % which is the remainder (or "mod" operator in other languages). The syntax is a variable followed by an equal sign followed by an expression using the above operators. Multiple equivalences may occur on one line, an economy of notation which can be useful.

```
a = b = c = d = 25 * 2 / (c + 1);
```

One interesting feature is that when a variable is set equal to itself plus something, the statement may be abbreviated:

```
a = a + 5;
```

Note that no space follows the equal sign in this form. In general the statement may be arbitrarily complex; however, note that one cannot assume a parenthesis, etc, in front of the implied variable.

Another very interesting feature is that any time a variable is used, it may have a prefix or a suffix of ++ or --. The ++ means increment, the -- means decrement. If used as a prefix, the operation is done before the variable is used; if used as a suffix the operation is done after the variable is used.

These features help make C the supreme pleasure it is to use. They do however have a detracting feature: one must be wary when composing a statement lest it do something unintended!

Note that an if or while statement's expression may be a simple equivalence or it may contain calls to other functions! This can result in some of the cleanest, nicest code you have ever seen. It does sometimes take a minute to fathom the meaning of a statement, however, and it is easy to forget the double equals in an if test, resulting in an equivalence!

```
A = B = C = D = 25 * 2 / (C + 1);
```

As an example, let's write a program to convert a file into lower case:

```
main(){
    char c;
    while((c = getchar()) != '\0'){
        if( 'A' <= c && c <= 'Z') putchar(c + 'a' - 'A');
        else putchar(c);
    }
}
```

The program gets a character and assigns its value to the variable c. If the character is not zero (end of file, i.e. \0) it executes the if statement. This checks to see if the character is upper case, and if so, converts it to lower case and outputs it. Otherwise it is already in lower case, and is output as is. The program continues with the next iteration of the while.

The Case and Switch Statements:

Another terribly useful C statement group contains the case and switch statements. They are used to replace tests of multiple options with if statements like:

```
if(c == 'a') ...
else if(c == 'b') ...
else ...
```

A = B = C = D = 25 * 2 / (C + 1);

One interesting feature is that when a variable is set equal to itself plus something, the statement may be abbreviated:

```
a = a + 5;
```

Note that no space follows the equal sign in this form. In general the statement may be arbitrarily complex; however, note that one cannot assume a parenthesis, etc, in front of the implied variable.

Another very interesting feature is that any time a variable is used, it may have a prefix or a suffix of ++ or --. The ++ means increment, the -- means decrement. If used as a prefix, the operation is done before the variable is used; if used as a suffix the operation is done after the variable is used.

These features help make C the supreme pleasure it is to use. They do however have a detracting feature: one must be wary when composing a statement lest it do something unintended!

Note that an if or while statement's expression may be a simple equivalence or it may contain calls to other functions! This can result in some of the cleanest, nicest code you have ever seen. It does sometimes take a minute to fathom the meaning of a statement, however, and it is easy to forget the double equals in an if test, resulting in an equivalence!

```
A = B = C = D = 25 * 2 / (C + 1);
```

As an example, let's write a program to convert a file into lower case:

```
main(){
    char c;
    while((c = getchar()) != '\0'){
        if( 'A' <= c && c <= 'Z') putchar(c + 'a' - 'A');
        else putchar(c);
    }
}
```

The program gets a character and assigns its value to the variable c. If the character is not zero (end of file, i.e. \0) it executes the if statement. This checks to see if the character is upper case, and if so, converts it to lower case and outputs it. Otherwise it is already in lower case, and is output as is. The program continues with the next iteration of the while.

The Case and Switch Statements:

Another terribly useful C statement group contains the case and switch statements. They are used to replace tests of multiple options with if statements like:

```
if(c == 'a') ...
else if(c == 'b') ...
else ...
```
NEW CHOICES

MICROPROCESSOR BASICS (Elphick) Selection and application info on 6 popular micro's, including the 8080, 6800, F8, IMP, and 6100. #5763-8, paper, $9.95

THE BASIC WORKBOOK: Creative Techniques for Beginning Programmers (Schuman) “Hands-on” learning of problem-solving using a computer. #5104-2, paper, $4.25

GAME PLAYING WITH BASIC (Spencer) Over 50 easy-to-learn and challenging games and puzzles for your personal computer. #5109-3, paper, $6.95

TELEPHONE ACCESSORIES YOU CAN BUILD (Gilder) Fully-illustrated, step-by-step instruction on building useful phone accessories at a fraction of the commercial cost. #5748-2, paper, $3.95

CRITICS' CHOICES

MICROPROCESSORS: New Directions for Designers (Torrero) “...a useful book for the electronics design engineer.” BYTE MAGAZINE. #5777-6, paper, $10.95

FUNDAMENTALS & APPLICATIONS OF DIGITAL LOGIC CIRCUITS (Libes) “A great book for use as a reference by people who are learning digital electronics.” PEOPLE’S COMPUTER COMPANY. #5565-6, paper, $6.95

BASIC BASIC: An Introduction to Computer Programming in BASIC Language (Coan) “...an excellent introduction...clearly written and well-organized.” COMPUTING REVIEWS. #5872-1, paper, $7.95

ADVANCED BASIC: Applications & Problems (Coan) “This one rates well above average.” DATA PROCESSING DIGEST. #5855-1, paper, $6.95

MORE CHOICES

GAME PLAYING WITH COMPUTERS, Revised Second Edition (Spencer) #5103-4, cloth, $16.95

COMPUTERS IN ACTION: How Computers Work (Spencer) #5861-6, paper, $5.50

COMPUTERS IN SOCIETY: The Wheres, Whys, & Hows of Computer Use (Spencer) #5915-9, paper, $5.50

PROGRAMMING PROVERBS (Ledgard) #5522-6, paper, $6.50

PROGRAMMING PROVERBS FOR FORTRAN PROGRAMMERS (Ledgard) #5820-9, paper, $6.50

COBOL WITH STYLE: Programming Proverbs (Chmura & Ledgard) #5781-4, paper, $5.45

MINICOMPUTERS: Structure & Programming (Lewis & Doerr) #5642-7, cloth, $12.95

DIGITAL SIGNAL ANALYSIS (Stearns) #5828-4, cloth, $19.95

FORTRAN FUNDAMENTALS: A Short Course (Steingraber) #5860-8, paper, $4.95

DIGITAL TROUBLESHOOTING: Practical Digital Theory and Troubleshooting Tips (Gasperini) #5708-3, paper, $9.95

DIGITAL EXPERIMENTS: Workbook of IC Experiments (Gasperini) #5713-X, paper, $8.95

COMPUTER MATHEMATICS (Conrad, Conrad, & Higley) #5095-X, cloth, $13.95

HAYDEN BOOK COMPANY, INC. 50 Essex Street, Rochelle Park, New Jersey 07662
The format of a switch is:

```
switch (variable) {
  case 'a':
    statements;
    break;
  case 'b':
    statements;
    break;
  default:
    statements;
}
```

The case statements label the various actions wanted; default is done if none of the other cases are satisfied. The default statement is optional. The break statement causes execution to continue with the next statement following the switch group. The break statements are optional, and if not used, execution falls through to the next case condition, which is often useful.

break may also be used with for and while statements, causing an immediate exit from the loop. (A continue statement may also be used to cause the next iteration of the loop to be started, in a for or while statement.)

Arrays

Arrays in C may be multidimensional and are subscripted by using square brackets. Array indexes begin at zero. Typical array declarations are:

```
int x[10];
int xy[10][20];
```

This example creates an array x, with ten elements, and an array xy with 200 elements. Subscripts can be arbitrary integer expressions.

Character Arrays (C Strings)

Character arrays in C are strings of ASCII characters terminated by a zero byte. This makes string handling simple. For example, when printing a character string using the printf format %s, printf prints the characters until it finds a zero byte.

Other Statements of Interest

The for statement is a generalized while of the format:

```
for (initialization; expression; increment) statement
```

where a typical example might be:

```
for(i=0; array[i] = getchar(); i++);
```

which copies characters from the input file into array, until getchar finds a zero byte. The for is usually used when a variable must be initialized before it is used and then modified in a regular way each time the loop is reiterated.

Structures, Pointers, Pointers to Pointers, Pointers to...

Briefly, a structure in C is a grouping of data declared as follows:

```
struct {
  variable declarations
} structure-name;
```

where the initialization section is optional. A structure member is referred to by:

```
structure-name.member
```

where member is one of the variable declarations of the declaration. Structures may be made to be arrays by following the structure-name with the array size in brackets. Note that the whole structure can be passed as an argument to a function, a feature most useful when complicated linkages are required.

Pointers are another way to refer to a member of a structure. Pointers in C are declared by preceding the variable name in the declaration with one or two (or more if you're a masochist) asterisks. One * means the variable is used as a pointer to something, two *s mean the variable is used as a pointer to a pointer (typically a pointer to an array of pointers). The address of a variable can be had in an expression by using the unary operator & in front of the variable. Pointers are usually used to "walk" along arrays efficiently. In fact, an array name represents the address of the zeroth element of the array, so it cannot be used on the left side of an expression, since you can't change the address of something by assigning to it.

The Crux of the Matter

It is difficult to learn much about a language by reading about it; I have only briefly skimmed the most important features of C in an effort to enable those who are not familiar with the language and are looking for a better high level language to gain some insight into it. It seems to take a bit of working with the language in order to be able to experience and understand the full power of it and to be able to effectively utilize that power.
CURSOR CONTROL. Forespace, backspace, 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 keypad.

REPEAT KEY. 15 cps repeat action.

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

SCREEN. 12 inch rectangular CRT – P4 phosphor.

NEW LOW PRICE
Assembled $975.00

Specials of the Month
North Star Micro Disk
with power supply and cabinet . . . . Kit – $699
Assembled – $799

IMSAI 8080 with TDL ZPU . . . . . . Kit – $825
Assembled – $975

Digital Systems FDS Disk Drive with
CP/M Software (assembled only) . Single – $1,750
Dual – $2,350

Mountain Hardware PROROM . . . . Kit – $145
Assembled – $210

Vector Graphic 8K RAM . . . . . . . Kit – $235
Assembled – $285

NOW WE’RE TOLL FREE
800-227-4440
(in California, Hawaii, and Alaska, call collect:
415/566-7472)

XIMEDIA OFFERS A FULL RANGE OF PRODUCTS FOR THE PERSONAL COMPUTER ENTHUSIAST AND THE SMALL SYSTEM DESIGNER. LET US QUOTE ON ALL YOUR HARDWARE AND SOFTWARE NEEDS.

OUR RETAIL STORE – THE COMPUTERIST™ – IS NOW OPEN IN SAN FRANCISCO. CALL US FOR DIRECTIONS.
All of this is of no use to microprocessor users unless it can be implemented on a microprocessor. C was designed with the PDP 11/45 hardware directly in mind, and it takes advantage of the indexed addressing mode and the increment and decrement instructions. It turns out that most C statements result in about three machine code instructions!

The power of this language becomes more apparent with several notes. The Unix operating system was originally written in assembly language. During the summer of 1973, it was rewritten in C. A number of features were added including multiprogramming and the ability to share reentrant code among several user programs. The size of the new system was about one third greater than the assembly language version. The programmers "considered this increase in size quite acceptable." Several benchmark programs written to compare the Digital Equipment Corporation's IAS operating system BASIC and FORTRAN with the Unix C language gave results which placed C roughly ten times faster than FORTRAN and more than 15 times faster than the BASIC.

Why not PL/I?

I don't wish to put down PL/I, since it certainly is a useful language and has its place; however, I would like to make a few comments in closing.

Special characters are fine and fun, but create all kinds of problems when trying to implement them; for example, few terminals have many of the special symbols resulting in all kinds of weird escape sequences to get them. Unfortunately many keyboards in use are not even full ASCII. Hardware to do 5 by 7 matrix printing tends never to get built by the builder and most of the special symbols have no ASCII value. (On a personal note: I have met with the most amazing resistance to the idea of expanding ASCII to eight bits. People will nearly resort to physical violence against the idea. But why not? Most machines use eight bits anyway, and parity [how often is it really used?] can be implemented vertically instead of horizontally. Greek characters seem to be the rage [though how often have you used them?] so why not implement 256 characters? I can see the mail now...) Mr. Skye notes that he will update PL/SKYE as one might a dictionary. Have you looked at the size of the Random House Dictionary lately? It is all very well to intend that it won't be too big, but doing it is very much the question. What is too big? That depends on your system. As memory prices go down, too will the "big" threshold continue to go up and up. At this time, too big for a disk operating system seems to be about 16 to 32 K with perhaps four or five overlays (and that is very big), while on a tape system, too big is 16 K and that is very small if the program is to be in core, too.

What to do, what to do… Well, you probably shouldn't be running a high level language with a tape operating system unless it is a block oriented system (like the Digital Group Phi Deck system when implemented like a disk). So, perhaps we could set a limit at between 16 to 24 K for the compiler itself, leaving room for several 512 byte blocks for IO in a 20 to 32 K system.

As for writing the compiler in its own language, that is the only way to go.

Relocatable code is a must; and with index registers it becomes very simple. Here is where the processor becomes important. We really need a processor with indexed and relative addressing modes. That looks like the Z-80 processor (scream, rant, rave...). That does not however mean that it could not be implemented on a 6800, 6502, 8080, Micro-NOVA or whatever. Actually it turns out to be relatively easy to rewrite the code generation portion of a compiler to run on another machine, though at some expense in speed. (Can you imagine C on a Turing machine?)

IO? Yes, IO should be completely device independent. But that should be the concern of the operating system. Simply write your operating system in C, making it device independent. Actually, it is best to design the IO routines to be function (subroutine) calls in the compiler code, so the user can change it to suit his or her needs.

This brings up an interesting point. How does one distribute the code? It can't be in C source code; one can't run that. I hate binary object code since it is so machine and device dependent. Well, everyone should have an assembler for their machine, so why not have the compiler (or at least a version) generate machine source code. The IO sections of this can be modified if necessary using a text editor, assembled with the users' assembler and then it's ready to run. It turns out that it's not quite that simple, if you think about it or try it, but it's close. From then on, the compiler can compile itself when changes are made.

This has been just an introductory commentary on the virtues of what I consider an excellent language which should be adaptable to the range and scope of personal microprocessor based systems. As for when and if versions for the microprocessor user will appear, all I can do is close with the statement "wait and C."
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 these features:

- 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.
- Three full days testing at 70° C (185° F).

**SPECIFICATIONS**

Capacity: 16384 bytes
Addressing: 16K boundaries
Bus structure: S100 - Plug compatible with IMSAI 8080, POLY 88, ALTAIR 8800, BYTE-8. SOL
Address and Data Buffering: <200 uA, special high impedance buffers - less than one low power Schottky load
Access time: 350 nSec
Memory chips: MM 5271 (National Semiconductor and others) 4K dynamic

**Dynabyte Inc.**
4020 Fabian Way, Palo Alto Ca. 94303

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

Dynabyte® brings to the S100 Bus a state of the art, industrial quality memory system. 16K on a single board for $485.00, Guaranteed for 1 year.
In order to produce a musical output, we must at least create a pitch output under control. This is but a starting point, since more complicated waveform and envelope generation is also useful in music.

The block diagram of a basic note and octave synthesis system is shown in figure 1. The top octave generator produces a square wave whose frequency (pitch) is determined by data sent out on the computer's data bus. Since the output of the top octave generator is a square wave, it can easily be divided by digital circuitry. Each time we divide the frequency by two, we end up with a note whose pitch is one octave lower than the input frequency. By using an ordinary TTL data selector as an octave selector, we can generate a musical scale covering many octaves, and we can also produce more than one pitch at a time, although these extra outputs will always be octave related to one another. The octave selector can also be controlled by data sent out on the computer's data bus, giving us more flexibility.

The octave selector can be easily implemented using an n-stage divider and several NAND gates. However, there are several methods of generating the top octave. We need 12 notes to produce a 1 octave chromatic scale. These notes must be accurate in frequency and drift free in order to produce a true chromatic or “equally tempered” scale useful in music.

One way of synthesizing the top octave is to use a digital to analog (D/A) converter controlling an oscillator. An 8 bit converter limits both resolution and range so that we cannot produce an acceptably accurate chromatic scale. If we use a converter of 10 bits or more, resolution and range are suitable, but such units are expensive and require stable voltage controlled oscillators for this type of application. This method of pitch generation is shown in figure 1. The one nice feature of the digital to analog converter method is that we have a continuously variable output frequency. This permits nifty frequency sweeping effects (known as “portamento” or “glide” effects to the musician).

To save money we can construct a rather crude digital to analog converter which, in conjunction with the voltage controlled
Figure 2. An Altair bus interface and frequency selection logic for the tunable digital-to-analog conversion method for the Altair 8800 bus.
oscillator, will produce the 12 notes required for the full top octave. This method is shown in detail in figure 2. By using surplus 10 turn trimpots and the voltage controlled oscillator, we can construct an inexpensive top octave generator. However, this method has its disadvantages: tuning the trimpots is a critical operation, and once the pots are tuned, they can easily detune themselves because of vibration or temperature variations. My present synthesizer uses this method and needs to be retuned about every two months or so.

A good alternate method of generating the top octave is to use an integrated circuit top octave generator such as the MOSTEK MK50240P. This chip can be had for under $10 and is second-sourced by General Instrument Corporation as the AY-3-0215.

There are several advantages to be had by using this chip. The chip nominally requires a 2.000240 MHz reference frequency which is approximated by the clock's circuitry of most Altair (S-100) bus systems. (The frequency is not exactly 2.000240 MHz, but will be close enough for this application.) This chip eliminates both the voltage controlled oscillator and digital to analog converter, and therefore puts an end to instability and tuning complications. The MK50240P generates the top 13 notes of the well-tempered music scale with an accuracy better than can be determined by the best musician.

### Hardware Considerations for Two Working Circuits

The circuit used for the pitch generator, using the tunable digital to analog converter with an Altair bus is shown in figure 2. Bus timing and address decoding are performed by IC1, IC2, and IC3. IC4 and IC5 latch and hold data sent to the board on the data out bus. IC8 is a 4 to 16 decoder with active low outputs. These outputs select which 10 turn trimpot is selected as the bottom leg of the voltage divider whose top leg is resistor R. These trimpots should all have a value about 5 R. The voltage produced by this divider is connected to the input of the voltage controlled oscillator. The output of the voltage controlled oscillator is divided by IC7. The outputs of counter IC7 are gated onto the output bus by IC6, the quad open collector NAND gates. IC8 and the voltage controlled oscillator comprise the top octave generator and IC6 and IC7 comprise the octave selector.

The circuit using the MK50240P for top octave generation is shown in figure 3. The board address, bus timing and latch circuitry are identical to the circuit of figure 2. The octave selector is also identical to the one in figure 2. The MK50240P, IC6, is a 12 V device and requires input signal conditioning and output buffering. The 2N2222 transistor and associated resistors bring the TTL level clock signal from the bus up to the 12 V level required by the MK50240P. The outputs of this chip are buffered by IC7 and IC8 before going to IC9 which is the data selector and multiplexer. The MK50240P generates all the notes in the top octave simultaneously and IC9 selects any one of these outputs depending on what data is present at the outputs of IC4. The output of IC9 is then connected to the two chips (IC10 and IC11) comprising the octave selector. An additional voltage regulator, a 7812, has been provided to supply the 12 V needed for the MK50240P. Note that for additional music channels, additional copies of the note selector IC9 and octave selector can be driven off the buffered outputs of the MK50240P.

A word of caution: the audio output of the circuits in figures 2 and 3 swings about 2 V peak to peak and should be attenuated with a potentiometer before you plug it into your stereo system or amplifier.

### Software Considerations

From a software point of view, the circuits of figures 2 and 3 are identical. Both circuits have an 10 device address of 300 octal. Outputting the proper data to 8080 port 300 octal will cause the synthesizer to audibly produce the note and octave(s) represented by that data.

The synthesizer can be considered as having two input nybbles, each nybble containing four bits. The least significant nybble determines what note is to be selected and the most significant nybble determines what one of four possible octave(s) is to be selected. One byte contains all the information necessary to set up any note and octave(s).

Bear in mind that the synthesizer will continue to produce the note and octave(s)
Figure 3: The complete circuit for an equivalent of figure 2, which uses the top octave generator chip with an Altair bus interface. This method uses the top octave generator and a note selector to drive the octave selection logic, while the digital to analog converter method uses a voltage controlled oscillator with a diode resistance selection of pitch.
you have selected until you send it new data.
To clear the synthesizer (no audible output) all you need do is output a 0 on data lines D4 thru D7, the most significant nybble.

Table 1 shows what the octal representations are for each note in the top octave. Table 2 shows the octal representations for each octave. To pack these two codes into one byte, they can either be added or ORed.

Table 3 shows a series of codes that, when moved to port 300 octal in sequence, will produce a 12 note musical scale in the synthesizer's highest octave. However, if you wish to hear this scale, you must insert a software time delay in between each note. Otherwise all you will hear is a very short "click" because of the processor's high speed of program execution. If you wish to hear this scale in four octaves simultaneously, all you need to do is keep all four bits in the octave enabling nybble in the high state.

Armed with this information and some simple software routines, you and your trusty computer are now capable of synthesizing all of J S Bach's "B Minor Mass," to say nothing of the many new types of musical expression you now have at your fingertips. I myself have synthesized Haydn's "Minuet in G," parts of Bach's "Toccata and Fugue in D Minor," Henry Mancini's "Pink Panther Theme," and "Hot Rod Lincoln!"

Table 1: Octal and hexadecimal notation for octave enabling bits sent to the interface in the high order nybble of the 8 bit word. Note that one or more of the octaves may be enabled simultaneously simply by adding the codes together (or using a logical OR).

<table>
<thead>
<tr>
<th>Octave</th>
<th>Octal Code</th>
<th>Hexadecimal Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 (highest)</td>
<td>020</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>040</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>40</td>
</tr>
<tr>
<td>1 (lowest)</td>
<td>200</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 2: Octal and hexadecimal notation for octave enabling bits sent to the interface in the high order nybble of the 8 bit word. Note that one or more of the octaves may be enabled simultaneously simply by adding the codes together (or using a logical OR).

<table>
<thead>
<tr>
<th>Octave</th>
<th>Octal Code</th>
<th>Hexadecimal Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 (highest)</td>
<td>020</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>040</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>40</td>
</tr>
<tr>
<td>1 (lowest)</td>
<td>200</td>
<td>80</td>
</tr>
</tbody>
</table>

The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, 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.

The Best of BYTE, Volume 1

The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, 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.

The Best of BYTE, Volume 1

The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, 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.

The Best of BYTE, Volume 1

The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, 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.

The Best of BYTE, Volume 1

The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, 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.

The Best of BYTE, Volume 1

The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, 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.

The Best of BYTE, Volume 1

The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, 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.

The Best of BYTE, Volume 1

The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, 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.

The Best of BYTE, Volume 1

The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, 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.
MORE BYTES FOR YOUR BUCK!

8K RAM...assembled, burnt-in, tested and warranted...only $197.50!

These are not kits, but completely assembled and 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 run. The 8K RAM has the same features and speed as our 4K RAM (500 nsec, no wait states) and if that isn't fast enough there's our 8K (Z) fast RAM that gives you a 250 nsec access time for only $217.50!

Because of the low power memory chips used, power requirements are lower than many other RAM boards. And the 8K RAM uses less power than two 4K boards. All RAMs are manufactured to military specification MIL STD-883-C, assuring greatest control over reliability.

Address selection is easily accomplished by our Visaddress™, an easy to read switch on the board top. The 8K board is designed to be selected as one of eight possible 8K RAM boards present on the S-100 bus.

To achieve address selection, the top address lines are decoded using the Visaddress switch. The switch will then show the selected starting address of the RAM card. (i.e. $0000-1FFF, 2=$2000-3FFF, etc. on the 8K board).

Both boards have fully buffered address and data lines, and extensive built-in noise immunity circuitry. And are plug-in compatible with the S-100 bus (Altair 8800, IMSA1 8080, etc.)

Quality, assembled boards at less than kit prices. But what else should you expect from a company whose prime products are electronic test instrumentation and microprocessing components?

Also available: 4K RAM; $107.00, Alpha-VDM; $107.00, Alpha-VDM-II; $145.00, Graphics-VDM; $137.00.

Order direct by check, BankAmericard or Master Charge (Add $1.50 shipping, credit customers give us all the card numbers, please and Ohio residents add 4½% sales tax) or contact us for more information.


Circle 407 on inquiry card.

