



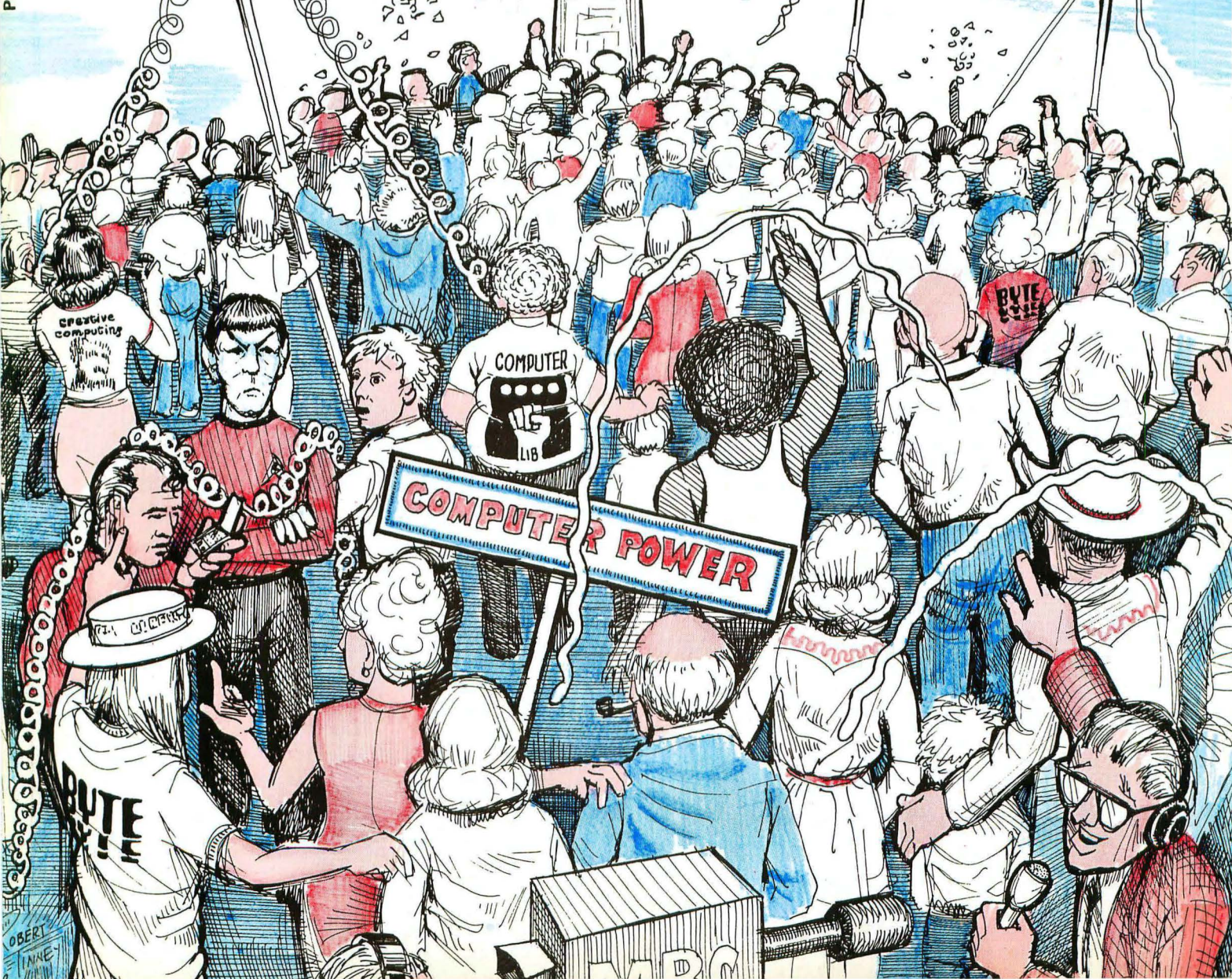
BYTE

0.01-CENTENNIAL ISSUE

PRINTED IN USA

STAMP OUT CYBER-**CRUD!**

2 COMPUTERS IN EVERY HOME!



COMPUTER POWER

COMPUTER LIB

Creative computing

BYTE

COMPUTER

ADAM

ROBERT FINNEY

SWTPC 6800

#1

IN PERFORMANCE

The word is getting around. There is simply no better processor available for general purpose computer work than the Motorola MC6800. This memory oriented processor is easier to program and makes possible more efficient, shorter and faster running programs than the old fashioned bus oriented processors. Have you been convinced that machine language, or assembler programs are only for the experts? Well not with a modern 6800 based computer. Anyone can learn very quickly with this simple straightforward hexadecimal notation processor. When you add to these advantages the unique programmable interfaces and the Mikbug® ROM you truly have a "benchmark" system.

Mikbug® eliminates the tedious and time consuming job of loading the bootstrap program from the switch console each time the computer is turned "On". With Mikbug® this is automatic and you simply don't have switches and status lights. It has been said (not by us) that a switch console is essential for "hardware development," (perhaps they meant "hardware debugging"). Anyway the SwTPC 6800 system has no need for either. This is a fully developed, reliable system with no strange habits. All boards have full buffering for solid noise immune operation. One crystal type clock oscillator drives everything, processor interfaces and all; so there are no adjustments and no problems.

FOR VALUE

The SwTPC 6800 in its basic form comes complete with everything you will need to operate the computer except an I/O device. This may be either a teletype of some kind, or a video

terminal. You get a heavy duty anodized aluminum case, a 10 Amp power supply large enough to power a fully expanded system, a mother board with seven memory/processor slots and eight interface slots, a 2,048 word static memory and a serial control interface. This kit is now only \$395.00. It was introduced at \$450.00, but when processor prices went down we reduced the price of the kit accordingly.

As an owner of our 6800 computer you will get copies of our newsletter with helpful information and software listings. We have a library of software including all the common computer games and our fantastic BASIC. This is available to you for the cost of copying, you don't have to buy anything to get this material.

What more could you want? Pay a visit to our nearest dealer and see the 6800, plus our new cassette interface, graphics terminal and printer. He will be happy to demonstrate our system and to supply you with a 6800 that will fit your exact needs.

Mikbug® is a Motorola Trademark

SWTPC 6800
Computer System

with serial interface and 2,048 words
of memory. \$395.00

Southwest Technical Products Corp.
219 W. Rhapsody
San Antonio, Texas 78216

The Computer Store, 820 Broadway,
Santa Monica, Calif. 90401, (213) 451-0713

Cyberdux, Microcomputer Applications,
1210 Santa Fe Dr., Encinitas, Calif. 92024
(714) 279-4189

The Micro Store, 634 South Central
Expressway, Richardson, Texas 75080
(214) 231-4088

ELS Systems, 2209 N. Taylor Rd.,
Cleveland Heights, Ohio 44112
(216) 249-7820

Microcomputer Systems Inc.,
144 S. Dale Mabry Ave., Tampa, Florida
33609, (813) 879-4301

William Electronics Supply, 1863 Wood-
bridge Ave., Edison, N.J. 08817
(201) 985-3700

Computer Mart of New York, Inc.
314 Fifth, New York, N.Y. 10001
(212) 279-1048

The Byte Shop Computer Store # 1,
1063 El Camino Real, Mountain View,
Calif. 94040, (415) 969-5464

The Byte Shop Computer Store # 2,
3400 El Camino Real, Santa Clara, Calif.
95051, (408) 249-4221

A-VID Electronics Co., 1655 E. 28th Street,
Long Beach, Calif. 90806 (213) 426-5526

Computer Warehouse Store, 584 Common-
worth Ave., Boston, Massachusetts 02215
(617) 261-1100

The Computer Workshop, Inc., 11308
Hounds Way, Rockville, Ind. 20852
(301) 468-0455

The Computer Store, Inc., 120 Cambridge
Street, Burlington, Mass. 01803
(617) 272-8770

Marsh Data Systems, 5405 B. Southern
Comfort Blvd., Tampa, Florida 33614
(813) 886-9890

Midwest Enterprises Inc., 815 Standish Ave.,
Westfield, New Jersey 07090
(212) 432-2066

The Milwaukee Computer Store, 6916 W.
North Ave., Milwaukee, WI 53213
(414) 259-9140

Control Concepts, P.O. Box 272,
Needham Heights, Mass. 02194

American Microprocessors, Equipment &
Supply Corp. at Chicagoland Airport, P.O.
Box 515, Prairie View, Illinois 60069
(312) 634-0076

The Computer Room Inc., 3938 Beau D'Rue
Dr., Eagan, Minn. 55122, (612) 452-2567

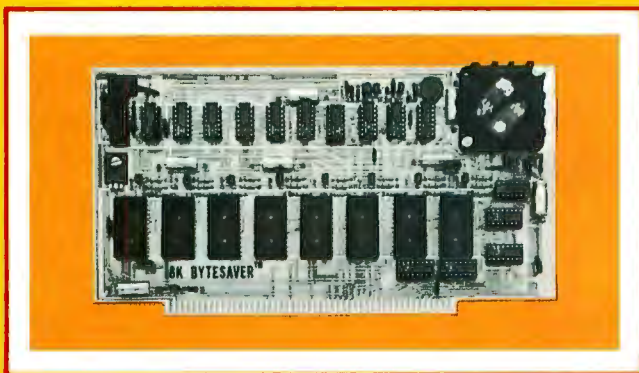
Computerware, 830 First St., Encinitas,
Calif. 92024 (714) 436-9119

Atlanta Computer Mart, 5091 B Buford
Highway, Atlanta, Ga. 30340
(404) 321-4390

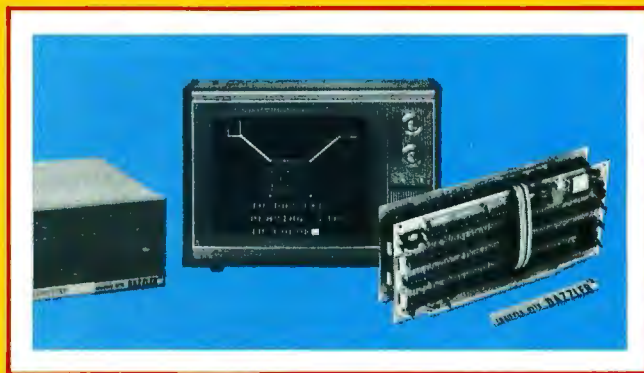
Four ways to get more out of (or into) your computer

Here are four of our most popular computer peripherals. They let you do a lot more with your Altair 8800 or IMSAI 8080. They are simple to use and simple to install. And they all have the combined quality and low price that has made Cromemco the leading name in microcomputer peripherals. Cromemco's delivery is prompt, too.

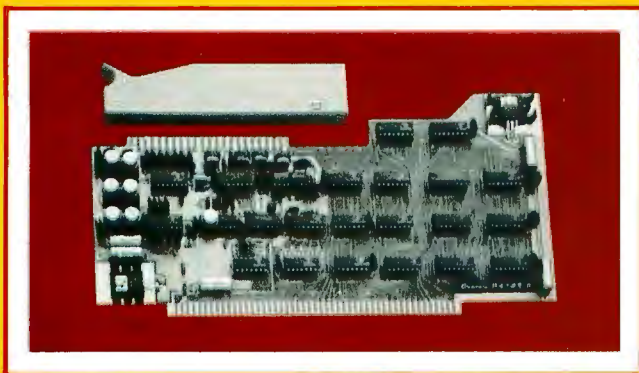
Watch this space for other exciting new Cromemco products to come.



The easy way to put programs into PROM. Cromemco's Bytesaver™ gives you a place for up to 8K of PROM memory using 2704/2708 PROMs. Also gives you a built-in PROM programmer (saves buying one separately). Enough memory capacity to hold powerful programs such as 8K BASIC. Kit (Model 8KBS-K): \$195. Assembled (Model 8KBS-W): \$295.



Let your color TV be your display terminal. You can have a full-color computer display terminal at unbelievably low cost with the Cromemco TV Dazzler™. You can display multi-colored charts, graphs, educational material, games. Requires only 2K-byte memory for 128 x 128-element picture. Kit (Model CGI-K): \$215. Assembled (Model CGI-W): \$350.



Fast analog I/O with 7 channels. Couples your digital computer to an analog world. This advanced board lets you input 7 channels of analog to your computer and output 7 channels of analog to feed to output devices. Also has an 8-bit parallel I/O port. Very fast conversion — only 5 microseconds. Kit (Model D+7A-K): \$145. Assembled (Model D+7A-W): \$245.

JOYSTICK ALSO AVAILABLE: Kit (Model JS-1-K): \$65. Assembled (Model JS-1-W): \$95.



Low-cost Optical Data Digitizer: This small, rugged camera is useful for image recognition, process control, and other industrial applications. Has f2.8 25-mm lens. Uses image sensors that produce 1024-element (32 x 32) picture. Controller boards also available to give software control of exposure, frame rate and memory allocations for picture storage. Camera kit (Model 88-ACC-K): \$195. Controller kit (Model 88-CCC-K): \$195. Camera assembled (Model 88-ACC-W): \$295. Controller assembled (Model 88-CCC-W): \$295.



Cromemco

Specialists in computer peripherals

2432 Charleston Rd., Mountain View, CA 94043 • (415) 964-7400

Imagine a microcomputer

Imagine a microcomputer with all the design savvy, ruggedness, and sophistication of the best minicomputers.

Imagine a microcomputer supported by dozens of interface, memory, and processor option boards. One that can be interfaced to an indefinite number of peripheral devices including dual floppy discs, CRT's, line printers, cassette recorders, video displays, paper tape readers, teleprinters, plotters, and custom devices.

Imagine a microcomputer supported by extensive software including Extended BASIC, Disk BASIC, DOS and a complete library of business, developmental, and industrial programs.

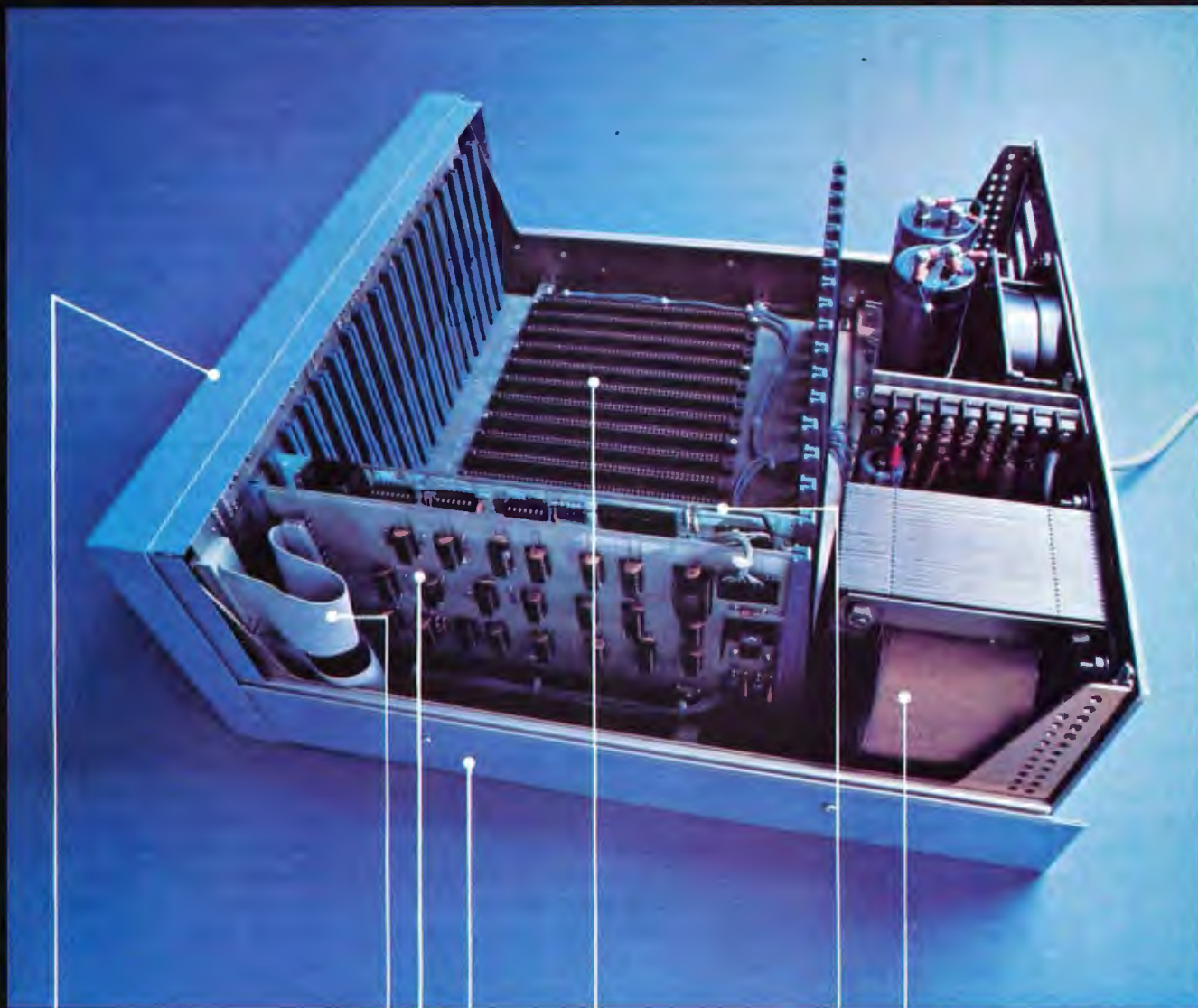
Imagine a microcomputer that will do everything a mini will do, only at a fraction of the cost.

You are imagining the Altair™ 8800b. The Altair 8800b is here today, and it may very well be the mainframe of the 70's.

The Altair 8800b is a second generation design of the most popular microcomputer in the field, the Altair 8800. Built around the 8080A microprocessor, the Altair 8800b is an open ended machine that is compatible with all Altair 8800 hardware and software. It can be configured to match most any system need.

NOTE: Altair is a trademark of MITS, Inc.





Redesigned front panel. Totally synchronous logic design. Same switch and LED arrangement as original Altair 8800. New back-lit Duralith (laminated plastic and mylar, bonded to aluminum) dress panel with multi-color graphics. New longer, flat toggle switches. Five new functions stored on front panel PROM including: DISPLAY ACCUMULATOR (displays contents of accumulator), LOAD ACCUMULATOR (loads contents of the 8 data switches (A7-A0) into accumulator), OUTPUT ACCUMULATOR (Outputs contents of accumulator to I/O device addressed by the upper 8 address switches), INPUT ACCUMULATOR (inputs to the accumulator from the I/O device), and SLOW (causes program execution at a rate of about 5 cycles per second—for program debugging).

Full 18 slot motherboard.

Rugged, commercial grade Optima cabinet.

New front panel interface board buffers all lines to and from 8800b bus.

Two, 34 conductor ribbon cable assemblies. Connects front panel board to front panel interface board. Eliminates need for complicated front panel/bus wiring.

New, heavy duty power supply +8 volts at 18 amps, +18 volts at 2 amps, 18 volts at 2 amps. 110 volt or 220 volt operation (50/60 Hz). Primary tapped for either high or low line operation.

New CPU board with 8080A microprocessor and Intel 8224 clock generator and 8216 bus drivers. Clock pulse widths and phasing as well as frequency are crystal controlled. Compatible with all current Altair 8800 software and hardware.

altair 8800-b

In This BYTE

About the Cover

BYTE began with its first issue dated September 1975. Since that time, a 96 page magazine has grown into a 128 page monthly compendium of information with a reputation of which we're quite naturally proud. That first issue was assembled from scratch in seven weeks of hectic activity starting May 25 1975. At that time, we had no real estimate of the way in which you, our readers, would respond. The goal was simply to put out the best product possible given the constraints and problems of a new enterprise. Since that time, much has changed as the people who bring you

this magazine have all grown and improved with experience.

The principles upon which BYTE is based, technical excellence combined with a sense of humor and a spirit of fun, have not changed. As a celebration of that combination, we commissioned Robert Tinney to implement a fanciful picture of the BYTE 0.01 Centennial Celebration. With this very personal anniversary, we look forward to the developments and improvements of the coming year.

In BNF notation,

```
<we := <We >
```

```
<the contents of the BYTE  
staff listing, page 5>
```

Whatever your stand on the questions of free exchange of software, one thing is certain: To write software of any form is an act of creation. The decision as to what is done with a work of software should reside with the creator. If you are a writer of software, find out about some of the legal aspects of your work by reading Calvin N Mooers' *Are You an Author?*

A multiprocessor system is a combination of two or more processors to accomplish more than what a single processor could do by itself. In his article *Build This Mathematical Function Unit*, author R Scott Guthrie describes a simple form of the multiprocessor concept: a scientific calculator unit controlled by an 8 bit microprocessor. The calculator comes preprogrammed with all the software you need to carry out floating point arithmetic operations and special functions, to say nothing of an arithmetic expression parser implicit in the parenthesis keys. The calculator peripheral in one fell swoop eliminates a lot of the software development required for an interpretive mathematically oriented computer language.

Learn how to *Randomize Your Programming* by reading Robert Grappel's discussion of pseudorandom number sequences along with practical software to implement 8 or 16 bit generators.

Well, here it is: the first version of *Star Trek* to be printed in full in BYTE. Gerald H Herd describes his version of *A BASIC Star Trek Trainer* as implemented on a Data General NOVA, but easily adaptable to any BASIC machine with about 5 K bytes of text area.

One of the choices open to readers familiar with the industrial OEM markets is to purchase computer products intended for systems engineering situations. In his product description article on *The MSC 8080+ Microcomputer as a Personal System*, BYTE reader Ken Barbier enthusiastically describes one such product and his experiences using it.

Binary, octal, hexadecimal or decimal? That is the question. Whatever your preference, however, James Brown will help you out with his article on *How to do a Number of Conversions*. By implementing the whole set of conversions, you can try each base on for size, depending upon your mood and idiosyncracies.

Last month, Burt Hashizume described the neat new architecture of the "super 8080" called Z-80 by its maker, Zilog Inc. In this issue, Dr Robert Suding brings the excitement down to earth with the complete details of *The Circuit for Z-80s*, a

complete central processor with some programmable memory and a dash of systems software in an erasable ROM thrown in for good measure.

What's an SC/MP? Find out by reviewing Robert Baker's *Microprocessor Update: SC/MP Fills a Gap*.

In the final instalment of our series of three reprints from Nat Wadsworth's *Machine Language Programming for the "8008" (and Similar Microcomputers)*, you'll find some information on the details of machine language programming in computers with limited resources.

Recycling pretested integrated circuits mounted on surplus printed circuit boards is an inexpensive way to obtain a good parts inventory. The main problem is getting the circuits off the board. Ralph Droms and Jonathan Bondy have dreamed up *A Flameless IC Recycling Trick* as one way to accomplish the recycling goal.

What does it take to program an 8080 debugging monitor? Joe Kasser and Richard Allen describe AMSAT's answer to this question in *AMSAT 8080 Standard Debug Monitor: AMS80 Version 2*. This is a complete assembly of a useful control program which can be adapted to any 8080 based microcomputer system.

In the Queue

Foreground

- 26 BUILD THIS MATHEMATICAL FUNCTION UNIT—Part 1
Hardware—Guthrie
- 36 RANDOMIZE YOUR PROGRAMMING
Software—Grappel
- 40 A BASIC STAR TREK TRAINER
Software—Herd
- 50 HOW TO DO A NUMBER OF CONVERSIONS
Software—Brown
- 62 THE CIRCUIT FOR Z-80s
Hardware—Suding
- 104 A FLAMELESS IC RECYCLING TRICK
Techniques—Bondy-Droms
- 108 AMSAT 8080 STANDARD DEBUG MONITOR: AMS80 VERSION 2
Systems Software—Allen-Kasser

Background

- 16 ARE YOU AN AUTHOR?
Software—Mooers
- 44 THE MSC 8080+ MICROPROCESSOR AS A PERSONAL SYSTEM
Product Description—Barbier
- 76 MICROPROCESSOR UPDATE: SC/MP FILLS A GAP
Hardware—Baker
- 84 MACHINE LANGUAGE PROGRAMMING FOR THE "8008"—Chapter 3
Software—Wadsworth

Nucleus

- 4 In This BYTE
- 6 Come One, Come All!
- 12 Letters
- 61 Software Bug of the Month 4
- 73 BYTE's Bits
- 74, 81, 96 What's New?
- 92 Classified Ads
- 98 Clubs, Newsletters
- 124 Programming Quickies
- 128 BOMB
- 128 Reader's Service

BYTE magazine is published monthly by BYTE Publications, Inc., 70 Main St, Peterborough, New Hampshire 03458. Subscription rates are \$12 for one year worldwide. Two years, \$22. Three years, \$30. Second class postage paid at Peterborough New Hampshire 03458 and at additional mailing offices. Phone 603-924-7217. Entire contents copyright 1976 BYTE Publications, Inc, Peterborough NH 03458. Address editorial correspondence to Editor, BYTE, 70 Main St, Peterborough NH 03458.