Kent-Moore
INSTRUMENT COMPANY
Structured Program Design

David A Higgins
Langston, Kitch and Associates Inc
715 E 8th St
Topeka KS 66607

In the world of electronics, no experimenter in his right mind would build a circuit by throwing a few parts together with some wire and some hope, then attaching a line cord and plugging it in to see if it works. Not only are you likely to destroy some very expensive parts, but it is also a good way to get fried, or at least get a new hairdo.

Yet, after all the trouble that a serious microcomputer hobbyist will take to insure that his circuit is put together correctly before he ever turns it on, he will invariably try to program his new computer by using a technique analogous to the one above. That is why his programs almost never run right the first time, if indeed they ever manage to run right at all. It is also why many microcomputer buffs stay up until odd hours of the night drinking coffee by the gallon in an effort to find that one little bug.

But there is hope. I'm sure that nearly everyone involved with computers has heard something about structured programming in one form or another. It is not really a new technique, having been preached about for many years. However, the tools and methodologies available to design programs have changed radically over the years.

In the beginning there were flowcharts, which looked like five-dimensional octopi or the corporate structure of a conglomerate. Despite the absence of a consistent approach that would enable everyone to design a program using flowcharts, those programmers who did bother to work out their problem with a flowchart first usually seemed to have more luck in getting programs to run sooner and better than programmers who did not.

Structuring Tools

The development of mathematics would surely have been stymied if Roman numerals had been retained as our number system. In much the same manner, the science of structured program design would have been mired down if only flowcharts had been
available for developing programs. It is not that calculus is impossible with Roman numerals, it's just that it's extremely difficult. Thus, over the years, a number of design and documentation tools were developed to better enable a programmer to understand the problem before going out to do battle with the program.

TOP-DOWN or GOTO-less programming, developed by Dijkstra and others, was probably the first major attempt to solve the design versus coding problem. Dijkstra simply observed that the more GOTO that were in a program, the less likely it was to run correctly. Dijkstra called such programs "spaghetti bowl" programs, because if you drew a line from each GOTO in the program to its destination, you ended up with a mess that looked like a bowl of spaghetti. He showed how any program could be written with just a few simple flow structures without any GOTOs. His techniques produced simple, readable code that was easy to test and maintain. So, the big push among design aficianados was to eliminate the GOTOs in their programming. Although TOP-DOWN programming was a big advancement over flowcharting, it was just that: programming. It was a technique for coding a program, not necessarily designing it.

Another technique, IBM's HIPO (and later HIPO-DB) entered the design field almost by chance, being primarily a documentation tool that was also being used for program design. The major drawback to HIPO techniques, besides the fact that they did not work well for designing a program, was their tendency to produce 50 pages of documentation for a 3 page program.

Warnier-Orr Diagrams — A New Approach

Within the last four years a new technique for program design has evolved from the work of Jean-Dominque Warnier (pronounced warn'-yay) in France, and Kenneth T Orr of Langston, Kitch and Associates in Topeka KS. The technique has foundations in set theory and Boolean algebra, and holds much promise for program design applications. Warnier-Orr diagrams, as we have called them here in the United States, allow programmers to design faster than ever before, to code programs with little or no effort, and produce programs that usually run correctly the first time. The approach is not limited to small programs. Nothing will make a believer out of someone quicker than a 20 page COBOL program which runs correctly the first time. The Warnier-Orr technique stresses design over coding and contends that once a problem is designed, it does not matter what programming language you code it in! At Langston, Kitch and Associates, people have used the technique to program in COBOL, PL/I, ALGOL, FORTRAN, BASIC, RPGII and assembler languages. It works equally well for all of them.

Warnier-Orr Diagram

The simplest way to learn about Warnier-Orr diagrams is to see examples of them. Warnier-Orr diagrams are very easy to learn
If the player rolls a 4, we first find the instructions to follow for a roll of 4 and check to see if the player has a BUG head. If he does, we then check to see whether or not the player already has two antennae...

and use; however, be forewarned that this is a technique that is sometimes deceptively simple, but not as trivial as it often seems.

Let’s consider the relatively simple game of BUG. In this game the computer rolls a die, once for itself and once for its opponent. Each number of the die corresponds to a part of the BUG’s anatomy: 1 = BODY, 2 = NECK, 3 = HEAD, 4 = ANTENNAE, 5 = TAIL, and 6 = LEGS. The object of the game is to finish your bug before the computer finishes its bug. Other rules: you must have a body before you can have legs, a neck or a tail; you must have a neck before you can have a head, and you must have a head before you can have antennae. One body, one neck, one head, one tail, six legs and two antennae are needed to complete a bug. Figure 1 is a Warnier-Orr diagram showing the basic structure of the BUG program.

The Warnier-Orr diagram is read left to right, top to bottom, just like conventional English text. The brackets enclose logically related operations, the largest of which is the program itself. The BUG program is composed of three logical sections:

- The BEGIN PROGRAM section, where the player’s name is requested and there is an explanation of the game rules. Note that the + symbol between the modules YES and NO denotes the exclusive OR function, meaning that one or the other but not both of the modules will be performed. Observe also that this is reflected in the number of times that each module may be performed: 0 if the condition is false and 1 if the condition is true.
- The process section, GAMES, where the playing of the game actually takes place. The (1,g) denotes that the section is to be performed at least once, and possibly many (g) times.
- The END PROGRAM section, which in this case is empty, but which usually contains things such as the closing of files, the goodbye message, etc.

The rest of the brackets decompose in a similar fashion. The GAMES procedure breaks down into the beginning of the game, (BEGIN GAME), the turns that each player takes (TURNS), and the end of the game (END GAME).

Notice that logically there are things that only happen at the beginning of the program and things that only happen during the playing of the game itself. The Warnier-Orr diagrams allow you to see very easily just where and when a particular event must take

![Figure 3: Warnier-Orr diagram for the ending of a turn or a game.](image)
Whether managing accounts at the local general store or aiding in the development of the latest Space War, the MiniTerm Dual MiniFloppy* System 80/2 can do the job best. Z-80 powered, and S-100 bus structured, the System 80/2 mainframe with regulated supplies for two mini floppies* is also available without cards and drives.

Dealer Inquiries Invited.

MiniTerm Associates, inc.
Box 268, Bedford, Mass. 01730 (617) 648-1200

*Mini Floppy is a registered trademark of Shugart Associates.
Listing 1, continued:

340 LET LEGS(1), LEGS(2) = 0
350 RETURN
360 REM TURNS SUBROUTINE
370 REM PLAYERS TURN
380 REM LET PLAYER START TURN
390 PRINT ‘HIT RETURN TO ROLL DIE’
400 INPUT A
410 LET PLAY = 1
420 GOSUB 520
430 LET ANTE(PLAY) = ANTE(PLAY) + 1
440 LET BODY(PLAY) = 1
450 GOSUB 820
460 REM END TURN
470 GOSUB 1060
480 RETURN
490 REM TURN SUBROUTINE
500 REM PLAY=1; PLAYERS TURN—PLAY=2; COMPUTERS TURN
510 REM ROLL DIE
520 LET ROLL = FIX((IRND(01) * 6.0)) + 1
530 PRINT: “ROLL IS A “, ROLL
540 IF ROLL = 1 THEN IF BODY(PLAY) = 1 THEN GOSUB 690 ELSE ; ELSE ;
550 IF ROLL = 1 THEN GOSUB 850
560 IF ROLL = 2 THEN IF BODY(PLAY) = 1 THEN IF NECK(PLAY) = 1 THEN GOSUB 760
570 IF ROLL=2 THEN GOSUB 650
580 IF ROLL = 3 THEN IF BODY(PLAY) = 1 THEN IF NECK(PLAY) = 1 THEN IF HEAD(PLAY) = 1 THEN GOSUB 820
590 IF ROLL=3 THEN GOSUB 650
600 IF ROLL = 4 THEN IF HEAD(PLAY) = 1 THEN IF ANTE(PLAY) = 2 THEN GOSUB 820
610 IF ROLL=4 THEN GOSUB 650
620 IF ROLL = 5 THEN IF BODY(PLAY) = 1 THEN IF TAIL(PLAY) = 1 THEN GOSUB 740
630 IF ROLL = 5 THEN GOSUB 650
640 IF ROLL = 6 THEN IF BODY(PLAY) = 1 THEN IF LEGS(PLAY) # 6 THEN GOSUB 1000
650 LET A = 3
660 RETURN
670 REM BODY SUBROUTINE
700 IF PLAY = 1 THEN PRINT : NAMES, “ ’S BUG HAS A BODY”
710 IF PLAY = 2 THEN PRINT : “COMPUTER’S BUG HAS A BODY”
720 LET CNT(PLAY) = 1
730 LET BODY(PLAY) = 1
740 RETURN
750 REM NECK SUBROUTINE
760 IF PLAY = 1 THEN PRINT : NAMES, “ ’S BUG HAS A NECK”
770 IF PLAY = 2 THEN PRINT : “COMPUTER’S BUG HAS A NECK”
780 LET CNT(PLAY) = CNT(PLAY) + 1
790 LET NECK(PLAY) = 1
800 RETURN
810 REM HEAD SUBROUTINE
820 IF PLAY = 1 THEN PRINT : NAMES, “ ’S BUG HAS A HEAD”
830 IF PLAY = 2 THEN PRINT : “COMPUTER’S BUG HAS A HEAD”
840 LET CNT(PLAY) = CNT(PLAY) + 1
850 LET HEAD(PLAY) = 1
860 RETURN
870 REM ANTENNAE SUBROUTINE
880 LET ANTE(PLAY) = ANTE(PLAY) + 1
900 IF PLAY = 2 THEN PRINT : “COMPUTER’S BUG HAS “, ANTE(2), “ ANTENNAE.”
910 LET CNT(PLAY) = CNT(PLAY) + 1
920 RETURN
930 REM TAIL SUBROUTINE
940 IF PLAY = 1 THEN PRINT : NAMES, “ ’S BUG HAS A TAIL”
950 IF PLAY = 2 THEN PRINT : “COMPUTER’S BUG HAS A TAIL”
960 LET CNT(PLAY) = CNT(PLAY) + 1
970 LET TAIL(PLAY) = 1
980 RETURN
990 REM LEGS SUBROUTINE
1000 LET LEGS(PLAY) = LEGS(PLAY) + 1
1030 LET CNT (PLAY) = CNT(PLAY) + 1
1040 RETURN
1050 REM END TURN SUBROUTINE
1060 IF CNT (1) = 12 THEN 1090
1070 IF CNT (2) = 12 THEN 1110
1080 GOTO 1130
1090 PRINT : NAMES, “ ’S BUG IS FINISHED’ YOU WIN”
1100 GOTO 1120

In figure 2, we have represented the logic for each of the players’ turns during the game. At the beginning of each turn, the die is rolled to determine the part of the BUG’s body that the player may receive. Whatever the roll, we then have a logical path to follow. Again, please note that the presence of the if between each of the possible rolls denotes mutual exclusion, ie: only one of the paths may be selected. This particular structure is known as a case statement.

If the player rolls a 4, we first find the instructions to follow for a roll of 4 and check to see if the player has a BUG head. If he does, we then check to see whether or not the player already has two antennae. If he does, then we do nothing. If he does not have two antennae yet, we give him one antenna. If he does not have a BUG head, then again we do nothing. In a similar fashion, all of the possible rolls and their associated procedures are explained. Now let’s move on to the Warnier-Orr diagram for the end of the turn, which is shown in figure 3.

If either player has won the game at the end of a turn, the computer declares the winner and ends the game. If neither player has won, the computer does nothing and cycles through for another turn.

Structured Programming

Having fully understood the problem, coding the BUG program is a simple and straightforward process. For this particular example I coded the program shown in listing 1 in a version of BASIC.

As you can see, each bracket of the original Warnier-Orr diagram roughly corresponds to a subroutine in the finished code; the process GAMES, for instance, becomes the subroutine at line number 180 which is called repeatedly by the branch at line 80 until EPGM equals 1, indicating that no more games are to be played; the process BEGIN PROGRAM is handled by the subroutine at line 110, and so forth. The resultant code is:

- easy to read and understand
- easy to change and maintain
- already documented
- logically correct.

It is also a program that will run correctly the first time, barring unforeseen syntax errors for those of us who can’t type or spell. All of this is possible because the
program was thoroughly designed before it was even partially coded.

Conclusion

Warnier-Orr diagrams are a giant leap in the right direction for structured programming. They represent an attitude which, for the first time since people have been playing with computers, can lead to consistently reliable software that is very easy to maintain. Currently, most data processing departments spend over 80% of their time and effort repairing old code that has suddenly gone bad. Warnier-Orr diagrams also provide the means to produce software of a quality that has never before been possible.

If you think that you are interested in using Warnier-Orr diagrams to help you solve some of your software headaches, by all means try them. But as I mentioned above, this technique looks deceptively simple, and you may not have much success. Understanding a diagram such as the one presented in this text is one thing; creating one from scratch is another.

If you do get bogged down, please feel free to write us for more information. If you try them, like them, and think you've done something exciting with them, again feel free to write us and tell us what you've done.

Listing 1, continued:

1110 PRINT: "COMPUTER'S BUG IS FINISHED, I WIN"
1120 LET EGAM = 1
1130 RETURN
1140 REM END GAME SUBROUTINE
1150 PRINT: "DOES ANYONE ELSE WANT TO PLAY?"
1160 INPUT ANS
1165 LET TEST = SCOMP (ANS$, 'YES')
1170 IF TEST = 0 THEN LET EPGM = 1
1180 RETURN
1190 REM EXPLANATION OF RULES SUBROUTINE
1200 PRINT "THE GAME OF BUG IS PLAYED AS FOLLOWS;"
1210 PRINT "A DIE IS ROLLED BY THE COMPUTER, AND EACH NUMBER"
1220 PRINT "ON THE DIE CORRESPONDS TO A PART OF THE BUG'S"'
1230 PRINT "BODY: 1=BODY, 2=NECK, 3=HEAD, 4=ANTENNAE, 5=TAIL"
1240 PRINT "6=LEGS. YOU NEED 1 BODY, 1 NECK, 1 HEAD, 2 ANTENNAE"
1250 PRINT "1 TAIL, AND 6 LEGS TO COMPLETE A BUG."
1260 PRINT "THE OBJECT OF THE GAME IS TO BUILD YOUR BUG"
1270 PRINT "BEFORE"
1280 PRINT "COMPUTER BUILDS HIS."
1290 PRINT "-HIT RETURN WHEN YOU ARE READY TO PLAY."
1290 INPUT A
1300 RETURN

BIBLIOGRAPHY


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
☐ Check Enclosed
☐ Bill Master Charge

No. ____________ Exp. Date ____________

Name __________________________
Address _________________________
City __________________ State ______ Zip ________
dise purchased. We temporarily FIX 0 to print the quantity as a nondecimal integer and make reading the tape easier. We store the quantity in data memory 01, print, restore the SR-52 to FIX 2 and HLT.

5. Key B is the list price of the product. It is stored in data memory 02 and printed. This time we did not use a HLT instruction, and we will see why in the next step.

6. Without a HLT in step 5, the SR-52 program counter automatically moves down to the next step. It encounters the command to multiply AxB, or more exactly that data we stored in memory 01 times the data we stored in memory 02. We print this result and then sum the result into data memory 19. Since we cleared the SR-52 memories in our initial housekeeping instruction, we know that the initial content of data memory 19 is zero. After all this we HLT.

At this point in our flowchart we are at a manual or human decision point as shown by the diamond. If we have more merchandise to price, we can go back and put in the next quantity A and price B. We would do this until we were done. Then we would go to the A' button. In our flowchart this is labeled as step 9. Putting A' near the end of the program may appear to be out of sequence, but the SR-52 doesn't care. It will automatically go looking for A' program instructions whenever that button is pushed. My programming style says that I like to finish defining the basic user-defined key functions before getting involved in the detailed program steps. Since we haven't defined key E yet, that will be the next step. Incidentally, claiming a programming style as a reason for doing something is an easy way of avoiding a lot of "no win" programming arguments. If it's a matter of taste, who can complain?

Since the flowchart represents the actual sequence the user follows, we see that the A' button is pushed to complete and print the totaling and discount of the list price merchandise. The details of this block will be discussed below. At this point we are only concerned with what we do next. The diamond again indicates a manual decision. Do we have any net price items on this invoice? If so, we will go to A again. This time, button A will be associated with quantities of the net price items. If there are no net price items we will push button B'. This will complete the invoice computation. At this point in the flowchart we define button E.

7. Key E is the net price of the merchandise. We will store this price in data memory 05 and print the price. Again as in step 5, since we don't have a HLT instruction, the SR-52 will automatically proceed to step 8.

8. This step will multiply AxE, or the quantity in data memory 01 times the net price in data memory 05. The result is printed and summed into data memory 19. Incidentally, the previous value of data memory 19 is stored with the net price computation that resulted when button A' was pushed. The diamond after this block is another manual decision. If there are more net price items go back to A; otherwise go on to B'.

9. This block, at first glance, appears to be complex. But it really isn't. It provides the following instructions:

```
RCL 19 Recall the list price amount that was summed into data memory 19 and print (PRT) the amount.
1-C Multiply the list price total by 1-C, the discount that was inputted, and stored originally with key C. Print the dollar amount of the discount.
-RCL 19 Subtract the list price amount from the discount; "+-"; change the sign since the result is a negative number and print the result. This is the net price of the merchandise.
Pap Advance the paper one line for format clarity.
STO 19 Store total net price in data memory 19 and HLT.
```

Not so bad, was it?

10. We now define the program instructions for the B' button:

```
Pap One line of paper advance.
RCL 19 Recall and print the total order net price that has been summed in previous
```
steps into data memory 19.

if we mark each program address line with

The RUN button is pressed and, if the list can be compared to the worksheet we will be a great help in finding the bug.

Analyze the key function name. If the program doesn't work, you may have

The revised flowchart is now complete and at the same time we have defined virtually every program instruction. The total development of this program has been a very straightforward translation of what we want to accomplish into precise terms understandable to our computer.

The last programming step is to set the SR-52 to receive program instructions. A key marked LRN (learn) is provided for this. Several other programming keys are also provided which dramatically simplify putting a new program into the machine.

At this point we will only need the LRN button. With the SR-52 in the LRN mode, we begin with the first instruction which is CMs (clear memories). We push that button on the keyboard. The program counter moves to 001 and we push the CLR button. The SR-52 stores the clear instruction and the program counter moves to 002. We continue pushing the buttons that we defined in our program until we reach the last instruction which is RST. Our program is now stored in the machine. We take the SR-52 out of the learn mode by pressing the LRN button again. The program counter is manually reset to 000. Listing 1 shows the entire program list.

We are now ready for the critical test. The RUN button is pressed and, if the display is not flashing, there is a good chance that our program might work. We test it by putting in a sample calculation. If it does work, our tape printout would look like listing 2. If there are problems, the printer will be a great help in finding the bug.

For the first time that can be done is to ask for a list of the program steps. This output list can be compared to the work sheet we used to enter the program. Analyzing the program list shown in listing 1 is made easier if we mark each program address line with the key function name. [In listing 1 this is done with typeset notations, but in SR-52 practice this would be done with handwritten notations ...CH/ In addition to doing this, groups of instructions should be bracketed so that they can be related to the flowchart in figure 2. The first three digits in the program list are the program counter address. The next two digits are the op code or the key codes for the SR-52 keyboard. If the program doesn't work, you may have omitted an instruction. The SR-52 provides an easy way to get to the point of omission so that we can insert the proper instruction. In a similar way, an incorrect instruction can be omitted.

The most painful debugging problem is where your program logic just isn't right. The printer will allow you to trace and print the steps the program actually is following as it attempts to solve the problem. (In our case, calculate the invoice.) If you are not too far off, the SR-52 has the facilities
that allow you to easily implement your program fix.

Listing 3 shows the printer output in the trace mode. Here the left digits are the printed equivalent of the SR-52 display. The right side data is the function the SR-52 is performing on the display data.

The data shown in listing 3 is the same data used for the printout shown in listing 2. Marking and grouping the trace data in the way shown will simplify comparing the trace output with the design flowchart of figure 2. Each step in the trace should match what you designed in the flowchart. If it doesn't, then the machine is being asked to do something incorrectly. If your flowchart logic is faulty, review the SR-52 instruction book before making corrections.

If your flowchart design has an improper premise, then the calculator will indeed proceed with perfect logic to an improper result.

Somewhere between the two "dump" methods, list and trace, lies the identification to the program problem. Listing shows program loading mistakes and tracing will reveal program logic errors.

I could go out on a limb with the truth and say that my invoice program was up and running on the first pass. But seriously, with the debugging facilities already discussed, it wasn't too difficult to find and correct the problem areas.

After the program has been debugged, you should make a new program list. Any flowchart changes that were necessary should also be documented. If you don't do this documentation, you will find that your memory will fail you miserably if you want to modify the program sometime in the future. The up-to-date list will also let you reproduce the program in case the magnetic program card ever becomes unusable.

Table 1: Functions of the user defined keys in the invoice program.

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Assignment when Button is Pushed</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Data entered on key is the quantity of the merchandise purchased.</td>
</tr>
<tr>
<td>B</td>
<td>Data entered is the suggested list price of the merchandise. (Used with A.)</td>
</tr>
<tr>
<td>C</td>
<td>Data entered is the customers' discount.</td>
</tr>
<tr>
<td>D</td>
<td>Data entered is the shipping charge for the order.</td>
</tr>
<tr>
<td>E</td>
<td>Data entered is the net price (no discount) of the second category of merchandise. (Button A is also used with E.)</td>
</tr>
<tr>
<td>A'</td>
<td>This is the second function use of A. When pressed it tells the SR-52 to total the list price items, compute and subtract the discount and leave the net amount.</td>
</tr>
<tr>
<td>B'</td>
<td>This is the second function use of B. When pressed it tells the SR-52 to add the total of the net price items, if any, to the results obtained when A' was pushed. It then adds the shipping cost and results in the total for the order.</td>
</tr>
</tbody>
</table>

Listing 3: Another sample listing; this one is in trace mode. Trace mode outputs each program instruction and the data associated with it so you can check each individual program step. This is the SR-52's equivalent of a symbolic high level language in a larger machine.
You should also write the user instructions at this time. Table 2 shows the user instructions for the invoice program. It is also helpful to the user if you attach a copy of a sample printout. The blank spaces on the magnetic program card should also be filled.

This invoice program, while being relatively simple, is certainly not a trivial program. It does involve many of the basic programming philosophies that would be applicable to writing a program for any system. With the prices of hand held calculators continuing to drop as they have in the past, the SR-52 and companion printer are not a bad way to get into the intricacies of computers. At the time of this writing, the combination can be bought for about $400. The fact that the SR-52 can be connected to the real life world (the printer), does hold a possible promise that a computer interface to the SR-52 might come along someday.

For those who have an SR-52 and need an invoice program, a listing of the program discussed in this article is provided in listing 1. The program uses 110 program steps, less than half of the total SR-52 capacity. This will allow space to modify the program to meet your specific needs.

### Table 2: User instructions for the invoice program. It is always necessary (for peace of mind) to write out these instructions, otherwise you may forget how to use a program.

<table>
<thead>
<tr>
<th>Step</th>
<th>Procedure</th>
<th>Press</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Load &quot;A&quot; side of card</td>
<td>CLR 2nd READ</td>
<td>Goes blank then 0</td>
</tr>
<tr>
<td>1a</td>
<td>Enter discount (see note below)</td>
<td>run A</td>
<td>Discount amt printed</td>
</tr>
<tr>
<td>2</td>
<td>Enter shipping</td>
<td>D</td>
<td>Shipping amt</td>
</tr>
<tr>
<td>3</td>
<td>Item quantity</td>
<td>A</td>
<td>Qty printed</td>
</tr>
<tr>
<td>4</td>
<td>Total list price of item</td>
<td>B</td>
<td>Individual and total price printed</td>
</tr>
<tr>
<td>5</td>
<td>Price printed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Total of list items</td>
<td>2nd A</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Enter item quantity</td>
<td>A</td>
<td>Qty printed</td>
</tr>
<tr>
<td>8</td>
<td>Enter net item price</td>
<td>E</td>
<td>Individual and total price printed</td>
</tr>
<tr>
<td>9</td>
<td>Net total printed, Then shipping and Invoice total printed</td>
<td>2nd B</td>
<td></td>
</tr>
</tbody>
</table>

Note: For 40% customers, enter .6; For 50% customers, enter .5; For 50+ 10+10, enter .405.
Computer Information Arrangement

Figure 1: This is a basic diagram showing the input arrangement for the information retrieval system. The text is entered from a keyboard into a buffer area, and sorted alphabetically. In the example the three text strings start with S, D and K. When a certain buffer area is filled the information it contains is dumped to an input cassette tape. The information on the tape is sorted in alphabetical order. When any one tape is filled, or an updated master file is desired, the input tape is added to the master file.

An examination of the small system computer field might lead the observer to take a limited view of the potential uses of small systems. This is unfortunate, because a computer, even a small one, can do more than play games or make lights blink.

One general application of computers is the information retrieval system. A classic goal of information retrieval is the construction of a system that absorbs the contents of books and can answer questions concerning the information contained in them. This goal has been unapproachable in even the largest of computer systems. The best approach is to put the burden of intelligence on the user's shoulders and make use of the computer's bookkeeping ability. This reduces the program to a large scale sorting system tailored to a microcomputer's capabilities.

Small systems have limitations in memory size, data transfer rates and throughput. To cope with these limitations, I propose a mass information handling system called the Computer Information Arrangement, or CIA. The basic hardware required for this system includes a processor, 8 to 16 K bytes of programmable memory, keyboard, TV interface and several cassette interfaces with a data rate of at least 300 bps. One cassette drive has to be controllable by the computer in a manner beyond that of simple motor control.

The main storage memory for the huge data base is magnetic tape. Tape is slow and serial, meaning that only the information physically located near the tape head can be dealt with. However, it is cheap. For the moment, our data base will be a dictionary, i.e.: a list of definitions sorted alphabetically.
Building a better computer wasn’t easy. But we did it.

Introducing the MSI 6800 Computer System

When we set out to build the new MSI 6800 Computer System, we knew we had our work cut out for us. It had to be at least as good as the now famous MSI FD-8 Floppy Disk Memory System which is also pictured below. So, the first thing we did was analyze all the problems and drawbacks we had encountered with other 6800 systems, and then put our engineers to work on solutions. The objective: Build a better computer.

We started with power supply. We had big ideas, so we used a hefty 18 amp power supply. You can run full memory and several peripherals without the worry of running out of juice. We also put it in the front of the cabinet so it’s out of the way.

The next step was the CPU Board. A separate baud rate generator with strappable clock outputs allows any combination of baud rates up to 9600. A separate strappable system clock is available and allows CPU speeds of up to 2 MHz. The new MSI monitor is MIKBUG software compatible, so you will never have a problem with programs. Additional PROM sockets are available for your own special routines and to expand the monitor. The CPU also contains a single step capability for debugging software.

When we got to the Mother Board, we really made progress. It has 14 slots to give you plenty of room to expand your system to full memory capability, and is compatible with SS-50 bus architecture. Heavy duty bus lines are low impedance, low noise, and provide trouble-free operation.

With all this power and potential, the interface had to be something special. So instead of an interface address in the middle of memory, we put it at the top... which gives you a full 56K of continuous memory. Interfaces are strappable so they may be placed at any address. An interface adapter board is compatible with all existing SS-50 circuit boards and interface cards. All MSI interface cards communicate with the rear panel via a short ribbon cable which terminates with a DB-25 connector. All baud rate selection and other strappable options are brought to the connector so they may be automatically selected by whatever plug is inserted into the appropriate interface connector. Straps may also be installed on the circuit board.

To complete the system, we used an MSI 8K Memory Board which employs low power 2102 RAM memory chips and is configured to allow battery back-up power capability. A DIP switch unit allows quick selection of a starting address of the board at any 8K increment of memory.

If you’re one of those people who understands the technical stuff, by now you’ll agree the MSI 6800 is a better computer. If you’re one who does not understand it yet, you’ll be more interested in what the system can do... play games, conduct research and educational projects, control lab instruments, business applications, or just about anything else you might dream up that a microcomputer can do. The point is... the MSI 6800 will do it better.

The MSI 6800 Computer System is available in either kit form or wired and tested. Either way, you get a cabinet, power supply, CPU board, Mother board, Interface board, Memory board, documentation, instructions, schematics, and a programming manual. Everything you need.

There is more to say about the MSI 6800 than space permits. We suggest you send for more information which includes our free catalog of microcomputer products.

Building a better computer was not easy. Becoming the number one seller will be.

See the MSI 6800 Computer System at Personal Computing ’77 - Atlantic City.

Midwest Scientific Instruments

220 West Cedar • Olathe, Kansas 66061
913/764-3273 • TWX 910 749 6403 (MSI OLAT)

NAME ______________________________
ADDRESS ______________________________
CITY ______________________________
STATE __________________ ZIP

Circle 411 on inquiry card.

0011077
by keyword. If the dictionary is closely packed on the tape, it will be difficult to add to it without shifting half of the data base. It would be more logical to spread out the entries on the whole tape to avoid later space problems. Unless the tape is getting full, the proper position of an entry is solely a function of keyword.

If entries are to be added in an efficient manner, close attention must be paid to differing data rates. A human can type two to five characters (bytes) per second, while a computer can take things on or off tape much faster. The typical personal computer can internally manipulate at least 250,000 bytes per second when programmed with an assembler. A video display can depict about 1,000 characters at a time, and can refresh itself 30 or 60 times a second, depending on the way the display handles the interface.

It will be the objective of the CIA system to put information onto the cassette tape in sorted order as fast as the user can type in the unsorted data. The user can therefore type the definition of “best” and of “machine” into our imaginary dictionary, and the computer will place both in their proper places on the tape.

A large part of main memory (at least 4 K bytes) is used as a buffer. As new data is typed in, it is added to the buffer and sorted on keyword. This sorting can be done by rearranging the data in the buffer in sequential order. An alternative is to keep items in unsorted order and maintain two pointers for each item, one pointing to the location of the item which is next in sorted order, the other pointing to the previous item in sorted order. The second system eliminates unnecessary searching in memory, but involves longer and trickier programming.

As the memory buffer accumulates data, it is important to keep track of how it is filling up. The alphabet is divided into eight sections for accounting purposes. The first section may be for words starting with A or B, etc. Eight counters keep track of how many bytes are taken up in the buffer by different ranges of the alphabet. When one counter exceeds certain limits, the cassette is moved to the region of tape corresponding to that range of the alphabet. Next, the data held in the buffer is transferred onto the tape in the proper location. Obviously the data would then be erased from the memory buffer to make room for more data. The end result is a cassette tape containing sorted information which generated at the same rate that the user is typing in the unsorted data (see figure 1). If the data is sorted as fast as it can be put in, what would be the advantage of greater throughput? The system works just as fast as it has to.

Notice how general the system is. It can be used to make huge mailing lists, keep track of books in a library, and so on. It has two principle limitations in addition to speed, size and the simple nature of the data that it can handle.

What can you do when you fill up a cassette? (It may take a while, since it is possible to fit as many as 500,000 characters on a digital cassette tape.) You could maintain a master set of 26 tapes, one for each letter of the alphabet. Once your tape is full, it would be merged with the master file, an unwieldy process at best. This procedure would mean putting master tape #1 in one cassette machine, your input cassette in another, and starting the merging program. After a while, the computer could signal that it had put all of the A entries onto master tape #1. Then you would take out tape #1, replace it with tape #2, and so on, up to tape #26. This process would happen rarely (or as often as you would require an up-to-date master file). But imagine how much data your system could hold. A friend of mine pointed out that if you were mechanically inclined, you could automate cassette manipulation. It would be a cross between a jukebox and large scale automated mass memory with media manipulation mechanisms. An automatic multimegabyte memory system for a few thousand dollars would be most impressive.

The question of data bases is a bit tricky. Data can be abstract, highly interrelated, and difficult to categorize. Your data will be interrelated in ways that the data base cannot show or represent. There are two approaches to follow. One way is to design very abstract data structures that show relationships inherently. The other is to maintain the simple dictionary alphabetic system, and add several cross-reference pointers. An example would be “Kennedy, Jackie: see Onassis, Jackie.” By pursuing all the pointers listed under a keyword, and checking out all the pointers listed there, a tree structure is developed. A multi-cassette data system implies a significant amount of tape manipulation, unless you have built the jukebox system.

Although a pointer system is a bit crude, it can be handled automatically. The following example illustrates a typical entry. The original data entry: “Beethoven,
Ludwig van, Symphony Number 3 (The Eroica)" would be filed under "Beethoven, Ludwig van". If the user wants to generate the cross reference pointer "Eroica Symphony, see Beethoven, Ludwig van", a special character could be typed before "Beethoven, Ludwig van" which the program would recognize. The program could then add "Eroica Symphony, see Beethoven, Ludwig van" to the text buffer. This would ensure that the pointer and the data match, eliminating a problem with typographical errors.

Later, if it is necessary to eliminate the entry, you would know that the cross-reference pointer is also in memory because of the existence of the special character. Other special functions can be implemented by special characters, such as labeling the data source of facilitating tabular data. This is left as an exercise for the reader. The power of this information handling system is limited mainly by the size of programs that can be stored in memory, and by the speed of the tape recorder.

The Computer Information Arrangement needs five separate programs to work properly. Note that it is not necessary for more than one to be in memory at any time. Program 1, the input program, is the biggest and most difficult. It accepts characters from the keyboard, edits them, adds the cross-reference, puts them in the buffer, recognizes when the tape machine is idle or part of the alphabet range is getting full in the buffer, and spreads the data on the input cassette by means of a linear hashing formula. It may be necessary for the tape recorder to be controlled by a separate microprocessor and 1 K bytes of programable memory shared by both processors, because of timing considerations. The second one is the merge program, which merges the input tape with the master set of cassettes. The third program, called the clean-up, goes through tapes, "unbunches" data, and straightens out any local area that gets "out of sort." The fourth program is the display program. The user can tell it to display the Richard Nixon file, whereupon it will display all the references and pointers that are filed under the keyword Nixon, Richard. The last program does a crucial, but easily forgotten job: altering or deleting outdated or incorrect data from the input tape or from a master tape.

The CIA is a general computer information arrangement, an answer machine, or a list maker. Put in randomly ordered data and it comes out neat and organized. The arrangement has many applications: small business, journalism, research, or help for folks who have trouble organizing things. This is the type of program which will sell small systems to the world.

GLOSSARY

Alphabet range: part of the contiguous alphabet used to decide where to store alphabetically sorted data.
Buffer: section of random access memory used to temporarily store data until enough is collected to pass on.
Cross reference: a notation to look elsewhere in the data base for more information.
Data base: collection of information and the system used to organize it for use by a computer program.
Entry: A block of data that stays together during the sorting routine. The end of an entry is recognized by a special termination character.
File: Set of entries with the same keyword.
Input tape: cassette that accepts the sorted data. For larger data bases it must be merged with the master tapes.
Keyword: Word in an entry used to sort the entire entry into the data base.
Master tapes: Set of cassettes that make up the entire data base. Each cassette covers a portion of the alphabet range.

A potent pair for AC power control

RELAY/OPTO-ISOLATOR CONTROL BOARD ($117)

This board is a natural for controlling audio systems, time lapse photography experiments, model trains, robot devices, or any application where you need a number of intelligent switches. 8 reed relays respond to an 8 bit word from your computer. Opto-isolators accept an 8 bit word from the outside world and send it back to your machine for further control purposes. Includes detailed instructions and application notes.

500 WATT CONTROL MODULE ($10)

Interfaces your Relay Control Board with power loads such as photofloods and motors. This module is about the same size as a plug-in wall transformer and plugs into any wall outlet: on the back is a female AC socket that accepts AC loads up to 500 Watts. Two isolated control wires trail back to your computer. Allow easy remote operation and keeping AC lines out of your machine.

MULLEN COMPUTER BOARDS

BOX 6214, HAYWARD, CA 94545

“Products that make your computer useful”

Chess Skill in Man and Machine is a series of eight essays dealing with four main topics, two major and two minor. These topics include the history and rules of computer chess tournaments, human chess skill, descriptions of the techniques of computer chess, and finally, an overview of the past, present and future of computer chess. The chapters of most interest to the computer chess enthusiast are those explaining human chess skill and the various computer chess techniques.

The first major topic is an analysis of chess skill in humans. Neil Charness in chapter 2 poses the question, "Should a computer be more like a human?" He describes the two schools of thought concerning this question. One school answers no, because computer hardware operates differently from the human mind and should therefore be operated in its most efficient mode, not like the human mind. The other school says yes, a computer should be like a human if it is to play and beat humans. The latter school believes that human thinking should be simulated in order to enable the computer to capitalize on any human miscalculations.

Charness believes that to write a better chess program, the programmer has to understand how humans solve the problem of choosing the best move. "Chess-specific" perception is the key. Searching through a tree of moves which consists of every possible move during an entire game of chess is an extremely inefficient way to play chess. An analysis of the legal moves for each side to a depth of six moves during a single turn of a player would involve over six billion different combinations of moves.

Becoming a good chess player demands that one learn to recognize types of positions. This chess-specific perception (or pattern recognition) can reveal weaknesses for further examination, thereby cutting down on the number of moves to be examined. This method of "pruning the search tree" is the human method; it should also be the computer method.

The second major topic deals with different theories of computer chess. Chapter 3, written by editor Peter W Frey, introduces opening move theory (whether to store standard openings or to simply play them from scratch every time), machine representation of the chessboard (using either bit patterns or the "mailbox" method), and move searching procedures.

There seem to be two general types of search strategies in use today: the A type (looking at all legal moves and counter-moves at each position), and the B type (looking at only a small subset of the potential legal moves). Frey prefers the B type strategy over the A type because of the number of positions to be examined.

The B type suggests that a minimum of moves from any position should be evaluated using von Neumann and Morgenstern’s minimax procedure. The idea of backward pruning using a heuristic method is also covered, involving evaluation of all capturing moves first, and so on down a list of types of moves. The various search strategies are given in detail in chapter 3 and are elaborated on in chapters 6 and 7.

Northwestern University's program, CHESS 4.5, is covered by the authors of that program. CHESS 4.5 is particularly relevant to the programmer who has a grasp of the theory and who is eager to design his or her own program. The creators of CHESS 4.5 discuss their older programs and the writing of their latest program. They discuss their particular searching strategy, how they came to decide upon their method of generating moves and evaluating them, and their reasons for reaching certain decisions. This discussion tends to bring one down from the theoretical level to the practical level of decisions concerning implementation of theories.

The book begins with a retrospective of past computer chess tournaments and their rules, and ends by asking the question, "Why program a computer to play chess?". I think there are two answers to this question. A computer chess program to simulate human methods of playing will greatly aid our understanding of human perception, learning and problem solving. Secondly, it can be fun solving a difficult problem in a creative, imaginative way.

Chess Skill in Man and Machine is an interesting book filled with the theoretical and the specific concerns of computer chess. It should give readers the background to get started, and its bibliography should provide a basis for further inquiry.
4800 BAUD CASSETTE RECORDER

An ASYNCHRONOUS NRZ type Recorder with remote motor start/stop. Error rate 10^-6 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' second 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.

MODEL CC-8 $175.00

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 30 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 3P+S). 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.

MODEL 3M3A $220.00

AVAILABILITY — Off the shelf.

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 Imsai. NO MORE BOOSTSTRAPPING. Loads and Dump's 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 BiiSync (IBM), BipPhase (Phase encoded) or NRZ modes with suitable recorders and interfaces.

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

Fits all S100 bus computers using 8080 or 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.

NOW AVAILABLE: 6800 I/O with ROM, same as our 2SIO (R) but fits SWTP bus. $190 tested. ($160 kit)

NEW: SWTP bus 8085 MPU board. Use 8080 software in your SWP 800 system. $190 tested. ($160 kit)

OVERSEAS: Export Version 220 volt 50 hz. Write factory or: Megatron-Datameg, 8011 Putzbrunn, Munich, Germany; Nippon Automation 5-18-7 Shiba, Minato-Ku, Tokyo, Japan; Hobbydata, FACK 20012, Malmo, Sweden; G. Ashbee, 172 Ifield Road, London SW 10-Sag: Trintronics, Ltd., 196 Queen Street W., Toronto, Ontario, Canada; EBASE, Enrique Barges 17, Barcelona 14, Spain; ARIES, 7, rue Saint Phillipe du Roule, 75009 Paris; Micromem 20131, Milano, Italy; Eagle Electric, Capetown, S. Africa.

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

National Multiplex Corporation
3474 Rand Avenue, South Plainfield NJ 07080 Box 288 Phone (201) 561-3600 TWX 710-997-9530

Circle 413 on inquiry card.
Personal Computing Expo will come to New York!

It's a brand new show (in the world's biggest economic center) specifically for manufacturers, buyers and those providing services to the personal computing enthusiast. For the first time, this booming field will have a New York Coliseum showcase. It is planned as the largest public show of its type and will attract attendance from the huge population areas of Boston, Washington, Philadelphia and Baltimore. New York 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.

More than 100,000 paid subscribers of BYTE magazine will be urged to view the exciting exhibits and to attend the BYTE-sponsored lectures. Circle the dates: October 28, 29, 30 — and make your plans now to be there when Personal Computing Expo comes to New York!

“Personal Computing: An idea whose time has come!”

Tutorials by the IEEE Computer Society, Mid-Eastern Area

Personal Computing Expo is also joined by the Institute of Electrical and Electronics Engineers Computer Society, Mid-Eastern Area Committee, whose experienced staff is presenting six day-long tutorials at a modest charge. If inconvenient for you to attend a tutorial during the show, simply sign up for follow-up tutorials on weekends after the show.

Whereas the lectures will provide you with information, the tutorials will teach important skills enabling you to use your own computer at home or at work.

TUITION:
The tuition fee for the tutorial program includes a one-day admission to the Personal Computing Expo.

<table>
<thead>
<tr>
<th></th>
<th>One Tutorial</th>
<th>Two Tutorials</th>
<th>Three Tutorials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students (with ID)</td>
<td>$30</td>
<td>$50</td>
<td>$75</td>
</tr>
<tr>
<td>IEEE Members</td>
<td>$40</td>
<td>$70</td>
<td>$100</td>
</tr>
<tr>
<td>Non-Members</td>
<td>$50</td>
<td>$90</td>
<td>$125</td>
</tr>
</tbody>
</table>

Tuition includes hand-out material, including text and/or hand-out materials. Participants will also receive a certificate of participation.

REGISTRATION:
In order to provide an interactive, learning environment between the participants and the lecturers, the number of registrants is limited. Registration is accepted on a first-come, first served basis. Early registration is therefore suggested. Cancellations received before September 15, 1977 will receive a full refund.

To register, make your check payable to the IEEE COMPUTER SOCIETY, and mail to:
Daniel R. McGlynn, Ph.D.
Tutorial Program Chairman
IEEE Computer Society
329 - 84th Street
Brooklyn, N.Y. 11209

FOR FURTHER INFORMATION:
on the technical content of the tutorials, technical background suggested to derive maximum benefit from the program, or information on the IEEE Computer Society, call
Cary Ringel
Chapter Chairman
IEEE Computer Society (212) 460-4600

TIME AND LOCATION:
The tutorials will be held from 10 AM to 5 PM on Friday and Saturday, and from noon to 5 PM on Sunday in the New York Coliseum, at a location to be announced and posted. Participation in the tutorials also includes a one-day admission to the exhibition area and other lecture programs.

EXPO TICKETS:
Will be sent to tutorial registrants about three or four weeks prior to the show.
Exciting lectures sponsored by

BYTE
The Small Systems Journal

Personal Computing Expo is endorsed by BYTE magazine, whose staff has contacted prominent speakers for an exciting series of lectures. Visitors will be able to attend these meetings free of charge. The lectures will not conflict with each other eliminating the worrysome choice among several equally important topics. In addition, they will be repeated on the next day to give you a second chance if you missed a topic. Lectures are typically 30 minutes, often with demonstrations and an additional 15 minutes for questions.

Portia Isaacson Ph.D. . . . Saturday 11 AM and Sunday 12 Noon
Co-owner of the Micro Store, a personal computer store, in Richardson Texas actively engaged for 12 years in the computing field in industry and at universities. Member of the ACM and IEEE, and chairman of the 1977 National Computer Conference. Author of many articles in professional journals and magazines. Received a Ph.D. in Computer Science from the Southern Methodist University.

1. Personal Computing: An Idea Whose Time Has Come
A review of what has happened so far in the personal computing field, and an outlook into future developments, including those in the computer assisted home. Slide demonstration.

Sol Libes . . . Friday 6 PM and Saturday 10 AM
President of the Amateur Computer Group of New Jersey. Teacher of electronics and computer programming at a community college. Author of 10 books (writing on the 11th) and several hundred magazine articles in electronics and computing. Received an award for "The Outstanding Amateur Computer Hobbyist of 1976" in Atlantic City by Personal Computing 1976 show, and BYTE.

2. How to Get Started
A discussion of typical home computer systems and their essential hardware and software components.

John H. Dilks Ill . . . Saturday 1 PM and Sunday 2 PM

Discussion of "for-out" applications of microcomputers and electronic technology for home use, such as a child locator and warning device, a home security system, etc. Slide demonstration.

Robert S. Jones . . . Friday 7 PM and Saturday 2 PM
Publisher of Interface Age Magazine. Prior experience in sales and marketing for the semiconductor industry, including Intel, National Semiconductor and Analog Devices Inc.

4. Personal Computing for the Business Man
Evaluating business applications for microcomputers, including slides showing selected applications.

Louis E. Frenzel . . . Saturday 3 PM and Sunday 3 PM
Director of Computer Products at Heath Company, involved in the planning of new computer products. Prior to Heath, with McGraw Hill in product planning and design of educational electronic kits. Prior experience including computer engineering for eight years. Author of several books, home study courses and numerous magazine articles in electronics and computers. Received a BS in electronics from the University of Houston and a MEd from the University of Maryland.

5. How to Build Personal Computer Kits
Tips for successful kit construction. Benefits of kit products for the personal computer user. Including slides showing selected computer kits.

Carl Helmers . . . Friday 9 PM and Saturday 4 PM
Editor-in-Chief and co-founder of BYTE magazine. Obtained computing experience as a personal way to accomplish artistic and technological goals in music. Graduated in 1970 with a BS in Physics from the University of Rochester, NY. Worked for several years at Intermetrics, Inc. in Cambridge, Massachusetts on the NASA Space Shuttle Project. Prior to working with BYTE, publication of a small computer newsletter on a part-time basis.
6. Computers and Music
How to create music with computers. Problems of performing electronic music, music under computer program control and computer music in conjunction with traditional instruments. Illustrations and examples from personal experience.

Jack L. Davies ... Friday 8 PM and Saturday 5 PM
President of Pan Atlantic Computer Systems GmbH, a distributor of various microcomputer systems in Europe.
Extensive experience in using minicomputers and microcomputers in the US Military Schools in Europe. Designed and developed numerous games and educational programs for students in these schools.

7. Microcomputers in Education
Discussion of the many possibilities of using microcomputer systems in schools. Effect of personal computers on students.

David Fylstra ... Saturday 6 PM and Sunday 5 PM
Member of the research staff of the Telecommunications Sciences Center at Stanford Research Institute for more than two years. Specialized in microcomputer software and computer simulation of speech processing systems.
Graduated in 1974 with a BA in English and Psychology, Stanford University, Phi Beta Kappa.
Active in the research on communication systems and devices for the deaf.

8. Speech Analysis and Synthesis for the Amateur
Using the personal computer as a device to analyze the acoustical foundations of speech and to formulate rules for the control of the speech synthesizer.

Max Mathews Ph.D. ... Saturday 7 PM and Sunday 4 PM
Director of Acoustical and Behavioral Research, Bell Labs
Author of The Technology of Computer Music, and numerous articles.
Scientific Advisor to the Institute for Research and Coordination of Acoustics and Music (IRCAM)
Dr. Mathews is often regarded as the "Father of Computer Music"

9. Pure Digital and Real Time Music Synthesis
The use of the digital computer as a musical instrument with which composers and performers create and play music. Slide and tape demonstration.

Carl L. Holder ... Saturday 8 PM and Sunday 6 PM
Director of Product Management, Planning and Communications at Information Terminals Corp. for five years.
Prior experience, including Memorex Corp., in the area of magnetic media development and testing.

10. Present and Future Storage Devices
Survey and discussion of current devices and media, including latest technological developments like the charge coupled devices and magnetic bubble memories. Costs, advantages and disadvantages of these devices for the personal computer user. Accompanied by slides.

--- DAILY TUTORIALS ---
There will be two tutorials offered each day, one aimed at those participants who have little or no experience with microprocessors, and the other for those already experienced with microprocessor systems.