PUBLISHERS

Virginia Peschke
Manfred Peschke

EDITOR

Carl T Helmers, Jr

GENERAL MANAGER

Manfred Peschke

PRODUCTION MANAGER

Judith Havey

CIRCULATION

Deborah R Luhrs

DEALER CIRCULATION

Deena Zealy

PUBLISHERS ASSISTANTS

Cheryl Hurd

Carol Nyland

ADVERTISING

Elizabeth Alpaugh

Virginia Peschke

TYPOGRAPHY

Custom Marketing Resources, Inc

Goodway Graphics

Mary Lavoie

Taimi Woodward

PHOTOGRAPHY

Ed Crabtree

Custom Marketing Resources, Inc

ART

Mary Jane Frohlich

Bill Morello

PRINTING

The George Banta Company

Custom Marketing Resources, Inc

ASSOCIATES

Bob Baker

Dan Fylstra

Harold A Mauch

Chris Ryland

PRODUCTS COORDINATOR

Floyd W Rehling

Come One, Come All!

Editorial by Carl Helmers

NCC 1976

BYTE shared a booth with David Ahl's *Creative Computing* at the 1976 NCC show June 7-10 in New York. For those who are not familiar with the computing trade, NCC is the big trade show sponsored by AFIPS (American Federation of Information Processing Societies) and attended by large numbers of people in the traditional data processing world. It features a strong technological information program with lecturers on numerous topics, as well as one of the most complete trade shows with booths manned by every major manufacturer and vendor in the computer industry.

This year's floor show was in the New York Coliseum, and the technical presentations were held in the New York Hilton and Americana Hotels. I attended a technical session on personal computing on Tuesday morning June 8 in the Americana Hotel. Speakers at this session included Stephen B Gray, Ted Nelson, and Dr Alfred Bork. The theme of Ted Nelson's talk was the idea of the computer becoming a home appliance, a necessity in the home in the same way that a lot of technological innovations have become "necessities." *Computer Lib* becomes a reality. To an audience of very sympathetic professionals, Ted emphasized the concept of the computer and its use as a "way of life," in the same sense that talking is a way of life. For those initiated into the art of computing, the truth of this view is quite evident. Ted also made great arguments for eliminating the term "microcomputer" — what we're all building, buying or using is not micro in any sense of the word, but simply an inexpensive computer of the general purpose variety. (Eliminating that term also gets rid of an ambiguity with

respect to microprogramming.) Ted made a strong case of comparison between the traditional "cuckoo" computer center concept and the medieval church with its priesthood and obscure Latin language. Personal computing as practiced by large numbers of people will help end the concentration of apparent power in the "in" group of programmers and technicians, just as the enlightenment and renaissance in Europe brought about a much wider understanding beginning in the 14th century. (See a forthcoming article by Dave Fylstra and Mike Wilbur for some further commentary on the subject.)

Ted also introduced his concept of the perfect computer store, when he started talking about the "itty bitty machine corporation" whose first computer store is to open soon in Evanston IL. He intends to become the McDonald's of computing. By way of formal legalisms, he entered several terms into interstate commerce, a first step toward obtaining a legally protected trademark: "FUNTRAN" is the itty bitty machine corporation's extensible function translation language, providing word processing, planning and figuring. "Simulatrix" is his name for a proposed library of games with educational and recreational values combined, a library to be maintained with royalties to authors. Interactive art works itty bitty machine corporation is to sell under the "Lady Lovelace" trade name (not a porn film, but the name of the world's first programmer), and the itty bitty machine corporation's first hardware product is to be called the "heaven eleven," an LSI-11 with an Altair compatible bus for peripherals. (For the upper crust, there is "heaven on wheels," a van to be equipped with a "heaven eleven.") [All the quoted words in

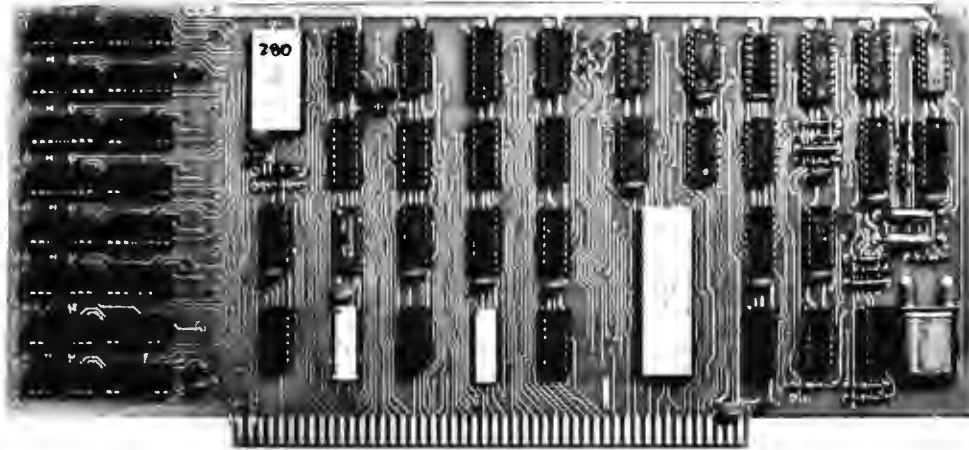
Articles Policy

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

Articles which are accepted are typically acknowledged with a binder check 4 to 8 weeks after receipt. Honorariums for articles are based upon the technical quality and suitability for BYTE's readership and are typically \$15 to \$30 per typeset magazine page. We recommend that authors record their name and address information redundantly on materials submitted, and that a return envelope with postage be supplied in the event the article is not accepted. ■

SUPER CHIP!

The Z-80 CPU by Zilog



From The Digital Group, of course.

If you are considering the purchase of an 8080-based system, look no further. The Z-80 has arrived. A new generation 8080 by the same individuals who helped design the original 8080 — combining all the advantages of the 6800, 6500 and 8080 into one fantastic little chip! And, the Z-80 maintains complete compatibility with 8080 software.

What's even better . . . the Z-80 is being brought to you by The Digital Group — people who understand quality and realize you expect the ultimate for your expenditure. With the Z-80, combined with the Digital Group System's video-based operation, you're at state of the art. There's no place better.

Take a look at some specifications:

Z-80 FEATURES

- Complete compatibility with 8080A object code
- 80 new instructions for a total of 158
- 696 Op codes
- Extensive 16-bit arithmetic
- 3 Interrupt modes (incl 8080), mode 2 provides 128 interrupt vectors
- Built-in automatic dynamic memory refresh
- Eleven addressing modes including:
 - Immediate
 - Immediate extended
 - Page Zero
 - Relative
 - Extended
 - Indexed
 - Register
 - Implied
 - Register Indirect
 - Bit
 - Combination of above

- New Instructions (highlights):
 - Block move up to 64k bytes memory to memory
 - Block I/O up to 256 bytes to/from memory directly
 - String Search
 - Direct bit manipulation
- 22 Registers — 16 general purpose
- 1, 4, 8 and 16 bit operations

DIGITAL GROUP Z-80 CPU CARD

- 2k bytes 500ns static RAM
- 256 bytes EPROM bootstrap loader (1702A)
- 2 Direct Memory Access (DMA) channels
- Hardware Interrupt controller
 - Supports all 3 modes of interrupt
 - Mode 2 supports 128 interrupt vectors
- Data and Address bus lines drive 30 TTL loads
- Z-80 runs at maximum rated speed
- Single step or single instruction step
- EPROM de-selectable for full 64k RAM availability (programs may start at location 0)
- Complete interchangeability with Digital Group 8080A, 6800 and 6500 CPUs

The Z-80 is here. And affordable. Prices for complete Digital Group systems with the Z-80 CPU start at \$475. For more information, please call us or write. Now.



THE DIGITAL GROUP INC.
P.O. BOX 6528
DENVER, CO 80206
(303)861-1686

MERLIN IS HERE!

Have you been trying to decide whether to spend your hard earned money on a fancy graphics display, or on an ASCII, alphanumeric (perhaps limited graphics) video interface? Now there is a third alternative! Get both with MERLIN the MiniTerm magician who can display your Altair or IMSAI memory in either format, or both. Of course MERLIN is plug-in compatible with both computers, and provides standard composite video output.

MERLIN has a 64 character generator chip to display ASCII coded data from your memory as 40 characters by 20 lines. And by a twist of magic (software control of a mode latch), MERLIN's hardware free format "memory saver" mode starts a new line after every carriage return. Change mode again and each point on the screen: 100 vertical by 80 or 160 horizontal, is controlled by a separate memory bit. Want both? Top 'n' lines can be ASCII data, the rest is displayed as graphics.

Software control of ASCII/Graphics mode is just the beginning. Think of some of the crazy (as well as useful) things that can be done with software control of: video polarity (black/white), carriage returns (blanked/displayed), control characters (blanked/video inverted), and cursor (on/off).

By the magic of DMA, MERLIN is super fast! Up to sixty completely different screens every second makes possible a whole new world of computer fun: *computer animation!* Worried about connecting your keyboard? Just plug it into MERLIN's keyboard I/O connector. Perhaps you think MERLIN is all magic and no smarts. NOT SO! Sockets and decoding for 4K ROM or 2K (2708) EPROM and our optional ROM software makes MERLIN the smartest video interface available. Our basic ROM (MBI*) contains all these keyboard editing functions for both fixed and free format modes:

- Cursor Up, Down, Right, Left, and 'Home'
- Delete Character or to 'End-of-Memory'
- Auto and Manual Scroll
- Insert and Replace Modes
- 4 Slave Cursor Operations
- 6 User Defined Functions

and MONITOR routines:

- HEX Dump and HEX Input
- ASCII Text Input
- Set Memory Display Area
- Set Display Mode
- Examine/Modify CPU Program Registers
- Examine/Modify Memory
- Memory Fill
- Execute User Program with Automatic Breakpoints

Our extended function ROM (MEI*) contains more editing functions, including a search routine, more MONITOR commands and graphics subroutines. MERLIN's Basic Intelligence comes with scratch pad RAM memory for monitor use.

With a lot of magic, we at MiniTerm are able to offer this fun and exciting interface for the low price of only \$249. All prepaid orders received before November 1, 1976 will receive free the MBI ROM, regularly sold separately at \$34.95.

A User Manual, including hardware and software details is available for \$8.00 (deductible from MERLIN order). Order now and receive a free listing of LIFE which runs in 1.2K including 800 bytes for the display.

Prices subject to change without notice. Mass. residents please add 5% sales tax.



*MBI - MERLIN's Basic Intelligence
MEI - MERLIN's Expanded Intelligence

this paragraph are claimed as trademarks of itty bitty machine corporation.] It will be interesting to see what comes out of Evanston in the coming months. Ted closed his talk with the following very quotable passage: "Using a computer should always be easier than not using it."

NCC is a hectic affair, and unfortunately I had to miss several of the other interesting technical sessions in order to work the booth. In the afternoon of June 8, David Ahl had organized a lengthy session on related topics.

But the 1976 NCC in New York is merely a taste of things to come. In the planning stages, under the overall guidance of chairperson Dr Portia Isaacson, is the 1977 NCC, which will be held in Dallas TX at about the same time next year. Portia is an enthusiastic personal computing user, and is a member of the North Texas Computer Hobbyist Group in the Dallas-Fort Worth area. Her enthusiasm for the concept that "personal computing is an idea whose time has come" will be reflected in the 1977 NCC program. One major interest area theme is scheduled to be that of the individual and computing. This will be reflected in the technical sessions, in programming and system design contests for amateurs and enthusiasts, and special exhibits. The show will be a major event on any small computer person's travel calendar for 1977.

On a Subject Nearer in Time, There is Personal Computing '76

As I write this month's editorial, the final preparations for the Personal Computing '76 show are being completed by John Dilks and Dave Jones, who are the principal persons responsible for the event. The list of exhibitors who signed up for booths as of June is shown in this month's advertisement for the show. The technical program will include numerous detailed seminars by computer users and manufacturers' representatives alike.

I'll be giving an opening talk at the start of the show, and other speakers will include Steven B Gray, founder of the original Amateur Computer Society, and Ted Nelson, author of *Computer Lib/Dream Machines*. I expect that, like the earlier and very successful shows at Trenton (Amateur Computer Group of NJ) and Cleveland (Midwest Affiliation of Computer Clubs) this year, the Personal Computing '76 show will be like a gigantic computer club meeting and will give ample opportunity for various manufacturers and vendors to meet potential users,

Continued on page 126

76



76

**Atlantic City, N.J.
August 28th—29th**

**Come To Personal Computing '76
And Meet Fellow Computer Enthusiasts and
Suppliers in Person . . .**

What it's all about!

AMSAT	Software Development
Computerized Music	Micro Computers
Video Terminals	Hardware Development
Kit Construction	Disc Memories
Printers	Computer Comparisons
Computer Games	Interfacing
Digital Tapes	Program Implementation

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

Weekend Fair admission \$5.00 advanced, \$7.50 at door. Admission includes Exhibits, Seminars.

Write for FREE TRIP-KIT to Personal Computing '76 Fair Headquarters, Box 1138 Boardwalk and Michigan Ave., Atlantic City, New Jersey 08404

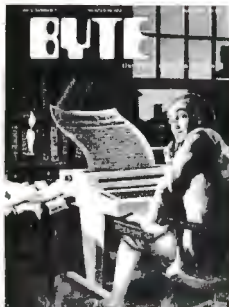
EXHIBITION BOOTHS STILL AVAILABLE—CALL (609) 927-6950

Personal Computing '76 is your opportunity to meet representatives of many of the manufacturers you have seen advertised in BYTE. Among the products you'll get to examine at firsthand are these... many of the items have been committed as door prizes for the drawing which will be held at the show.

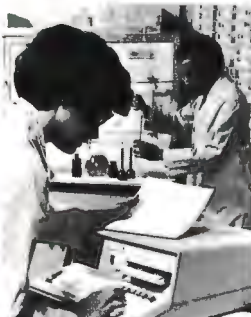
And then, of course, we at BYTE will do our part toward filling the door prize pot, by contributing one lifetime subscription to BYTE beginning with issue No. 1. The first 16 issues will be delivered in a bound volume sometime in 1977, although we'll start the subscription with the current issue if the winner is not presently a subscriber.

IBM Corporation will be present at the Personal Computing '76 show, to demonstrate "live" the 5100 System. This machine is a high technology combination of video display, keyboard and mass storage hardware with high technology software of a complete APL interpreter and BASIC interpreter. Evaluating its features, it is perhaps the ultimate in a small programming and applications oriented computer system. MOS Technology, 950 Rittenhouse, Norristown, PA 19401, will supply this KIM-1 board as a door prize at Personal Computing '76.

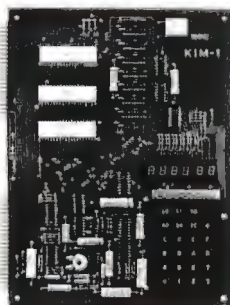
Here is a preliminary list of exhibitors as of June 30, 1976. [Door prize offerings committed as of June 30 are in brackets following the exhibitor's name.]



BYTE



IBM Demonstration



MOS Technology

Manufacturers

Cromemco [TV Dazzler]
 Digital Equipment Corporation
 Digital Group
 EBKA [Familiarizer]
 Economy Company
 E & L Instruments [MMDI-K]
 HAL Communications [MCEM-8080]
 Heathkit
 IBM Corporation
 Lear Siegler [ADM Kit]
 Microterm
 MITS [Altair 680 Kit]
 MOS Technology [KIM-1]
 MOSTEK [F-8 Evaluation Kit]
 National Multiplex
 Ohio Scientific
 [Model 300 Computer Trainer]
 Processor Technology [VDM Kit]
 Prolog
 RCA Laboratories
 Seals Electronics [8K memory board]
 Southwest Technical Products
 [6800 system kit]
 Sphere [310 Kit]
 Technical Design Labs
 [Z-80 CPU Board]
 Vector Electronics Co.
 Wave Mate [Wire wrap gun & wire]
 Williams Electronics
 Wintek

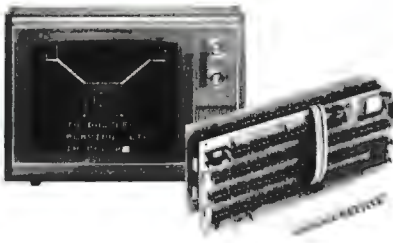
Stores and Retailers

Computer Mart of NJ
 [\$25 Gift Certificates]
 The Computer Store (Boston) [Books]
 Computer Store (LA)
 Computer Systems Center
 Digital Computer Services
 [6502 processor]
 Hoboken Computer Works
 Itty Bitty Machine Corporation
 [Computer Lib]
 NBC Imports [T-shirts]
 Russ Banks Computer Store
 SD Sales

Publishers & Organizations

ARRL, QST
 AMSAT
 Benwill Publications
 BYTE Publications
 [Lifetime subscription]
 The Computer Hobbyist
 Creative Computing [Books]
 Hayden Books
 Interface
 Microcomputer Digest
 Peoples Computer Company
 SCCS

Some of the Door Prizes



Cromemco, home of the TV-Dazzler will supply a TV-Dazzler kit as a door prize at Personal Computing '76. You can probably expect to see two or three dazzlers in operation as you walk around the show, since no product presently available so epitomizes the fascination of personal computing as this color graphics display.



Southwest Technical Products Corporation, 219 W. Rhapsody, San Antonio, TX will provide an SWTPC 6800 computer system kit as a door prize for the show.



Processor Technology Corporation, 6200-B Hollis Street, Emeryville, CA 94608, will give out a VDM-1 module as a door prize at Personal Computing '76 which plugs directly into the Altair/IMSAI/Polymorphics backplane bus structure to produce EIA video such as that shown in the monitor.



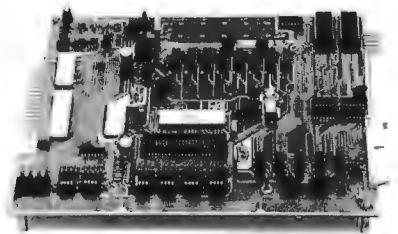
E & L Instruments, 61 First Street, Derby, CT 06418 will make available one MMD-1 computer kit, shown here, as a door prize.



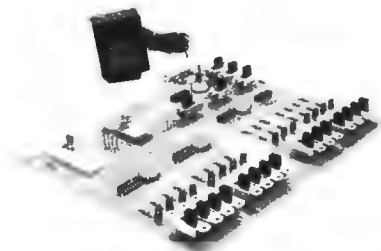
MITS, 2450 Alamo SE, Albuquerque, NM 87106, will provide this Altair 680 in kit form to the lucky winner of one of the door prize drawings.



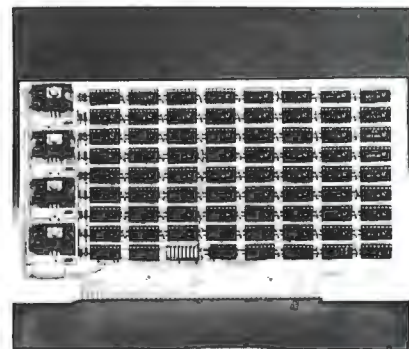
Wave Mate, 1015 West 190th Street, Gardena CA 90248, manufacturer of a wire wrapped computer kit, will provide this wire wrap gun and unwrapping tool as a door prize at the show.



This is the HAL Communications Corporation's MCEM-8080 microcomputer system, a single board computer containing a monitor in 1K bytes of ROM, and 8080-A processor, 1K bytes of programmable memory and system buffering. HAL Communications, 807 East Green Street, Box 365, Urbana, IL 61801 will provide an MCEM-8080 as a door prize at the Personal Computing '76 Show.

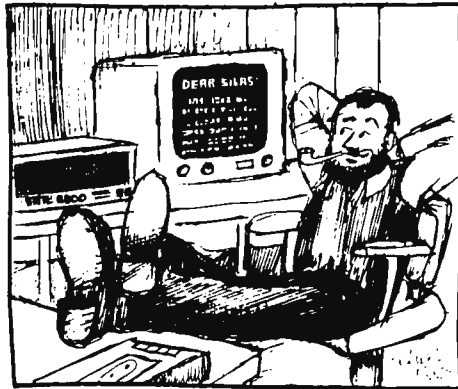


Ohio Scientific Instruments, 11679 Street, Hiram, OH will provide a Model 300 Computer Trainer as a door prize for Personal Computing '76.



MJB Research & Development Corporation's contribution to the stack of door prizes at Personal Computing '76 will be this 8K Altair compatible "Seals" memory board. MJB is located at 36 W 62nd Street, New York, NY 10023.

Letters



STRUCTURE'S WHERE IT'S AT!

In regard to Ronald Herman's article [page 22, June 1976 *BYTE*] on "Programming for the Beginner:" Right On!! I for one am fed up with articles and letters of the "my code is two bytes shorter than yours" variety. It's about time somebody jumped in with a word or two in favor of sanity and rationality in programming style. While the techniques Mr Herman presents are by no means new, their acceptance among programmers of smaller systems is shall we say not as widespread as one might wish. This is detrimental not only to the individual practitioner of the "dirty tricks" school of programming, but to the home computing community as a whole, for such practices can severely restrict the utility and shareability of the software produced, and thereby work to defeat the purpose of hobbyist software interchange standards.

I think that to a large extent it is up to such people as the editors of *BYTE* to encourage neophyte programmers to develop structured, top-down programming practices rather than bit-pinching, memory-grudging trickery. Novices in particular should be warned that code of the latter type can be nearly impossible to debug, and just try and understand it a year from now! Since you, the editors, have been pushing for standardization of various sorts, I think that it would not be too unreasonable for you to exercise a little discretion as to what sort of programming style is advocated in the articles and letters you select for publication. I might also suggest that software-related articles include metalanguage descriptions of the algorithms involved, similar to Mr Herman's pseudo code.

For those old-timers already past the point of no return, by all means don't let me interfere with your work. If the "quick and dirty" approach to coding is your bag, then

go right ahead. Just don't ask me to debug it for you, and please, please don't tempt neophytes down your primrose path.

Gregory P Kusnick
3532 Ramona
Palo Alto CA 94306

P.S. In case you haven't figured this out yet, all my BOMB points for this month go to Ronald Herman.

We're all for structured programming, just surprised at how long it took for someone to write an article on the subject. Ron Herman's article shows a very useful technique for organizing one's programming thoughts. Let it be known: Articles which use a pseudo code representation for programs are highly desirable. Of course, in the cases where what might be called a "dirty trick" is required, we can always partition the problem so that the tricks are off isolated in some subroutine.

DISPLAY WANTED

I recently picked up the May and June issues of *BYTE* at the "Rochester Hamfest" at Rochester NY. I was very impressed with *BYTE* as it appears to be an excellent magazine for the "computer hobbyist" or "ham RTTY operator."

I would like to see an article on a 72 character per line TV display which would then be compatible with Teletype line lengths such as the model 32 and 33s, etc.

Vincent R Staffo
Rochester NY

ATTENTION HAMS! REQUEST FOR A HAM NETWORK

I am a "charter subscriber" to *BYTE* and have been in and around the radio/electronics business for quite a while, since 1941 in fact. I have an Amateur Extra Class license and operate all bands, SSB, CW and RTTY. I am also CE for a St Louis directional AM and stereo FM station (20 years).

It seems to me that the opportunities in the microprocessor field are virtually unlimited but also that the pitfalls are of the same order. I also believe that a large number of your readers must also be hams and RTTYers. I would like to see *BYTE* magazine promote a net type of operation on the ham bands to promote the exchange of information regarding microprocessor systems and peripherals. A few minutes of

It's about time somebody jumped in with a word or two in favor of sanity and rationality in programming style.