BASIC COURSES

1. Development of Microcomputer Systems for Business Use ... Friday 10 AM to 5 PM
Sy Rainier, Citibank, N. A.
- system design concepts
- economic efficacy of microcomputers versus large centralized computer systems
- distributed processing definitions
- advantages and problem areas
- network design and architecture
- data communications aspects
- case study: design of a stand-alone workstation for data entry and retrieval

2. Development of Microcomputer Systems for Home Use ... Saturday 10 AM to 5 PM
Cory Ringel, Con Edison
- survey of simple microcomputer control systems for home use
- selection of hardware: IC's, boards, kits, development systems
- interfacing and software aids
- interfacing: A/D and D/A conversion
- examples: design of a home control system: microcomputers for a music synthesizer: computer TV games
- case study: use of the Motorola 6800 in design of a microcomputer system

3. Survey and Comparison of Microprocessors ... Sunday Noon to 5 PM
Donald Lewis, Standard Microsystems Inc. and other speakers
- definitions and distinctions between ALU-chips, controllers, microprocessors, microcomputers
- current applications
- microprocessor architectures (bit-slice, 4-bit, 8-bit, 16-bit, microcomputer-type)
- vendor survey
- performance evaluation and criteria for selection

INTERMEDIATE/ADVANCED COURSES

4. Microprocessor Interfacing ... Friday 10 AM to 5 PM
Donald Lewis, Standard Microsystems Inc.
- interface components (peripheral interface chips, UARTS, etc.)
- interface standards (IEEE 488, RS 232C, S-100, etc.)
- interfacing to keyboards
- interfacing to casette and floppy disk drives
- interfacing to display devices
- case study: how to design a CRT terminal

5. Microprocessor Programming and Software ... Saturday 10 AM to 5 PM
Donald Lewis, Standard Microsystems Inc.
- software design: flow-charting, setting breakpoints, documentation, etc.
- assembly language for the Intel 8080, 8085, 2-80, Motorola 6800
- instruction types and addressing techniques
- use of the stack
- interrupt handling and direct memory access (DMA)
- software development aids
- high level languages for microcomputers

6. Technology Analysis and Forecast of Future Microprocessor Structures ... Sunday Noon to 5 PM
- emergence of specialized computational elements (CSE)
- architectural evolution (stack processors, reconfigurable architectures, multi-level logic)
- resource management techniques
- software evolution (micro-programming, extensible instruction set, super-high-level languages)
- evolution of semiconductor technology of microprocessors
- microprocessor architecture at the chip level
- case studies: design of MOS Technology's new 8-bit and 16-bit processors
Personal Computing Expo to be produced by
H.A. Bruno & Associates

H.A. Bruno & Associates, Inc., has been prominent 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.

Interesting, educational exhibits of Personal Micro Computers

The lectures and tutorials are not just theory! You will be able to see a multitude of micro computers. Try out various demonstrations: fascinating games, even in color; small business applications; computer produced speech; music; color graphics; and many more. Micro computers are not only small and portable, also you will need only a “micro budget” to be able to take them home.

Show Hours and Admission

Personal Computing Expo hours are as follows:
Friday, Oct. 28 — Noon to 10 p.m.
Sat. Oct. 29 — 10 a.m. to 10 p.m.
Sunday, Oct. 30 — Noon to 7 p.m.

General Admission: $5.00 (includes free BYTE lectures) per day.
Two-day Tickets: $9.00 (advance sale only)
Three-day Tickets: $13.00 (advance sale only)

Advance sale of tickets available . . . Three days ‘13
Two days ‘9 . . . General Admission ‘5
MAIL THIS CONVENIENT ORDER FORM NOW!

Personal Computing Expo admission is $5.00 per day. Advance reservation eliminates waiting in line. Order advance tickets with this coupon. Admission ticket includes access to exhibits, lectures and tutorials.

Please send me _________________ advance registration tickets for three days, October 28-29-30. Total cost $13.00 per person.
Please send me _________________ advance tickets for two days, October _____ and October ______. Cost is $9.00 per person.
Please send me _________________ advance tickets for one day, October _____. Cost is $5.00 per person.

Make all checks payable to PERSONAL COMPUTING EXPO, and mail to:

Name ____________________________ Amount enclosed $_______
Address _____________________________________________________
City _________________________ State ______ Zip ______

General Information
You may find it advantageous to purchase two or three-day admission tickets in advance. These are available by mail only, no later than October 10, 1977. Use coupon below.

Group rates (10 or more persons) qualify for $1.00 off regular prices. Arrangements must be made by mail prior to October 10, 1977.

Special arrangements have been made if you desire to stay overnight. Our headquarters hotel, the Barbizon-Plaza, is located on Central Park South, two blocks from Columbus Circle. Single rooms available at $34.00 per night; $40.00 double, plus tax. There’s a weekend plan: $22.95 daily, plus tax per person, double occupancy . . . includes breakfast (brunch on Sunday) and meal gratuities. Children under 14 in same room with parents, free.

For hotel reservations and information, call toll free (800) 223-5493. From New York State call (800) 223-5963.

For those traveling to New York by air, American Airlines offers a convenient service through arrangement with Personal Computing Expo. For information, call toll free (800) 433-1790. In Texas the number is (800) 792-1150. From the West Coast, round trip fare via American is only $227.00.

20,000 persons are expected to attend and view the more than 200 exhibits by personal computer manufacturers and retailers.

Personal Computing Expo will occupy the 4th floor of the New York Coliseum. It is located on 59th Street and Columbus Circle — the geographical center of New York City. Garage parking in the building is available.

For answers to any questions pertaining to your attendance at Personal Computing Expo, contact the Show Manager, Ralph Ianuzzi, at Area Code 212/753-4920.
Analyze Your Car’s Gas Economy with Your Computer

In a note accompanying this article, author Bauernschub reported: This program was developed to be displayed on a video monitor using the SWTPC CT-1024. However, I borrowed a Texas Instruments Silent 700 to generate the listing and was happy at the ease of interfacing it to the SWTPC MC-6800 and AC-30 cassette interface.

A few comments about myself. First, I had never put an electronic kit together before this. I spent six months studying the microcomputer market and decided on Southwest because they appeared to have “put it all together.” It was enjoyable and fairly straightforward assembling the kits, and they all worked.

How well is your car performing? Is it matching the EPA miles per gallon ratings that the manufacturer posted on it? Is it time for a tune-up? Here is a system that will produce these answers and provide a worthwhile application for your personal computer.

The first step in this system is to gather data. What we’re out to do is automate a familiar procedure: analysis of data about your car. But analysis requires data. Every time you purchase gas, record the quantity and the mileage. It helps to keep a small notebook in the glove compartment of your car for this. The system will be the most informative if the tank is filled each time. However, the fill-up is not necessary every time since the program totals the gallons and miles for an overall rate. You may also wish to note when you do an extraordinary amount of highway or city driving so you can identify its effect on your normal miles per gallon.

If you want to check your fuel performance and have not been recording detailed fill-up data, you can recover past data from gas receipts (on your credit card bills) and get a starting mileage from a shop maintenance record.

When the program is initiated you have the option of displaying the instructions. Next you are asked for the mileage before the first gas purchase. This is either the mileage when you first started recording the data or the last mileage used the previous time the program was run. For the greatest accuracy, the tank should be full for the first and last recordings. This value is stored by the program in the variable L to be used in the final statistics, and assigned to M2 for the first computation.

Column headings are displayed and you are requested to input mileage and gallons for the first fill after the mileage used as the start. The program will then display the mileage, the number of gallons, the number of miles since the last purchase and the miles per gallon for this data. You are again prompted for miles and gallons and the cycle repeats itself.

If you detect an error in the most recent entry, you can back it out by entering a negative number for the next mileage and zero for gallons. When this is done, the number of gallons for the most recent entry, G4, will be subtracted from the total gallons, T, (line 370); and the most recent mileage variable, M2, will be set equal to the second most recent mileage variable, M4. When you are prompted for mileage and gallons again, enter the corrected amount. If this is also in error, it can likewise be backed out. However, you can only back out the most recent entry. If you tried to back out multiple levels of entries by entering successive negative values for mileage, nothing would happen because after the first back out G4 was set to zero (line 380) and M2 equals M4.
After you have entered your last data and are prompted for miles and gallons, enter a zero for each and your computer will tell you how many miles you drove, how many gallons were used, and your average miles per gallon. This miles per gallon figure is computed using the total gallons and miles driven and not an average of each previously computed miles per gallon.

This program was written for a SwTPC 6800 and will run with either 4 K or 8 K BASIC. Its output was designed to fit the SwTPC CT-1024 video screen and will therefore appear squeezed on a wider screen or printer. This can be adjusted by changing the spacing of the column headings in line 700 and the tabs in line 320. The statement PRINT CHR$ (16);CHR$ (22); in lines 450, 520 and 690 are the computer cursor commands (home up, erase to end of frame) for the CT-1024 TVT and should be changed or left out if some other terminal is utilized. Also, all REM statements can be left out at the risk of losing some documentation value.

Carry this idea forward in your computer to help keep track of the performance of your car.

Listing 1: The BASIC program for analysis of automobile mileage data. This program is written for the SwTPC 6800 system's 4 K or 8 K BASIC interpreters, and can easily be adapted to any interpreter which implements a minimum of decimal arithmetic and string output operations for formatting. This program is not recommended for use with "tiny" BASIC interpreters.

```
10 REM ANALYSIS
20 INPUT TO DECLARE ON RELEASE
30 GOSUB 2000
40 END

2000 REM START
3000 PRINT "MILEAGE ANALYSIS"
3100 PRINT "MILES GAS";
3200 IF MILE<0 THEN 30
3300 PRINT MILE;
3400 IF MILE>0 THEN 31
3500 MILE=0
3600 REM L HAS NUMBERS, byte 41
3700 REM L-contextualized, byte 42
3800 IF L<0 THEN 37
3900 PRINT L;
4000 IF L>0 THEN 38
4100 L=0
4200 PRINT "0"
4300 GOTO 2000

MILE: 0
```

Micronics Inc.
BOX 3514, 123 WEST 3RD ST., SUITE 8
GREENVILLE, NC 27834 • (919) 758-7757
Mastermind has become one of the most popular games around these days. (Games and Puzzles magazine gave it their highest rating.) Among other applications, it is used by a number of computer companies to test the programming skills of potential employees. The game was originated by Invicta Plastics; Mastermind is a trademark of that firm. In its original form, this game of skill requires a mind, but no computer.

The number of black pegs corresponds to the number of correct colors in correct positions, and the number of white pegs gives the number of correct colors in wrong positions. An important rule is that no position in the try is counted more than once. For instance, if you have Green in columns 1 and 3, and the secret code has Green in 3 only, you get a black peg for Green in position 3, and the secret code has Green in 3 only, you get a white peg for Green in position 3. If you fail to break the code in ten rows, the codebreaker will tell you the answer and let you play again. It helps to study a missed game to see where you made faulty deductions.

One of the most interesting conventional (ie: noncomputer) games on the market is "Mastermind," distributed by Invicta Plastics, Suite 940, 200 5th Av, New York NY 10010, and available in many local stores. Mastermind involves deductive logic, hypothesis testing and probabilistic inference. In Mastermind, the players take turns as "codemaker" and "codebreaker." The codemaker sets up a concealed row of four colored pegs from a set of Red, Blue, Brown, Green, Yellow and Orange pegs. It is acceptable to use the same color or colors more than once. In version 2, a more advanced game, empty Spaces are also permitted. (The bold face characters are abbreviations recognized by the BASIC program.)

The codebreaker has ten tries (rows) in which to discover the secret arrangement of colors in the concealed row. To input a row to the computer, type the legal abbreviation for the leftmost color on the first line, the next to left on the second line, and so on. After each row has been typed in, the program evaluates the try and types back the following message:

YOU GET \text{n} BLACK AND \text{m} WHITE PEGS ON ROW \text{i}.

Listing 1: Text of BASIC Mastermind Codemaker. Minor modifications may be required for other versions of BASIC.
RUN

MASTER MIND CODEBREAKER

PLEASE BE PATIENT. SOMETIMES I TAKE A FEW MINUTES ON MY MOVE

WHICH VERSION (1 OR 2)?

MY MOVE FOR ROW 1 IS
YELLOW  RED
HOW MANY BLACK PEGS? 1
HOW MANY WHITE PEGS? 1

MY MOVE FOR ROW 2 IS
RED  RED
HOW MANY BLACK PEGS? 0
HOW MANY WHITE PEGS? 1

MY MOVE FOR ROW 3 IS
BLUE  GREEN
HOW MANY BLACK PEGS? 0
HOW MANY WHITE PEGS? 2

MY MOVE FOR ROW 4 IS
GREEN  YEL L OW
HOW MANY BLACK PEGS? 1
HOW MANY WHITE PEGS? 1

MY MOVE FOR ROW 5 IS
YELLOW  BLACK
HOW MANY BLACK PEGS? 3
HOW MANY WHITE PEGS? 1

MY MOVE FOR ROW 6 IS
YELLOW  BLACK
HOW MANY BLACK PEGS? 4
THANKS FOR THE GAME

ANOTHER GAME? Y
STOP AT LINE 900
READY

Listing 4: Sample run of BASIC Mastermind Codebreaker. Values in response to the queries "HOW MANY BLACK PEGS?" and "HOW MANY WHITE PEGS?" are typed by the player and correspond to correct colors in correct positions and correct colors in wrong positions in program's try.

strategy used by the program is main strength calculation involving little conceptual sophistication, it is nevertheless a powerfully effective strategy. Indeed it might be said that the BASIC Codebreaker program is a Mastermind Master.

After you have played against both of the programs you may want to link the two programs together so they play against each other. It would be interesting to watch the computer's strategy against itself.

Mastermind is marketed in several forms. MiniMastermind is identical to Mastermind except that only six rows are permitted for completing a game. The programs are easily adapted to play any version by altering index variables, subscripts, etc. It is recommended that a playing set be used when playing against the computer, as the game is somewhat difficult to play in your head. However, paper and pencil can be used if no playing set is available. Interesting computer games usually require long and complex programs. Mastermind is a logical and challenging game which can be programmed in a small system with minimum memory and a simple version of BASIC.

Special 5% Discount during October with this ad

Get the Best in Business Data Processing Systems

Accounting, Inventory, General Ledger, Timesharing Applications

Also a complete line of personal computers and peripherals, including:

- Processor Technology
- IMSAI
- Digital Group
- Compucolor
- TDL
- Books/Magazines

HARDWARE/SOFTWARE SUPPORT
Our experienced staff is ready to help you configure the best system to suit your needs

MAIL ORDERS WELCOME
(We accept MasterCharge/BankAmericard)

THE COMPUTER ROOM

124 H Blossom Hill Road • San Jose, CA 95123
(408) 226-8383

OUR PRICES REVEALED

BOARDS

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z-80 CPU</td>
<td>$35.00</td>
</tr>
<tr>
<td>2708/16 EPROM</td>
<td>$25.00</td>
</tr>
<tr>
<td>PROTOBOARD</td>
<td>$25.00</td>
</tr>
<tr>
<td>8k STATIC RAM</td>
<td>$25.00</td>
</tr>
<tr>
<td>16/64 k DYNAMIC RAM</td>
<td>$25.00</td>
</tr>
<tr>
<td>RAM</td>
<td>$35.00</td>
</tr>
</tbody>
</table>

Please add $5.00 shipping per order

CHIPS

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z-80 2 mHz</td>
<td>$25.00</td>
</tr>
<tr>
<td>4 mHz</td>
<td>$35.00</td>
</tr>
<tr>
<td>2708 EPROM 450 ns</td>
<td>$20.00</td>
</tr>
<tr>
<td>21L02-1 RAM 450 ns</td>
<td>$1.30</td>
</tr>
<tr>
<td>21F02-1 RAM 250 ns</td>
<td>$1.50</td>
</tr>
</tbody>
</table>

Circle 416 on inquiry card.
### Byte's Bits

#### Multiplication

One of the applications of my home-brew 6800 based system requires a large amount of complex number crunching. A noticeable improvement in performance can be realized by reorganizing complex multiplication.

Conventional complex number multiplication of two numbers requires two additions and four multiplications. This can be reduced to three multiplications and five additions. Brute force multiplication yields the product directly as the cross product sum. That is:

\[(a+ib)(c+id) = (ac-bd) + j(ad+cb)\]

However, these products can be expanded by the addition and subtraction of a constant to shorten the computation time.

\[
\begin{align*}
(a+ib)(c+id) &= (ac-bd+ad-ad) + j(ad+cb+ac-ac) \\
&= ((a-b)d + (c-d)a + j((a+b)c - a(c-d))) \\
&= (y+x) + j(x-z)
\end{align*}
\]

Where:

- \(x = (a-b) \quad c\)
- \(y = (a-b) \quad d\)
- \(z = (c-d) \quad a\)

William Jackson  
24 Coldstream Dr  
Munster Ontario  
CANADA K0A 3PO

#### Music and Poetry by Computer?

The Association of Computer Machinery (ACM) in cooperation with New York University is presenting a Conference on Computing in the Arts and Humanities to be held in Warren Weaver Hall at New York University, October 21 to 23, 1977. The programs include: computer generated video tape and film works; a concert of music composed with the aid of computers; poetry by computer; a computer fashion show; literary analysis using computers; and more.

For registration information, contact Dr. Naomi Sagar, Conference Chairperson, NYU Linguistic String Project, 251 Mercer St, New York NY 10012, (212) 598-2294, ext 5.

#### An IEEE Hands-on Microcomputer Tutorial

The Nuclear and Plasma Sciences Society of the IEEE, in conjunction with their annual symposium, are sponsoring a hands-on microcomputer applications tutorial. It will be held Saturday, October 22 1977 from 9:00 to 5:00 at the Sheraton Palace Hotel, San Francisco. The tutorial is slanted toward people with computer backgrounds who would like a hands-on experience on actual problems. Suitcase trainers based on the 8080 will be used. Eugene Fisher, Lawrence Livermore Laboratory, and Michael Maples, M and E Associates will head a group of microcomputer systems engineers, teachers and consultants in leading the workshop. Projects include a stopwatch timer display and a set point temperature controller. For further information on registration, write to A J Stripinka, Lawrence Livermore Laboratory, POB 808, Livermore CA 94550. The registration fee is $40 (lunch and coffee included). Material will be sent out for advance study. To obtain maximum benefit from the tutorial it is recommended that attendees study the material in advance. Class will be limited.

#### The Microcomputer Connection

The 24th annual IEEE Fall Conference will be held at the Town House Convention Center in Cedar Rapids IA

---

### INTERNATIONAL DATA SYSTEMS, INC.

#### S100 Bus Cards (ALTAIR/IMSAI Compatible)

<table>
<thead>
<tr>
<th>Card</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, data 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 500 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 Originator/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. Autodialer is included so your computer can call other computers under software control. Operates at 110, 333, 150, 300, and 600 baud.</td>
<td>$245.00</td>
</tr>
<tr>
<td>MCTK Morse Code Trainer/Keyer</td>
<td>Hard/Software package which allows your computer to teach Morse Code, key your transmitter, and send prestored messages. Uses “NEW CODE METHOD” for training.</td>
<td>$29.00</td>
</tr>
<tr>
<td>TSM Temperature Sensing Module</td>
<td>Use it to measure inside and/or outside temperature for computerized climate control systems, etc.</td>
<td>$24.00</td>
</tr>
<tr>
<td>DAC8 Eight Bit Digital to Analog Converter</td>
<td>Requires one eight bit TTL level latched parallel output port. Use it to produce computer music or to drive voltage controlled devices.</td>
<td>$19.00</td>
</tr>
</tbody>
</table>

Terms: Payment with order. Shipment prepaid. Delivery is stock to 30 days. Write or call for detailed product brochures.
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.

*Assembled and tested, $119.95. Add 5% for shipping. Texas residents add 5% sales tax. BAC/MC available.

Ahem. AMI Makes a Sewing Machine LSI Control Chip

The new Singer Athena sewing machine features a novel LSI integrated circuit that has 24 different patterns permanently stored in its read only memory. The chip was designed by American Microsystems and has a capacity of 6 K bytes. It also controls the sequences of variable length pulses going to linear servo actuators that control needle and fabric movement.

The computerized appliance is here to stay, and we can look forward to many happy hours of digital darting and byte based basting.

New Micro Interfacing Workshop

Wintek has added a 2 day “Hands on Interfacing Workshop” to their standard 3 day “Hands on Microprocessor Short Course with Free Take Home Microcomputer.” The new interfacing workshop includes analog to digital and digital to analog conversion, signal conditioning, keyboard scanning and decoding, LED display driving, motor position, velocity control and related topics. Tuition is $299. The fall 1977 schedule includes workshops in Dallas, Houston,

on September 29 and 30 1977. The topic will be “The Microcomputer Connection” and 12 speakers will deliver talks on microcomputer applications in industry, government and the home. Also featured are product and equipment exhibits from a variety of manufacturers. The Fall Conference is open to the public; the fee is $4 for nonmembers, $2 for IEEE members and $1 for students.

For more information, contact Firooz Etemad, IEEE Fall Conference, POB 451, Marion IA 52302.
Programming Tidbit

The normal mode of comparison for equality between two numbers, A and B, is simply the equivalent of the following statement in whatever language is used:

\[ \text{IF } A = B \text{ THEN . . . .} \]

However, in many contexts this is not necessarily the test to do. Suppose in a pattern recognition program, for instance, you merely want to find out if the two variables A and B are within the right "ballpark" of each other, which is the range \( \text{DELTA} \). The semantics of the test computation must then be changed, and the first impulse might be to use the form:

\[ \text{IF } (A-B) < \text{DELTA} \text{ THEN . . . .} \]

However, what happens when B is greater than A and \( \text{DELTA} \) is a positive number? This would give a true result all the time, independent of the true magnitude of the difference between the two. Since \( \text{DELTA} \) is a positive number by assumption, we must force the difference being checked to be a positive value also. This is where it is handy to have a language with an absolute value function \( \text{ABS} \) to force a change of algebraic sign if its argument is negative. Rewriting this comparison gives:

\[ \text{IF } \text{ABS}(A-B) < \text{DELTA} \text{ THEN . . . .} \]

If your interpreter or compiler has no absolute value function, all is not lost, however. You can still check for a differential range by coding the equivalent:

\[ \text{IF } (A-B < 0 \text{ and } (A-B) > \text{DELTA}) \text{ or } ((A-B) > 0 \text{ and } (A-B) < \text{DELTA}) \text{ THEN . . . .} \]

Here the terms and and or have been used for logical functions relating different arithmetic expressions and parentheses are used for precedence of operations. Add this thought to your bag of programming tricks and techniques; it might be quite useful.

IBM Selectric Interface

Regarding "Interfacing The IBM Selectric Keyboard Printer" by Dan Fylstra, in June 1977 BYTE, page 46, see a letter from Beardsley Ruml on page 32 of this issue, paragraph 4, for comments about:

1. Proper wiring of IBM 73x contacts for the purpose of implementing closed loop feedback. Mr Ruml states that Dan's suggestion of a proposed method for implementing feedback will not work. (Dan's interface was open loop, worst case timing.)
2. The IBM specified current and voltage used with operating the switches should not be ignored, since above the minimum 10 V, 10 mA specification there is enough current to "keep the contacts clean."
Program Structure

The BASIC Mastermind Codemaker shown in listing 1 requires only 2 K bytes for the source code including remarks. RND and INT (random number and integer) are the only special functions used; no string functions are needed. This program can be easily modified to run under a version of BASIC that does not support string variables by substituting numbers for the color abbreviations. The secret row of colored pegs is set up by a call to the random number generator in line 290. Note that for version 1 games, R=6 and for version 2, R=7 permitting the assignment of empty spaces. Black peg responses (exact correspondences) are counted from lines 490 to 510, and white peg responses (correct color, wrong position) are counted from lines 512 to 555.

An example of play against the computer program is given in listing 2. The first row of input is a guess. The second row is designed to test whether there are any repeated colors from the first try in the secret code. The codemaker’s response tells us that one of the colors in the first try is repeated. In row 3 we hypothesize that Blue is the correct color in row 2 and that Orange is the repeated color from row 1. The codemaker’s response constitutes a stroke of luck for us as we learn that Brown is the correct color on row 2 and that Red and Orange can be eliminated from row 1. In row 4 we hypothesize that Green is the correct color in the first row, but alas, we are mistaken. Now only two arrangements are possible. We have a 50% chance of getting it on row 5, but fate is cruel.

After playing the Codebreaker half of the Mastermind game you will learn how to get the most information out of each move. I leave these discoveries to the reader.

After trying to break the codes produced by the computer our next step will of course be to get even with it and see how good it is at breaking codes. The Mastermind Codebreaker program attempts to decode input codes. A word of warning; the Codebreaker program is an excellent player.

Listing 2: Example of play against BASIC Mastermind Codemaker. The text gives an analysis of the moves.

Listing 3: Text of BASIC Mastermind Codebreaker. Minor modifications may be required for other versions of BASIC.
### Listing 3, continued:

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>320</td>
<td>PRINT 'HOW MANY BLACK PEGS *1'</td>
<td>Print message</td>
</tr>
<tr>
<td>325</td>
<td>PRINT S(1,0)</td>
<td>Print row 1</td>
</tr>
<tr>
<td>330</td>
<td>IF S(1,0)=0 THEN 320</td>
<td>If no black pegs, go back</td>
</tr>
<tr>
<td>330</td>
<td>PRINT 'THANKS FOR THE GAME'</td>
<td>Print closing message</td>
</tr>
<tr>
<td>330</td>
<td>END</td>
<td>End program</td>
</tr>
</tbody>
</table>

### Strategy

Two simple rules determine the Codebreaker program’s strategy: row 1 is a random try (all possible arrangements are equally probable); each subsequent row is an arrangement which cannot be disproven on the basis of previous results. When playing Mastermind the human player may try an arrangement which is known to be incorrect in order to obtain specific information; this kind of strategy is not used by the BASIC program.

### Program Structure

The BASIC Mastermind Codebreaker shown in listing 3 requires less than 3 K bytes for the source code, including remarks. RND and INT are the only special functions used. Play begins with a call to the random number generator for row 1 at line 200.

After the program types a row it requests feedback information about the try. The number of correct colors in correct positions is requested at line 270. If there are two or fewer black pegs for the try, the program asks for the number of correct colors in incorrect positions at line 360.

After receiving this information the program constructs a table of all possible arrangements, one row at a time (lines 390 to 730). Each row is internally hypothesized to be the hidden code and the question is asked: “How many black pegs and how many white pegs would have been awarded to each try from the first to the current row if the hypothesis were true?” These values are calculated internally by the subroutine shown at lines 910 to 1100. This is the same subroutine used by the Codemaker program. If there is a discrepancy between the calculated value and the actual value awarded for any row, the current hypothesis is rejected and the next row in the table is constructed. The variables L0, L1, L2 and L3 mark the program’s place in the table of possible arrangements between tries so that rejected arrangements are not considered again.

Since row 1 is constructed randomly and therefore occupies an unknown position in the table, it is necessary to check each try to ascertain that it does not duplicate the first row. This check is conducted from lines 610 to 670. If the program exhausts ten tries without breaking the code, it prints the message at line 860. I would be interested to hear if this ever happens.

When evaluating the program’s try it is necessary to count black and white pegs carefully. If you make a mistake counting the number of exact or inexact correspondences, the program may exhaust the table of all possible arrangements without finding a possibly valid try. In this event, the message at lines 740 to 750 is printed.

### Summary

An example of the computer’s play is given in listing 4. Note that although the
Mind
Massagers
from BITS

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 innumerable complex and 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 with a minimum of technical jargon, this book discusses the capabilities of modern digital computers and how they are being used in contemporary AI research. It discusses the progress of AI, the goals, and the variety of current approaches to making the computer more intelligent. $6.95.

The Thinking Computer: Mind Inside Matter by Bertram Raphael. Artificial intelligence, or AI, is the branch of computer science concerned with making computers "smarter." It is a growing, vital field that is, unfortunately, the subject of much popular misunderstanding. The Thinking Computer: Mind Inside Matter is a lucid introduction to AI that does much to overcome this misunderstanding. With a minimum of technical jargon, this book discusses the capabilities of modern digital computers and how they are being used in contemporary AI research. It discusses the progress of AI, the goals, and the variety of current approaches to making the computer more intelligent. $6.95.

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 sourcebook of information for the person who is inclined to manipulate numbers. $4.95.

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 theory. It includes a detailed description of the best computer chess program to date (Northwestern University's CHESS 4.5), the 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.

How to Solve Problems by Wayne 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.

Send to:
BITs, Inc
70 Main St
Peterborough NH 03458

Total for all books checked $ 
Postage, 50 cents per book
for _______ books $ 
Grand Total $ 

Check payment method:

My check is enclosed
Bill my MC No. ___________________________ Exp. date __________

Bill my BAC No. ___________________________ Exp. date __________

Name ___________________________
Address ___________________________
City State Zip ______________________

Prices shown subject to change.
You may photocopy this page if you wish to leave your BYTE intact.

All orders prepaid.
In unusual cases processing may exceed 30 days.

Circle 383 on inquiry card.
References and Robots from BITS

Game Playing With Computers by Donald D. Spencer, published by Hayden. What does it mean to play games using a computer? Read this book to get an introduction into numerous recreational uses of the computer to program and play mathematical and logical games. Topics include numerous mathematical problems, casino games, board games, unusual gambling games, and miscellaneous logic games. Numerous BASIC language programs and listings are included to show details, $16.95.

Scientific Analysis on the Pocket Calculator by Jon M. Smith, published by John Wiley & Sons. This book is another in a set of source books for mathematical analysis using the contemporary products of technology. It is oriented to the pocket calculator, yet it will provide you with algorithms and methods useful with any personal computer which implements the scientific and analytical functions found on a good pocket calculator. For a more complete description, see the book review on page 120 of the December 1976 BYTE; or order its 392 pages of detailed technical information and review its use for yourself. $13.75.

Build Your Own Working Robot by David L. Heiserman, published by Tab Books. This book will not tell you how to build Robbie, the robot of Forbidden Planet, or a classical android of science fiction. What it will introduce you to is the problems of making a robot mobile device called Buster III, using pre-microprocessor TTL integrated circuits for all logic functions. It is a must book for background reading, but much of the logic can be extremely simplified using today's microprocessor technology. Use this book as a first look at these problems from which you can build further and more elaborate solutions. Softbound, $5.95.

Software Design for Microprocessors. This stand alone guide to microprocessors has been designed by the people at Texas Instruments to convey knowledge to the first time user of microprocessors. This excellent source book of computer concepts begins with an outline of the basic principles of the general purpose computer, its machine architecture, software, and methods of addressing. It proceeds to discuss how to build software, what is involved in documenting what you've done once you've done it, the mechanics of programming, and specific examples using the TI TMS-1000, TMS-8080, TMS-9900 and SBP0400 designs. You'll find a thick hardcover textbook filled with over 370 pages of useful information including a comprehensive glossary of microprocessor terminology, among several other detailed appendices. $12.95.

A Dictionary of Microcomputing by Philip E. Burton. In the opinion of BYTE's editor, Carl Helmers, "This is one of the best designed and executed dictionaries of computer related terms yet seen on the market. It is of particular relevance to those individuals who want a good general reference to numerous technical terms, broadly covering hardware and software fields as currently practiced." This new hardbound edition is part of the Garland Reference Library of Science and Technology, $12.50.

References and Robots from BITS

Send to:
BITS, Inc
70 Main St
Peterborough NH 03458

Check payment method:
My check is enclosed
\[\square\]
Bill my MC No. \[\square\]
Bill my BAC No. \[\square\]

Total for all books checked $__________
Postage, 50 cents per book for _______ books $__________
Grand Total $__________

Prices shown subject to change.
You may photocopy this page if you wish to leave your BYTE intact.
All orders prepaid.

Send to:
BITS, Inc
70 Main St
Peterborough NH 03458

Check payment method:
My check is enclosed
\[\square\]
Bill my MC No. \[\square\]
Bill my BAC No. \[\square\]

Total for all books checked $__________
Postage, 50 cents per book for _______ books $__________
Grand Total $__________

Prices shown subject to change.
You may photocopy this page if you wish to leave your BYTE intact.
All orders prepaid.

Send to:
BITS, Inc
70 Main St
Peterborough NH 03458

Check payment method:
My check is enclosed
\[\square\]
Bill my MC No. \[\square\]
Bill my BAC No. \[\square\]

Total for all books checked $__________
Postage, 50 cents per book for _______ books $__________
Grand Total $__________

Prices shown subject to change.
You may photocopy this page if you wish to leave your BYTE intact.
All orders prepaid.
Amateur Computer Club—Essex ENGLAND

In comparison to most hobbyist groups the English-based Amateur Computer Club is an ancient giant. They are now in their fifth year and are approaching 1000 members. Most of course are located in Great Britain, but many live in other European countries, and some in such remote corners of the world as Singapore and North Carolina.

Communication is via a newsletter which is published six times per year and contains articles on hardware, software and other pertinent technical data. The subscription rate is £2, although four American dollars would be equally acceptable.

A word about the club's name. When it was founded five years ago the only other computer group they had heard of was Stephen Gray's Amateur Computer Society. At that time they had no idea that interest in computers would take off to the extent that local clubs would become viable, so they thus have the distinction of being the Amateur Computer Club.

Write Amateur Computer Club at 7 Dordells, Basildon, Essex ENGLAND.

Apple Core Computer Club

According to Apple Core president Jerry Starzinski, AC³ was formed in May of this year, can boast 14 members, two complete SwTPC 6800 systems, two partial SwTPC 6800s in need of terminals, and various individual members looking at some newer entries like Jupiter, Compucolor and Apple II.

The Apple Core Computer Club holds regular meetings at 7:00 PM on the last Wednesday of the month at J M Perry Technical Institute’s Industrial Science Building, 2011 W Washington in Yakima WA. For more information write Jerry P Starzinski, 220 N 2nd St #17, Yakima WA 98901, or call (509) 248-1620 between 9 and 9:30 AM.

A Strange and Clandestine Group

An article by managing editor Joel Miller in Homebrew Computer Club’s Newsletter has disclosed the existence of a new and potentially dangerous group of underground hackers. At least part of their covert activity centers around the publication of The Pit. The text of Mr Miller’s article reads as follows:

A few days ago, a mysteriously unsigned envelope was placed on my desk in the Fairchild Publications Department and, to my delight, it contained the last five issues of an informal underground microprocessor newsletter called the Effete Pit, “a journal for and by a small underground band of F8 users who can’t afford a Formulator.” An introductory statement informs us that “the chief users...are engineers working for a major semiconductor company...” and that “you can’t even get on the mailing list except by asking around in the right places.” The issues contained a number of very interesting applications-oriented articles by such nefarious authors as “The Mole,” “Captain Midnight,” “The Scrounger” and “Poto.”

After a little detective work con-
STATE OF THE ART

BOTH FORMS

There are two forms of the "state of the art." One form is the personal growth attained by most professionals who realize they must stay in step, intellectually, with new concepts and new techniques. Too often, a tragedy occurs when the professional neglects the second form, his career development. The fatal mistake occurs when working in an environment that provides a continuous parallel to industry but neglects to provide the professional growth that is necessary to insure career development and avoid potential future frustrations. To insulate yourself against this happenstance, check with our professional staff. They will advise you on your career development as it relates to your technical development.

Our areas of specialization are:

- Software Development
- Computer Graphics
- Data Base
- Computer Design
- Simulation and Modeling
- Telecommunications
- Hardware/Software Interface

For further information either call or send resume to:

PERRI-WHITE ASSOCIATES
50 FRANKLIN STREET
BOSTON, MASS. 02110
(617) 423-1900

PERRI-WHITE & ASSOCIATES, INC.
5373 W. ALABAMA PLACE
HOUSTON, TEXAS 77065
(713) 960-0350

Management Consultants
Specializing in Data Processing/Systems Engineering
All Replies Held In Strictest Confidence

Try our SOFT PRICES

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSP-40</td>
<td>$575</td>
</tr>
<tr>
<td>MP-40</td>
<td>$425</td>
</tr>
<tr>
<td>KP-40KIT</td>
<td>$179</td>
</tr>
</tbody>
</table>

SSP-40 connects to your Heathkit AX-900 for easy connection to your RS-232 serial port or TTY current loop.
MP-40 interfaces to your microcomputer parallel output port for ASCII data transfer.
KP-40KIT contains microcomputer PC board, all components, transformer, and complete assembly instructions for parallel data transfer.

Need Hard Copy?

Send for FREE LITERATURE

Contact your numerous informants, snitches, and ratfinks scattered throughout the company, I finally made contact with The Mole, the newsletter's editor, who-despite my not knowing this week's password—was eager to talk. "Although F8 parts are easy to get here," says The Mole, "a lot of us are starting to fool around with other stuff, like the Z-80. (So we've) solved the problem of newsletter content by changing the name of the rag (to The Pit) and moving further underground to avoid creditors and others of a similar interfering bent."

The Pit has published articles such as "A Twelve IC TV Typewriter," "Parallel Handshake Interface From An ASCII Keyboard," "A Simple-But-Slow Analog Output For The F8," "Dynamic RAM For The F8," "Keypad Data Loader" among others. An especially interesting article by The Mole uses the F8 as a controller for surplus Diablo printer mechanisms that are now available on the hobby market.

Thanks for the story, Joel.

Should The Mole and his accomplices care to surface, BYTE would of course be happy to use its good offices to spread the word.

And Speaking of Homebrew...

Bay area hobbyists who haven't done so already should look into Homebrew, an inventive and well-established organization. Drop a line to Homebrew Computer Club, POB 626, Mountain View CA 94042 and find out more.

Nebraska Clubs

OMAHA, the Omaha Microprocessor Amateur Hobbist Association, has recently celebrated its first anniversary. Club accomplishments to date include the financing of an Altair 8800, an audio cassette interface and an E & L Instruments MMD1—a micro designed for educational and experimental purposes. The group currently meets twice a month at Northern Natural Gas Company in Omaha.

One of OMAHA's board members, Lt Tom Smith, has formed a club in Bellevue NE called MACH (no explanation of the acronym was given). The club meets the second Tuesday of the month in the Commercial Federal Bldg, Hwy 73-75 and Galvin Rd, Bellevue NE.

Interested persons should contact
OMAHA, c/o Rita Bianchi, S&DP—4th Floor, Northern Natural Gas, 2222 Dodge, Omaha NE 68102, or MACH, c/o Thomas Smith, 2708 Calhoun St, Bellevue NE 68005.

Portland Computer Society

Portland Computer Society has recently been in touch. Meetings are held on third Saturdays at 1:00 PM at Portland Community College. For details write to PCS's new address, 3763 SE Division, Portland OR 97202.

Houston Amateur Microcomputer Club

Obviously there is strong interest in personal computing in the Houston area. One reflection of this is the growth of the Houston Amateur Microcomputer Club. Membership is now over the 100 mark, and according to their newsletter Nybble, they are preparing to make a significant contribution both in Houston and nationally. Contact HAMC, c/o Troxel Ballou, 3842 Grennoch, Houston TX 77025, or call (713) 661-6806.

Philadelphia Area Computer Society

Like Houston and countless other clubs PACS has grown dramatically with the personal computing boom. Their newsletter, The Data Bus, has doubled in size and improved in quality over the past four months. Club activity centers around the sponsoring of the annual Trenton Computer Festival, an active group purchase effort and a major club project on 6502 applications.

Meetings are held on the third Saturday of the month at 2:00 PM at the LaSalle College Science Building, 20th and Olney St, Philadelphia; workshops, courses and subgroups meet prior to the club meeting, usually at 12:00.

You can write Philadelphia Area Computer Society at POB 1954, Philadelphia PA 19105, or call (215) 923-3299 or (215) 829-6745.

PET User Group

A user group has been formed for people interested in the Commodore PET 2001 computer. For those unfamiliar with the PET it is a compact computer with integrated keyboard, CRT with character and graphics capability, cassette drive, 14 K ROM operating system including full 8 K BASIC, and 4 K RAM user space.

The purpose of the PET user group will be to share and exchange applications,
programs and hardware expansion techniques, and to provide general user feedback. The first year membership is $5 and will include the User Notes publication. Contact Gene Beals, POB 371, Montgomeryville PA 18936.

"Name the Users' Group" Contest

A 5 year membership is the prize in a "name the users' group" contest sponsored by computer hobbyists exchanging information on the use of Heathkits. Contestants should of course avoid submitting entries which have been registered as trademarks, since these will not be considered.

Entries and requests for further information should be sent to Charles A Floto, 267 Willow St, Apt 23, New Haven CT 06511.

Southern Nevada

Southern Nevada Personal Computing Society was formed in June of this year and as of this writing (July) was still in the embryonic stage with meeting times and places still unresolved. Membership is open to Clark county residents and nonresident students of Clark County educational institutions. More information should be available by the time this issue of BYTE reaches its readers. To learn more about SNPCS write 1405 Lucilee St, Las Vegas NV 89101, or call (702) 642-0212.

Hex Users

A new Hex Users Club has been announced for systems with 16 to 24 keys, EPA, Motorola Evaluation Kit II, restricted to 6800 based systems, for now. For more information write to Charles C Worstell, 36012 Military Rd S, Auburn WA 98002, enclosing a stamped, self-addressed envelope, or phone (206) 927-6038.

Inland Empire Computer Club—Spokane WA

The Inland Empire Computer Club is a 50 member organization. Meetings are held at 7:30 each fourth Wednesday at the Washington Trust Bank at 3830 E Trent. To find out more about the Inland Empire Computer Clubwrite POB 1434, Spokane WA 99210.

IBM 5100 Users Group

Those interested in contacting an IBM 1500 Users Group are invited to write 5541 Parliament Dr, Suite 104, Virginia Beach VA 23462, or call (804) 490-0154.
The memory management technique of LIL is to be as shown in figure 1.

Objects in a program may encompass the following: real simple variables, integer simple variables, homogeneous arrays (real or integer), strings, numerical constants, string constants, subprograms, functions, parameters, pointers and labels.

Numerical representation has been chosen so as to be as flexible as possible. Integer values are signed magnitude binary; real values have a binary coded decimal (BCD) mantissa with excess 64 binary exponent, a hybrid form which lends itself to the goal of memory conservation.

At present, the only segment of LIL which is actually implemented (on an MCS 6502-based KIM) is the arithmetic package, a mixed mode, variable word length package of great power and flexibility. Integer values may be from one to seven bytes in length; real values may be from two to 14 bytes in length. These word lengths may be curtailed to smaller values as LIL is actually implemented; much depends upon the demands made by language features for space in zero page.

As Mr Stavely suggests, LIL is being taken "one step at a time." However, it has been found that some of the more complex control features planned for LIL make it necessary to look ahead and allow for their implementation without "starting from scratch" with each new feature. LIL is to have control features which lend themselves to the concept of structured programming.

Object code, along with its accompanying reference table, is relocatable to any area of memory. Subprograms and functions may be compiled separately and used with different main programs. LIL has a COMMON statement through which global objects may be linked to a subprogram or function. All linkages in LIL are to take place through descriptor addresses; this implies execution time type checking for all operations. With the exception of labels, any global object may appear in a COMMON statement, including constants. Thus, heavily used objects may be included in the main program's COMMON statement and be repeated in a COMMON statement at the

Constant voltage power supplies by Parasitic Engineering.

Give your Altair, IMSAI or custom system the power it needs to keep your programs up and running through brownouts and power fluctuations. The famous Parasitic Engineering Constant Voltage Power Supplies deliver full output with line voltages as low as 90 volts or as high as 140 volts. 12-amp kit designed for Altair 8800-A, $90 postpaid. Universal 20-amp kit, $200 postpaid. (Cal. res. add tax.)

Parasitic Engineering

Constant Voltage Power Supplies deliver full output with line voltages as low as 90 volts or as high as 140 volts. 12-amp kit designed for Altair 8800-A, $90 postpaid. Universal 20-amp kit, $200 postpaid. (Cal. res. add tax.)

Tarbell Floppy Disc Interface Described for Hobbyists and System Developers

• Plugs directly into your IMSAI or ALT AIR* and handles up to 4 standard single drives in daisy-chain.
• Operates at standard 250K bits per second on normal disc format capacity of 243K bytes.
• Works with modified CP/M Operating System and BASIC-E Compiler.
• Hardware includes 4 extra IC slots, built-in phantom bootstrap and on-board crystal clock. Uses WD 1771 LSI Chip.
• 6-month warranty and extensive documentation.
• PRICE: Kit $190...Assembled $265

* ALTAIR is a trademark/tradename of MITS, INC.

Tarbell Electronics 20620 South Leopold Avenue, Suite P Carson, California 90746 (213) 538-4281

Circle 418 on inquiry card.
finally, a telephone with byte!

6800 automatic telephone dialer program ................ $9.95 postpaid
Have your 6800 system dial your phone • Uses only 5 external components • Stores 650 variable length phone numbers • Operates in less than 1K bytes of memory
Includes: Paper tape in Mikbug format and object code • Circuit diagram and instructions • Instructions for adapting to other 6800 systems

6800 telephone answering device program ........... $4.95 postpaid
Have your 6800 system answer your phone and record messages automatically. Compatible with any 6800 system.
Includes: Assembly listing and object code • Circuit diagram and instructions

Write to: SOFTWARE EXCHANGE
2681 PETERBORO
W. BLOOMFIELD, MICH. 48033

Mikbug is a registered trademark of Motorola Inc.

beginning of each subprogram or function. The COMMON statement of LIL is to cost zero additional bytes, no matter how lengthy the statement, with a maximum of 256 common objects.

The concept of which Mr Stavely speaks seems to be ideal for the limited resources of the microcomputer. This is the software simulated compiler, much in the same manner that all arithmetic operations more complex than single byte addition or subtraction must be software simulated.

As Mr Stavely rightly points out, the object code of programs may be shared between owners of different brand microcomputers; all that is necessary is that each person have a loader and interpreter which defines the proper "abstract machine." This is an exciting concept which bodes well for those microcomputer owners who wish to establish a shared program network, and interchange ideas about the abstract machine.

LIL is simply one version of the abstract machine. However, the concept of control bytes in the code seems to be a very general idea which would be useful for any abstract machine. A control byte is to appear as the first item of executable code (discounting a few overhead bytes such as 'number of bytes in code'), and control bytes are to appear after each 8 unit subdivision of code, or after each END OF STATEMENT, whichever comes first. The interpreter is to pick up control bytes "on the fly" as it executes the code. A copy of the control byte is rotated left each time the program counter is incremented; thus, by examining bit 7 of the copy of the control byte, the interpreter can determine whether the next unit of code is a 2 byte address or a 1 byte offset. In LIL, END OF STATEMENT is a routine which checks for certain valid conditions, then turns control back over to the program monitor which loads the next control byte and proceeds merrily upon its way. Upon encountering code such as WHILE . . ., the program counter and current state of the control byte are pushed onto the hardware stack for preservation; by later popping this data off the hardware stack and restoring the values of the program counter and control byte, execution is returned to WHILE . . . .

Mr Stavely's article contains a most interesting concept which holds the promise of truly high level programming languages for the microcomputer, as well as portability of code from one brand of machine to another.

Please let us see more such stimulating discussion of software concepts in the pages of BYTE.
by Rodnay Zaks. Ref. C201

THE BOOK (MICROPROCESSORS: from chips to systems)

This book is based on the author's experience in teaching microprocessors to more than 2000 persons during the past 5 years, both in a professional, and in a university environment. The seminars have received the highest ratings for their educational and technical value, and have been taught worldwide.

This step by step book stresses:
1. How to assemble a complete system
2. Design trade-offs: advantages, disadvantages, problems, costs.

About 250 pages, over 70 illustrations. Available August 1977

CONTENTS
A comprehensive introduction to all the essential aspects of microprocessors and microcomputers:

THE AUTHOR
Dr. Rodnay Zaks has been responsible for the design of industrial microprocessor systems since their inception in 1972. He is the author of 11 educational books in the field and more than 20 scientific publications.

CASSETTES
The 2 basic microprocessor courses are now available on cassettes. Each course includes 2 cassettes (2½ hours) and a special book:
S1 — An Introduction to Microprocessors ............................................... $29.95
S2 — Programming Microprocessors ....................................................... $29.95

OTHER BOOKS
- MICROPROCESSOR ENCYCLOPEDIA
  2 complete reference books, containing manufacturers' data and description of operation for every microprocessor. A necessary tool for any comparative evaluation.
  E8 - 8 BITS MICROPROCESSORS: Am 9080, AMI 6800, EA 9002, Fairchild F8, GI LP 8000, Intel 8080 A, Intersil 6100, Mos Tech 650 X, Mostek 3880, 3850, M 6800, NS 8080 A, SC/MP, RCA CDP 1802, PPS-8, Signetics 2650, TMS 8080, Western Digital MCP 1600, Zilog Z-80. ....................................................... $29.95
  E5 - BIT SLICES: AMD 2901, Fairchild Macrologic, Intel 3000, MMI 5701, Motorola M 2900, M 10800, NS IMP, Signetics 8X02, 3000, TI S8P 0400. ............................................... $18.00
- IMD — International Microprocessor Dictionary
  10 Languages: English — German — French — Spanish — Italian — Swedish — Norwegian — Danish — Hungarian — Polish. 30 Pages. ............................................... $1.95
- ACR-Microcomputer Acronyms — 4-Page guide to abbreviations ................ $1.00
- SEMINAR BOOKS (condensed notes)
  B2 - Microcomputer programming ...................................................... $39.95
  B3 - Military microprocessor systems ............................................... $39.95
  B5 - Bit Slic e ................................................................. $39.95
  B6 - Industrial microprocessor systems ............................................... $39.95
  B7 - Microprocessor interfacing techniques ........................................ $39.95

IN-HOUSE TRAINING
SYBEX is independent from manufacturers and provides in-house training and education on all the above topics, as well as on specialized microprocessor techniques.