Continued on page 93

If you thought a rugged, professional yet affordable computer didn't exist,

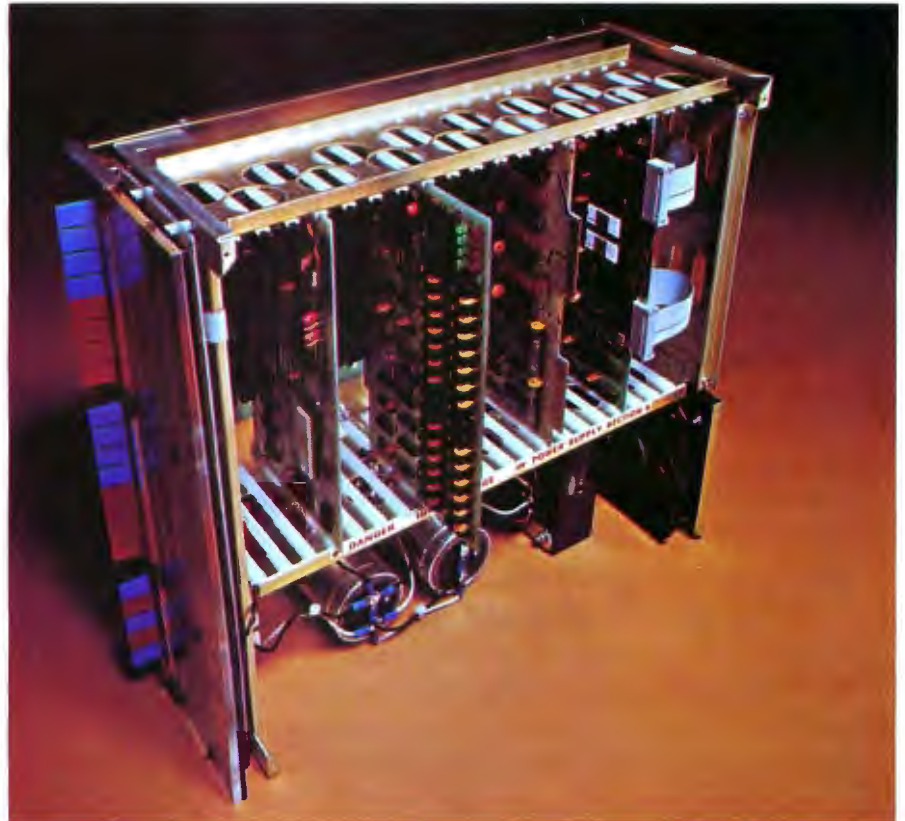
think IMSAI 8080.

Sure there are other commercial, high-quality computers that can perform like the 8080. But their prices are 5 times as high. There is a rugged, reliable, industrial computer, with high commercial-type performance. The IMSAI 8080. Fully assembled, it's \$931. Unassembled, it's \$599. And ours is available now.

In our case, you can tell a computer by its cabinet. The IMSAI 8080 is made for commercial users. And it looks it. Inside and out! The cabinet is attractive, heavy-gauge aluminum. The heavy-duty lucite front panel has an extra 8 program controlled LED's. It plugs directly into the Mother Board without a wire harness. And rugged commercial grade paddle switches that are backed up by reliable debouncing circuits. But higher aesthetics on the outside is only the beginning. The guts of the IMSAI 8080 is where its true beauty lies.

The 8080 is optionally expandable to a substantial system with 22 card slots in a single printed circuit board. And the durable card cage is made of commercial-grade anodized aluminum.

The IMSAI 8080 power



supply produces a true 28 amp current, enough to power a full system.

You can expand to a powerful system with 64K of memory, plus a floppy disk controller, with its own on-board 8080—and a DOS. A floppy disk drive, an audio tape cassette input device, a printer, plus a video terminal and a teleprinter. These peripherals will function with an 8-level priority interrupt system. IMSAI BASIC software is available in 4K, that you can get in PROM. And a new \$139 4K RAM board with software

memory protect. For the ultimate in flexibility, you can design the system for low-cost multiprocessor, shared memory capability.

Find out more about the computer you thought didn't exist. Get a complete illustrated brochure describing the IMSAI 8080, options, peripherals, software, prices and specifications. Send one dollar to cover handling.

Call us for the name of the IMSAI dealer nearest you.

Dealer inquiries invited.



IMSAI

IMS Associates, Inc. B-9
14860 Wicks Boulevard
San Leandro, CA 94577
(415) 483-2093

New Intel microcom system costs, increase

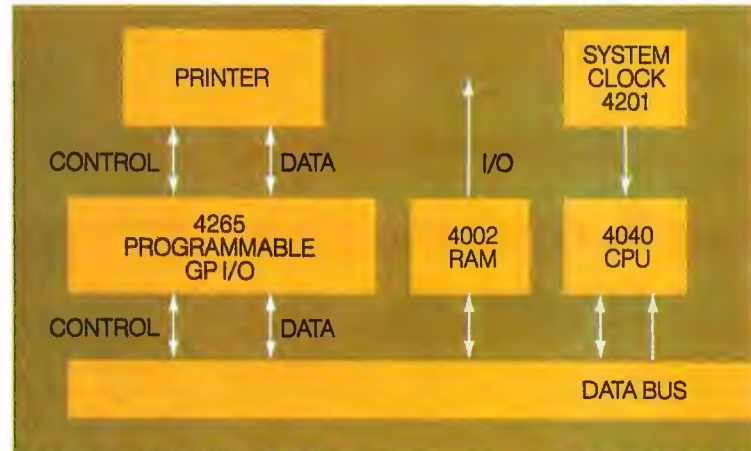
Intel has two new LSI components for the MCS-40™ microcomputer system which will help you cut system costs, increase throughput and reduce the number of components you have to stock for I/O interface requirements. The new Intel 4269 Programmable Keyboard Display and the 4265 Programmable General Purpose I/O devices eliminate the large number of discrete SSI/MSI components previously required for keyboard, control panel, indicator array, alphanumeric display, printer, communications and other I/O interfaces. These new LSI parts increase system throughput up to 50%, and make it easy to add standard Intel memory and system peripherals.

The 4269 Keyboard Display can be software programmed to interface to various keyboard and display elements and makes it possible for you to eliminate fifteen or more discrete components.

It significantly increases system throughput since it performs the scan, storage, refresh, and other simultaneous keyboard/display tasks previously required of the 4004 or 4040 CPU.

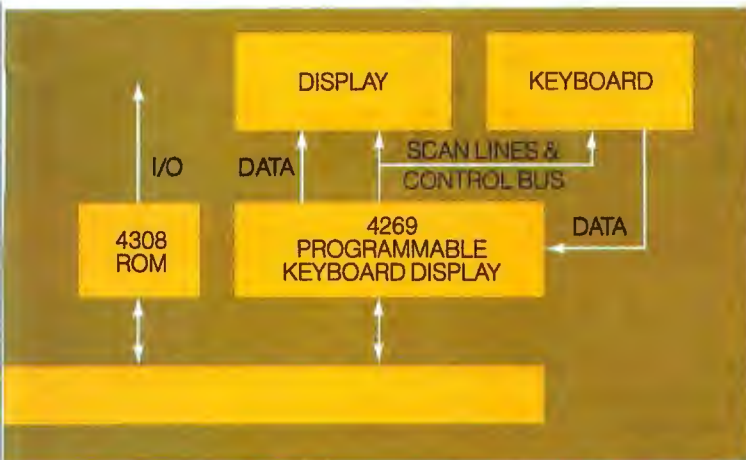
When programmed as a keyboard or line sensor input interface, the 4269 can scan up to 64 key closures or lines. When a key closure is detected, the 4269 generates a system interrupt and stores up to eight characters in its first-in/first-out buffer before requiring CPU service.

In alphanumeric applications, the 4269 eliminates the need to use the CPU



* Self-Scan is a registered trademark of the Burroughs Corporation.

puter I/O devices cut throughput up to 50%



and system memory for display refresh since the necessary memory and control are built in. One 4269 can operate and refresh alphanumeric displays or indicator arrays with up to 32x4 digits, 16x8 characters or any configuration of 128 elements or lights, including a 20-character Burroughs Self-Scan* Display.

The 4265 General Purpose Programmable I/O is ideally suited to implement custom interface requirements. Up to four devices can be controlled by the CPU. Each 4265 has 16 I/O lines organized into four ports which can be used in 14 different data transfer and control/interface organizations. The 4265 provides synchronous/asynchronous control, buffer inputs and outputs, bit set and bit reset capability on output port lines and byte transfer control. It can be used to add industry standard RAM memory such as Intel's 5101 CMOS RAM. And the 4265 lets you use system peripherals such as the 8251 Programmable Communications Interface (USART), the 8253 Programmable Interval Timer or the 8214 Priority Interrupt Control Unit.

To order, contact our franchised distributors: Almac/Stroum, Components Specialties, Components Plus, Cramer, Elmar, Hamilton/Avnet, Industrial Components, Liberty, Pioneer, Sheridan or L.A. Varah. For your copy of our MCS-40™ System brochure, use the bingo card or write: Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051.

intel[®]

Microcomputers. First from the beginning.

Are you the proud author of a piece of hobbyist software? If so, you are in the same class as the author of a novel, a play or any other kind of salable writing.

Are You an Author?

Calvin N Mooers
Rockford Research Inc
140 1/2 Mt Auburn St
Cambridge MA 02138

Softlifting is a white collar (no gun) crime which is easy to commit and hard to detect.

Are you the proud author of a piece of hobbyist software? If so, you are in the same class as the author of a novel, a play or any other kind of salable writing. If your software is good, and other people want it, it could be worth something to you. Dollars! Money!

This is the first of several articles in *BYTE* describing details of an owner's rights in computer software. It is written from the hobbyist and software writer's standpoint. While I am not a lawyer with a formal legal degree, I have some practical credentials for discussing these matters. I have 30 years experience in studying this topic and in advising my lawyers concerning my own patent, trademark and software copyright problems. I have previously written on this subject as it concerns the professional data processing markets (in particular see my article "Computer Software and Copyright" in the March 1975 issue of the *ACM Computing Surveys*). Any lawyers or individuals who read my views on the subject and have comments to make are invited to respond in writing to me or via the *BYTE* letters column.

Maybe you are not yet an author, but only a user of personal computing software. Then you, like everyone else, need more and better software in order to use and enjoy your computer. Yet where are the suppliers?

Why do some of the potential suppliers seem to be holding back?

There is a partial answer to this question. You undoubtedly know that a few bad apples are rapidly giving all computer hobbyists a very bad name. For example, it now appears that more copies of Altair's BASIC have been pirated than have been legally sold. (See the letter by Bill Gates on page 3 of the February 1976 edition of *MITS Computer Notes*, the March April 1976 issue of *People's Computer Company* and widely published elsewhere in newsletters and club bulletins.)

Software piracy is a white collar (no gun) type of crime. It is easy to commit and hard to detect. As a crime it is in the same class as shoplifting, or the use of a "blue box" on a telephone to make illegal freebie worldwide telephone calls. Software piracy is a crime ethically because it extracts creativity and effort from the author(s) of software against their will and thereby robs them of their property. It is a crime legally to the extent that existing legal mechanisms are available for the protection of software by owners who desire some form of recompense through sales.

As a software user who has come by his or her software honestly, what can you do with your software? What can you not do? From my experience talking to many people, I have concluded that very few persons really know what they can do and cannot do within the present legal definitions. There is much misinformation and little real knowledge. One target of this article is you who are software users.

Furthermore, it is you — the great majority of honest users — who will by your

A Note of Interest

The concern with protection of software creations is by no means confined to the personal computing field. At about the time this article was being edited, a survey entitled "Development of an International System for Legal Protection of Computer Programs" by Oliver R Smoot appeared in the April 1976 edition of Communications of the ACM, page 171 of the volume 19 number 4 issue. The content of the report was an informal account of a recent (June 1975) meeting of an international committee named the Advisory Group of Non-Governmental Experts on the Protection of Computer Programs, held under the auspices of the World Intellectual Property Organization in Geneva, Switzerland. . . CH

peer pressure provide the most effective way for putting a stop to the bad apples who steal software. If the software piracy threat can be stopped, more and better software on the market will be the result.

Methods of Protection

So you are an author, and you have this great little piece of hobby software (or business software). It is so new, it is still a secret between you and your computer. Even your best friend hasn't been provided with a copy. It is all yours.

(We presume that this software was created in your own basement, and not on your company's time or computer. We presume that your business arrangement with your employer allows you to hold ownership in your own out-of-hours software creations. (Maybe you had better dig out that copy of your employment contract, and read the fine print on this matter.) We also presume that your creation wasn't copied from someone else's copyrighted software or documentation.)

For the moment this new software is all yours, and you legally and completely own it. The courts will back you up to the hilt. So much for the good news.

Now for the bad news. Exactly what is it that your own? Should you try to protect your new software? If so, how? How can you take your software out of your basement without losing your ownership? If you can find a buyer, what is it that you really sell? What steps (patent, trademark, copyright, trade secret, or other method) should you take to protect your new property?

The easiest way out is for you to give your software away, thereby forfeiting ownership. You won't have any problems as an owner. For some kinds of hobby software, this is the preferred course. After all, a

hobby is mainly for the fun of the thing, and you don't really expect to make any money.

But what if you really did put in an awful lot of time, and worked up some documentation, got all the bugs out, and have something that you think others would really like to pay some money for. What then?

If you can find a buyer for your software, someone who is willing to purchase all rights to your software, sight-unseen, for some nice round number, then your worries are also completely over. The buyer can worry about protecting it, and selling it. However, most buyers will want to examine the goods before buying, so you are back to where we began: If you want to sell your software creations, how do you protect your property before and after the sale? Thus we get down to basics.

Secrecy

If you don't let your software out of your basement, and you don't let anyone else see or have access to it, even by data line, and you tell no one about it, then you are probably completely protected (barring a computer-oriented burglary). This is the method of protection by secrecy. It is completely effective. The best people do it: IBM is reputed to have many more secret developments filed away in their labs than all they have ever published or marketed.

Can the "idea" of your new software be protected? The answer is clearly "No," particularly if the software is to be marketed to a number of customers. Forget it.

Patent

Can the new software be patented? This method of software protection might seem to be a hopeful way, since a patent protects the processes or devices used to carry out an

When you create or purchase software, exactly what is it that you own?

The easiest way out of software protection problems is for you to give your software away, thereby forfeiting ownership. This is guaranteed to cure any problems you may have as an owner. For some kinds of software this is the preferred course. After all, a hobby is mainly for the fun of the thing and you don't really expect to make any money.

**With software, trade secret protection is not likely.
You simply cannot sell copies of the secret and keep
the secret at the same time.**

inventive new idea. In practice, the answer is again “No.” There are two reasons. The first is that your software is almost certainly not sufficiently original in concept to be patentable. The second is that during the past ten years the courts — including the Supreme Court — have had as much trouble in agreeing about how to deal with software patents as they have had with the equally intriguing topic of pornography (What is it? Should it be allowed?). Finally, getting a patent will cost you an arm and a leg (more than \$1000), and will take a minimum of two to three years (if ever) to get. Again, forget it. Let the big corporations fight this battle.

Trade Secret

If you are going to sell your software to more than one customer for hobby computer use, you can also forget the “trade secret” method of protection. This method works for large companies if it is a manufacturing process or formula that can be kept behind locked doors (like the formula for Coca Cola). But with software, you simply cannot sell copies of the secret, and keep the secret at the same time.

Trademarks

Trademarks are another fascinating legal device for your protection as an entrepreneur. Trademark laws protect your use of a special mark (your trademark) on your goods or services. The purpose of your trademark on your software is to inform the buying public that the goods or services so marked and sold are manufactured or provided by you, and not by someone else. If you are interested in the game of software selling, you should seriously consider using a trademark (or service mark) to help protect you from unfair imitators, since there are legal sanctions to prevent them from using your mark. However, useful as it is, a trademark cannot be used to protect the software itself from theft.

Copyright

How can your new software be sold, and still be protected from “soft-lifting” (equivalent to “shoplifting” in another context)? How can it be protected from the pirates? In my estimation, the best tool we have is copyright, which is the same legal tool used

by all other authors — authors of novels, plays, and other kinds of creative written works.

Copyright is unbelievably swift and cheap. What you do is to place the magic incantation “Copyright 1976 J Jones” (if your name is J Jones) at the top of the first page or title page of your software listing, and then give a copy so marked to a friend. You now have a copyright!

It is like magic. The very instant that you place your program, or listing, or tape, or documentation on sale or put it into distribution *with this notice on it* you become the proud owner of a US copyright in the software so marked.

You do not need to file papers anywhere to obtain your legal copyright protection! (However, more about this topic later, and about filing a copyright claim in the US Copyright Office.)

However, if you first distribute any copies of your software without this copyright notice, then you have lost your rights forever. By first distributing your software without a copyright notice, you thereby tell the world that you renounce your ownership, and that anyone thereafter may copy your software at any time with no need for permission.

It is best if the notice “Copyright 1976 J Jones” (with the correct name and date) is placed in a comment line at the head of the program. It should also be stamped or handwritten on all tapes and boxes containing cards or tapes for the software. It is important that it be placed on the *title page* of all documentation. All copies, whatsoever, going out should bear your copyright notice.

What Copyright Means

A copyright means that no one, without your permission, is legally authorized to make copies of your copyrighted software. In the language of the law, you now have:

“the exclusive right to print, reprint, publish, copy, and vend the copyrighted work; to translate the work into other languages or dialects, or to make any other version thereof . . .”

Since one can't run a computer without first using a copy of the software to make a data pattern inside the computer, you can begin to see how copyright can protect you.

In my estimation, the best tool we have is copyright, which is the same legal tool used by all other authors — authors of novels, plays, and all other kinds of creative written works.

SCELBI SOFTWARE



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

A complete language for "8008"/"8080" systems including source listings, routines, flow charts and more!

SCALBAL. SCientific ELementary BAasic 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** ppd.



Here's SCELBI's First Book of Computer Games for the 8008/8080. Action-packed. And fun.

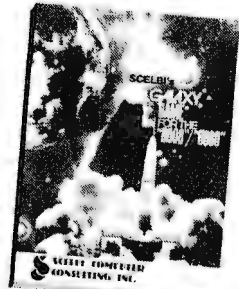
Try to beat the computer at its own game. Here's the first complete machine language computer manual for computer games to include source listings, flow charts, routines and more. **Space Capture** — You against the computer using "search and destroy" strategy to shoot down roaming alien spaceships in outerspace. **Hexpaw** — a mini-chess game that lets the computer make mistakes... but only once. **Hangman** — an updated version of the great kid game. Computer selects words at random from long, expandable list. Try to beat it in 8 moves or less. Illustrated. Fun extras to put your computer to challenging, competitive, fun use.

Order yours today! **14⁹⁵** ppd.

SCELBI's new microcomputer Game Book of Outer Space...

GALAXY

Captain your own crusading starship against the logic of your "8008" or "8080". You must destroy a random number of alien ships without running out of stardates, out of fuel, out of ammunition or out of the galaxy. Plan your mission. How much fuel is used for your warp factor? Don't run into roaming stars that can damage your ship. **Suddenly! Condition Red! Alien in sight!** How big is he? Fire a phasor or torpedo! He's damaged or destroyed. But, you've used valuable fuel. That's just the beginning. A new game every time. Complete program book in machine language for 4K memory, including source listings, flow charts, routines, etc. Great intergalactic adventure and fun.



Order "GALAXY" today! **14⁹⁵** ppd.

"8080" SOFTWARE MANUALS

Three new basic, complete "MUST" manuals! "8080" Assembler Program operates easily in 4K bytes of RAM (including symbol table). Unique feature: How to accept mnemonics related to "8008" based machines on "8080". Includes all major routines, pertinent flow charts, highly com-

mented assembled listing and more. \$17.95. "8080" Editor Program identical to "8008" Editor, with machine codes for "8080". \$14.95. "8080" Monitor Routines same functions as "8008" specifically developed to utilize expanded capabilities of "8080". \$11.95.



17⁹⁵ ppd.

14⁹⁵ ppd.

11⁹⁵ ppd.

SAVE \$5.35! BUY ALL 3 FOR ONLY \$39.50!

MACHINE LANGUAGE PROGRAMMING FOR THE "8008"

(and similar microcomputers)

Here's the detailed, basic manual you need to develop today's machine language programs. 170 pages. Illustrated. Easy-to-read, understand. Most techniques applicable to other micro-computers, including "8080". Floating-point arithmetic package is worth the price alone!

Order today!

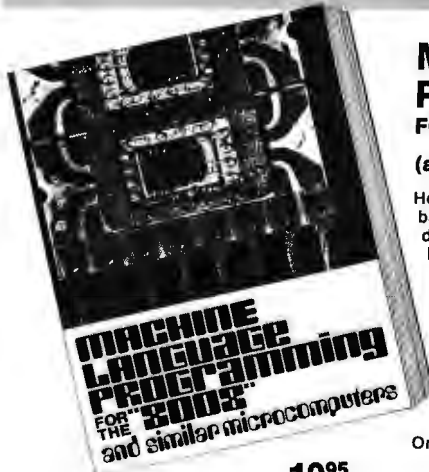
- Detailed presentation of "8008" codes
- Flow Charts
- Mapping
- Floating-point Package
- Debugging
- Basic programs: loops, counters, masks
- Organizing Tables
- Editing/Assembling
- Math operations
- I/O, Real Time Programming
- Maximizing memories
- And lots more.

Master Charge, Postal and Bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing, specifications, availability subject to change without notice. Prices for U.S. and Canadian delivery at book mailing rate. Add \$2.50 for each publication if Priority Air Service (U.S.) desired. Foreign orders add \$6.00 for each publication.



SCELBI COMPUTER CONSULTING INC.

1322 Rear Boston Post Rd., Milford, CT 06460
Telephone: 203/874-1573



19⁹⁵ ppd.

The best way to explain what this language of the copyright law means is to describe what you as the purchaser of the software can and cannot do with copyrighted software if you wish to stay within the law.

After you have bought the copyrighted software, you may read your copy, throw it away, re-sell it, give it to a friend, memorize it, burn it – or do just about anything except to “make a copy.” You own the paper it is written on, you own this particular copy of a program; but you don’t have the legal right to make further copies! This is what copyright is all about.

Of course, computers were not with us in 1909 when the current copyright law was written. But even back in 1909 they had “high technology” for the time— linotypes and high speed printing presses. Our present computers are merely another form of high technology machines, and they also use and produce printed material. The copyright law applies to computers too.

Making a computer listing is both making a “copy” and “printing” or “reprinting” copies of a program in the language of the copyright law. Giving a listing to a buddy is “publishing,” even if no money is involved in the act. Selling the listing, say for 50¢, is “vending” a copy. Unless you have permission, in one way or another, from the copyright owner, doing any of these things is called an “infringement” of the copyright.

If you have a copyrighted program listing in 8080 language, and you convert it to 6800 language, this, in my opinion, is also an instance of “translating the work into another language or dialect.” Taking the program and making major changes in it is, in my opinion, an instance of “making another version thereof.” Again if you have not been authorized to do so, these actions would also be infringements.

It is evident that when you buy some software for running in your computer, you must be allowed to do a number of things that might otherwise be forbidden by the copyright laws. Thus when you buy copyrighted software you should get from the seller, either as a definite written statement, or more usually as an implied understanding between the software seller and yourself, a clear indication of what things you can do with your purchased software.

The major computer manufacturers seem to take particular pains to make these matters very clear. For example, they tell you that you can copy the software into your computer (they may even specify the serial number of the computer) for purposes of running the computer. They may also

permit you to make a limited number of copies of the listing, but only for purposes of your own study and maintenance. They will often tell you that you are not authorized to furnish copies of the software, listing, or documentation to any other person. To maintain their control, they may even retain ownership of the physical listings, tapes, and documentation.

These generally-authorized actions permitted with copyrighted software are now usually understood by the sellers and buyers of software in the data processing field at large. Therefore if you as a buyer have any intent of making a wider use of the software, you should be sure to get your license or sale agreement to specify all the actions that you wish to be permitted.

Copyright, What You Can't Do

A person has bought some software. What can he or she do, and what can't he or she do with it? What are the penalties?

The usual understanding between the seller and the buyer of the software is that the person buying the software is allowed to copy the software into his computer to make it run, that he or she is allowed to make a limited number of private copies for safety back-up or maintenance, but that the buyer shall not furnish copies of any kind to any other person.