AVAILABLE SOON:
Microprocessor programming techniques (with 8080 and 6800 examples) ............................................... $7.95

ORDER FORM

Name
Position
Company
Address

□ Send me

□ check □ P.O. □ money order □ other

copies of. = $
copies of. = 
copies of. =
Subtotal

In California, add tax +

Postage and Handling

TOTAL

Payment included:

Handing (fourth-class rate): no charge. Faster delivery: $1.40 one book, $0.90 each additional one. ($2.50 overseas)
1. Fill attached coupon, or send P.O.
2. Enclose payment (prices are for U.S. only). In California, add tax.
3. Mail to SYBEX Inc, Publications Dept, 2161 Shattuck Ave, Berkeley, Ca 94704, Tel: (415) 848-8233.
Telex: 336311

In Europe: SYBEX - Europe, 313 rue Lecourbe, 75015 Paris, France
Tel: (1)8265202, Telex: 2008858

Circle 429 on inquiry card.
WHO HAS THE MOST STORAGE?

I read your April 1977 article, "A Software Controlled 1200 bps Audio Tape Interface," page 40, and found it extremely refreshing to find magazine people really practicing low cost hardware methods with "nice" software. It really surprised me to see a machine state diagram. However, the most interesting comment is that about a 3M tape controller and drive. I am not really interested in working the electronics out from scratch as you are. But, like so many of your readers, I am very much interested in having a 9 track NRZI peripheral for my machine. My main interest is the huge software availability from "big machines." For me, this software is accessible to run on my machine. I certainly would be interested in constructing the interface hardware for a 9 track drive and would like to see some information on that subject. My machine at present is a Z-80 with 128 K of memory. Soon, like now that the price is right, fast fixed point multiply and divide is available and the microprocessor is now a decent minicomputer in performance. The next thing for me logically is multiprocessing when I add the other 128 K bytes.

R Plemich
307 Hatten
Mt Prospect IL 60056

With a main memory region of that size, you need at least a 9 track drive or a multiplexer hard surface disk. Here is an interesting and informal contest: let each reader who reads this note send in a postcard giving the amount of memory presently integrated (i.e.: plugged in, checked out and used as a programming resource) in his or her system. Give a short description of the system in use as well as the type of memory technology broken down by regions in address space. Address your cards to BYTE Memory Census, 70 Main St, Peterborough NH 03458.

10,000,000,000 BITS

Could you tell me when I might expect to be able to purchase a Philips/MCA optical video disk system with write and read capabilities suitable for use with a microcomputer?

Jack M Mann Jr
105 Yosemite Dr
Ontario CA 93030

Rare circumstance falls in your favor. It happens that I sat next to Dr Jerry S Sullivan of Philips Laboratories on a plane flight from New York City to San Francisco April 19. We were both going to Monterey CA to attend the 1977 IEEE Asilomar workshop on microprocessors. One of the topics of the wide-ranging conversations was the application of video disk technology in a meaningful way to personal computing. The excitement is there: if a writing method could be achieved we'd have a media cost of $5 to $10 per disk with perhaps 10 billion (10^10) bits. Jerry could not predict an exact date of course, but Philips is indeed working on the concept of a field programmable digital version of the disk. It is conceivable that such a device might be ready within the next five years. The Philips/MCA video disk which is supposed to be on the market sometime this year is a read only device which will only be available programmed with movies and other visual materials. Functionally a digital version of the disk is similar to a fusible link read only memory or an ASR 33 paper tape with rubout: you can only write the information once. But with 10^10 bits per disk and inexpensive disks, there is plenty of space to allow rewriting of files and updating by overwriting with the logical equivalent of "rubouts." Cost is an open question at this time, of course... CH

TAKING STOCK

I want a small computer. I need it for stock market research. I can wangle the money. I'm satisfied I can learn to program. But how can I get stock market data into my memory? Are there services that provide stock market data for that purpose? Do you buy it on tapes? On disks? Can you tie your own computer into someone else's data book through a telephone connection?

Shouldn't a greenhorn locate his data source first before he can decide what type of computer will be best for him? Why don't I ever see any ads for data sources?

Charles M Larson
Investment Consultant
11513 Havenwood Dr
Whittier CA 90606

It is a good point that one should tie down the data source before assuming one can use it. Without any knowledge of the charges and contractual arrangements, I do observe that the stock brokerage office which is located in our office building in Peterborough uses

CANADIANS!
Eliminate the Customs Hassles.
Save Money and get Canadian Warranties on IMSAI and S-100 compatible products.
IMSAI 8080 KIT $ 838.00
ASS. $1163.00
AUTHORIZED DEALER
Send $1.00 for complete IMSAI Catalog.
We will develop complete application systems.
Contact us for further information.

Rotundra
Cybernetics
Box 1448, Calgary, Alta. T2P 2H9
Phone (403) 283-8076

Circle 430 on inquiry card.

COMPUTER MART
OF NEW HAMPSHIRE

DEALERS FOR
TDL ICOM IMSAI
DIGITAL GROUP
KIM-1

SOFTWARE INCLUDES:
8K BASIC EXT. BASIC
TEXT EDITOR DISK BASIC
WORD PROCESSOR
MACRO-ASSEMBLER

170 MAIN STREET NASHUA
(603) 883-2386

Circle 431 on inquiry card.

$ $$

IF YOU ARE GOING TO MAKE YOUR
FIRST PURCHASE IN EDI, YOU REALL Y
OUGHT TO YOURSELF TO GET IN ON
THE TREES OF THE EDI PROFESSIONAL
WHY WASTE VULNERABLE TIME LEARNING
THE TREES OF THE TRADE TOO HARD UP?
LEARN IN THE COMFORT OF YOUR OWN
WORKPLACE. ENJOY THE JOURNEY.
THESE PREMIUM HOW TO MANUALS EXHIBIT
TREES NEVER BEFORE AVAILABLE TO YOU-
ABSOU UTLY GUARANTEED ON YOUR MOST
SEND $6.95 PLUS $3.00 PRODUCTION AND
HANDBOOK TO EDI PROFESSIONAL DIV.
2846 DEFORD ST NW, ARLINGTON, VA 22205

Circle 432 on Inquiry card.
BARGAIN: IBM CARDS
$4.50 each

They once printed in beige on a navy T-shirt with an indelible pen to punch your own message! How about equations? Maxwell's, Legendre's, Cauchy, Schrodinger or Einstein. We have those on shirts, too. How about a flip-flop or an analog circuit, a Shakespeare sonnet or an H-R diagram, the periodic table, or a Chapin, Enright, Fitch, Brown, Chlorophyll molecular diagrams, the Krebs cycle, the Carbon-Nitrogen cycle, early Earth geology or Frog anatomy, the human circulatory system, a Feynman diagram or a structured flowchart, Sties, S.M.L and XL. Include 50 cents pst & hub, Circle reader card or write for free illustrated catalogue.

OUTER PRODUCTS Dept N Box 88
LAFFAYETTE HILL, PA. 19444

Circle 433 on inquiry card.

Circle 434 on inquiry card.

Circle 435 on inquiry card.

WASHINGTON, D.C.'S
Newest Computer Store

Intelligent Business Machines, Inc.

Cromemco, IMSAI, TDL, Administrator I, SEALS, DIP Chips, etc.

* * * * * * *

SWTPC Business Programs

Mesh Package, 5 byte mantissa, 1 byte exponent. 11% digit accuracy, Forsythe style formatting (integer, floating points and floating or both read and write) $25.00

Letter Writing Editor. No line numbers needed. Subset of Honeywell's 1484 Time Sharing Editor. $25.00

Payroll. Phone for details

Accounts Receivable. Phone for details.

Stock Club - Marketing Evaluation. This is the same program used by Custom Computer Services to provide accounting for clubs belonging to the National Association of Investment Clubs.

Disk Based SWTPC Business Systems. Phone for details.

Dealer for SWTPC, Smoke Signal Broadcasting, Permcom, Teledata 6, 7, 11, Centronics, Motorola, etc.

To Order: Include 3% postage. Illinois residents add 5% sales tax.

AAA Chicago Computer Center
3007 W. Wadsworth Avenue
Chicago, IL 60618
(312) 539-5833 (9 AM - 6 PM)

Circle 436 on inquiry card.

Circle 437 on inquiry card.

Circle 438 on inquiry card.
telecommunications equipment supplied by GTE Sylvania with a number of different video terminals and printers. It is all pretty standard communications equipment, although the size of the GTE equipment case in the hall (a full height 19 inch rack cabinet) suggests a dedicated minicomputer as the local satellite of a network. Based on this observation, it should in principle be technologically feasible for you to hang onto a stock market wire and accumulate data via an RS-232 port and modem as it is generated day by day. There is also a whole host of stock market analysis firms, many of whom may offer some form of machine readable data as a by-product. Consult the advertisements in the Wall Street Journal or Barron's. If any readers can help by providing specifics in answer to Mr. Larson's request, please write so that we can make the data available. Once the source of data is better detailed, it is possible to make some more specific comments.

As far as you don't see any advertisements for data sources, perhaps it is because none thought of selling the data before. Perhaps your letter will help create a market.

WHERE DO I START?

I am currently a college student enrolled at a community college. I have several issues of your magazine and have been reading about all the small systems of computers. I am very interested in owning and operating a small computer system except for one thing: how much do I need to know to set one up? Specifically, how much digital electronics, electronic assembly, computer science and programming knowledge do I need? Assuming I need a lot of this type of knowledge, how long might it take me to master it so I can set up my system?

I intend to use my computer for games, mathematical modeling, digital to analog devices and simulation. I would love to set up a system to do all these things, but I don't want to wait until I have a bachelor's degree in computer science to do it. I would greatly appreciate any guidance you could give me on what kind of knowledge or experience I would need in order to set up a workable system.

John Graffio
1094 Quail Creek Rd
Fallbrook CA 92028

The minimum "set up" time for a personal computer, by far, is through purchase of an "appliance" computer which comes really to use, in the same manner as a stereo receiver comes ready to use. In this category are the latest calculators (such as the brilliant new SR-59 from Texas Instruments with its printer and ROM-software attachments), or true general purpose computers such as the Apple-II, Commodore PET 2001, or Naval 760. All of these come complete and assembled. If you want kits, those products range in complexity from simple peripherals to complete systems. Kits naturally take more time to get into operation and debug. Kits such as those Heathkit computer products assume no initial familiarity with electronics, and provide tutorial materials to get the user into operation.

CONVERSIONS?

As a newcomer to the field of computers, I am just beginning to understand the hardware, but my software problems still exist en masse. While thinking hexadecimal is nice, I have been brought up thinking in a base 10 numeral system (whatever that is). I know not whether this is an extremely simple and trivial or a vastly complex question, but tell me, how does one get all those rows of 1s and 0s into groups of four BCD bits? I can understand actually counting up to the number with a high frequency counter (on a 7490 BCD counter paralleled by eight 7490s in binary format driven by a 10 MHz clock), some complex software method that applies tables, or even successively subtracting out exponents of 10. But I figure some genius, somewhere, has an algorithm. Can you help me?

Whit Smith
606 Brookwood Ln
Goldensboro NC 27530

Conversion of a binary number to a decimal number was presented in some detail for the 8080 processor in an article entitled "How to Do A Number of Conversions" by James Brown on page 50 of our September 1976 issue of BYTE. There is no magical short cut with most processors: you compare the integer successively with values of 10 raised to various powers (stated in a table of binary numbers) and derive each digit of the converted BCD number from the binary by successive subtraction of the appropriate power of 10 (or division if you have such an processor). Conversion of the BCD form to an external ASCII representation is easier, and can be done by adding the hexadecimal constant 30 to each 4 bit BCD digit value, giving an 8 bit ASCII value from hexadecimal 30 to 39, which prints as the character values 0 to 9 respectively. Converting BCD inputs or ASCII inputs to binary internal forms is fairly simple and after each successive digit entered, multiply the previous entry sum by 10 (add the old value to the sum ten times for a less than optimal brute force technique), then add the new entry (carefully limited to values from 0 to 9) to the previous entry sum giving the new entry sum.
BLOS and ZAPS

BLOS (rhymes with gloss) and ZAPS are two operating systems announced by Algorithmics for the 8080 and the Z-80 respectively. They are directly loadable on Digital Group systems via cassette. BLOS requires 12 K and ZAPS, 13 K of memory. ZAPS assembles all Z-80 instructions; BLOS assembles only instructions executable by the 8080. BLOS and ZAPS are identical otherwise. Programs written in the 8080 subset are transferable from one system to the other.

The text editor within BLOS and ZAPS is much like those used on time-sharing systems. Because it is a full context editor, no line numbers are required. You may edit any of up to 16 files in memory. Only the actual data bytes are stored. Editor commands include Input, insert, delete, replace, change, global change, up, down, top, bottom, string search, print and print current line number.

The assemblers feature user-defined commands, switchable number bases, and a full set of cassette operations.

Contact Algorithmics Inc, POB 56, Newton Upper Falls MA 02164.

A Microprocessor Development Kit

Aivex Inc has a new product that should be of interest to those who work with the PDP-11 computer: the MAX-11 conversion kit. The package, which converts a PDP-11 into a microprocessor development system, consists of a cross assembler, simulator and PROM programmer. The cross assembler is said to feature a user defined macro library, local symbols, symbolic cross reference, listing controls and conditional assembly. The simulator features eight break points, single step with trace, external device simulation, interrupt simulation and real time cycle counter. The programmer accepts data from a PDP-11 disk through an RS-232C Interface. The kit is presently available in 4040, 8080 and 6800 versions. The price for the simulator plus assembler is $1250; the programmer is $2300.

Contact Aivex Inc, 6 Preston Ct, Bedford MA 01730, (617) 275-2333.

A Hardware and Software Catalog

TDL's new catalog features a variety of hardware and software products, including the Z-16 memory module, a 16 K byte static programmable memory card. Maximum access time is 200 ns, and the kit price is $574. Also described is the XITAN series of mainframes and the Zapp monitor and text editor. A relocating macroassembler is offered along with several BASIC packages and disk software.

For a free copy of the catalog, contact Technical Design Labs Inc, Research Park, Bldg H, 1101 State Rd, Princeton NJ 08540 (609) 921-0321.

An In-Depth Software Package

Sublogic Company of Culver City CA has announced 3D graphics for microcomputers. The Sublogic 3D micrographics package allows the user to view two-dimensional perspective projections of three-dimensional scenes from any location in space. Driving and flying simulations, artistic projections, design projections, engineering analysis and advanced games are some of the applications.

Two versions of the graphics package are offered: a minimal subset BASIC version for general purpose slow speed graphics on any microcomputer system, and a 6800 assembly language version with dynamic graphic capabilities for advanced simulation and complex graphics.

Adaptation instructions, program listings, applications, interface and testing information are supplied with each package. The BASIC version costs $22; the 6800 package is priced slightly higher. Dealer inquiries are welcome. For further information contact Sublogic, POB 3442, Culver City CA 90230.

LSM Engineering's New Text Editor and Operating System

Need a text editor or an operating system for your computer? LSM Engineering, POB 3243, Orange CA 92665, now offers both of these items.

EDIT is a text editor designed to edit source files for BASIC, FORTRAN, assembly language, etc, which utilize the standard 96 character ASCII set. Source files can be manipulated by character, string, line and page to create additions, insertions, replacements and deletions to text.

LSM's Small Operating System, called SOS, has commands for saving, loading and verifying data blocks in conjunction with a Tarbell cassette interface, as well as memory fill, memory block transfer, block verify and memory dump routines. SOS resides in 2 K bytes of read only memory and is supplied with a commented source listing and information to enable the user to patch in up to four new keyboard commands without reassembly.

EDIT is available for $22.50 on Tarbell block form cassette or paper tape, and SOS costs $15 on Tarbell block form cassette.

LSM Engineering, Research Park, Bldg H, 1101 State Rd, Princeton NJ 08540 (609) 921-0321.
16K E-PROM CARD
S-100 (IMSA1/ALTAIR) BUSS COMPATIBLE

Imagine having 16K of software on line at all time!

Kit Features:
1. Double sided PC Board with solder mask and silk screen and gold plated contact fingers.
2. Selectable wait states.
3. All address lines and data lines buffered!
4. All sockets included.
5. On card regulators.

Kit includes all parts and sockets! (Except 2708’s)

Add $25 for assembled and tested

16K E-PROM CARD $69.95 (Kit)

Special Offer: Our 2708’s (650 NS) are $12.95 when purchased with above kit.

DEALER INQUIRES INVITED

2708 S-100

1KX8 2708

Prime new units from a major U.S. mfg. 650 N.S. access time. Equivalent to four 1702A’s in one package!

Going into business sale! $15.75 each

8K LOW POWER RAM KIT
(S-100 BUSS COMPATIBLE)

The last word in RAM Kits. Uses 21L02-1 RAM’s. All address and data lines fully buffered. PC Board is solder masked and silk screened and has gold plated contacts. Four regulators with heatsinks included.

Add $30 for assembled and tested

2708 $149.00

(Kit of all parts and sockets)

741C OP AMPS
MINI DIP. Prime new units. Has computer MFG’s house number.
12 FOR $2
100 FOR $15

TALUM CAPACITOR
1 MFD. 35 V. Kemet. Axial Lead. Best Value.
10 FOR $1

EDGE CONNECTOR — $1.50

TERMS: ORDERS UNDER $15 ADD $.75. NO C.O.D. WE ACCEPT VISA AND MASTER CHARGE CARDS. MONEY BACK GUARANTEE ON ALL ITEMS. TEXAS RESIDENTS ADD 5% SALES TAX.
A New Plotter for Microcomputers

Axiom Corp has introduced a compact graphic plotter, called the EX-810, which is designed to operate with microcomputers. The unit prints 8,192 dots per second with up to 512 dots per row. It can function as an 80 column alphanumeric printer with a speed of 160 characters per second.

The EX-810 is designed to be driven by an 8 bit microprocessor and is equipped with a TTL compatible controller which takes care of all the internal timing functions necessary to drive the printhead and advance the paper. Printout is initiated by a single input command which causes the eight track non-impact printhead to move across the printing field in approximately 240 ms. Margin and printhead position marker signals are fed back to the microcomputer and may be used to synchronize an eight bit scanning raster to the printhead driver input. The user can vary the horizontal dot resolution by generating an external timing cycle for the input raster.

The EX-810 is a self-contained unit with case power supply, paper holder, infrared low power detector and bell. The dimensions are 9.625 by 4 by 10.875 inches (24.45 by 10.16 by 27.62 cm), and 5 inch (12.7 cm) wide electrosensitive paper is used for printing. Applications include printing forms, tickets, logos, maps, pictures and charts.

The price is $795 in single quantities. OEM discounts are available.

Contact Simon Harrison, vice president of marketing, Axiom Corporation, 5932 San Fernando Rd, Glendale CA 91202. (213) 245-9244.

Circle 604 on inquiry card.

The Signal Laboratories Inc Programmable Gain Amplifier card (PGA) is a Zilog Z80 MCB compatible accessory offering two channels of computer controlled amplification. Each channel has filtered and unfiltered outputs. Under control of the Z80 MCB, the gain of each channel may vary from unity (0 db) thru 54 db (70 db optional) in 2 db increments, with gain accuracy of 0.1% (0.01% optional). Similarly, the Z80 MCB may select any one of eight user-selectable bandwidths for each channel's filtered output. Unfiltered outputs are 3 db down at 50 KHz. The Programmable Gain Amplifier (PGA) is one of a family of Zilog Z80 MCB compatible accessory cards available from Signal Laboratories Inc, 202 N State College Blvd, Orange CA 92668. Price for quantities under ten is $395, available from stock.

Circle 605 on inquiry card.

A State(ment) of Flux

There are many sources of information available to the experimenter which teach the art of soldering, but few of these sources give a detailed explanation of the metallurgy and physics involved in the process. M W Dutton Company, Consumer Products Division, 350 Kinley Av, POB 6205, Providence RI 02940, has bridged the solder gap with a well-written brochure entitled "Dutton's Nokorode Soldering Guide." Topics covered include the composition of solder, solder alloy phase diagrams, mechanical properties of solder and a discussion of solder flux. The brochure is free for the asking.

Circle 606 on inquiry card.

EPA Micro-68a Computer Now Available in Kit Form

Electronic Product Associates Inc, 1157 Vega St, San Diego CA 92110, has announced the Micro-68 computer kit for $385, which includes power supply and cabinet.

This computer kit comes complete with 16 key hexadecimal keyboard and 6 character hexadecimal display. It uses the Motorola 6800 processor design. Sockets are provided for 768 bytes of volatile programmable memory (128 words supplied with kit). The MON-1 keyboard operating system is supplied in PROM form so that the computer is ready to run upon completion of construction. Commands include inspect and change, load user's program, run user's program, insert break points and save stack. Sixteen bits of IO are provided to the side connector, while the main bus is accessible at the front connector. Full bus buffering provisions are provided on the board. A piggyback Teletype/video display/audio cassette adapter is available.

The Micro-68a is fully compatible with the Micro-68b for later upgrading into a larger system and is available from stock.

Circle 607 on inquiry card.

National's New 16 Bit Microcomputer: the IMP-16L

Anyone looking for an alternative to the many existing 8 bit word microcomputers will be interested in National Semiconductor's new 16 bit microcomputer called the IMP-16L. The unit has a high speed asynchronous bus and a data controller that allows direct memory access (DMA) data transfer rates of up to 16 million bps. It comes complete with 4 K by 16 bits of programmable memory expandable to 64 K by 16 bits. A 20 mA interface is also provided.

There are 60 general purpose instructions, including multiply, divide, double precision add and subtract, and numerous bit oriented instructions. A control read only memory is also provided in two different versions to speed calculations.

The IMP-16L has four general purpose registers and a 16 word last-in-first-out stack. IO lines include eight general purpose flags, one general interrupt, one vectored interrupt, and four general purpose jump condition inputs. Cycle time is 1.4 ìs. Typical register to register addition time is 4.9 ìs, and memory to memory addition time is 8.4 ìs. One direct memory access transfer takes 1.05 ìs.

Software support includes diagnostic routines, a resident assembler, plus a variety of cross-assemblers. The system's architecture permits the installation of up to three central processing units (CPUs) for multiprocessing. The unit costs $825 and up, depending on memory size and options, and is available either from distributors or directly from the manufacturer. For more information, write to National Semiconductor, 2900 Semiconductor Dr, Santa Clara CA 95031.

Circle 606 on inquiry card.
Features, service, price!

...and a selection of S-100 compatible kits to match.

8Kx8 Econorom II™

3 kits - 24K of memory! $450

Configured as two separate 4K blocks for maximum flexibility. Individual protect for each block; provides interrupt if write attempt is made into protected block. Buffered data in, address in, and outputs. Tri-state outputs. Guaranteed under 1.5A and faster than 450 ns. Use with 1 wait state (logic implemented on board) with 4 MHz I-Z-80. Writes on either PWR or WR, your choice. Low power Schottky support ICs. Lots of bypass caps. Connectors for all ICs. Legended, solder masked double-sided board. ALSO: 8K ECONORAM II, assembled, tested, warranted for 1 year......$388.50

4K ECONORAM in kit form......$100.00

4K ECONORAM, assembled, tested, warranted for 1 year......$120.00

Active Terminator $29.50

Plugs into any S-100 Motherboard whose bus lacks active terminations. Cleans up noise, crosstalk, overshoot, and other bus problems that can scramble data unpredictably.