To go beyond these authorized limitations, and to do other things (make and distribute other copies, other versions, or translations) with such purchased software, is wrong. It is not a city ordinance or state law that is broken. It is a Federal law duly passed by Congress. It is called “copyright infringement.”

Three things can happen to a copyright infringer:

First. The infringer may get a guilty conscience, but otherwise get away with an illegal act. As with shoplifting, which is also difficult to detect, this is the usual outcome. However if enough people in the hobby field go in for “soft-lifting,” the software producers might get the message, and stop making software available to hobbyists. In the hobby field, and for good reason, this is already beginning to happen in at least one well known case.

Second. If the software owner finds out about the infringer’s illegal acts, the infringer may be in for big trouble, especially if the owner feels sufficiently damaged by the illegal activities. The software owner may decide that there has been so much infringement that it is now time to “give those guys a lesson” and to make an example of one of



Now, you can buy an Altair 8800 or Altair 680 computer kit right off the shelf. Most all Altair options, software and manuals are also available. The MITS Dealer List below is just the beginning:

off the shelf

RETAIL COMPUTER STORE, INC.
410 N.E. 72nd
Seattle, WA 98115
(206) 524-4101

COMPUTER KITS
1044 University Ave.
Berkeley, CA 94710
(415) 845-5300

THE COMPUTER STORE
(Arrowhead Computer Co.)
820 Broadway
Santa Monica, CA 90401
(213) 451-0713

THE COMPUTER SHACK
3120 San Mateo NE
Albuquerque, NM 87110
(505) 883-8282

GATEWAY ELECTRONICS
2839 W. 44th Ave.
Denver, CO 80211
(303) 458-5444

GATEWAY ELECTRONICS
8123-25 Page Blvd.
St. Louis, MO 63130
(314) 427-6116

BYTE'TRONICS
Suite 103 - 1600 Hayes St.
Nashville, TN 37203
(615) 329-1979

CHICAGO COMPUTER STORE
517 Talcott Rd.
Park Ridge, IL 60068
(312) 823-2388

MARSH DATA SYSTEMS
5405-B Southern Comfort Blvd.
Tampa, FL 33614
(813) 886-9890

MICROSYSTEMS
6605A Backlick Rd.
Springfield, VA 22150
(Washington DC area)
(703) 569-1110

THE COMPUTER SYSTEMCENTER
3330 Piedmont Road
Atlanta, GA 30305
(404) 231-1691

THE COMPUTER STORE, INC.
120 Cambridge St.
Burlington, MA 01803
(617) 272-8700

THE COMPUTER STORE OF ANN ARBOR
310 East Washington St.
Ann Arbor, MI 48104
(313) 995-7616

COMPUTER PRODUCTS UNLIMITED
4216 West 12th St.
Little Rock, AR 72204
(501) 666-2839

the COMPUTER STORE, INC.
63 South Main St.
Windsor Locks, CT 06096
(203) 871-1783

the COMPUTER STORE of NEW YORK
55 West 39th St.
New York, NY 10018

THE COMPUTER ROOM
3938 Beau D'Rue Drive
Eagan, MN 55122
(612) 452-2567

NOTE: Altair is a trademark of MITS, Inc.

MITS, Inc. 2450 Alamo S.E. Albuquerque, N.M. 87106

the infringers. This could lead to a lot of rancor, and you certainly wouldn't want that to happen to you!

More about the third possibility later.

Getting Caught Can Be Bad News!

Scenario: A US marshal appears at the infringer's door. (Copyright is a federal law, so it goes to a federal court which uses federal marshals to convey its "greeting.") The marshal carries a legal paper called a "complaint." The infringer has now "had it." Like it or not, he must accept the complaint. A full legal response to it is required within 20 days. Whatever the infringer does now could cost plenty of money, time, and grief. Also bad publicity if he or she was chosen to be an example.

Probably the best thing an infringer can do at this stage is to make peace — but fast — with the owner of the infringed copyright. All the other alternatives have high risk and cost. Trying to fight the case (assuming a real "softlifter" is involved) will cost buckets of money for legal fees, whether the suit is won or lost. To not fight, or to default on the complaint, means the infringer will receive an automatic default judgment which could cost a minimum of \$250 for each alleged instance of copyright infringement.

In cases of infringement, the copyright law provides for mandatory damages to be paid to the owner which "shall not be less than \$250 nor more than \$5,000" for each act of infringement. Each time a copy had been run off for a friend would probably constitute a separate act of infringement. The fact that no money was involved in the deal does not excuse the infringement. Neither is ignorance of the law an acceptable excuse.

Profits and Prison

What about the guy who knew the software was copyrighted, but still made and *sold* copies of the software hoping to make a tidy profit? Maybe he wanted to buy some super disk system with his ill-gotten gains. When he copies wilfully and sells the stolen copyrighted software, it is a federal crime (yes, I said *criminal*). The newspapers recently had accounts of the FBI going after a nationwide audio-tape bootlegging ring. Audio tape bootlegging is a similar kind of copyright infringement.

Although a hobby computer program pirate would probably have to go large scale before the software owner could get the reluctant federal authorities to move, it could happen. The law provides that if a copyright infringement is wilful and for

profit, the convicted infringer "shall be punished by imprisonment for not exceeding one year or by a fine of not less than \$100 or more than \$1,000, or both, in the discretion of the court"

The Effects of Software Piracy

But what if the software owners don't take the legal sanctions route. What then? After all, it is terribly expensive for them to do so. What we can expect to happen — and now is happening — is that software producers will shun the hobby software field. Already the bad apples in the hobby field have produced a devastating impact. Bill Gates, producer of the Altair BASIC, in his "Open Letter to Hobbyists" mentioned earlier, tells of his sad experience.

According to Bill, he and two associates produced the Altair BASIC, investing three man years and burning up \$40,000 in computer time. It was to be sold on commission through MITS for use with Altair computers. Gates now finds that many of the "users" he talks to praise his BASIC very highly, but few of them can admit that they bought the copy they use. He says, "As most of the hobbyists must be aware, most of you steal your software." He is bitter, and says that the return for his group was less than \$2 an hour for the great amount of time they put into the programming, debugging, and documentation required to make a first class package. He continues, "We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists."

The software people I have talked to agree with him. This means that prudent software producers will begin to sell complex packages only to commercial and business buyers, since such buyers would be less likely to break the law. This means that soon only the hardware and kit manufacturers will be willing to supply hobbyist systems software. If so, hobby equipment prices must rise to absorb the cost of the bundled software. Another result will be that the hobbyist will be severely limited in choice of software. Hobby systems will be limited mainly to what the manufacturer wants to produce.

The hobbyists can do a lot to clean up this situation. They can first learn what the owner's rights are in his software. They can expose the bad apples who are peddling stolen property. They can exclude bad apples from hobby meetings until they change their ways. The pressure from one's "peers" can be very powerful if it is properly applied. ■



Rickey's tackling the SDK-80 microcomputer kit for his next science project.

Rickey likes soccer, lizards, hot fudge sundaes, skateboards and microscopes. He can't decide if he'd rather be Franco Harris, Bobby Fischer or Jonas Salk.

When his Dad brought home the Intel SDK-80 microcomputer systems kit, Rickey helped him put it together. It took only four hours. Everything was there. The 8080 CPU, RAM, PROM, programmable, I/O, a printed circuit board with all those capacitors and resistors and the other things that go with it. The

best part was the instruction manuals. Every step was clearly explained. It was easy. The programming part looked especially interesting. So simple. Just imagine talking to a computer.

The big thrill came on Saturday when they went to his Dad's office to use a terminal. When they connected the SDK-80 to the teletypewriter they got a printout. That was exciting. Within an hour they were talking to the computer, then inventing games. They stayed all day.

Now Rickey is building a micro-

computer of his own. He may be the first kid on his block with his own computer. Thanks to a \$350 low interest loan from his Dad.

If you're interested in being the first on your block to have a microcomputer, contact your Intel distributor: Almac/Stroum, Component Specialties, Components Plus, Cramer, Elmar, Hamilton/Avnet, Industrial Components, Liberty, Pioneer, Sheridan, or L. A. Varah.

Microcomputers. **intel**[®]
First from the beginning.
3065 Bowers Ave., Santa Clara, California 95051.

How Far Can You Go

Find out—by adding 8080 compatible

3P+S Input/Output Module

Our 3P+S is a simple inexpensive answer to handling a variety of peripheral devices. It has **two 8-bit parallel I/O ports**, with full handshaking logic. Plus, it has a **serial I/O port**, with a data rate range from 35 to 9600 Baud. Simultaneously handle a paper tape reader, a keyboard, a TV terminal, a tape punch, **and** a telephone coupler! Available with premium grade low-profile IC sockets, **\$149**. Owner's Manual, **\$4.95***

ALS-8 Assembly Language Operating System

Just turn on the switch, and instantly you have the power to **write, edit, assemble, de-bug, and run your own programs**. The **ALS-8** is the most **useful** software development tool available today. Optional firmware includes **SIM-1**, an interpretive simulator, that simulates programs without running them in real time. Thus, errors encountered during testing do not endanger your entire system. The **TXT-2** firmware adds the dimension of text editing. Insert, delete, move entire lines or single characters, and much more! **ALS-8** (assembled only), **\$425**. **SIM-1**, **\$95**. **TXT-2**, **\$95**.

VDM-1 Video Display Module

If you're setting up just the system you really want, don't fall short by limiting its communicating ability. The **VDM-1** is an **ultra-high speed** video output device. Its **16 display lines** have **64 characters each, upper and lower case**. 1024 bytes of random access memory are on the card. It scrolls up or down, **even to 2000 lines per minute!** Any combination of the 1024 cursors can be displayed as black-on-white or vice versa. **Free terminal mode software is included**, along with premium grade, low-profile IC sockets. **\$199**. Owner's Manual, **\$4.95***

8KRA Static Memory Module

We now offer a low-power static memory module, with a **full 8192 bytes of memory**.

It has an exclusive KSET™ address selection DIP switch, so you can conveniently set address boundaries in increments of 1K. **Our low-power RAM's typically require one-third less power** than those commonly used by our competitors. They'll even retain memory for 4-5 hours when powered by two "D" flashlight cells. **On-board recharging circuitry and battery connectors** make it possible to protect your data against sudden power loss. Each RAM has its own IC socket, too, for easier assembly and repair. **\$295**. Owner's Manual, **\$4.95***

2KRO EPROM Module

The **2KRO Read Only Memory** will accept up to eight 1702A or 5203 EPROM's (not included), providing 2048 eight-bit words of non-volatile storage for monitor, executive, loader and other programs. Programming services available from your dealer or write us for details. **\$65**. Owner's Manual, **\$4.95***

4KRA Static Memory Module

The 4KRA (4096 bytes) was our first static memory module. It's still very popular, and uses the same **low-power static RAM's** as the 8KRA. Plus, we've added a **DIP switch**, and **every RAM now has its own premium grade, low profile IC socket**. On-board **recharging circuitry** (with battery backup) makes it possible to retain memory for 8-10 hours during power failure. **\$159**. Owner's Manual, **\$4.95***

MB-1 Mother Board

Our single piece **Mother Board** for the Altair 8800 gives you 16-card capacity in one single installation. Available with 12 (**MB-12**) or 16 (**MB-1**) slots. **\$70**. (Discontinued; limited to stock on hand.)

Wire Wrap Board

Do your own wire wrap prototyping with the **WWB Wire Wrap Board**. Up to 62 16-pin sockets or various combinations of 14, 16, 24 and 40-pin sockets. **\$40**.

EXB Extender Board

The **EXB Extender Board** allows accessibility in servicing any 8080-compatible module. **\$35**.

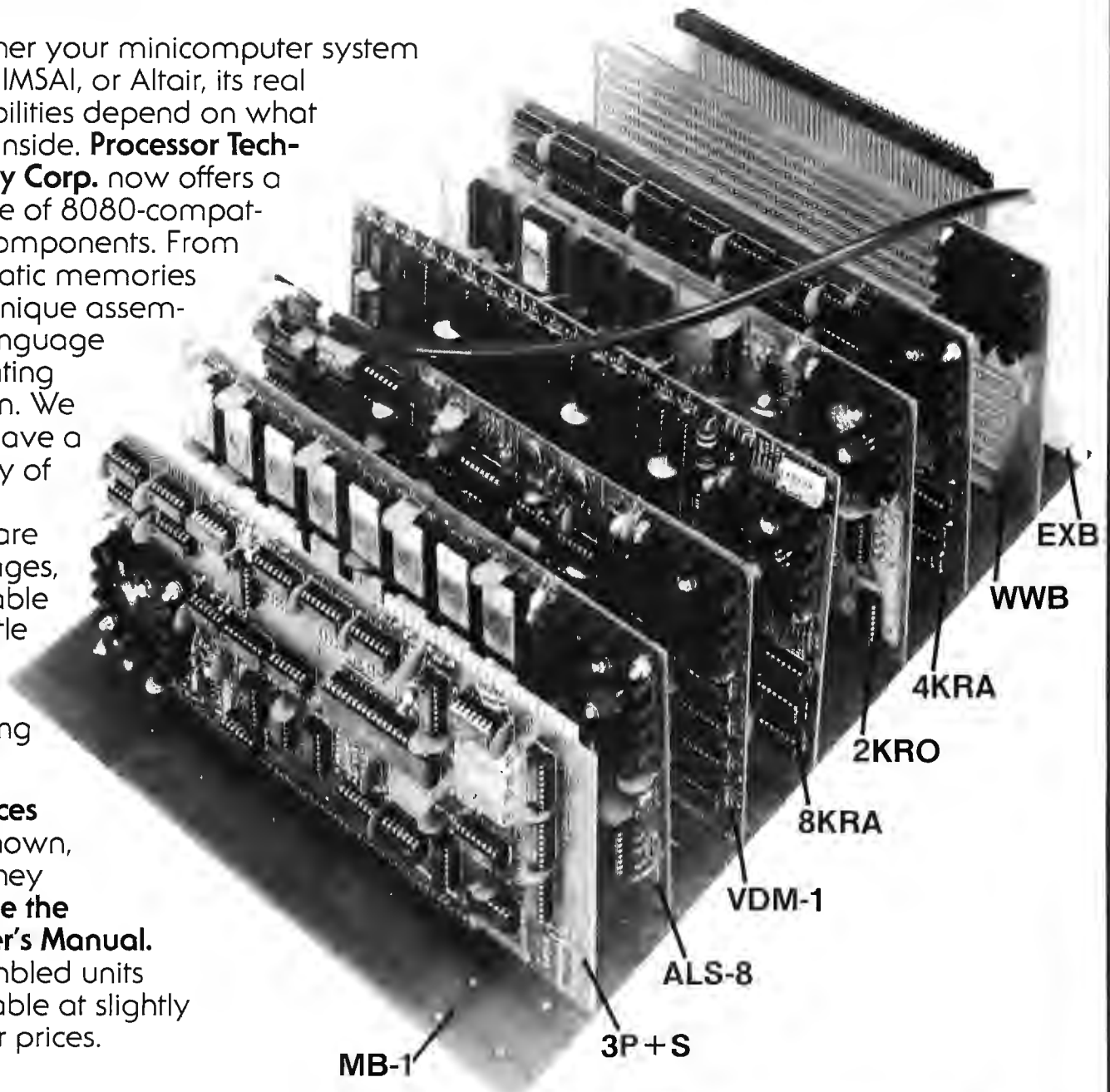
*Fully descriptive Owner's Manual available separately. Price refundable with purchase of kit.

On The 8080 Bus?

products from Processor Technology.

Whether your minicomputer system is Sol, IMSAI, or Altair, its real capabilities depend on what goes inside. **Processor Technology Corp.** now offers a full line of 8080-compatible components. From fine static memories to a unique assembly language operating system. We also have a variety of useful software packages, available for little more than copying costs.

Kit Prices are shown, and they include the **Owner's Manual**. Assembled units available at slightly higher prices.



Write Us for further details on all our 8080-compatible products, and for the dealer nearest you.

Processor Technology
6200-V Hollis Street
Emeryville, CA 94608

Build This Mathematical Function Unit

Part I: Hardware

R Scott Guthrie
1374 Franchere Pl
Sunnyvale CA 94087

Why Use a Calculator?

The small computer system designers and computer hobbyists have a wide variety of 8 bit, 12 and 16 bit microprocessors to choose among for their design problems. Most of these units are capable of performing binary fixed point arithmetic. However, there are many applications which require floating point arithmetic operations with greater precision and extended mathematical functions. Sophisticated software routines or complicated hardware must usually be developed and debugged in order to achieve these goals. However, without undue complication the floating point arithmetic func-

tions can be achieved through the use of the existing hardware found in calculator chips.

The more sophisticated calculator chips are capable of trigonometric, logarithmic and other special functions as well as the standard add, subtract, multiply and divide capabilities and can be easily interfaced to microcomputer systems.

My objective with this project was to develop a method of interfacing a calculator chip to a microcomputer, thus relieving the problems of writing the software for arithmetic and mathematical operations. This two part article describes how you can combine the calculator's hardware with a conventional 8 bit microcomputer. The result is a multiprocessor system: the microcomputer and its ROM programmed calculator slave.

Hardware

The Mathematical Function Unit is built of standard TTL, MOS, and CMOS logic components, and uses TTL compatible input

Figure 1: Mathematical Function Unit System Block Diagram. The Mathematical Function Unit accepts inputs from the computer at the left, simulates keystroke inputs to the calculator, and decodes BCD outputs which can be read by the computer. The result of connecting this peripheral to your microprocessor is a multiprocessor computer system, for the calculator chip is nothing but a form of microcomputer which comes with a fixed program load needed to perform mathematical and arithmetic operations.

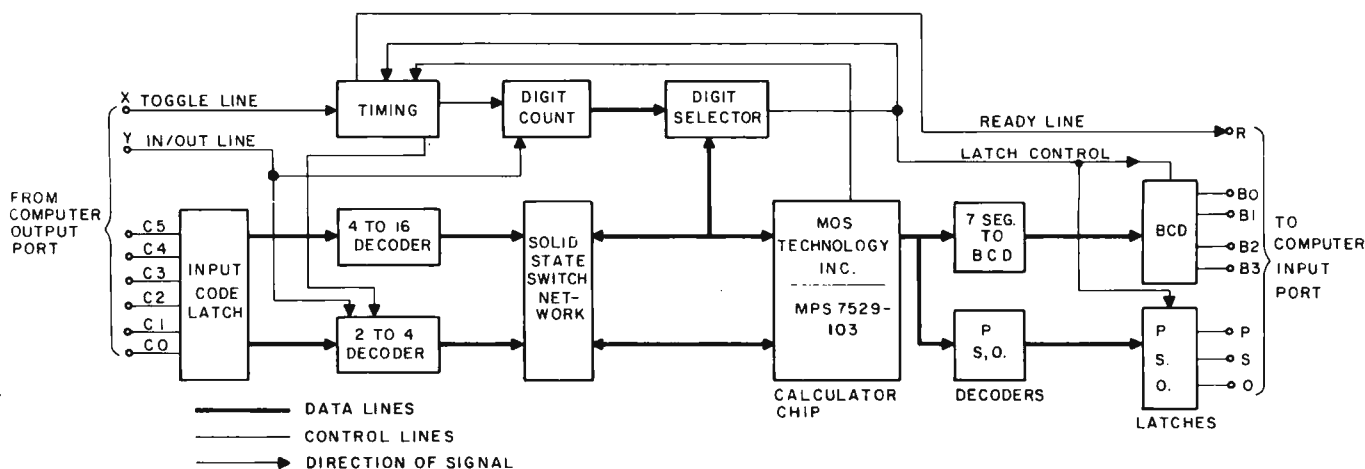
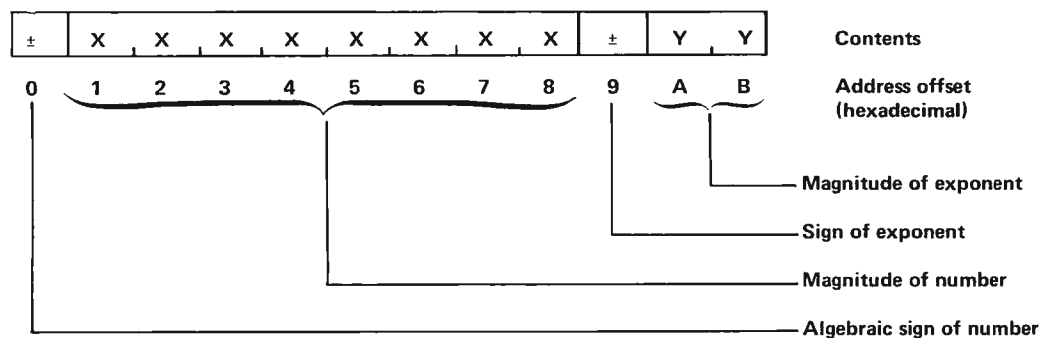


Figure 2: Memory formats of the Mathematical Function Unit data. When transferring data from the calculator to the microprocessor's main memory, one byte at a time is read, starting with the algebraic sign. A natural way to store the coded numbers read is in the form of 12 bytes in ascending order in the address space of your computer. Each byte's low order nybble is a BCD number in the magnitude positions (Xs or Ys in the figure). The high order portion of each byte contains the content of the decimal point, sign and overflow bit lines at the time the byte is read from the calculator. For output to the Mathematical Function Unit, the low order bits of a byte are used to drive the 6 key selection lines C0 to C5.



and output lines. Although not shown in the schematic diagrams, my version included an internal power supply, so the connections to the microcomputer are limited to signal and ground lines. Any 8, 12, or 16 bit machine's input and output ports can potentially be used with this interface.

The basic operations of the Mathematical Function Unit consist of the input of a code to be interpreted as a "function desired," the processing or calculating required to perform that "function," the decoding and output of the result, and the internal timing needed for control.

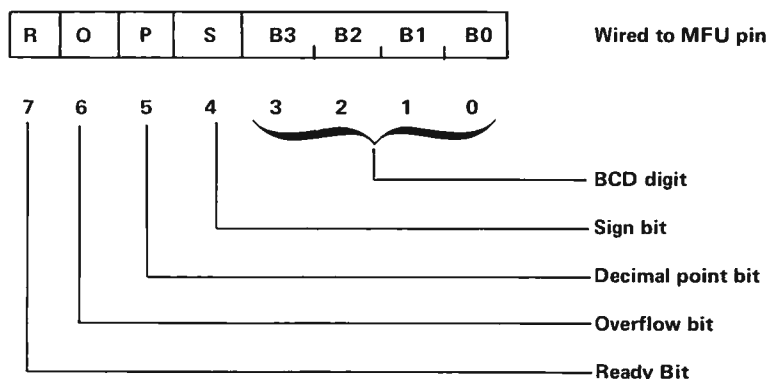
The functional block diagram of figure 1 shows how these sections are related, and provides a basic knowledge of the internal operation of the Mathematical Function Unit.

Input Section

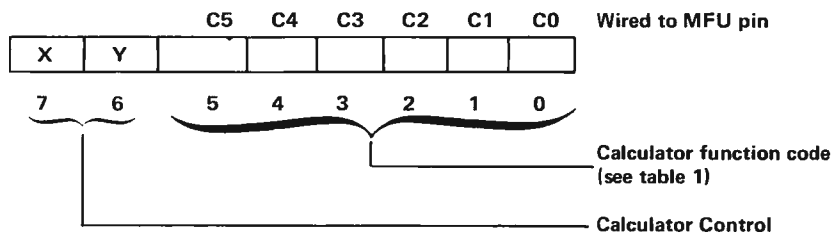
The input to the Mathematical Function Unit from an external device such as a microprocessor consists of 8 lines from an output port of that device. These lines are labeled X, Y, C₀, C₁, C₂, C₃, C₄ and C₅. The input section stores the new data supplied, and decodes this to the "function desired." See table 1 for a complete list of the functions and their codes. The function is applied to the calculator chip in the form of the correct "pushed button." (Since this is all done electronically, the pushing of buttons is simulated using solid state switches, and no push buttons really exist.)

The X line is used to distinguish new data from old. As this line is raised from logic level 0 to 1, the data on lines C₀ to C₅ is

Byte Format: Calculator Outputs



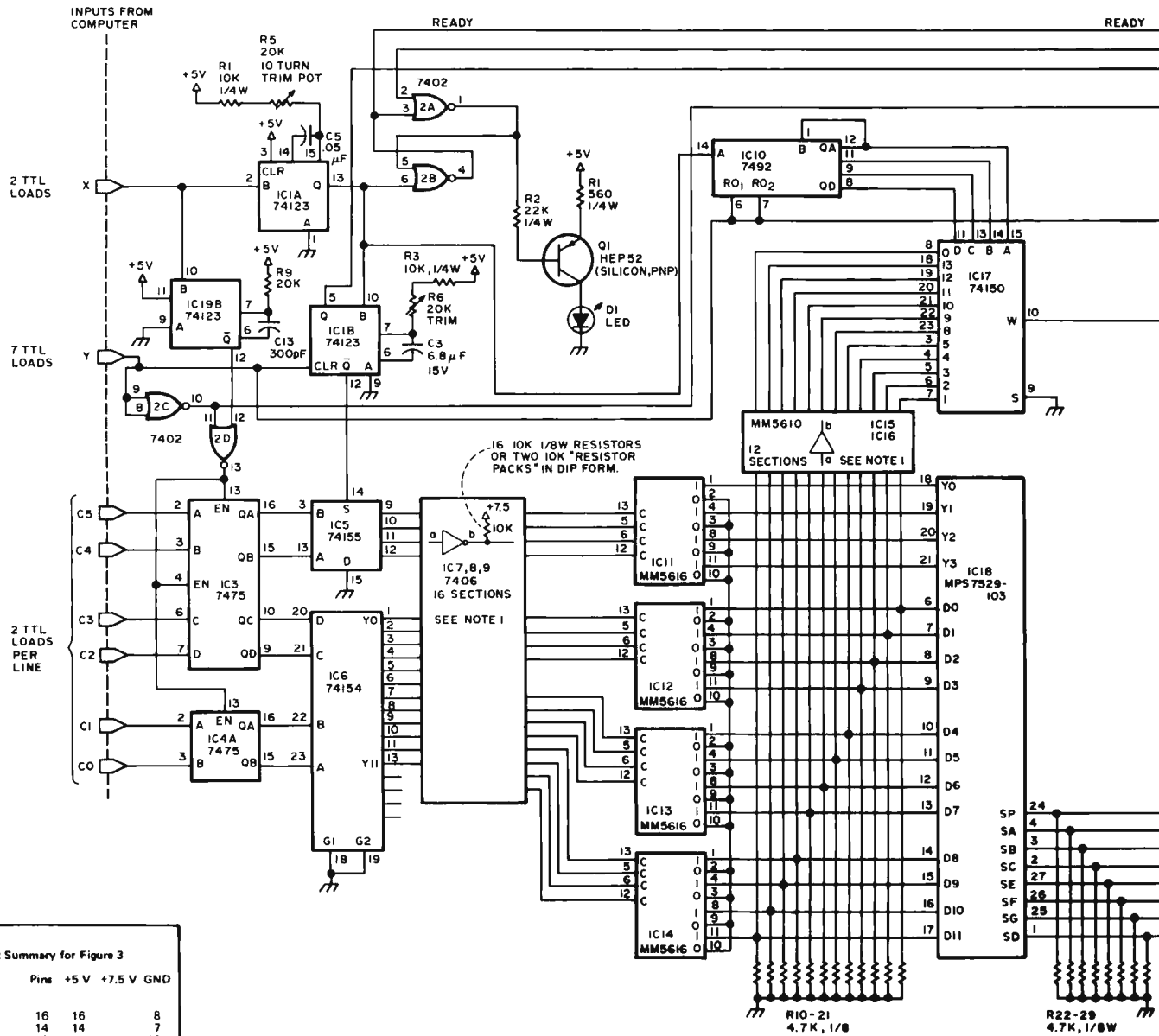
Byte Format: Calculator Inputs



ACKNOWLEDGEMENTS

I would like to thank J C Hertsch of MOS Technology, Inc, for his assistance by supplying information and donating the calculator chip used in this project, which was carried out under the auspices of California Polytechnic State University, San Luis Obispo.

Figure 3: Schematic Diagram of the Mathematical Function Unit. A total of 25 integrated circuits is required to accomplish the floating point and mathematical functions of a scientific calculator.



accepted as new information, and gated into the input buffer. After this data has been accepted, the data on the C lines is ignored until the next X line transition from 0 to 1.

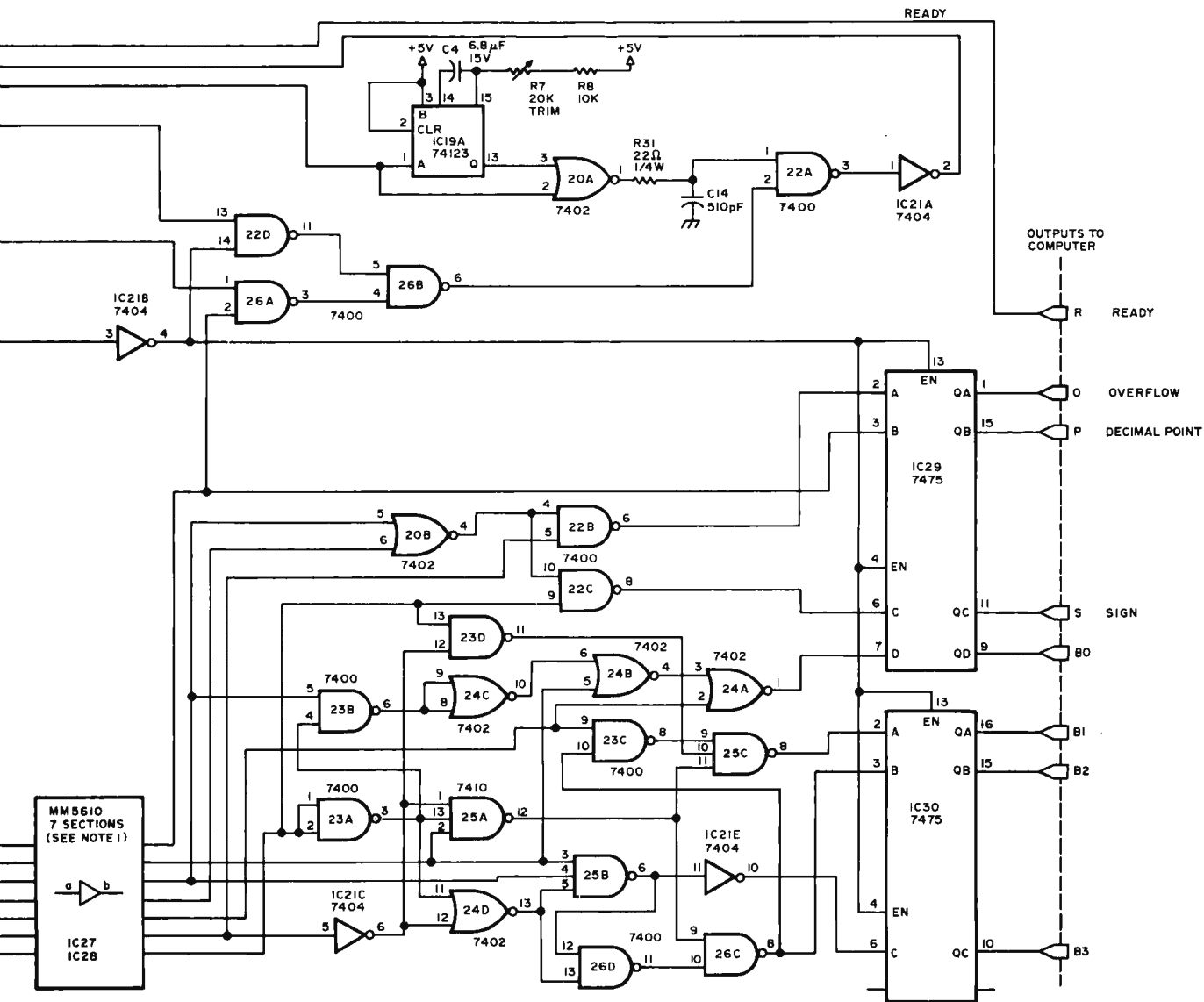
The Y line is used to determine whether an input or an output of information is to be performed by the Mathematical Function Unit. A logic 1 on this line indicates an input operation, while a logic 0 indicates that an output of information is to occur.

Input lines C₀ through C₅ are used to convey the codes for the different functions from the controlling device to the MFU input circuitry. A unique combination of 0 and 1 levels on these lines at input time is taken as a "key pressed" code. It may be

noted that 64 total combinations are possible with these 6 input lines; however, only 40 combinations are used, with the other 24 codes being invalid. These C lines are not used during the output state of the Mathematical Function Unit, when Y is low.

Processing Section

The processing of the required function is done by a large scale integration single chip, 40 key scientific calculator array, (MPS 7529-103) made by MOS Technology, Inc. This calculator chip has roughly the same set of available functions as some of the more sophisticated non programmable hand held calculators on the market today. When used



in the Mathematical Function Unit with the proper hardware and software, your micro-processor system can outperform the best of them.

Some of the operating features of the MPS 7529-103 calculator chip include:

- Number entry in floating point or scientific notation.
- Automatic selection of correct notation for result (scientific or floating point).
- Algebraic problem entry.
- Two levels of parenthesis.
- Full chain calculation with any function sequence.

The functional features include:

- Basic arithmetic (+, -, *, /)

Note 1: The pin connections for buffers using 7406 and MM5610 integrated circuits are not shown in detail on the diagram. They are shown in terms of "a" (input) and "b" (output) pins. The connections are as follows:

7406, six sections per package

Section	"a" pin	"b" pin
A	1	2
B	3	4
C	5	6
D	9	8
E	11	10
F	13	12

MM5610, six sections per package

Section	"a" pin	"b" pin
A	3	2
B	5	4
C	7	6
D	9	10
E	11	12
F	14	15

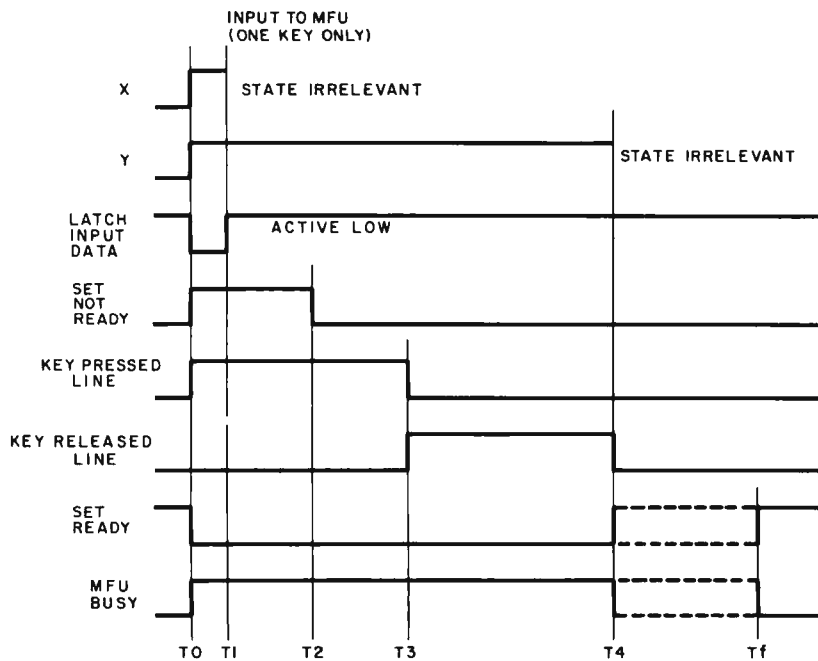


Figure 4: Mathematical Function Unit input timing sequence. This diagram shows typical relative timing of several signals during an input operation:

- T_0 is the time of an X line transition from 0 to 1.
- T_1 is 1.5 μ s after T_0 . The data must be stable from T_0 until after T_1 .
- T_2 is 300 μ s after T_0 . At this time, the ready flip flop is reset.
- T_3 is 50 ms after T_0 . At this time, key pressed is reset, key released is set.
- T_4 is 100 ms after T_0 . At this time, key released is reset.
- T_f is the delay until the calculator is again ready. The actual time interval depends upon the calculator function selected.

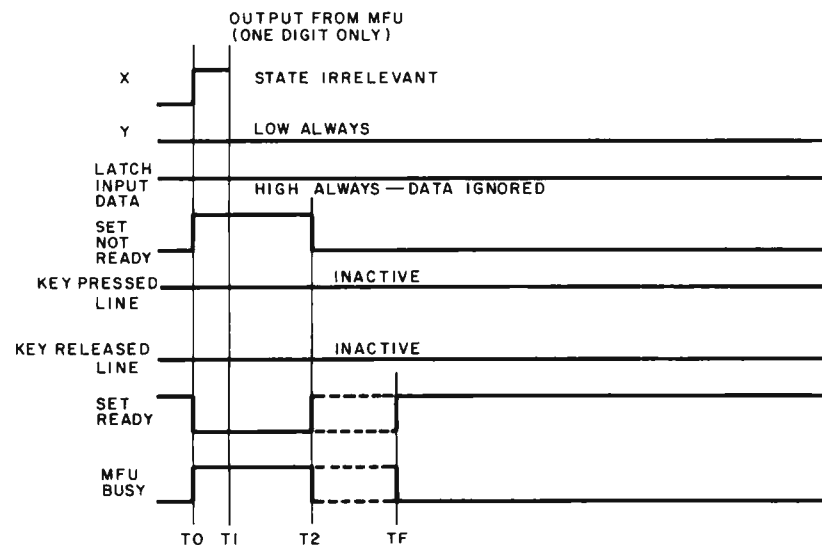


Figure 5: Mathematical Function Unit output timing sequence. This diagram shows typical relative timing of several signals during an output operation:

- T_0 is the time of an X line transition from 0 to 1.
- T_1 is shown to indicate that no data latch pulse occurs in this mode.
- T_2 is 300 μ s after T_0 . At this time, the ready flip flop is reset.
- T_f is the delay until the calculator is ready again, the maximum time before a digit is available in the output buffers.

- Trig functions (sine, cosine, tangent)
- Inverse trig functions (arc sine, arc cosine, arc tangent)
- Logarithms (Ln, Log)
- Anti-logarithms (e^x , and 10^x)
- Exponentiation (Y^X)
- Factorials (N!)
- Convenience Functions ($1/X$, X^2 , \sqrt{X} , Pi)
- Full feature memory (store, recall, sum)
- Exchange operation ($X \leftrightarrow Y$)
- Degree or radian selection for trig functions
- Automatic error detection
- Clearing operations (clear entry, clear all)

The calculation range includes positive or negative numbers with absolute values between 1×10^{-99} and 9.999999×10^{99} . Any number in this range may be entered and all results must fall within this range or an overflow will be indicated.

The output format of the calculator chip consists of 12 digit positions organized as shown in figure 2. Each output digit occupies one byte of memory when the microprocessor reads information from the Mathematical Function Unit.

The expected decimal point will be indicated in one of the digit locations 1 through 8, and a decimal point will be indicated in digit location 0 if the calculator chip's degree radian mode has been set to the radian mode.

Output Section

The output section of the Mathematical Function Unit is connected to the controlling processor through 8 output lines to an 8 bit input port. The output section is responsible for the decoding of the data supplied by the calculator chip after the required actions have been completed. The output section also generates the correct sequence for information presented to the controlling computer.

A handshaking signal is provided by the ready (R) line. This line is at a logic 1 level when the Mathematical Function Unit is not performing any input output or calculation operations. This line is used as a signal to the controlling computer as to the status of the slave. The ready line could be used to generate an interrupt upon completion of the calculations, or it could be connected directly to an input port line which would be polled until the Mathematical Function Unit has set it high indicating completion of its tasks.

The other 7 lines are data lines to the controlling computer and contain the information normally seen on the display of a calculator. The B lines contain one BCD digit of the normally displayed number, while the O (overflow), S (sign), P (decimal point) lines contain other necessary information.

The unit is designed to output one digit per request, where a request consists of a transition from a logic 0 to a 1 on input X line while input Y line is held at a logic 0 level. This means that only one digit is transferred at a time, slowing down the maximum speed of the system. This greatly simplifies both the supporting hardware and software handling of the 12 digits of the "displayed" number which is sent to the computer.

These 12 digits are generally loaded into the controlling computer's main memory in 12 sequential locations. This leads to the question of using a direct memory access operation to transfer this data. Due to the small amount of data (12 bytes), the calculator chip's slow speed, and the added hardware required, using direct memory access for the loading of the generated information would probably not be efficient.

The overflow line is high (logic 1) if the digit displayed exceeds the capacity of the calculator chip. The sign line is high if the digit position contains a negative sign, at which time the B lines are invalid. The decimal point line is high if a decimal point accompanies the digit on the BCD lines, and positionally goes to the right of the digit.

The output of the calculator chip is in seven segment notation and the decoding of this to the MFU's output format of BCD is done by ICs 20b, 21c and e, 22b and c, 23, 24, 25, and 26c, as shown in the schematic diagram of figure 3. This decoder circuit

Table 1. Hexadecimal Codes for the Mathematical Function Unit operations. The low order six bits of an 8 bit byte determine the function presented to the Mathematical Function Unit according to this table. On hand calculators, these functions correspond to the mnemonics of the keytops. A simple "program" for the calculator would be a string of bytes sent one by one with meaningful selection of these operation codes, followed by reading the outputs, formatting them and displaying them on a TV typewriter or Teletype.

Hex Code	Function
00	Zero (0)
01	One (1)
02	Two (2)
03	Three (3)
04	Four (4)
05	Five (5)
06	Six (6)
07	Seven (7)
08	Eight (8)
09	Nine (9)
0A	Arc Function
0B	Display Restore
10	Decimal Point
11	Add
12	Subtract
13	Multiply
14	Divide
15	Y to the X power
16	Equals
17	Left Parenthesis
18	Right Parenthesis
19	PI (3.1415927)
1A	Change Sign
1B	Enter Exponent
20	Sine
21	Cosine
22	Tangent
23	Natural Log (ln)
24	Log (base 10)
25	Square Root
26	Recall From Memory
27	Add to Memory
28	Swap X with Y
29	Degree - Radians
2A	Store in Memory
2B	Clear Entry - Clear All
30	1/X Inverse
31	X ²
32	10 ^X
33	e ^X
34	N! Factorial

(All other Hex codes are invalid)

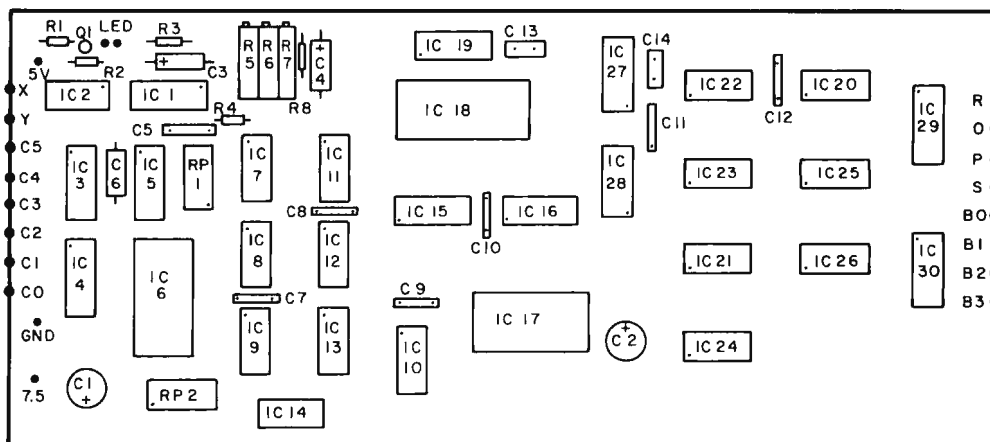


Figure 6: Parts placement in the author's prototype of the Mathematical Function Unit. The unit was constructed on a piece of Vector P pattern Vector-board (.1 inch grid, 2.54 mm grid) as depicted in photo 1, with this layout.

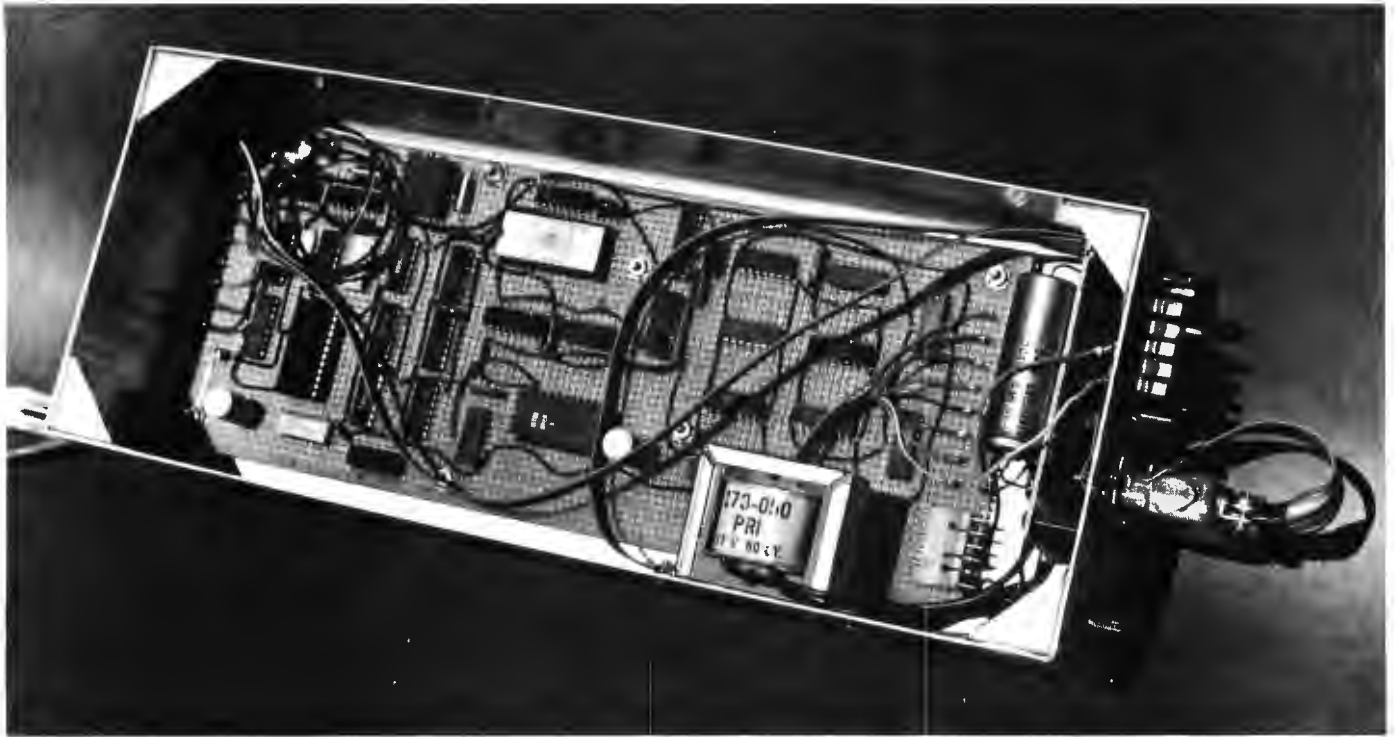


Photo 1: The physical assembly of the prototype Mathematical Function Unit. A multiple conductor ribbon cable is used to route ground and signal information to the microprocessor system which drives the Mathematical Function Unit through a parallel IO port. A separate power supply (not shown in the schematics of this article) was built into the box.

interprets a blanked digit position as a zero, so all digit positions contain a digit, overflow symbol, or negative sign with decimal points included when applicable.

Timing and Control

The basic timing and control problem for the Mathematical Function Unit is to prevent the external device from overrunning the unit with information, holding signals for the required length of time, controlling the input and output buffers, and controlling the ready flip flop used for handshaking.

The timing pulses are generated by a series of monostable multivibrators, and the ready flip flop is made of NOR Gates, IC 2a, and 2b in figure 3.

A transition from a logic 0 to a logic 1 level on the X input line is responsible for

initiating both the input and output sequences. These two sequences are determined by the state of the Y input line, where a 1 signifies the input procedure, while a logic 0 means an output of information is to be performed.