We also distribute these other fine products:

The WunderBuss™

This 20 slot motherboard, that includes all edge connectors, is designed to function in even the most adverse electrical environments. Includes active termination circuitry and double-sided board. $154

The Speakeasy

More than a cassette interface: handles 3 cassette 1/0 channels, but also has RS-232/teletype interface, 8 bit parallel port with handshaking signals...and includes integral RAM and ROM to drive it all......$120

Morrow Front Panel

This integral front panel/CPU card offers maximum debugging and programming flexibility, thanks to the ability to examine and alter memory, 1/0 ports, and more while your program is running......$250

MotherBoard $90

...also 18 SLOTS $124

Includes all edge connectors, plus active terminations to minimize crosstalk, noise, overshoot, and ringing that may be present with unterminated buses. Excellent for stand-alone system, or add on to existing systems.

8080 Software Board $265

Guts you away from talking to your 8080 in machine language by giving editor, assembler, and monitor routines programmed in EROM. Low power.

Unprogrammed EROM boards are also available; use for custom routines. Do the programming yourself or take advantage of our programming services.

2K (smaller) ECONORAM ..... $195.00

4K (basic) ECONORAM ..... $250.00

8K (larger) ECONORAM ..... $350.00

Mullen Extender Board $35

Allows you to troubleshoot and inspect boards outside of the system in question. Integral logic probe uses 3 different colored LEDs to indicate hi, lo, and pulse. Special edge connector makes for easy clip lead probing & lead identification. Non-skid probe.

Godbout Electronics

Box 2355, Oakland, CA 94614

One of the reasons we can offer these kits for such reasonable prices is that we do a big business in parts...far too many parts, in fact, to list in an ad this size. But if you are looking for a source of DB-25 connectors, edge connectors, and more...send for our flyer.

Circle 456 on inquiry card.
NEW EPROM PROGRAMMER

for 2708's and 2716's

The new low cost PP-2708/16 PROM programmer from OAE is the only programmer all in these features:

• No complex interface to wire, just plug the programmer into a 2708 memory socket and clip one wire to the "wait" bus!
• Driving software is short and simple!
• 2 programmers for the price of 1
• No turn cermet trimmers for precision voltage and pulse width adjustment
• 2708s and 2716's
• Driving software is short and simple!
• 2 programmers for the price of 1

Programs both the popular 2708 (1024x8) and the new TMS 2716 (2048x8) EPROMs.

The OAE PP-2708/16 module turns your computer into a powerful full feature PROM programmer. OAE's exclusive interfacing technique makes it a snap: simply plug the PP-2708/16 into a 2708 read only socket. A short software routine sends data over the address lines to program the PROM.

SAVE $50.00. For a limited time we are selling the assembled, tested and aligned unit at the kit price! Only $249.00!

1Let us know whose PROMs you are using and we will align our programmer for optimum date retention and longest PROM life.
2Also available is our Model PP-2716 programmer for Intel's unique 2716 EPROM. Same low price.

Includes $3.50 for domestic shipping and handling. California residents include 6% tax.

NEW LOCATION!

OAE
Oliver Audio Engineering, Inc.
676 West Wilson Avenue
Glendale, CA 91203.
(213) 240-0080

Circle 457 on inquiry card.
### MICROCOMPUTER

#### SUPPORT DEVICES
- 8212: 4.00
- 8214: 12.95
- 8216: 5.25
- 8224: 6.00
- 8226: 9.25
- 8238: 8.20
- 8251: 12.00
- 8253: 28.00
- 8255: 12.00
- 8257: 22.00
- 8259: 22.00

#### 8200 SUPPORT
- 6810P: 6.00
- 6820P: 8.00
- 6828P: 9.60
- 6850P: 12.00
- 6880P: 17.00
- 6886P: 15.00
- 6860P: 18.00
- 6860P: 2.70

#### Z80 SUPPORT DEVICES
- 3881: 15.95
- 3891: 14.95
- 3852: 14.95

#### FLOPPY DISC CONTROLLER
- PD372D: 65.00
- 1771: 69.95

### DYNAMIC RAMS
- 4140 (16P): 5.50
- 10103 (16P): 1.50
- 2104 (16P): 6.50
- 2107B (22P): 4.50
- 2107B-4 (22P): 4.00
- TMS4050 (18P): 4.50
- TMS4060 (22P): 4.50
- 4096 (16P): 5.50
- MM5262 (22P): 3.00
- MM2570 (18P): 5.00
- MM2580 (22P): 6.00

#### STATIC RAMS
- 31L01: 2.00
- 91L11A: 4.25
- 91L12A: 4.25
- 1101A: 4.25
- 2101: 3.00
- 2102 (10S): 1.25
- 2102-1 (500NS): 1.50
- 2M1A-4: 4.45
- 2112A-4: 3.00
- 25018: 1.45
- 3107: 2.95

#### DYNAMIC RAMS (Continued)
- 4102 (200NS): 11.95
- 4984: 20.00
- 5101: 20.00
- 74C89: 3.00
- 74S201: 4.75
- 91102A: 2.00
- 7489: 2.25
- 8225: 1.50
- 8599: 1.50
- 82509: 9.00

#### STATIC RAMS (Continued)
- *Limited supply.

#### FİFO
- 5213A: 6.75
- 2812-D: 11.95

#### TV GAME CHIPS
- AMS1955 (6 Games): 10.95
- AY-5-3600 (6 Games): 10.95

#### PROMS
- 1702A: 5.00
- 1702AL: 7.00
- 2704: 20.00
- 2708: 24.00
- 2716: 75.00
- 3601: 4.50
- 5203AO: 7.00

#### CHARACTER GENERATORS
- B59: 1.50
- MCM6571: 10.80
- MCM6571A: 10.80
- MCM6572: 10.80
- MCM6581: 8.75

#### MICROSHELFER COMPONENTS
- F-8: 19.95
- Z-80: 36.95
- Z-80A: 49.95
- CDP1802DC: 29.50
- MDS9520: 22.95
- 6502: 24.95
- 6800: 24.95
- 8008-1: 8.75
- 8008A: 15.95
- 8008B: 16.95

### CUSTOM SHOP

#### S-100 COMPATIBLE

#### JADE Z80 KIT
- WITH PROVISIONS FOR ONBOARD 2708 AND POWER ON JUMP
- $135.00 EA.

### IMSAI/ALTAR

### ELECTRONICS FOR THE HOBBIEST AND EXPERIMENTER

**Circle 458 on inquiry card.**
**HEAT SINKS**

680-76A $2.50

680-76B $1.88

341-6 Black Aluminized Aluminum for T0-3...$2.35

BIT-6 Black Aluminized Aluminized Editions.

BLK 10-3...=25 KIN 4...$2.35

**DIGITAL SWITCHES**

SPST Switches

2-State (4 pin diode) 4 switch unit...$1.76

2-State (8 pin diode) 6 switch unit...$2.25

**SLIT-N-WRAP WIRE WRAP TOOL**

3600...$5.95 ea

Slits and opens insulation exposing wires...$2.40

**SLIT-N-WRAP WIRE WRAP TOOL**

3602...$6.65 ea

6 5/8 x 5 1/4 x 1 1/2 split ring class...$2.05 ea

2244...$10.95 ea

Universal Microcircuit Pinheader - 24 x 1 1/2 x 1 1/2...$10.95 ea

**1/16 VECTOR BOARD**

1/16 VECTOR BOARD - 1000 dots...$1.21

8000 dots...$2.45

1/16 VECTOR BOARD - 16000 dots...$6.29

**HEXDECIMAL ENCODER 1-KEY PAD**

1...$0.00

2...$0.00

3...$0.00

4...$0.00

5...$0.00

6...$0.00

7...$0.00

8...$0.00

9...$0.00

0...$0.00

A...$0.00

B...$0.00

C...$0.00

D...$0.00

E...$0.00

F...$0.00

**S3 KEY KEYBOARD**

$24.95 ea

**TOOLS**

ARM5...89000 5000 1000 100 MHz...$9.50

A1D5...89000 1000 1000 100 MHz...$9.50

T-5...89000 5000 5000 5000 MHz...$9.50

T-3...89000 3000 3000 3000 MHz...$9.50

CS-5...89000 5000 5000 5000 MHz...$9.50

**PERMIGL**

P-20 PLUS...89000 2000 2000 2000 MHz...$9.50

Circle 459 on inquiry card.
IMSAl Offers New Microcomputer with One Megabyte Memory Capacity

The new IMSAl 8080 Megabyte Micro is one manufacturer's answer to the ever increasing need for more programmable memory space in microcomputers. The unit's chassis has the capacity for one million bytes of programmable memory. The problem of addressing this much memory is solved by using a total of 20 address lines. Memory modules may be purchased in units of 16 K, 32 K or 64 K bytes of dynamic programmable memory.

The heart of the memory control logic is the Intelligent Memory Manager/Interrupt (IMM) board, which provides write protect for each 1 K byte block of programmable memory, read protection, fully vectored interrupts, and time of day clock. Memory expansion is implemented by increasing the number of address lines from sixteen to twenty, and using a form of block switching to control the four added lines. The extended address space is divided into 64 16 K byte blocks of which four may be on-line at one time. The "switch" is implemented by defining a map which associates a state for address lines 16 through 19 with each state of address lines 12 through 15. The map defaults to 0000 out for all states of address lines 12 through 15 when the system is reset. It is modified and maintained under software control.

For more information, write to Michael Stone, IMSAl Manufacturing Corporation, 14860 Wicks Blvd, San Leandro CA 94577.

The ACI-33 is an audio cassette interface designed primarily for the Southwest Technical Products Corporation SwTPC 6800, the control interface and a terminal. The unit will also operate with any RS-232 terminal and computer serial IO which can supply +5 V and +12 V for the RS-232 interface. When used with the SwTPC, the ACI-33 supports all functions of the control interface.

The ACI-33 uses the self-clocking redundant Manchester scheme of encoding, sometimes called the "Byte Standard." The two logic states used for encoding are represented by 1200 Hz and 2400 Hz frequencies which are written onto and read from the tape.

Priced at $59.95 for an assembled model, the interface is available from Personal Computing Company, 3321 Towerwood Dr, Suite 101, Dallas TX 75234.

Circle 572 on inquiry card.

MARQUIS, a New Video Display Unit from Dataview

The MARQUIS, announced by Dataview Inc, 23A Dana St, Malden MA 02148, is a new video display unit featuring both 20 mA current loop and RS-232 interfacing capability. The user can select any of 13 different data transmission rates ranging from 50 to 9600 bps. Full or half duplex operation and parity generation are included and are switch selectable. The unit displays 80 upper case characters per line, using a 5 by 7 dot matrix. Characters are taken from a 64 character ASCII set.

For more information, write to Dataview Inc, POB 73, Malden MA 02141.

Circle 573 on inquiry card.

A 3 Base Triple Play

Texas Instruments' new TI Programmer calculator could be a boon for the long suffering programmer: it performs arithmetic in octal, hexadecimal and decimal, and can be used to convert automatically from one of these bases to any other.

Typical applications of this calculator include converting memory addresses from hexadecimal to decimal, calculating relative address locations and so on. Expressions may have mixed bases within them, and parentheses may be nested up to four deep. Negative numbers are displayed in two's complement form in both the hexadecimal and octal bases: a one's complement key is also included.

A particularly useful feature on this unit is its ability to perform bit by bit operations on hexadecimal or octal numbers, including AND, OR, exclusive OR and SHIFT operations.

The TI Programmer sells for $49.95 complete with 8 digit LED display, rechargeable battery, AC adapter and carrying case, and is being offered initially by mail only from the company. For more information, write Texas Instruments Inc, Inquiry Answering Service, POB 5012, M/S 84 (Attn: TIP), Dallas TX 75222.

Circle 574 on inquiry card.

News of Interest to "Computers"

SEI Publications has announced a new monthly management report called the Small Business Computer. The term "computer" was coined by the company to describe the man who computes, rather than the machine. The report is to be in newsletter format with no advertising. Its goal is to educate and assist small businessmen and women in the use of computers, computing and data processing as they apply to small businesses.


More information can be obtained by writing to SEI Publications, POB 145, Newington CT 06111.
PDR-27 GEIGER COUNTER
Just released by the US Navy San Diego. They appear to be in excellent condition and come in fitted aluminum carrying case not shown. Batteries easily obtained except for 1.5 mercury cells which you can substitute with external AA cells, 4 ranges from 0.5 to 500 mr/hr, detects beta & gamma rays. Visual indication and if phones are purchased, audio as well. With no facilities to test, we are selling “as is” visually OK, with schematic.
Phones $5.00  Inst. Book $5.00  PDR-27 $35.00

TOUCH TONE ENCODER KIT $12.95
Simply solder the chip to the PC board on back of the touch pad. It’s done. Add 9 volt battery, small speaker, and you have touch tone audio output. We provide specs and instructions.

SPL-149-B  $12.95

IR VIEWER $199.00 COMPLETE
Custom made with manufacturers guarantee. Complete with built-in light source. Permits viewing in total darkness. Operates from 6 volt lantern battery. Great for scientific experimentation, viewing birds, animals, criminal detection, just plain snooping. (We cannot ship to Calif. residents.
SPL-21  $199.00

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
SP-324-A $1.00/ft.  10 ft/$9.00
SP-234-A $1.00 ft 50 cond.  10 ft/$9.00
SP-234-B .90 ft 32 cond.  10 ft/$8.00

TOUCHTONE ENCODER CHIP
Compatible with Bell system, no crystal required. Ideal for repeaters & w/specs.
SPL-21  $6.00

WIRE WRAP WIRE
TEFZEL blue #30 Reg. price $13.28/100 ft. Our price 100 ft $2.00; 500 ft $7.50.

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
24 '' #24  5.00  20.00  30.00
29 ''  22  7.50  28.00  45.00
Great savings as these are about 1/4 book prices. All fresh & new.

CHARACTER GENERATOR CHIP
Memory is 512x5 produces 64 five by seven ASCII characters. New material w/data
SPL-21  $6.00

Please add shipping cost on above. Minimum order $10
FREE CATALOG SP-9 NOW READY
P.O. Box 62, E. Lynn, Massachusetts 01904

Meshna

Circle 460 on inquiry card.
A New Altair That Looks Different

The new Altair Turnkey microcomputer is a significant departure in appearance from the rest of the Altair line. The front panel is virtually blank, sporting only two toggle switches, five LED indicators, and a key operated power switch. All the functional units of this Turnkey model of the 8800b are contained on one circuit board called the Turnkey Module board; this includes the central processing unit (CPU), programmable memory, sense switches and serial IO. All the hardware for this model is compatible with preceding Altair computers.

The Turnkey Module board consists of a serial IO channel that can operate a variety of peripheral devices, plus 1 K byte of programmable memory and provisions for another 1 K bytes of programmable read only memory and logic for the power-on-start feature.

The available software includes a multipurpose bootstrap loader and monitor, both resident in the programmable read only memory. The 8800b is software compatible with the rest of the Altair line. For more information, contact MITS, 2450 Alamo SE, Albuquerque NM 87106.

Industrial Users: Upgrade MDS-800 to Develop Z-80 Code

For designers wanting the additional capability of the Zilog Z-80 processor, but who presently have Intel MDS-800 hardware and 8080 software, Relational Memory Systems Inc (RELM) announces the Z-80 System Adaptor Module, Z-80-SAM. Completely hardware and software compatible with the 8080, 8080A, 8085 and Z-80 microprocessors, the Z-80-SAM enables an MDS-800 to support and develop both 8080 and Zilog Z-80 software. The Z-80-SAM contains a single board and associated software and firmware. The SAM microcomputer board supplants the corresponding MDS-800 board.

Besides the hardware board, Z-80 SAM system monitor programmable read only memory (PROM) firmware replaces corresponding MDS-800 read only memory (ROM) firmware (on monitor ROM board).

The Z-80-SAM extends the useful life of the capital investment 8080 users have in their MDS-800 system while enabling the user to utilize the faster, more efficient Z-80 microprocessor in future hardware/software development.

Enhanced debugging capability is afforded by hardware features such as display LEDs and single cycle switch which allow the user to freeze the MDS-800 data bus, examine the bus contents, and single step the program.

The Z-80-SAM is priced at $1495 in single unit quantities which includes the Z-80 microcomputer board, monitor ROM board, and Z-80 SAM assembler on diskette.

For more information, contact Douglas B Kelley, director of sales and marketing, Relational Memory Systems Inc (RELM), POB 6719, Santa Clara CA 95150.

A New Way to Label Integrated Circuit Pins

Here is an efficient method for labeling dual in line (DIP) integrated circuit pins on the pin side of printed circuit boards:

- Adtech Inc, POB 10415, Honolulu HI 96815, has announced DIP-1 DIP Pin Out Labels. The labels, which come in a variety of sizes from 8 pin to 40 pin, are printed on an 8½ by 11 inch sheet of clear mylar with adhesive backing. The accompanying reproduction illustrates a portion of the sheet. Each label is imprinted with the appropriate pin out pattern. The user cuts out the desired label and presses it onto the circuit board. Each 8½ by 11 inch sheet costs $5.95, postpaid, in the US. This technique can save a significant amount of time by cutting down on wiring errors and speeding up troubleshooting. We highly recommend this technique for prototyping, having used a crude manual approximation for years to help eliminate wiring errors.

- Century Industries, POB 348, Blue Bell PA 19422, has announced a new power supply kit called the Mighty 4000, designed for use with microprocessors. The unit features four regulated and short circuit protected outputs: +5 V at 3.5 A and -5 V at 2.2 A (both fixed output), +5 V at 1.9 A to +12 V at 3.3 A, and +12 V at 0.23 A to +28 V at 0.33 A.

The supply is 10.66 by 4.43 by 3.80 inches (27.07 by 11.25 by 9.65 cm) and weighs 8.5 pounds. The price of the kit is $99; it is also available assembled for $129.

Power for Your Microprocessor

Minicomputer Press has announced forthcoming publication of the first of several new periodicals devoted to minicomputer end users in business, industry and the home. Minicomputer Workbook is devoted to the interests of engineers, educators and professionals who are familiar with data manipulation and computer concepts, but who need a periodical that integrates the two areas together in a meaningful way. For more information, contact Charles Moore Associates, POB 6, Stump Rd, Southampton PA 18966.

New Minicomputer Periodical Offered

- Circle 576 on inquiry card.
- Circle 578 on inquiry card.
- Circle 577 on inquiry card.
Now low-cost memory stacks up in reliability!

Introducing a new generation of ECONORAM™ dynamics with SynchroFresh™ reliability

Meet ECONORAM™ III with SynchroFresh™ the 8Kx8 dynamic memory for S-100 bus computers that really works. And uses less than half the power of static designs. And costs just $188 for an assembled 8K.

Unlike previous attempts at building a low-cost dynamic memory, ECONORAM™ III is entirely reliable... because of SynchroFresh™, a new approach to memory refresh that is simple, elegant and totally effective.

SynchroFresh™ was invented by George Morrow, designer of the original ECONORAM™. Instead of arbitrarily interrupting your CPU to perform memory refresh cycles, Morrow designed SynchroFresh™ to weave refresh invisibly into the natural timing of the S-100 bus. SynchroFresh™ circuitry simply monitors your computer's machine states, utilizing all of the normal opportunities for memory refresh. It's that simple.

And simplicity means reliability and dramatically lower cost. That's why a SynchroFresh™ design was chosen for the first ECONORAM™ dynamic, to follow in the footsteps of the largest-selling static memories for personal computers.

ECONORAM™ III with SynchroFresh™ is an 8Kx8 dynamic board, configured as two individually addressable 4K blocks for flexibility. It is available assembled, tested and warranted for one full year for just $188. This unprecedented warranty offers a full refund of purchase price if ECONORAM™ III does not run reliably with your S-100 CPU—evidence of our confidence in its performance.

It is also available as a kit with complete assembly instructions and documentation for $159.

ECONORAM™ III with SynchroFresh™ in assembled or kit form, may be ordered directly from Thinker-Toys™. Write 1201 10th Street, Berkeley CA 94710 or call (415) 527-7548. Call BAC/MC orders toll-free to 800-648-5311. Or ask your computer store to order it for you.

$188
Assembled, tested & warranted

ECONORAM™ III with SynchroFresh™

A product of Morrow's Micro-Stuff for

Thinker Toys™

*ECONORAM is a trademark of Godbout Electronics.
Bally introduces new programmable game unit

The new Bally Professional Arcade home TV entertainment center is a well-engineered example of the new breed of programmable game modules which use microprocessors for logic and control functions.

The unit features a printing keyboard calculator and can generate video games in color. The games are stored on cassettes; users can change games when desired by simply plugging a different cassette into the front of the unit. A large variety of games is available, including Checkmate, Dodgem, Seawolf, tennis, elementary math drills and poker; up to four players can participate. All games come complete with sound effects and music.

The unit sells for $299, which includes two game cassettes. Contact Robert E. Wiles, director of marketing, consumer products, Midway Manufacturing Corporation, 10750 Grand Av, Franklin Park IL 60131.

OAE announces new PROM programmer

OAE's new programmable read only memory (PROM) programmer should be of interest to the experimenter. This is the PP 2708/16, designed to program the 2708 and the 2716 PROMs which are available from a variety of sources. A parallel interface is used to connect the unit to any microcomputer; an internal address unit is built into the unit, plus timing and control logic and DC to DC power conversion from unregulated 8 VDC input.

The PP 2708/16 comes complete with a black anodized aluminum case, 5 foot ribbon cable with connectors, and software. The price is $249 for the kit, or $299 for the assembled and tested unit. Contact Oliver Audio Engineering Inc, 7330 Laurel Canyon Blvd, North Hollywood CA 91605.

Innovex introduces new series 400 floppy disk drives

Innovex Inc, 75 Wiggins Av, Bedford MA 01730, has announced their new series 400 floppy disk drives which feature automatic head unload and stepper motor timeouts, bidirectional write protect, six LED activity indicators, and 50 pin ribbon cable or twisted pair interfacing capability.

Both the Model 410, which is soft sectored and IBM compatible, and the Model 420, which is hard sectored, provide single and double density recording capability. A proprietary data separator design is said to offer improved data integrity margins in comparison with other available units.

Prices for the Innovex Series 400 start at $575 in single quantities. Contact Innovex for further information.

Turn your microcomputer into a programmable scientific calculator

From Artisan Electronics, 5 Eastmans Rd, Parsippany NJ 07054, comes news of a new microcalculator called the Model 85. This new unit is the single board equivalent of a hand held scientific calculator but without a keyboard or case, designed to interface directly to a microcomputer.

Instruction entry to the microcalculator Model 85 is under microprocessor software control. The Model 85 accepts instructions, provides a means to detect busy status, and outputs the full 14 digit display back to the microprocessor for storage or display.

Software for controlling the Model 85 in both a read and write mode requires less than 256 bytes of microprocessor system memory.

The Model 85 has scientific calculation capabilities for handling scientific, engineering, mathematical or statistical problems. Problem solving capability includes transcendental functions, such as logarithms, sines and tangents; polar to rectangular coordinate conversions for handling complex arithmetic, vector; multiple storage registers, selecting operating mode and also constants for π and e are provided as well as four metric to English unit constants for conversions between inches and centimeters, liters to gallons, and so on. Statistical routines are also provided for calculation of means and standard deviations. The price is $189.

Burr-Brown has announced their new MP20, a microprocessor compatible analog input which consists of a 16 channel analog multiplexer, high gain instrumentation amplifier, 8 bit analog to digital converter, plus all necessary address, data and control bus interfaces.

The MP20 is timing and logic level compatible with the 8080A and the 8008 processors. No external logic is needed. Gain and offset are internally laser trimmed, eliminating the need for external adjustments. Absolute accuracy is better than ±0.4% on the ±5 V or 0 to +5 V ranges.

Low level signals such as thermocouple outputs can also be handled directly with reduced accuracy. The instrumentation amplifier can be programmed with a single external resistor to provide input signal ranges as low as ±10 mV full scale.

In actual use, the MP20 is treated as memory. Each analog input channel occupies one memory location. Any memory reference instruction can be used to access data. Thus an LDA instruction will input data from one channel to the accumulator. Two adjacent input channels can even be acquired with one LHL instruction. The MP20 can also be interfaced as IO.

Power requirements for the unit are ±15 VDC and ±5 VDC. The unit is $195 in single unit quantities. Contact C. R. Teeple, product manager, Burr-Brown, International Airport Industrial Park, Tucson AZ 85734.
From the same people who brought you the 599.95 K RAM KIT. We were not the first to introduce an IMSAI/Altair compatible 280 K RAM Card, but we do feel that ours has the best design and quality for the lowest price! The above features all the 280 K card in an expanded set of 156 instructions, 60000 ROM, and a complete software set. Software compatibility, and operation from a single 5VDC supply, are all well known. What makes our card different is the extra care we took in the hardware design. The CPU card will always go on an M1 state. We also supply COMPLETE TRUE 5VDC, in case that the rest of our system functions properly. Dynamic memory refresh and NMI are built out for your use. Believe it or not, not all of our components have gone to the extreme of being this. As always this kit includes all parts, all sockets, and complete instructions for ease of assembly. Because of our past experience with our K1 kit we suggest that you order early. All orders will be shipped on a first come first served basis. Kit includes 1400 Manual and all parts. Kit shipped with 2 MHz crystals.