Input Timing Sequence

The normal environment of the calculator chip is in a hand held calculator with a human operator pushing the buttons. This allows the calculator enough time to scan the keyboard several times, determining whether a key is being pressed, or if a noise spike on the line caused an unwanted pulse during a couple of scan times. This is the method used by many calculator chips for debouncing the push buttons used. The calculator chip used in this project requires about 40 ms for a pressed key to be recognized. (About 7 keyboard scan times.) This 40 ms delay is virtually impossible for a human hand to beat, assuring a key will be recognized every time a button is pushed. There is always going to be a sufficient delay between different key pressings due to the

Note: The author and two friends have gotten together in order to make a printed circuit board available for this design. The product is a two layer board with plated through holes, and is designed to be compatible with the Altair 8800 or IMSAI computers, interfacing through an IO port. The price is quoted as \$24.95 plus \$1.23 for postage and handling. California residents please add 6% sales tax. Write RSG Electronics, PO Box 13, Santa Margarita CA 93453. (Price is subject to change without notice.)

slow human controller; however, all of these delay times do not necessarily hold true when interfacing with a much faster controlling device, such as a microcomputer. This requires that a timing network be implemented to insure that the calculator chip receives the proper signals in the proper sequence, with the proper delays.

The "key pressed" delay is provided by monostable multivibrator, IC 1b, and is adjustable by R6. When this 40 to 50 ms delay is completed, IC 19a, also a monostable multivibrator, is triggered as a "key released" delay providing the system with a short delay between key pressings. These two delays form the minimum time required for the Mathematical Function Unit to become ready for the next sequence; however, for some of the more time consuming functions such as the trigonometric, factorial and logarithmic functions, the calculator would not be finished after the two delays had passed. To insure calculations are completed before setting the Mathematical Function Unit state to "Ready," both delays must be completed and a decimal point be sensed by the output circuitry. Since a decimal point is the only character present in all output displays, and is not present until all calculations are complete, sensing the decimal point indicates end of calculation. (The gates used for detection of the decimal include 20a, 21a and b, 22a and d, and 26a and b in figure 3.) When these requirements are met, a condition is placed on the ready flip flop, setting it to the "Ready" state. The timing signals for the input sequence are shown in figure 4.

The length of the delay between T_4 and T_f (of input routine; see figure 4) will depend on how the calculator chip's internal scan timing coincides with the surrounding hardware. For the 7529-103 calculator chip used in this project, this delay will not exceed 3.3 ms after the calculations or required actions are complete.

Output Timing Sequence

The output sequence is specified by placing a logic 0 level on the Y input line. This low level inhibits the operation of the key pressed and key released delays which are not needed for output. The X line transitions are used to clock a counter, IC 10, which with IC 17 selects the next digit to be placed in the output buffers. This digit is decoded as previously mentioned, and latched in the buffers while the ready flip flop is set to "Ready."

The next digit is found by pulsing the X line again while keeping the Y line low. The

Y line must be kept low during the entire output procedure since a high state on this line resets the counter, which will then point to the first digit again.

A pictorial description of the output timing signals is shown as figure 5.

The length of the delay between T_2 and T_f will depend on how the calculator chip's internal scan timing coincides with the surrounding hardware. This delay will not exceed 3.3 ms for the calculator chip used in this project.

TTL – MOS – TTL Interface

The power requirements for the Mathematical Function Unit are 5 volts at approximately 0.5 Amps, and 7.5 volts at close to 200 mA. The 5 volt supply is used for all TTL gates, and the 7.5 volts is used to operate the MOS calculator chip. The signal levels are buffered and adjusted from the TTL input levels to 7.5 volts through the open collector, high voltage output inverters ICs 7, 8 and 9, using 10 k ohm pull-up resistors in the two resistor packs connected to the 7.5 volt supply.

These higher voltage signals are applied to the switch matrix made from CMOS Quad Bilateral Switches (ICs 11-14) operating at that higher voltage. The signals are then directly compatible with the calculator chip used.

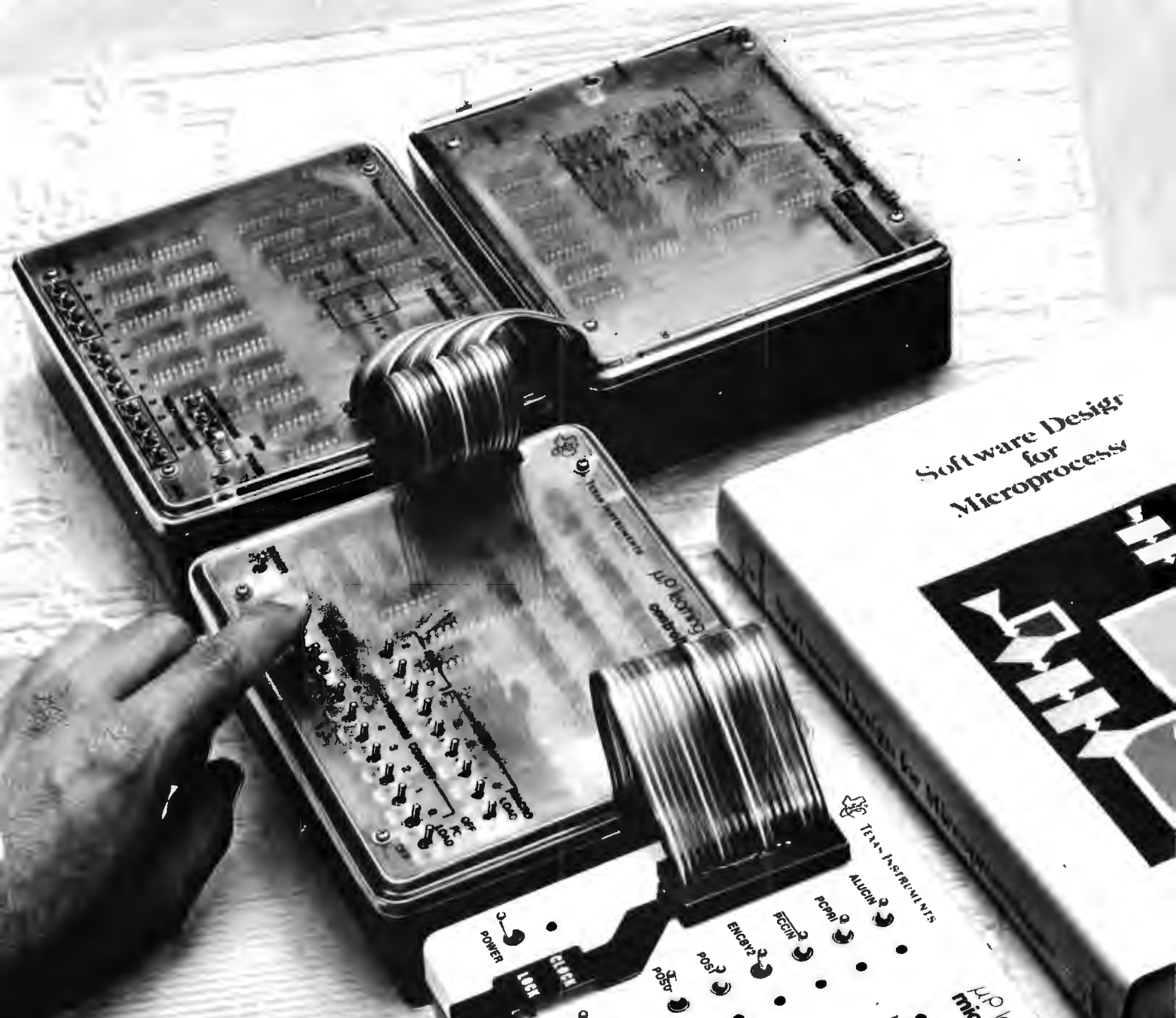
The signals coming from the calculator chip to the output circuitry are buffered to the 5 volt level through the use of CMOS Hex Non-inverting Buffers, ICs 15, 16, 27 and 28.

Construction

The Mathematical Function Unit is constructed on an 11 by 4 inch (27.94 by 10.16 cm) piece of Vectorboard with all wiring done point to point. A parts placement diagram with all of the visible parts, with numbers referencing figure 3, is found in figure 6. The two 24 pin multiplexors and the 28 pin MOS Calculator Chip are placed in sockets. Photo 1 shows the hardware mounted in its case; refer to figure 6 to identify components.

The main component board is bolted in a 13 by 5 by 3 inch (33.02 by 12.7 by 7.62 cm) aluminum chassis with a piece of clear plexiglas on the top. Contained also in this chassis are the simple 5 volt and 7.5 volt power supplies for the unit, with the 5 volt regulator, power switch and the IO data line connector mounted on the rear of the box. The "Ready" indicator is mounted on the front panel, indicating the state of the Mathematical Function Unit.■

In part 2 of his article, which will be published next month, you'll find details of the software needed to drive this calculator interface from an Intel 8080 based computer system . . . CH



At your fingertips.
Microprocessor concepts...
from Texas Instruments

Software Design
for
Microprocessors

Texas Instruments

µP learning module
microprogrammer

POWER
MICROCONSTRUCTION
ALUOUT
MSB DATA LSB
MSB Address LSB
ALUIN
PCIN
PCOUT
ENCLV2
POS2
POS1
DATA IN
D
D
D
D
D
D
S
S

New. The complete microprocessor learning system. Ready-to-use. Economical. User-paced.

Introducing three new add-ons for TI's basic Microprogrammer Learning Module.

Microprocessors. Now they're simpler to understand, easier to cope with, because of TI's complete new microprocessor learning system.

It's modular...consists of four, portable components: the basic LCM 1001 Microprogrammer Learning Module and three new add-ons: Controller. Memory. And Input/Output.

It's a down-to-fundamentals system for getting "hands on" experience with microprocessors. It focuses your attention on learning microprocessor concepts—not on individual products. You progress from module to module in a logical sequence.

Each module is self-contained. Complete and ready-to-use. No kits to assemble. Nothing to build. Each has its own battery/charger power system. Each has its own instruction manual.

Microprogrammer Module

This is the one you start with. The basic building block (LCM-1001). It demonstrates the most fundamental level of microprocessor operation: Single-clock-step microinstruction. It contains a 4-bit static parallel processor with manual switch inputs, VLED monitors, and pushbutton manual clock. A 40-pin socket lets you link up with the controller. \$149.95.*

Controller Module

Add the LCM-1002 Controller Module. Progress from micro to macro level programming. Learn instruction set development and microcomputer architecture. Use the Controller with RAM or additional program memory—12 basic instructions are programmed into the PROM (which may be redefined). Using these, write your own programs. Or, expand the instruction set and customize the system to your needs using the blank locations you define yourself. \$189.95.*

Memory Module

Add an LCM-1003 Memory Module to the Controller. Now you're into fully automated digital system operation. The read/write memory is configured as a 1K word by 12-bit structure, for both data and program storage. Switches are provided for manual loading. The memory automatically increments the address. This feature spares you the tedium of cycling the entire memory when loading or changing memory. \$189.95.*

Input/Output Module

With the final module, you're in contact with the outside world. There are four 4-bit input ports and four 4-bit output ports with buffers for data transfer synchronization. Operate the ports independently, or in conjunction with each other. For example, two ports can recognize an 8-bit code approach. \$109.95.*

Building-block System.

This approach to microprocessor self-instruction is sensible and economical—add another module when you're ready to learn more.

To order your Microprocessor Learning Modules, call your nearest TI distributor. Use the coupon to get a detailed brochure.

New, authoritative book on microprocessor software design. Only \$12.95.*

A working knowledge of microprocessor software is essential. Acquiring such knowledge is now simplified with **Software Design for Microprocessors**.

Helps you fully understand basic microprocessor machine code and assembly language. Suitable for non-technical professionals as well as the technically trained.

In 500 pages are the graphs, tables and data needed to learn the language, special terms, and the underlying concepts that lead to understanding the complex facets of microprocessor software. The text concludes with four fully-worked examples for "real world" situations.

Use the coupon below to order your copy.

* Manufacturer's suggested list price. Subject to change without notice.

To: Texas Instruments Incorporated

Mail checks and money orders to: P. O. Box 3640, M/S 84, Dallas, Texas 75285
Postage paid. Add state and local taxes where applicable. †

Mail company purchase orders to: P. O. Box 5012, M/S 84, Dallas, Texas 75222
Postage and taxes will be added to your invoice.

Please send me

more information on the TI Microprocessor Learning Modules.

Copies of *Software Design for Microprocessors* @ \$12.95** ea. _____

I enclose check money order for \$ _____

Name _____

Title _____

Firm _____

Address _____

City _____ State _____ Zip _____

**Subject to change without notice

† AL, AZ, CA, CO, CT, FL, GA, IA, IL, IN, KY, MA, MD, MI, MN, MO, NC, NJ, NM, NY, OH, PA, TN, TX, UT, VA, WA, WI.



B-7

Randomize Your Programming

Computers are supposed to produce predictable results; when a program acts unpredictably, it is usually time for head-scratching and debugging. There are times, however, when unpredictability is exactly what the programmer wants. A class of examples is the simulation of games with a chance element. How can a computer simulate a coin toss or the throw of a die or the picking of a card? In each of these cases, the application requires a value within set limits which is unpredictable in its sequence.

As a second class of examples, suppose one wants to test a program or piece of hardware with random data. How can a microcomputer generate the appropriate numbers? In this case, it would be nice to be able to repeat the sequence of values at will,

so that any errors that are found can be repeatedly tested.

This article describes simple programs which can provide the capability of generating appropriate sequences. The programs are written for the Motorola 6800 processor, but are simple enough that conversion to other instruction sets should pose no major problem.

Pseudorandom Sequences

To be precise, the programs described in this article generate "pseudorandom" sequences. These are sequences which give each value in the range of the sequence (except the value 0, which will be discussed later) exactly once before the sequence repeats. Given the starting value, the se-

Robert Grappel
148 Wood St
Lexington MA 02173

Listing 1: The "RANDOM" subroutine specified in the symbolic assembly language of the Motorola 6800 processor, along with hexadecimal machine code. The origin was set arbitrarily to address 1000 for this assembly, but any other address could be used without change since all branches use relative addressing and data is passed to RANDOM via a pointer in the index register.

Line	Addr	Hex Code	Label	Op	Operand	Commentary
1	1000	A6 00	RANDOM	LDAA	0,X	load A from random byte source;
2	1002	26 01		BNE	NOTO	if random byte not zero then proceed;
3	1004	4C		INCA		else force non zero state A = 1;
4	1005	16	NOTO	TAB		begin exclusive OR procedure;
5	1006	C4 8E		ANDB	#\$8E	mask out feedback bits;
6	1008	27 06		BEQ	SKIP	if no feedback then exit without setting bit;
7	100A	58	SHIFT	ASLB		loop to find a set feedback bit;
8	100B	24 FD		BCC	SHIFT	if not set then reiterate;
9	100D	5D		TSTB		is there more than one feedback bit?
10	100E	27 04		BEQ	EXOR	if not then shift in a new bit;
11	1010	48	SKIP	ASLA		shift without setting bit;
12	1011	A7 00		STAA	0,X	restore pseudorandom value to argument;
13	1013	39		RTS		return to the caller;
14	1014	48	EXOR	ASLA		shift and set bit;
15	1015	4C		INCA		increment pseudorandom number value;
16	1016	A7 00		STAA	0,X	restore pseudorandom value to argument;
17	1018	39		RTS		return to the caller;

quence is fixed. If the sequence is long enough, however, the result is an apparently random sequence. For example, the sequence based on four bit values is:

1,2,4,9,3,6,13,10,5,11,7,15,14,12,8

This looks random when only two or three values are considered. A longer sequence can be chosen and only a portion of each value used. This will heighten the apparent randomness of the sequence.

Pseudorandom sequences are usually generated in hardware through the use of a shift register and feedback from certain stages of the register through an exclusive OR gate to the beginning of the register. This hardware configuration is easily simulated in software, and such a program is shown as the subroutine RANDOM found in listing 1. The origin of the subroutine was chosen to be hexadecimal 1000 for no especially strong reason. Any other origin would work as well since all branches within RANDOM use relative addressing. RANDOM generates eight bit values, and the sequence is 255 values in length. It is designed to be called with the address of a byte in which the random value is to be generated located in the index register. If the same sequence is desired every time, this byte should be initialized to a fixed value such as zero. If the byte is not initialized, the sequence will be based on prior memory content and therefore unpredictable.

Random Logic

RANDOM begins by loading an accumulator with the present value of the random byte accessed via the index register of the

Table 1: Properties of Pseudorandom Generators.

Stages	Period	Feedback Bits	Mask
2	3	1,2	03
3	7	2,3	06
4	15	3,4	0C
5	31	3,5	14
6	63	5,6	30
7	127	6,7	60
8	255	2,3,4,8	8E
9	511	5,9	0110
10	1023	7,10	0240
11	2047	9,11	0500
12	4095	2,10,11,12	0E02
13	8191	1,11,12,13	1001
14	16383	2,12,13,14	3802
15	32767	14,15	6000
16	65535	11,13,14,16	8400

6800. The value zero is illegal here, because the sequence will not set any bits in the byte if all bits ever go to zero simultaneously. This case is eliminated by the increment instruction, line 3, which sets the byte to 1 if it ever gets to be zero. The accumulator is being used as an eight bit shift register. The appropriate feedback loop must be simulated. This is done by the instructions between NOTO and SKIP. The feedback loop is an exclusive OR function of certain bits. The AND instruction chooses the bits to be exclusive ORed. The exclusive OR function is true, if and only if exactly one of its inputs is true. Hence, if the AND leaves no bits set, the exclusive OR is false. If at least one bit is set, the SHIFT loop is entered. This loop shifts the feedback bits left until one has moved out of the left of the accumulator and into the carry flag. The contents of the accumulator are again tested. If the bit that was shifted to the carry was

Listing 2: Using the RANDOM routine to generate a 16 bit pseudorandom number. This extension is obtained by calling RANDOM once for each byte of a 16 bit number. The origin used here is arbitrarily chosen as hexadecimal 2000.

Line	Addr	Hex Code	Label	Op	Operand	Commentary
1	2000	00 00	RAND2	RMB	2	reserve two bytes for 16 bit key;
2	2002	CE 20 00	RND16BIT	LDX	#RAND2	point index at 16 bit random number;
3	2005	BD 10 00		JSR	RANDOM	calculate high order 8 bit part;
4	2008	A6 00		LDAA	0,X	get high order part to A;
5	200A	08		INX		point to low order part of 16 bit number;
6	200B	85 01		BITA	#1	should carry be propagated down?
7	200D	27 03		BEQ	RETRND	if not then skip low order randomizer;
8	200F	BD 10 00		JSR	RANDOM	calculate low order 8 bit part;
9	2012	39	RETRND	RTS		return with new 16 bit number in RAND2;

Listing 3: The "DICE" program specified in the symbolic assembly language of the Motorola 6800 processor, along with hexadecimal machine code. This subprogram is written to exercise the pseudorandom number generator, print the results (or display them) using the OUTHR subroutine of the Motorola MIKBUG program. After output, the INEEE subroutine is called to wait for a keyboard response before generating another "roll." An arbitrary starting address is used, 2000 hexadecimal, and RANDOM is assumed to start at address 1000 hexadecimal.

Line	Addr	Hex Code	Label	Op	Operand	Commentary
1	2000	00	DIE	RMB	1	one byte for pseudorandom number seed;
2	2001	CE 20 00	DICE	LDX	#DIE	point index at the random number value;
3	2004	BD 10 00	ROLL	JSR	RANDOM	go define next random digit;
4	2007	A6 00		LDAA	0,X	move random digit to A;
5	2009	84 0F		ANDA	#\$0F	force low order value 0 to 15;
6	200B	81 02		CMPA	#2	is digit less than 2?
7	200D	2D F5		BLT	ROLL	if so then roll again;
8	200F	81 0C		CMPA	#12	is digit greater than 12?
9	2011	2E F1		BGT	ROLL	if so then roll again;
10	2013	A6 00		LDAA	0,X	move random digit to A for output;
11	2015	BD E0 6B		JSR	OUTH	display using MIKBUG Rev 9 OUTH
12	2018	BD E1 AC		JSR	INEEE	operator response via MIKBUG Rev 9 INEEE
13	201B	20 E4		BRA	DICE	then restart the program

the only bit set, then the test for zero is passed and the exclusive OR is true. Otherwise, there was more than one bit set in the feedback loop and the exclusive OR is false. If the exclusive OR test was false, then the contents of the accumulator are shifted left once and the low order bit is left zero. If the test was true, then the accumulator is shifted and the low order bit is made one. The accumulator is restored into the random byte and the subroutine returns to the program which called it.

By changing the feedback connections, pseudorandom generators of varying periods and value size can be made. Table 1 shows the feedback points for generators of lengths 2 to 16. The number of stages is the number of bits in the shift register, which determines the maximum value the pseudorandom number will be. The period of the sequence (the number of values generated before the sequence repeats) is given by the formula $2^n - 1$, where n is the number of stages. RANDOM can generate sequences corresponding to generators of 2 to 8 stages, simply by changing the mask value of the AND instruction (line 5 in listing 1) to pick out the proper bits to exclusive OR. The appropriate masks are given in table 1.

Longer Sequences

The eight bit accumulator of the 6800 processor limits RANDOM to eight stages. A more complex program could be written to directly generate longer sequences, and for those who wish to write such programs, the table gives the masks for up to 16 bit generators. There is another way to generate longer pseudorandom sequences, and that is to generate them in smaller sections. RAND2 shows a simple way to generate 16

bit sequences using RANDOM twice (see listing 2). The high order byte is generated by RANDOM directly. If the byte is even (which will be the case half of the time), the low order byte is unchanged by this call. If the byte is odd, then the low order byte is changed by another call to RANDOM. Longer sequences yet can be generated by extension of the ideas in RAND2.

Rolling the Die

An example of a practical use for RANDOM is shown in the little test program DICE. This program, given in listing 3, simulates the tossing of a pair of dice. It uses RANDOM to generate pseudorandom values between 2 and 12. RANDOM generates an eight bit value which is truncated to the low order four bits by an AND masking operation. If this value is not between 2 and 12, the process is repeated. It should be clear how to modify DICE to simulate one die . . . or two dice, or the flipping of a coin. In general, the longer the pseudorandom sequence in use, the more apparently random the values returned by these routines will be. Eight bits should be quite adequate for many games of chance. Thorough testing of programs or hardware might profit from a longer sequence.

Now, generation of nearly random values should no longer be a problem. This should ease the job of programmers trying to simulate games of chance, and help in other areas of simulation, statistical work, and testing. These are just about the simplest programs which can perform the necessary tasks. Try them. And, when your computer gives you unpredictable results, you don't have to look for the bug. It's your programmable "randomness" generator! ■

Learn to use a microprocessor.

This portable computer teaches you how.

Low cost, battery operated Intercept Jr. Tutorial System instructs you on today's most advanced LSI semiconductor devices, then serves double duty as an operating solid-state computer!



Learn microprocessors in hours.

With its detailed Owner's Handbook, Intercept Jr. teaches you basics of microprocessors, Random Access Memories (RAMs), Read Only Memories (ROMs) and input/output interfacing, all in less than 8 hours.

Contains everything you need.

Intercept Jr. is a complete all-CMOS computer on a 10- by 12-inch board. Contains batteries, entry keyboard, 8-digit LED display, RAM and ROM memory ... all controlled by an Intersil IM6100 12-bit CMOS microprocessor.

An inexpensive teaching computer.

For \$281.00 you get both the handbook and the computer, plus provisions for expansion with 1024 words of additional memory, from 256 to 2048 words of added program, and input/output interfacing with terminals.



Send in coupon today.

Intersil, Inc.,
10900 North Tantau Ave., Cupertino, CA 95014.

- | | PRICE | QTY. | TOTAL |
|--|----------|------|----------|
| <input type="checkbox"/> I'm interested. Send me Intercept Jr. Tutorial System complete with Owner's Handbook. | \$281.00 | ___ | \$ _____ |
| <input type="checkbox"/> I need to know more. Send full technical details on Intercept Jr. | FREE | ___ | _____ |

Plug-in Expansion Modules Also Available

- | | | | |
|---|--------|-----|-------|
| <input type="checkbox"/> RAM Module, with 1024 words of added non-volatile memory. | 145.00 | ___ | _____ |
| <input type="checkbox"/> P/ROM Module, with sockets for 256 to 2048 words of program. | 74.65 | ___ | _____ |
| <input type="checkbox"/> I/O Module, for interfacing TTY keyboard/reader and printer/punch. | 81.70 | ___ | _____ |

(California residents add 6% sales tax.) _____

B9

TOTAL PAYMENT DUE _____

- Check enclosed.
 Money order enclosed.
 Bill me. (Please list Purchase Order. no _____)

SHIPPING ADDRESS (Please print)

name _____
 company _____
 address _____
 city _____ state _____ zip _____

A BASIC Star Trek Trainer

Listing 1: The text of the Star Trek Trainer, specified in BASIC for a Data General NOVA. For users of other BASIC definitions, modifications will be required, depending upon the specification of your particular implementation.

```
10 REM SHIP 1 IS ENTERPRISE, SHIP 2 IS KLINGON BATTLECRUISER.
20 REM EACH VESSEL REPRESENTED BY 7 NUMBERS IN THE STATE ARRAY S.
30 REM S(1, ) IS FORWARD PHASOR, 1=OPERATIVE, 0 INOP
40 REM S(2, ) REAR PHASOR; 1 OP, 0 INOP.
50 REM S(3, ) PHOTON TORP.; S(4, ) WARP DRIVE.
60 REM S(5, ) IS DEFLECTOR SHIELD EFFECTIVENESS. 1.TOTALLY EFFECTIVE, 0
70 REM MEANS SHIELDS GONE. S(7, ) IS DAMAGE COEFFICIENT, A MEASURE OF
80 REM TOTAL SUSTAINED DAMAGE. S(6, ) IS IMPULE ENGINES STATUS.
90 DIM S(7,2)
100 REM INITIALIZE VARIABLES
110 FOR I=1 TO 2
120 FOR J=1 TO 6
130 LET S(J,I)=1
140 NEXT J
150 LET S(7,I)=0
160 NEXT I
170 REM GEN RANGE AND BRNG
180 PRINT "ENTER STARDATE"
190 INPUT A
200 LET A2=-ABS(A/1000)
210 LET R=1017*RND(A2)
220 LET B=180*RND(A2)
230 LET I=1
240 REM ODD MOVES ENTERPRISE'S, EVEN KLINGNON
250 PRINT "KLINGNON APPROACHING:"
260 PRINT "R="R"KM, BEARING="B" DEG."
270 PRINT "COMMAND"
280 INPUT C
290 PRINT C
300 LET I=1
310 LET J=2
320 LET H=0
330 GOSUB C OF 790,920,1040, 1140, 1180, 1250, 1310
340 GOTO 1360
350 REM KLING. MOVE SELECTION
360 LET I=2
370 REM CHECK KLINGNON PHASOR STATUS
380 LET W=S(1,I)+S(2,I)
390 LET J=1
400 IF S(3,I)=1 AND R>3E+06 AND B<100 THEN 710
410 IF W>0 AND R<3E+06 THEN 540
420 IF W#0 THEN 480
430 REM ATTEMPT TO BREAK CONTACT
440 PRINT "KLINGNON ATTEMPTS TO BREAK CONTACT"
450 LET C=5
460 GOSUB 1180
470 GOTO 1920
480 PRINT "KLINGNON MANEUVERING TO ATTACK"
490 LET C=4
500 GOSUB 1140
510 LET B=0
520 LET H=0
530 GOTO 1920
540 PRINT "KLINGNON FIRES PHASOR"
550 LET C=1
560 IF B<90 THEN 640
570 IF S(2,I)=0 THEN 600
580 GOSUB 920
590 GOTO 1360
600 LET B=B-90
610 GOSUB 790
620 LET B=B+90
630 GOTO 1360
640 IF S(1,I)=0 THEN 670
650 GOSUB 790
660 GOTO 1360
670 LET B=B+90
680 GOSUB 920
690 LET B=B-90
700 GOTO 1360
710 PRINT "KLINGNON FIRES PHOTON TORPEDO"
720 LET C=3
730 LET B2=B
740 LET B=40
750 GOSUB 1040
760 LET B=B2
770 GOTO 1360
780 REM FIRE FWD PHASORS
790 LET H=0
800 IF S(1,I)#0 THEN 830
810 PRINT "FWD PHASORS INOP"
820 IF I=1 THEN 260
830 LET R1=1-0.2*(R/1E+06)
840 LET B1=(90-B)/90
850 IF B1<0 OR R>5E+06 THEN 900
860 REM GEN RANDOM NUMBER IN INTERVAL 0 TO 1
870 IF R1<RND(1) THEN 900
880 LET H=1
890 LET P=4
900 RETURN
910 REM REAR PHASOR
920 IF S(2,I)#0 THEN 960
930 PRINT "REAR PHASORS INOP"
940 REM
950 IF I=1 THEN 260
960 LET H=0
```

Gerald H Herd
742 Valley Dr
Pensacola FL 32503

While complex Star Trek and Space War games, complete with space warps, fleets of enemy ships, and starbases currently exist, they generally require a sophisticated system to support them. For the microcomputer hobbyist who does not have the resources of Star Fleet at his disposal for the purchase of hardware, the following short version of Star Trek is offered.

The program was developed in BASIC on a NOVA 1200 minicomputer and emulates a Star Trek game I originally encountered on the Univac 1108 system at Georgia Tech. The program requires about 2200 16 bit words in its current form, although considerable savings of memory are possible by simply deleting the remarks. While lacking many of the trappings of larger games, the BASIC Star Trek Trainer offers the following advantages:

- 1) A choice of weapons, phasor banks or photon torpedoes, each turn.
- 2) Maneuvering commands.
- 3) Deflector shields which weaken as the number of hits on each craft increases.

Listing 1:

```

970 LET R1=1-0.2*R*1E-06
980 REM RANGE BEARING CHECK
990 IF R>5E+06 OR B<90 THEN 1030
1000 IF R1<RND(1) THEN 1030
1010 LET H=1
1020 LET P=2
1030 RETURN
1040 LET H=0
1050 IF S(3,I) #0 THEN 1080
1060 PRINT "PHOTON TORPEDO INOP"
1070 IF I=1 THEN 260
1080 IF R<2E+06 OR B>90 THEN 1130
1090 LET R2=(1-B/90)*(1-2*R/1E+08)
1100 IF R2<RND(1) THEN 1130
1110 LET H=1
1120 LET P=8
1130 RETURN
1140 LET R=R*(1-0.5*S(4,I))
1150 LET B=0
1160 LET H=0
1170 RETURN
1180 LET R=R*(1+S(4,I)+0.05*S(6,1))
1190 IF R>1E+08 THEN 1230
1200 PRINT "CONTACT NOT BROKEN"
1210 LET H=0
1220 RETURN
1230 PRINT "CONTACT LOST"
1240 STOP
1250 PRINT "SELF DESTRUCT ACTIVATED"
1260 FOR I=10 TO 1 STEP -1
1270 PRINT I
1280 NEXT I
1290 PRINT "BOOM"
1300 STOP
1310 PRINT "MESSAGE FROM KLINGON *** I ACCEPT YOUR SURRENDER"
1320 PRINT "PREPARE TO BE BOARDED."
1330 PRINT "***MESSAGE FROM STAR FLEET COMMAND"
1340 PRINT "YOU DIRTY COWARD"
1350 STOP
1360 LET J=3-I
1370 IF C>3 THEN 1920
1380 REM DAMAGE ASSESSMENT
1390 IF H#0 THEN 1440
1400 PRINT "MISS"
1410 GOTO 1920
1420 REM P= DESTRUCTIVE POWER OF WEAPON. D= INCREMENTAL DAMAGE DONE, LIMIT
1430 REM TO A MAX VALUE OF 2.
1440 PRINT "HIT"
1450 LET D=P*(1-S(5,J))
1460 IF D <= 2 THEN 1480
1470 LET D=2
1480 LET S(7,J)=S(7,J)+D
1490 LET S(5,J)=S(5,J)-P/100
1500 LET Z=10-INT(S(7,J))
1510 IF J=1 THEN 1580
1520 REM DAMAGE DONE TO KLINGON
1530 PRINT "SCANNER REPORT KLINGON"
1540 IF Z>1 THEN 1610
1550 PRINT "KLINGON DESTROYED"
1560 STOP
1570 REM DAMAGE TO ENTERPRISE
1580 PRINT "ENTERPRISE DAMAGE RPT"
1590 IF Z>0 THEN 1610
1600 LET Z=1
1610 GOTO Z OF 1620,1640,1850,1730,1730,1780,1780,1780,1830,1830
1620 PRINT "ENTERPRISE DESTROYED"
1630 STOP
1640 PRINT "WEAPON SYST. DESTROYED"
1650 PRINT "WARP DRIVE DESTROYED"
1660 PRINT "MAJOR STRUCTURAL DAMAGE"
1670 PRINT "SHIELDS BUCKLING"
1680 LET S(1,J)=0
1690 LET S(2,J)=0
1700 LET S(3,J)=0
1710 LET S(4,J)=0
1720 GOTO 1920
1730 PRINT "PHASORS DESTROYED"
1740 PRINT "MINOR DAMAGE AMIDSHIPS"
1750 PRINT "SHIELDS WEAKENING"
1760 LET S(1,J)=S(2,J)=0
1770 GOTO 1920
1780 PRINT "FOREWARD PHASORS DESTROYED"
1790 PRINT "MINOR DAMAGE AMIDSHIPS"
1800 PRINT "SHIELDS WEAKENING"
1810 LET S(1,J)=0
1820 GOTO 1920
1830 PRINT "SHIELDS HOLDING NO DAMAGE"
1840 GOTO 1920
1850 PRINT "WEAPONS SYST. DEACTIVATED"
1860 PRINT "DILITHIUM CRYSTALS OVERHEATING"
1870 LET S(1,J)=0
1880 LET S(3,J)=0
1890 LET S(3,J)=0
1900 GOTO 1920
1910 REM NEW RANGE, BEARING
1920 LET R3=0.5*(S(4,1)+S(4,2))+0.05*(S(6,1)+S(6,2))
1930 LET R=R+R3*(RND(1)-0.5)*1E+06
1940 LET R=ABS(R)
1950 LET B=ABS(B-150+20*(RND(1)))
1960 IF B>180 THEN 2000
1970 LET I=J
1980 LET J=3-I
1990 GOTO I OF 260,360
2000 LET B=ABS(360-B)
2010 GOTO 1970
2020 END

```

- 4) Warp and impulse drive engines, the status of which are taken into account to compute the incremental changes in range between turns.
- 5) Evasive maneuvering to escape out of sensor range and end the game in a draw.

The player, as captain of the Enterprise, alternates moves with the Klingon battle-cruiser. After a readout of the range and bearing to the enemy vessel, the player is queried for his command. After each move the player receives a status report of the Enterprise or the Klingon.

Each ship is armed with a forward twin phasor bank, a single rear phasor, and a forward firing photon torpedo tube. These weapons have relative destructive powers of 4, 2 and 8 respectively. The probability of achieving a hit with the phasors is given by (line 830):

$$P_H = 1 - R/(5E06) \text{ where}$$

R is the range in kilometers between ships. For ranges over 5 million kilometers the phasors are useless. The forward phasors may be used for bearings 0 to 90 degrees, the rear phasors for bearings 90 to 180 degrees. The photon torpedo tube may be used for targets bearing 0 to 90 degrees for which the range is at least 2 million kilometers. The probability of a hit is given by (line 1090):

$$P_H = (1 - B/90) * (1 - 2R/1E08).$$

While the phasors are range dependent, the photon torpedo is almost entirely bearing dependent.

Both ships have options for maneuvering to attack and trying to break contact. Maneuvering to attack halves the range and brings the bearing to zero. This command appears most useful when used to close in on

SUMMARY OF INSTRUCTIONS FOR THE GAME

The player will engage a Klingon battle cruiser and will alternate moves with the Klingon.

When the prompting message "STARDATE?" appears, enter any random number to initialize the game. This is a seed for the pseudorandom number generator, and using a different number each game prevents repetition of the same battles.

Enter the command after the prompting message "COMMAND" appears. Select commands from the following list:

Command 1 fires forward phasors, of which there are two.

Command 2 fires the rear phasor.

Command 3 fires the photon torpedo. The photon torpedo fires forward. The minimum photon torpedo range is 5 million kilometers.

Command 4 means "maneuver to attack." The bearing to the target Klingon and the range are reduced.

Command 5 means "attempt to break contact" by using the warp drive. If the range goes greater than 100 million kilometers, contact is lost and the game ends.

Command 6 is the suicide command, the end game maneuver used to prevent capture by Klingons.

Command 7 is surrender to the Klingons.

The relative strength of a photon torpedo is 8, the relative strength of the rear Phasors is 2, and the relative strength of the forward phasors is 4.

a fleeing or crippled foe. Attempting to break contact opens the range. (Along about the time your weapon systems are deactivated, your shields are half gone and the Klingon is closing in, it is time to get it in gear and haul for Alpha Centauri.) The game ends in a draw when the range exceeds 100 million kilometers.

Damage assessments are provided any time a vessel is hit by a phasor or photon torpedo. The amount of damage done depends on the relative strength of the weapon (8 for a photon torpedo, 4 or 2 for phasors) as well as the effectiveness of the deflector shields. The amount of damage done is computed and added to the cumulative damage, and the deflector shield effectiveness is reduced.

Two other commands, surrender and self-destruct, are included for defeatists.

The program is written in a version of BASIC which permits GOSUB...OF... and GOTO...OF... statements, and may require minor reprogramming for other BASIC languages. The random number function, RND (X) generates a random variable with uniform distribution in a range (0,1). The argument X, when negative, is used as the random number seed; when positive the argument is ignored and an internal seed is used by the generator. By entering a "stardate" at the beginning of each game, a unique pseudo-random number series is generated for that game.

The Enterprise and the Klingon use the same routines for command processing. Lines 780 – 1030 determine if a phasor shot produced a hit or a miss. Lines 1040 – 1130 process a photon torpedo command. Maneuvering to attack is handled in lines 1140 – 1170, while attempting to break contact transfers control to lines 1180 – 1240. The damage assessment routine in lines 1380 – 1890 prints out the scanner reports or damage control reports. At the end of each move the subroutine in lines 1910 – 2000 is called to change the range and bearing.

The computer selects the Klingon's move in lines 350 – 770. Presently the Klingon is programmed to be somewhat aggressive. For novice Star Trek players the Klingon wins about 75 percent of the first several dozen games. ■

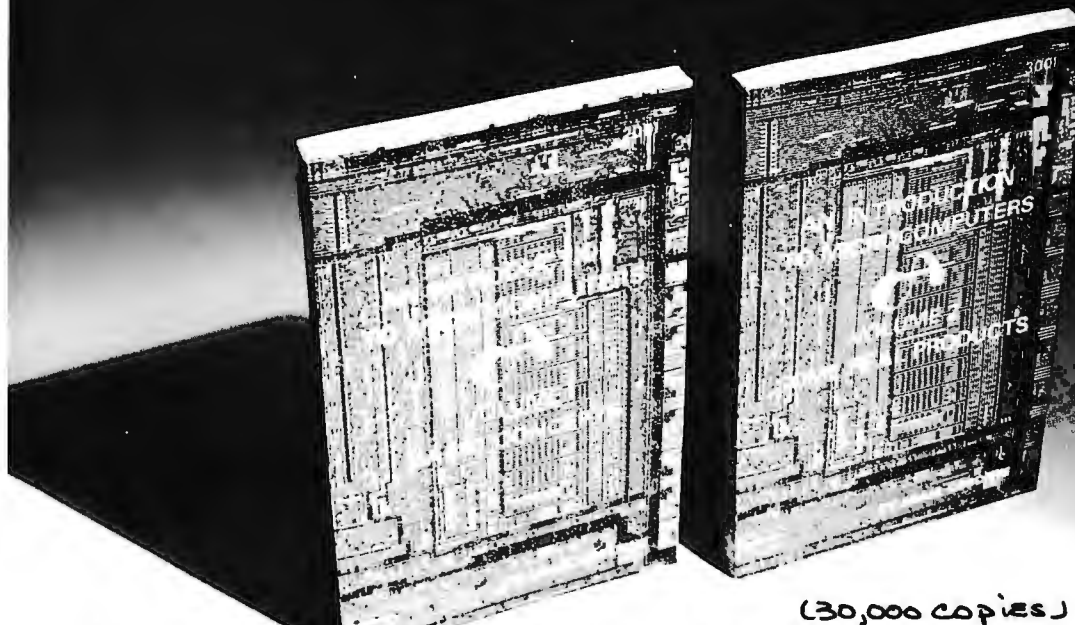
Listing 2: A sample run of this version of Star Trek.

```

ENTER STARDATE
 6091.1
KLINGON APPROACHING
R= 3849000 KM. BEARING= 69.282 DEG.
COMMAND
 3
MISS
KLINGON FIRES PHOTON TORPEDO
MISS
R= 3661890 KM. BEARING= 56.722 DEG.
COMMAND
 4
KLINGON FIRES PHASOR
HIT
ENTERPRISE DAMAGE RPT
SHIELDS HOLDING NO DAMAGE
R= 2021835 KM. BEARING= 15.44 DEG.
COMMAND
 3
HIT
SCANNER REPORT KLINGON
SHIELDS HOLDING NO DAMAGE
KLINGON FIRES PHASOR
HIT
ENTERPRISE DAMAGE RPT
SHIELDS HOLDING NO DAMAGE
R= 1658325 KM. BEARING= 8.48 DEG.
COMMAND
 3
MISS
KLINGON FIRES PHASOR
HIT
ENTERPRISE DAMAGE RPT
SHIELDS HOLDING NO DAMAGE
R= 2092815 KM. BEARING= 10.48 DEG.
COMMAND
 3
HIT
SCANNER REPORT KLINGON
FOREWARD PHASORS DESTROYED
MINOR DAMAGE AMIDSHIPS
SHIELDS WEAKENING
KLINGON FIRES PHASOR
MISS
R= 2266905 KM. BEARING= 13.12 DEG.

```

Seeing Double?



No it's just that the first edition of this popular book sold out in 5 months flat.

The second edition of **AN INTRODUCTION TO MICROCOMPUTERS** has been revised and expanded to two volumes, to keep up with the fast-moving microcomputer industry's pace of change.

Have things changed that much in just 5 months? You probably know the answer to that already: a flood of new and second-source CPU chips and a whole host of new LSI support packages. Nowhere else can you get all this vital information. **VOLUME I — BASIC CONCEPTS** surveys the subject, from elementary concepts to system configuration and design. In **VOLUME II — SOME REAL PRODUCTS** we take a long, hard look at the products available today and about to be announced tomorrow. If you're actually putting together your first system, you may find some help in the latest addition to the microcomputer library: **8080 PROGRAMMING FOR LOGIC DESIGN**.

Whether you are planning new products, updating existing ones, or simply need to keep abreast of the latest developments in this revolutionary new technology, **AN INTRODUCTION TO MICROCOMPUTERS** has become the industry's standard reference on the subject.

VOLUME I — BASIC CONCEPTS, takes you by the hand, from elementary logic and simple binary arithmetic through the concepts which are shared by all microcomputers. It tells you how to take an idea that may need a microcomputer and create a product that uses one. This book is complete — every aspect of microcomputers is covered: the logic devices that constitute a microcomputer system; communicating with external logic via interrupts, direct memory access, and serial or parallel I/O; microprogramming and macroprogramming; assemblers and assembler directives; linking and relocation — everything you need to know if you are going to select or use a microcomputer. Volume I is equivalent to Chapters 1 through 6 of **AN INTRODUCTION TO MICROCOMPUTERS**, first edition, but with extensive new sections on chip slice products and serial I/O. Order publication number 2001. Date available: May 31, 1976.

VOLUME II — SOME REAL PRODUCTS, covers real microcomputers, in considerable detail. Every major microcomputer: 4-bit, 8-bit or 16-bit, is described, including some soon to be announced products. Major chip slice products are also covered. More than 20 microcomputers in all. Order publication number 3001. Date available: July 15, 1976.

8080 PROGRAMMING FOR LOGIC DESIGN, is a completely new book on a totally new subject: implementing digital and combinatorial logic using assembly language within an 8080 microcomputer system. What happens to fan-in and fan-out? How do you implement a one-shot? This book simulates well known digital logic devices using assembly language; next it shows you how to simulate an entire schematic, device by device, keeping the assembly language simulation as close to the digital logic as possible. But that is the wrong way to use a microcomputer; the book explains why, then shows you the correct way. This book describes the meeting ground of programmer and logic designer; it is written for both readers. Order publication number 4001. Date available: June 15, 1976.

For ordering and pricing information in Europe contact:

SYBEX

Publications Department
313 Rue Lecourbe
75015 — Paris, France
Telex: 200858 Sybex

Everywhere else:



OSBORNE & ASSOCIATES INC.

P.O. Box 2036 • Berkeley, California 94702 • (415) 548-2805

Copies of each book are available for \$7.50 per book plus sales tax for California residents. Discounts are available on orders of 100 books or more. Please be sure to include publication numbers with order. Add 30 cents for surface mail, \$2.50 for airmail shipping charge when ordering from outside the U.S.A. Dealer, distribution and translation inquiries welcome.

BankAmericard and Master Charge orders accepted • Please include in mail order: card number, expiration date, full name and address

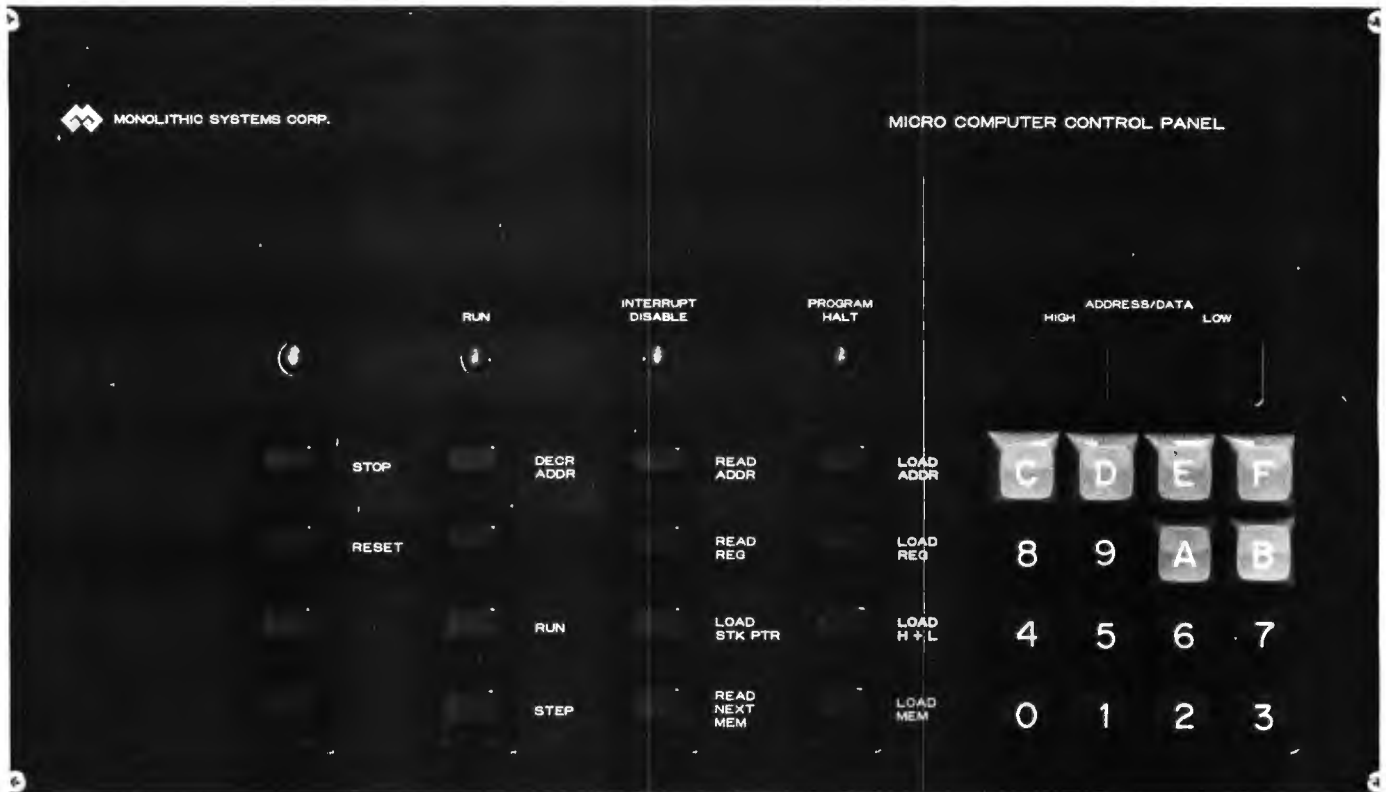


Photo 1: One of the keys to the ease of use of this Monolithic Systems Corporation "8080+" microcomputer is its "smart" control panel. Instead of row after row of blinking lights, this panel uses software to drive a hexadecimal display, with a hexadecimal data entry keypad and several control function buttons. The photographs accompanying this article are supplied courtesy of Monolithic Systems Corp, 14 Inverness Dr E, Englewood CO 80110.