Z-80 Chip & Manual - $49.95: Add $5.00 for Z-80A
Z-80 Manual - $7.00 Separately.

Complete kit - $149.

NEW! DESIGN CONSOLE KIT - $89.95

S.D. Sales announces the inexpensive way to beat the wire wrap jungle. Our latest kit gives you 124 solderless quick connect terminals, enough for eight 16 pin IC's and provides 50 x 8 common bus matrix. Has regulated +5VDC and +15VDC, all at 1 AMP. Voltage regulation at 100°.

Also includes a pulse generator variable (from 0Hz to 500mHz and 01 sec to 100 nano seconds). Generator output is +5V. In kit form only and includes all parts, sockets, front panel measures 15" x 22", and hardware case not available.

CAR/BOAT KIT $34.95

Music to your Ears!

Limited Quantity! $9.95 kit

MUSICAL HORN


THE WHOLE WORKS $89.95 kit

THE WHOLE WORKS - INCLUDES CAR/BOAT KIT $26.90 HORN KIT - INCLUDES CAR/BOAT KIT $5/00 $1.00 $6/00 $1.00

6 DIGIT ALARM CLOCK KIT

We made a fantastic kit even better! redesigned to take advantage of the latest advances in IC technology. Features: Lithium Dual 9 display, Master 50 x 24 pin clock chip, single 1 C segment display, SCI chip drivers. Greatly simplified construction. Most reliable and durable to build. A kit which eliminates all necessary parts, silicon chips. For P.C. board add $3.00, 4C XRAM add $1.50. Do not confuse with non-alarm kits sold by other companies. Eliminates the hassle - avoid the $3101.

4/$1.00 12/$1.00 75c ea. 200/$2.00 SPECIAL! 3/$1.25 11,000 MFD 50VDC

3.579545

PC BOARD ASSORTMENT

28 PIN SOCKETS $2.60

FACTORY PRIME! 21L02-1

500ns 8/$12.95

5/00 250ns 8/$15.95

MOS 6 Digit Up/Down Counter

40 PIN DIP. Everything you ever wanted in a counter chip. Features: Direct LED segment drive, simple power supply (12 VDC TYPE), six decades, unidirectional, 64 step counter, separate pre- adjustable compare register with compare output, BCD and seven segment outputs, internal clock oscillator, MOMOS compatible, limiting zero blinking. 1MHZ, count up frequency.

$12.95

orders over $15. - Choose $1. FREE MERCHANDISE!
Nybbllng Thinly Sliced Bytes

Texas Instruments has announced a new set of 4 bit slice microcomputer circuitry which will be of great interest to experimenters wanting to try their hands at designing a microcomputer from scratch.

The S481 chip set is a new series of Schottky TTL microprogrammable building block integrated circuits which are said to offer up to ten times the throughput of conventional microprocessor systems.

The S481 chip set can select and operate on two operands, generate status and store results in 100 ns. The set consists of one or more 4 bit slice processors, one or more 4 bit slice controllers, an appropriate number of field programmable logic arrays (FPLAs), and a variety of Schottky programmable read only memories and programmable memories.

The experimenter can choose word size to be any multiple of four, making of general purpose registers which can be created is limited only by the number of directly accessible memory locations. A memory to memory system places the register file function in main memory; this system can be designed to handle interrupts or context switches faster than conventional register to memory systems.

For more information, contact Texas Instruments Inc, Inquiry Answering Service, POB 5012, M/S 308 (Attn: S481), Dallas TX 75222.

Circle S86 on inquiry card.

Periodical Guide for Computerists

The Periodical Guide for Computerists is a new 20 page book that indexes over 1000 personal computing articles from 15 magazines (including BYTE) for the period from January thru December 1976. The articles are indexed under more than 100 subject categories. Indexed are magazine articles, letters from readers, book reviews and editorials from both hobbyist and professional publications.

The books are available from E Berg Publications, 1360 SW 199th Ct, Aloha OR 97005, for $2.50 each, postpaid, and also from local computer stores.

Circle S87 on inquiry card.

A New Product Guide from RCA

RCA has announced the MPG-180, a new 40 page guide to their COSMAC family of microprocessor products. Hardware products described include the CDP1800 series of CMOS integrated circuits. This series features central processing units (CPUs), programmable memories, read only memories, IQ circuits and interfacing circuitry. The MPG-180 describes the CDP1802 microprocessor and covers its features, architecture, ratings, characteristics, timing diagram and instruction summary.

Among the additional items described are a Microtutor which teaches programming techniques, a floppy disk system, a binary arithmetic package, timeshare software development packages and a cross-reference guide to other manufacturers' integrated circuits.

The free MPG-180 product guide may be obtained by writing to RCA Solid State Division, POB 3200, Somerville NJ 08876.

Circle S88 on inquiry card.

Is Your Computer System Half-Baked?

We received the following information from The Computer Center Inc, 321 Pacific Av, San Francisco CA 94111:

A new Microcomputer Recipe Book is available free from the Computer Center, listing everything from soup to nuts that you need to put together your own microcomputer operating system for personal, business or scientific use. Items listed under "Ingredients" include computers, semiconductor and floppy disk memories, video displays and printers. Suggested menus for complete systems range from the "Big Mac," a simple 8080-based computer used in conjunction with the family TV set to the "Beef Wellington," which includes a Processor Technology Sol-20 microcomputer, dual floppy disk drives, a DEWriter II printer and a video monitor. Technical books dealing with hardware design, software development and "How To Do It" books are also offered.

Circle S89 on inquiry card.

A Hand Held 4 Channel Logic Analyzer

A problem with multichannel digital logic analyzers for most computer experimenters is cost. Digital Broadcast Systems Inc, 4306 Governors Dr, Huntsville AL 35805, has provided a relatively low cost solution to this problem with their new 45-B Logic Analyzer. The 45-B is a hand held instrument which attaches to any conventional single trace oscilloscope having an external trigger to produce four simultaneous digital waveforms. A typical application would be to display the clock from a given circuit on one channel, and selected logic gate inputs and outputs on the other three channels. Complete instructions are printed on the analyzer itself.

The unit can be used with TTL, DTL, RTL or CMOS logic families. Two potentiometers control waveform amplitude and multiplexed display rate; once these controls are set, the scope time base control operates normally. The price, including a 9 V battery and AC adapter, is $149.95.

Circle S90 on inquiry card.

Continental Specialties' New Catalog

Continental Specialties Corporation, 44 Kendall St, POB 1942, New Haven CT 06509, have announced their new 16 page catalog. It contains a variety of breadboarding aids such as integrated circuit test clips, solderless breadboards (some with self-contained power supplies), and logic probes. A line of instrumentation is also featured, including a function generator, RC bridge and pulse generator. There is a question and answer section in back of the catalog which answers frequently asked breadboarding questions. The catalog is free.

Circle S91 on inquiry card.

A New Floppy from Digital Systems

Digital Systems announces a new floppy disk system featuring Shugart Associates drives and Digital Systems FDC-1 controller. Disk formatting is IBM compatible, and diskette initialization capability is provided. The Digital Research CP/M disk operating system software is available as an option, as is an Altair bus compatible interface. The price for the assembled and tested unit is $1845. Contact Digital Systems, 6017 Margarido Dr, Oakland CA 94618.

Circle S92 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 serial devices on one RS 232 device.

Typical use: You can use your computer ports to run an RS 232 printer, video terminal and two other RS 232 devices at once, without constantly connecting and disconnecting your terminals.

Example: Out store to printer — Voltage requirement +5V and ±5V or ±12V depending on your RS 232 device.

We supply — board, connectors, documentation and components. Sorry, we do not supply case or power supply.

WHERE IT MAKES SENSE, MAY BE USED WITH ANY 8080, 6800, Z80 or F8 COMPUTER

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 60 cycle primary. The outputs with local regulators are ±5V to 10A, in one amp increments, ±5V at 1A, ±12V at 1A. Separate TIL to RS 232 devices. See accessory requirement ±5V and ±12V depending on your RS 232 device.

This power supply kit features a high frequency torroid transformer with switching transistors in order to save space and weight. 115V 60 cycle primary. The outputs with local regulators are ±5V to 10A, in one amp increments, ±5V at 1A, ±12V at 1A. Separate TIL to RS 232 devices. See accessory requirement ±5V and ±12V depending on your RS 232 device.

32-2102-1 static RAM's, 16 address lines, 8 data lines in, 8 data lines out, all buffered. Onboard decoding for any of 64 pages, standard 44 pin, 156" buss.

FOR FAIRBUG 4K F8 BASIC ON PAPER TAPE $25.00

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-lup.

- 32-2102-1 static RAM's, 16 address lines, 8 data lines in, 8 data lines out, all buffered. Onboard decoding for any of 64 pages, standard 44 pin, 156" buss.

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 PSU

FOR FAIRBUG 4K F8 BASIC ON PAPER TAPE $25.00
A New Printed Circuit Board for the

Experimenter

Midgard Electronics Inc, 26 Walnut St, Waterstown MA 02172, has announced the UMB, an epoxy glass board with 1380 plated through holes in a two-sided foil pattern. There are four power distributions, and pad spacing is 0.1 inch (0.254 cm).

All four power distributions are bus bars to each metalized circuit position to provide short interconnection distances. A reverse foil pattern minimizes characteristic impedances.

The UMB will accommodate all dual in line integrated circuit packages as well as discrete components. Exterior connection is provided by a 22 pin edge connector compatible with the Amphenol type 225-2221-101 connector. The price is $17.15 in quantities of one to six.

Circle 569 on inquiry card.

A New Business Software and Hardware Package

Computer Mart of New Jersey Inc, 501 RT 27, Iselin NJ 08830, has announced a new business oriented small computer system. It stores information on floppy disks for random access, and features both video display and hard copy printing of business reports. A custom software package allows the user to communicate in simple English phrases. Contact the Computer Mart for further details.

Circle 509 on inquiry card.

Electrolabs

POB 6721, Stanford, CA 94305

415-321-5601

TRICK OR TREAT!!

For each C2716, or TMS2716 purchased before 1 Jan, 1978 we will donate $2.00 to UNICEF in your name, and issue you an equal amount in credit towards your 1978 purchases.

UVEPROMs
C2716 5V Only - 99.00
TMS2716 3 Supplies - 49.00
C2702 Indus. Std. - 18.95
C2708 l50nSec. - 14.85
C1702A 2K - 4.95
C1702 2K - 2.25
M52030 2K - 2.25
M52040 4K - 0.95

Removable Magnetic Storage Media

manufactured by
Information Terminals Corp.

MiniDisks
5 Per Box, Ea. Box: 10 Boxes, Ea: 50 Boxes, Ea
MD525 1 (Soft Sector) 29.95 24.25 24.25
MD525 10 (Hard Sector: 10 Holes) 29.95 24.24 24.95
MD525 16 (Hard Sector: 16 Holes) 29.95 24.95 24.95

Flexible Disks
2 Per Pkg.
FD34-1000 (Soft Sector IBM Std) 11.95 10.49 9.49
FD32-1000 (Hard Sector, Inner) 12.49 10.19 9.19
FD65-1000 (Hard Sector, Outer) 12.49 10.19 9.19

Flippable Disks
10 Per Box:
FD34-2000 (Soft, IBM) 89.30 72.30 72.30
FD35-2001 (Hard, Inner) 92.80 84.40 75.90
FD65-2000 (Hard, Outer) 92.80 84.40 75.90

Digital Cassettes
2 Per Pkg.
I-150 For all "Kansas City" and other digital recording with audio decs.
9.90 8.50 7.99

Circle 506 on inquiry card.

The A 242A features:
 Flush mounted acoustic cups to lock in hand set.
 A vibration isolation technique utilizing direct microphone hand set coupling.
 Quartz crystal control of both transmitter and receiver.
 A carrier detector designed to sense valid data regardless of the carrier level.
 RS-232 and 20 mA interfaces provided.
 A rubber foot design for vibration isolation.

The A 242A is priced at $365 in single quantities. Contact Anderson Jacobson Inc, 521 Charcot Av, San Jose CA 95131, (408) 263-8520.

Circle 504 on inquiry card.
A powerful new tool for every serious 8080 user - professional and novice alike. Priceless timesaver for engineers, technicians, and programmers. SAVES time and money in the lab, on the production line, or in the field.

Convenient pocket size — 3½ by 7½ inches — gives quick and easy access to all vital reference data. No more searching here and there for codes, instructions, or definitions. It’s all there — at your finger tips — everything you need to successfully use the 8080A and — Intel's new 8085 microprocessor.

Features cross listing, for rapid assembly and disassembly, of MACHINE CODES and MNEMONICS. Concise description of 8080 and 8085 OPERATIONS, SIGNALS, PINOUTS, and INSTRUCTIONS. Convenient cross conversion of OCTAL, HEXDECIMAL, DECIMAL, ASCII, and EBCDIC codes. Easy-to-read tables of powers of two, eight, and sixteen ... and much more ...

Sturdy • Handsome • Easy-to-use • Data Packed

Your timesaver will give many years of professional service.

$12.95 each (plus postage & California sales tax) — 25% discount for 4 or more.

MoneyBack Guarantee: You must be fully satisfied or simply return the guide within 15 days for full and prompt refund.

URBAN INSTRUMENTS • 4014 CODY ROAD • DEPARTMENT C1 • SHERMAN OAKS • CALIFORNIA 91403

PLEASE SEND YOUR ORDER TO: CHECK □.M.O. ENCLOSED □ BANK AMERICAN □ 1 Master Charge □

NAME .......................................................
CARD NUMBER ...........................................
STREET .....................................................
CITY .........................................................
STATE ....................................................... ZIP ....................................................... 
SIGNATURE ........................... 

NEW! All IC's, sockets & hardware for WAMECO MEM-1 includes prime 2102AL-4's. $1.44
Order PCB only separately below
Special 2102AL-4 1K x 1 ram is less than power of 21102 type rams, with power down, prime from NEC. Ea. 2.20; 32 ea. 1.80; 64 ea. 1.70; 128 ea. 1.60; 256 ea. 1.50.

CLAUDIO M. ALCAMO

2708 EROM board, S-100, 8KX8 or 16KX8 kit without PROMS .. $65
2708B EROM board, S-100, 8KX8 or 16KX8 kit without PROMS .. $65
IO-2 S-100; 9 bit parallel I/O, 4/0 of board is for rgbkering.
$55 PCB
$30

NEW! All IC's, sockets & hardware for WAMECO MEM-1 includes prime 2102AL-4's. $1.44
Order PCB only separately below
Special 2102AL-4 1K x 1 ram is less than power of 21102 type rams, with power down, prime from NEC. Ea. 2.20; 32 ea. 1.80; 64 ea. 1.70; 128 ea. 1.60; 256 ea. 1.50.

VARIABLE RESISTOR BOARD, 96S102, 2.00
VARIABLE RESISTOR BOARD, 96S102, 2.00

MoneyBack Guarantee: You must be fully satisfied or simply return the guide within 15 days for full and prompt refund.

URBAN INSTRUMENTS • 4014 CODY ROAD • DEPARTMENT C1 • SHERMAN OAKS • CALIFORNIA 91403

PLEASE SEND YOUR ORDER TO: CHECK □.M.O. ENCLOSED □ BANK AMERICAN □ 1 Master Charge □

NAME .......................................................
CARD NUMBER ...........................................
STREET .....................................................
CITY .........................................................
STATE ....................................................... ZIP ....................................................... 
SIGNATURE ........................... 

NEW! All IC's, sockets & hardware for WAMECO MEM-1 includes prime 2102AL-4's. $1.44
Order PCB only separately below
Special 2102AL-4 1K x 1 ram is less than power of 21102 type rams, with power down, prime from NEC. Ea. 2.20; 32 ea. 1.80; 64 ea. 1.70; 128 ea. 1.60; 256 ea. 1.50.

CLAUDIO M. ALCAMO

2708 EROM board, S-100, 8KX8 or 16KX8 kit without PROMS .. $65
2708B EROM board, S-100, 8KX8 or 16KX8 kit without PROMS .. $65
IO-2 S-100; 9 bit parallel I/O, 4/0 of board is for rgbkering.
$55 PCB
$30

NEW! All IC's, sockets & hardware for WAMECO MEM-1 includes prime 2102AL-4's. $1.44
Order PCB only separately below
Special 2102AL-4 1K x 1 ram is less than power of 21102 type rams, with power down, prime from NEC. Ea. 2.20; 32 ea. 1.80; 64 ea. 1.70; 128 ea. 1.60; 256 ea. 1.50.

VARIABLE RESISTOR BOARD, 96S102, 2.00
VARIABLE RESISTOR BOARD, 96S102, 2.00

MoneyBack Guarantee: You must be fully satisfied or simply return the guide within 15 days for full and prompt refund.
<table>
<thead>
<tr>
<th>Reader Service</th>
<th>Reader Service</th>
<th>Reader Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>441 AAA Chicago Computer Center p. 185</td>
<td>360 Extensys p. 31</td>
<td>354 PERCOMP 78 p. 5</td>
</tr>
<tr>
<td>382 Alpha Digital p. 82</td>
<td>419 Floto, Charles p. 177</td>
<td>387 Peripheral Vision p. 99</td>
</tr>
<tr>
<td>378 Artec Electronics p. 73</td>
<td>372 Information Terminals p. 51</td>
<td>* PolyMorphic Systems p. 7</td>
</tr>
<tr>
<td>446 AVTAR Systems p. 187</td>
<td>371 Integral Data Systems p. 49</td>
<td>355 Processor Tech pp. 8, 9, 10, 52, 53</td>
</tr>
<tr>
<td>383 BITS pp. 83, 144, 151, 175, 176</td>
<td>439 Intelligent Business Machines p. 185</td>
<td>368 Quantronics p. 45</td>
</tr>
<tr>
<td>392 BPI p. 109</td>
<td>417 International Data Systems p. 172</td>
<td>* Radio Shack of Fort Worth p. 43</td>
</tr>
<tr>
<td>409 Byte Shop East p. 155</td>
<td>435 IOR p. 185</td>
<td>406 RHS Marketing p. 139</td>
</tr>
<tr>
<td>451 Byte Shop of Miami p. 187</td>
<td>416 Ishaca Audio p. 171</td>
<td>430 Rotunida Cybernetics p. 184</td>
</tr>
<tr>
<td>* BYTE Subscription p. 93</td>
<td>458 Jade p. 195</td>
<td>372 Scribe Computer Consulting p. 121</td>
</tr>
<tr>
<td>374 Canada Systems p. 59</td>
<td>352 James Electronics Cover IV</td>
<td>* Scientific Research pp. 44, 63</td>
</tr>
<tr>
<td>399 Compuart Inc p. 123</td>
<td>395 John Fluke Mfg p. 117</td>
<td>376 Seals Electronics p. 67</td>
</tr>
<tr>
<td>363 Computalker p. 38</td>
<td>453 JVR Computer Store p. 187</td>
<td>381 Smoke Signal Broadcasting p. 81</td>
</tr>
<tr>
<td>444 Computer Corner p. 186</td>
<td>460 Meshrai p. 199</td>
<td>375 Software Records p. 61</td>
</tr>
<tr>
<td>431 Computer Mart of NH p. 184</td>
<td>369 Microwave Systems Corp. p. 47</td>
<td>350 SWTPC Cover II</td>
</tr>
<tr>
<td>442 Computer Playground p. 186</td>
<td>443 Microtex Inc p. 186</td>
<td>429 Sybex p. 183</td>
</tr>
<tr>
<td>415 Computer Room p. 171</td>
<td>465 Mikos p. 207</td>
<td>401 Synchro Sound pp. 128, 129</td>
</tr>
<tr>
<td>437 Computer Systems Store p. 185</td>
<td>408 MiniTerm p. 149</td>
<td>448 Sierzep Enterprises p. 187</td>
</tr>
<tr>
<td>363 Cromemco pp. 1, 2</td>
<td>461 Morrow's Micro Stuff p. 201</td>
<td>373 Technical Design Labs p. 57</td>
</tr>
<tr>
<td>379 DataByte p. 75</td>
<td>421 mpi p. 178</td>
<td>403 Technical System Consultants p. 133</td>
</tr>
<tr>
<td>436 Electravalue p. 185</td>
<td>411 MSI p. 157</td>
<td>402 Tecnico Inc p. 131</td>
</tr>
<tr>
<td>432 Electronic Mail p. 119</td>
<td>412 Mullen Electronic Boards p. 159</td>
<td>388 TEI p. 101</td>
</tr>
<tr>
<td>438 Disc 3 p. 185</td>
<td>359 North Star Computer pp. 21, 77</td>
<td>358 TLF p. 17</td>
</tr>
<tr>
<td>455 DRC p. 191</td>
<td>384 Objective Design p. 85</td>
<td>466 Urban Instruments pp. 207</td>
</tr>
<tr>
<td>351 ECD Cover III</td>
<td>365 Ohio Scientific pp. 38, 39, 40, 41</td>
<td>393 Vector Electronics p. 111</td>
</tr>
<tr>
<td>450 Economy Terminals p. 187</td>
<td>457 Ohio Audio Electronics p. 194</td>
<td>385 Vector Graphic p. 86</td>
</tr>
<tr>
<td>386 E &amp; L Instruments p. 97</td>
<td>399 Osborne &amp; Associates p. 125</td>
<td>422 Vectron p. 179</td>
</tr>
<tr>
<td>432 EDP Professionals p. 184</td>
<td>434 Outer Products p. 185</td>
<td>424 Worldwide p. 180</td>
</tr>
<tr>
<td>436 Electroluminescent p. 185</td>
<td>428 PAIA Engineering p. 182</td>
<td>405 Xemedia p. 137</td>
</tr>
<tr>
<td>440 Electronic Discount Sales p. 185</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Reader service inquiries not solicited. Correspond directly with company.

**BYTE Subscription** p. 93

**BOM-BombWorks**

The results of the July 1977 BOMB voting found the winner to be James R. Boddie's "Speech Recognition for a Personal Computer System," page 64. James' article placed first in the voting by BYTE readers, at a distance of 1.75 standard deviations above the mean vote for 14 articles in the July voting. Second place was a tie between W. Douglas Maurer's "How to Pick Up a Dropped Bit," page 72, and Robert Grapple's "Give Your Micro a Megabyte," page 78. Bonus checks of $100, $50 and $50 were sent to authors Boddie, Maurer and Grapple respectively.
WE'RE OUT OF OUR MINDS!
OUR MICROMIND COMPUTER HAS MET
WITH SUCH CONSUMER ENTHUSIASM
THAT WE'RE BUSY PARING DOWN
A SIX MONTH PILE OF ORDERS....
PLEASE BE PATIENT.... IT'LL BE WORTH IT!
FOR INFORMATION ON MICROMIND CALL (617) 661-4400

ECD
196 BROADWAY, CAMBRIDGE
MASS. 02139
INNOVATIONS IN ELECTRONICS

IT'S TH' BACKLOG
THAT USUALLY GETS YA'!!
ATTENTION DEALERS:
Announcing

Jim-pak®
electronic components

One-Stop Component Center

- Over 200 quality items including integrated circuits, resistors, diodes, transistors, capacitors, connectors, switches, sockets, LEDs and Data Books covering all JIM-PAK® items.
- Immediate delivery on all orders
- Store display racks available
- Stock rotation and return policy
- Direct mail program available from list of active electronic buyers in dealers' area.
- National advertising campaign in leading electronics magazines to include list of qualifying dealers
- Nationally known manufacturers' products at prices every dealer can afford
- Guaranteed products
- Standard industry part numbers

A component line of proven sellers developed for the independent dealer. Ideal for computer shops, school stores, electronic dealers, hobby shops, or any location where there is a potential market for electronic sales.

A product line which supplies most of your needs from one distributor with a reputation for fast and efficient service. Attractive and compact display racks make initial installation of the JIM-PAK® line easy.

Your customers deserve the best. Now you can profitably retail name brand components at competitive prices. Be the first in your area to announce and sell the JIM-PAK® line. Write or call today.

FOR MORE INFORMATION AND PRICING SCHEDULE CONTACT:
, a division of James Electronics, 1021 Howard Avenue, San Carlos, California 94070, (415) 592-8097

Circle 352 on inquiry card.