The MSC 8080 + Microcomputer as a Personal System

Ken Barbier
PO Box 1042
Socorro NM 87801

In the beginning there were rows and rows of little lights and little switches. Incredible as it may seem, some micro, mini, and mega computers still come with rows and rows of little lights and little switches. After wearing out countless eyeballs, fingertips, and four letter words, Mankind finally asked: "Isn't there a better way?"

Along came the monitor program in read only memory, allowing instant power up and communication with a console terminal device. This is an excellent solution when higher level languages are available, and not too bad a solution when text editors and assemblers are available. But for entering machine language routines of any length,

and for debugging programs which are intimately connected with hardware, such as device drivers, the terminal has its limitations.

Enter the intelligent control panel, with hexadecimal or octal keyboard and readout. Machine language programs or data blocks can be swiftly entered, and debugging by single stepping through a program while monitoring memory or registers becomes a snap. For the designer of small systems working down at the machine language or hardware level the intelligent control panel can be an attractive alternative.

The MSC 8080+, from Monolithic Systems Corp, Englewood CO, is an Intel 8080

based microcomputer with what is undoubtedly one of the most complete "smart" panels on the market. It includes other unique features which make it a good choice for the person building or experimenting with small systems.

Some Features of the MSC 8080+

Unpack the MSC 8080+, connect a single +5 VDC @ 2 A power supply, press the RESET switch on the control panel, and you have an operating microcomputer. This ease of setup was one of the reasons I chose this product as my personal computer. The control panel monitor program takes up two of the four 1702A EROM positions on the processor board, and uses a maximum of 64 bytes of the 1 KB static programmable memory, also on the board. The panel monitor program has its own hardware interrupt via the RESET switch, and does not interfere with use of the eight RST instructions which are provided for interrupts by the 8080 processor itself. (A separate CPU RESET switch provides access to interrupt 0, once your program is loaded.)

Other goodies supplied include a 4.5 inch by 7 inch (11.43 cm X 17.78 cm) wire wrap area right on the processor board, with room for 42 sixteen pin sockets (12 of these are used up by the 1 KB RAM). Surrounding this area are nine 26 pin connectors, accepting either wire wrap directly, or ribbon cable connectors, to provide access to peripherals. All connections to the 8080 processor are already buffered before reaching the wire wrap section, simplifying the addition of peripheral interfaces.

On the MSC 8080+ processor board there is also a DC to DC inverter to provide the +12, -5, and -9 VDC required for the 8080 integrated circuit and the 1702As, so that only the single +5 VDC supply is necessary. Connectors are provided for a number of

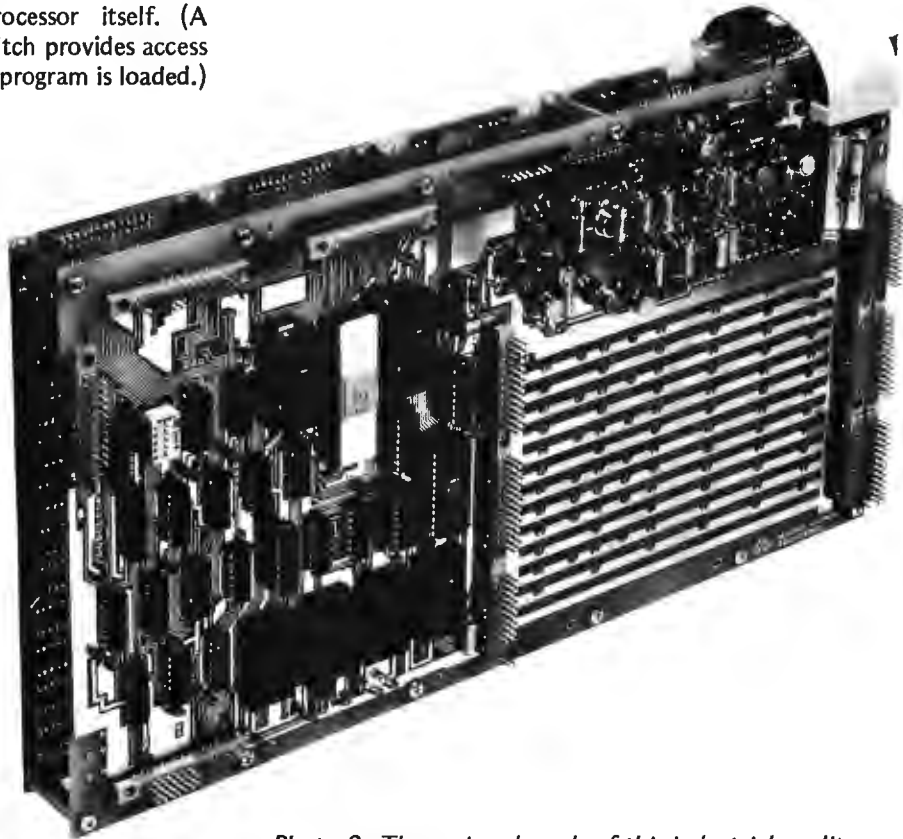


Photo 2: The various boards of this industrial quality product are designed to be stacked using 1 inch (2.54 cm) spacers. This view illustrates the processor board (front) and dynamic programmable memory board (rear) mounted together with spacers; interconnections throughout an MSC 8080+ system are made using 26 conductor parallel ribbon cable assemblies like the one in the upper right hand corner of this picture. Also note the uncommitted wire wrapping area which can be used for custom logic designs oriented to a specific application system.

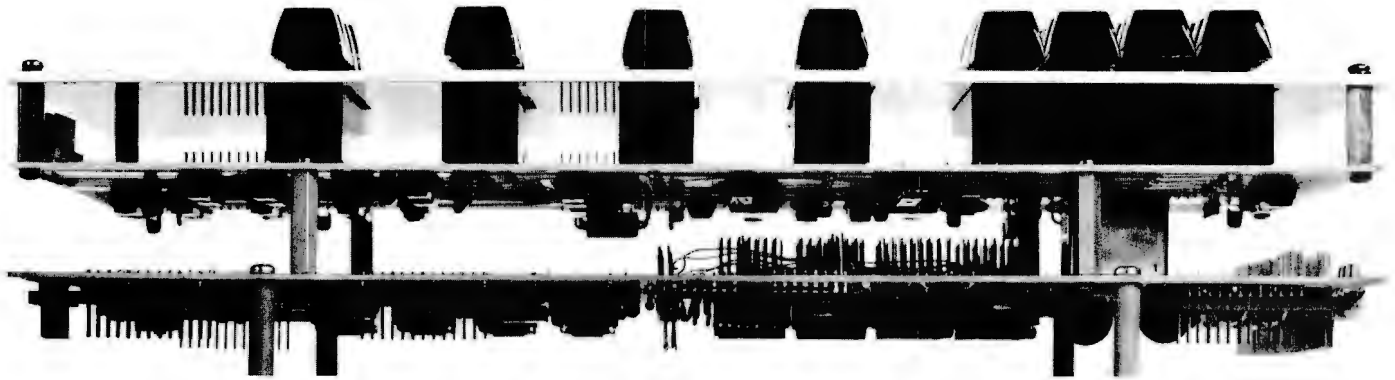


Photo 3: A side view of the control panel and processor board stacked together for a minimal system. The control panel consists of the metal cosmetic panel (top) and a circuit board to which key switches are attached (middle). The processor board is shown at the bottom of this assembly. For purposes of photography, interboard connector cables have been omitted in this view. Other boards of the family could be added to this stack. (The author's system has an additional 4 K CMOS programmable memory board with battery backup added to the two boards shown here.)

memory options, which are detailed later in this article.

The control panel is what I found to be the outstanding feature of the MSC 8080+. It has a 16 key hexadecimal keypad, a four digit hex display, 16 function keys, and four status indicator LEDs. A 20 mA current loop TTY interface is provided on the panel, but an additional 1702A (optional) is required to drive it. This 1702A Teletype interface contains a program with timing loops to perform the parallel to serial conversions, a software UART algorithm.

The MSC 8080+ is intended for the industrial market, so the quality of manufacture and components is first rate and the unit comes assembled. Industrial quality design is one of my reasons for choosing this processor. In spite of this, the goal of the under \$1000 computer is met.

Control Panel Operation

As supplied, the memory address of the panel monitor program is (in hexadecimal) from 0200 through 03FF, and the 1 KB RAM can be found at 0400 through 07FF. Some functions of the monitor use the stack, so the next operation after RESET should be to initialize the stack. This is done by entering 07FF on the hexadecimal keypad and pressing the LOAD STK PTR function key. As the digits are entered they will appear, shifting into the hex display from the right, and will disappear when loaded into the desired register. The functions LOAD STK PTR, LOAD ADDR, and LOAD H+L use 16 bit (4 digit) entries; all other entries are eight bits (2 digits).

A user's program can now be keyed in. The start address is set by entering four digits and using the LOAD ADDR key. The address entered will disappear from the display and the eight bit contents of the addressed location will appear in the two low order display digits as LOAD ADDR is pressed, indicating proper operation of the system. Now enter two digits of data or program and press LOAD MEM. At this time an address register in the panel will be incremented and the contents of the next sequential memory location will be displayed. If a load error occurs (the panel reads back each entry from memory to verify it) an error indication of "FF" appears in the two high order digits of the display. This gives an instant indication if you are trying to write into ROM, or a non-existent address, or hardware that is malfunctioning.

Loading each sequential memory location from the panel thus consists of entering and verifying two hex digits of data and pressing LOAD MEM. At any time during loading, the address of the next sequential location can be displayed by pressing READ ADDR. When loading is complete you can verify the program by entering the start address, then using READ NEXT MEM to examine each location in turn.

Once your program is entered, initial values of any register can be set using the LOAD REG, LOAD H+L, or LOAD STK PTR keys. Enter your starting address using LOAD ADDR, and you are ready to run.

Unless you have infinite confidence in your infallibility, you may want to single step through the program the first time. Just

press STEP to execute each instruction in turn. The address of the next instruction will then be displayed. The contents of any register can be examined (READ REG) or changed (LOAD REG) as you step through your program. DECR ADDR will allow you to back up the program counter one byte at a time.

When you are confident the program is fully debugged, enter the start address and press RUN to execute it. If things do not go as planned, press STOP to halt the program and display the address of the next instruction. Registers and memory can then be examined.

Larger segments of programs, or long loops that would take all day to single step through, can be run by temporarily patching in the HALT instruction where traps are desired. After starting the program with the RUN key, the PROGRAM HALT indicator will light when you reach the HALT instruction. Then simply press STOP to display the next program address and enable all of the other panel functions.

The 8080+ control panel uses a combination of hardware and software, but its operation is transparent to the user's program. If the user's program should end up in the illegal combination of disabled interrupts and program halt, the panel RESET key will restore operation without it being necessary to turn the power off.

It is hard to believe without experiencing it how easily a program can be keyed in and debugged using the MSC 8080+ control panel. It makes an expert out of a novice in minutes.

Hardware Configuration

One unusual aspect of the MSC unit is the absence of edge connectors on the boards. All connections between the control panel, processor board, and optional memory boards are through 26 conductor ribbon cables and matching connectors. The boards can be physically stacked in endless combinations using #4-40 X 1 inch threaded spacers, or can be mounted in Augat 8170 series frames. As the components of the system are intended to be a part of the user's industrial hardware, no cabinets or power supplies are furnished.

The control panel, processor, and dynamic programmable memory boards are all 7 1/2 inches by 13 1/2 inches (19.05 X 34.29 cm). The CMOS programmable memory is slightly smaller on the long dimension but has compatible hole patterns for the spacers or frame mounting.

Currently available options include the

processor board without programmable memory, and no EROMs installed in the four sockets; a dynamic programmable memory board with room for 32 KB; and a nonvolatile 4 KB CMOS static programmable memory board with built in NiCad batteries which are kept charged during normal operation. In the works, according to MSC, is a compatible EROM board with pre-loaded software including a text editor and assembler.

A User Comments on the MSC 8080+

For years I had been waiting for the price of some old worn out mini to come within reach of a meager hobbyist budget, but before that could occur the age of microprocessors was upon us. I didn't feel that I had the time to spare to put together a system from a handful of parts, so I watched the "processor on a board" market develop with much interest. Prices were still high, but falling rapidly, when the Altair explosion occurred. I was instantly tempted by the first Altair ad, but since I had no TTY or other terminal to go with it the investment required for any sort of useful configuration was still several kilobucks. And there were all those rows of lights and switches! I had too many of those to contend with while earning a paycheck; I resolved that any system I had at home would have minimal blinking lights! So I watched, and waited, and collected specifications sheets, and compared instruction sets.

I think too little has been said about the relative merits of micros and minis when comparing instruction sets. It is not enough to have bunches of instructions and memory addressing methods. It is not enough to have all kinds of tricks to conserve memory. To be truly useful a machine must have a set of instructions that are easy to learn, easy to remember, easy to use, and suited to the task at hand. A calculator will beat any micro at number crunching, but is lost as a controller. (How long must we wait for the micro-controlling-a-calculator chip?)

Having worked with machines from big IBM size to hand held calculators, I had a pretty good idea of what I wanted for a home controller, game player, and accounting system. The Intel 8008 didn't quite make it, but when I saw the instruction set of the Intel 8080, I flipped! All that simplified CALLing and RETurning, PUSHing and POPping, and decimal adjust too!

So now I knew my system would use the 8080. I started trying to design a "smart" control/display panel. From the day I saw the first Altair ad til I found what I wanted

For more information on the MSC 8080+ contact: Monolithic Systems Corp, 14 Inverness Dr E, Englewood CO 80110. Their phone number is (303) 770-7400.
--

Altair Owners...

Have you heard our new hit single?

THE FIRST DOUBLE DENSITY FLOPPY DISK SYSTEM FOR MICROCOMPUTER SYSTEMS!



MSI is proud to introduce the FD-8 floppy disk memory system for use with the Altair 8800, Motorola 6800, and other micro computer systems.

The FD-8 requires only one PIA chip for interfacing to any microcomputer system. One 8 bit bidirectional data port and one 8 bit control port and that's it! Complete FDOS software for 8080 and 6800 systems is provided and can operate from either RAM or ROM as desired.

Format Specifications:

- 77 Tracks/Diskette
- 16 (32) Sectors/Track*
- 256 Bytes/Sector
- 315,392 (630,784*) Bytes/Diskette

*Model FD-8-II with Double Density

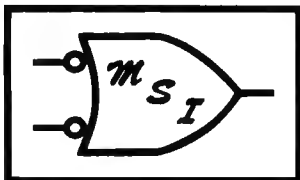
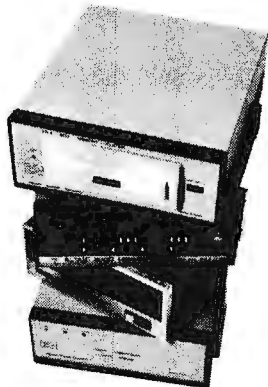
Introductory Prices:

SINGLE DRIVE MODELS:

SINGLE DENSITY, KIT.....\$ 950.00
 DOUBLE DENSITY, KIT.....\$ 1250.00

DUAL DRIVE MODELS:

SINGLE DENSITY, KIT.....\$ 1750.00
 DOUBLE DENSITY, KIT.....\$ 2050.00



Midwest Scientific Instruments

BASIC FOR THE 6800

We now have full blown BASIC ready for delivery. It has full floating point arithmetic, strings, subroutines, arrays, and USER DEFINED VERBS!
 Price \$65.00 per copy (Includes documentation and paper tape)

PROM/RAM BOARD FOR ALTAIR 8800 AND IMSAI SYSTEMS

Contains 3,840 bytes of ROM and 256 bytes of scratchpad RAM on one board. Extensive monitor software provides memory examine/change using octal or hex, trap functions, bootstrap loaders for MITS BASIC and monitor software, paper tape and cassette dump and load functions using hex or BNPF formats. Also contains a DISASSEMBLER and a RELOCATOR.
 Price \$245.00

ADDITIONAL PRODUCTS FOR THE SWTP 6800

Wire Wrap Board

Accepts 40 pin, 24 pin, 16 pin and 14 pin sockets as well as discrete components. Contains 7 805 on-board regulator for +5V power bus. Plugs into the SWTP 6800 mother board.
 Price \$25.00

PROM/RAM Board

Contains 3,840 bytes of 1702A ROM and 256 bytes of RAM on one board. May be used to contain MSI-FDOS software and scratchpad area.
 Price \$95.00

MSI SOFTWARE PRODUCTS:

12K 6800 Basic.....\$65.00
 Relocating loader program...\$15.00
 Disassembler program.....\$25.00
 1K Mini Assembler program...\$25.00

We've made several other hits at MSI, including our floppy disk system for programmable calculators, CRT terminals, PROM programmers, high speed printers, and instrumentation interfaces.

For more comprehensive product information, write MSI at the address below. Watch our monthly ad, and we'll introduce you to our new products for the computer hobbyist.

from MSC, I spent long hours of free time trying to design the perfect control panel. All my designs were bogged down in excessive amounts of hardware, so too expensive. I gave up on the panel idea, and started building a CRT terminal, although I had nothing to connect it to yet.

Suddenly there appeared before me (in one of the electronics trade magazines) a description of Monolithic Systems Corporation's 8080 based processor board with single +5 VDC power supply and room left over for all my interface circuitry. Ideal! By the time I called them, they were announcing the MSC 8080+ system, with that neat processor board *and* a smart panel. I dug out my old panel design sketches and sure enough they had stolen all my ideas by long distance telepathy! And added lots of functions I would never have thought of.

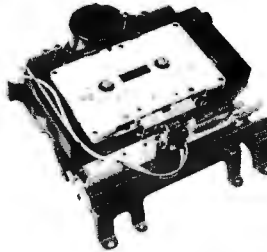
I wasted no time in ordering an MSC 8080+. When it arrived it took me only one weekend to connect up my CRT terminal hardware, key in and debug the software I had previously written, and have a smart CRT terminal in operation. This is a tribute both to Monolithic's interfacing documentation, and the speed of operation possible with this control panel.

A 4 KB CMOS board arrived later, and after hooking it up I was able to turn things off without losing all my software. Of course it is still possible to blow my programs by writing stupid mistakes into them, but the ability to single step through program segments has all but eliminated that problem. (Most debug programs used with a console terminal have a limited number of settable traps, or breakpoints, and it is too easy to sneak past them all and get totally lost. Not so with single stepping.)

This combination has proved to be an ideal solution to the problem of putting together an inexpensive home computer, especially as I had no method of program storage with the power off. While the cost is not as low as some systems advertised in BYTE, there are many tangible benefits that come with the small extra expenditure. The panel has all the functions you'll ever need for machine language programming. The system was factory assembled and tested, built of the best quality components and fully guaranteed. All of the "works" are hidden behind a professional appearing front panel, so it doesn't look like a collection of surplus parts. And, delivery was on a realistic schedule.

The least I can say is that I am completely satisfied with this product, and don't hesitate to recommend it to other computer-heads. ■

All-new Phi-Deck,[®] precision remote controlled cassette transports starting at under \$100!



- Featuring:
- Re-engineered precision parts
 - New cast frames
 - 4 motor reliability
 - Remotely controlled
 - Precise, fast head engage/disengage
 - Quick braking
 - Various speed ranges

Electronic packages for control or read/write

- For application in:
- | | |
|-----------------------------------|---------------------------------------|
| 1. Micro processing | 6. Data duplicating |
| 2. Data recording/logging/storage | 7. Security/automatic warning systems |
| 3. Programming | 8. Test applications |
| 4. Instrumentation | 9. Audio visual/education |
| 5. Industrial Control | 10. Hi-Fi |
| | 11. Others |



Triple I A Division of The Economy Co.

1901 North Walnut P.O. Box 25308
Oklahoma City, Oklahoma 73125 (405) 521-9000

- I am interested in application no. _____
 Have Representative call Send application notes

Name _____ Title _____
 Company Name _____
 Address _____
 City _____ State _____ Zip _____
 Phone Number _____

BROWN-OUT PROOF your ALTAIR 8800

With the unique **Parasitic Engineering** constant voltage power supply kit. A custom engineered power supply for your Altair. It has performance features that no simple replacement transformer can offer:

- *BROWN-OUT PROOF: Full output with line voltage as low as 90 volts.
- *OVER-VOLTAGE PROTECTION: Less than 2% increase for 130 volt input.
- *HIGH OUTPUT: 12 amps @ 8 volts; 0.5 amps @ ±16 volts. Enough power for an 8800 full of boards.
- *STABLE: Output varies less than 10% for any load. Regulators don't overheat, even with just a few boards installed.
- *CURRENT LIMITED: Overloads can't damage it.
- *EASY TO INSTALL: All necessary parts included.

only \$75 postpaid in the USA.
calif. residents add \$4.50 sales tax.

Don't let power supply problems sabotage your Altair 8800.

PARASITIC ENGINEERING

PO BOX 6314

ALBANY CA 94706

How to Do a Number of Conversions

James Brown
2518 Finley St #636
Irving TX 75062

Table 1: Hexadecimal Codes of Selected ASCII Characters (high order bit assumed zero).

Hexadecimal Code	ASCII Character	Hexadecimal Code	ASCII Character	Hexadecimal Code	ASCII Character
00	NUL	30	0	40	@
:		31	1	41	A
0A	line feed	32	2	42	B
:		33	3	43	C
0D	car. ret.	34	4	44	D
:		35	5	45	E
20	space	36	6	46	F
:		37	7	47	G
2B	+	38	8		
2C	,	39	9		
2D	-	3A	:		
2E	.				
2F	/				

Listing 1a: The BIN Routine Specified for an 8080. This listing, as all the listings of this article, shows the symbolic code and absolute machine code for an 8080 version of the routine. The notes refer to absolute addresses which must be adjusted when relocating the code to some address in memory address space. BIN reads the '1' and '0' characters of an ASCII encoded binary string, leaving up to 8 bits of input in B.

	Rel. Addr.	Code	Label	Op.	Operand	Commentary
	0000	06 00	BIN:	MVI	B, 0	ANSWER := 0;
Note 1	0002	CD xx xx	BINLOOP:	CALL	GET	A := INPUT [character];
	0005	FE 30		CPI	'0'	is A LT '0'?
	0007	D8		RC		if so then return;
	0008	FE 32		CPI	'2'	is A LT '2'?
	000A	D0		RNC		if not then return;
	000B	1F		RAR		CARRY := A ₀ ;
	000C	78		MOV	A, B	A := ANSWER;
	000D	17		RAL		rotate carry into A;
	000E	D8		RC		overflow: if CARRY = 1 then return;
	000F	47		MOV	B, A	ANSWER := A;
Note 2	0010	C3 xx xx		JMP	BINLOOP	reiterate for next bit;

Note 1: address of GET should replace "xx xx".

Note 2: "xx xx" should be the address of BINLOOP.

Perhaps one of the more difficult tasks on any small computer is the conversion from numeric characters to a form usable by the machine and back again. That is, given some type of input output device (Teletype or TV typewriter) connected to your computer, it would be desirable to have the capability of entering a string of numeric characters (consecutive digits) through the keyboard. The computer would then perform some operation on that number. Finally, the result of that computation is displayed back on the IO device. Since the computer's natural language is *N bit* binary (i.e., ones and zeros), how can such a string be converted? An example of the problem is: How do I convert the three character decimal string '196' into the binary integer equivalent 1100 0100 (or octal 204, or hexadecimal C4)?

Converting a decimal (base 10) number into binary can be a long and involved operation. Let us work our way into decimal conversion by considering what would be

necessary to do the following conversions in order of increasing complexity:

1. Binary character strings (ASCII 0 or 1) to or from unsigned 8 bit integers.
2. Octal character strings (ASCII 0 to 7) to or from unsigned 8 bit integers.
3. Hexadecimal character strings (ASCII 0 to 9, A to F) to or from unsigned 16 bit integers.
4. Signed decimal character strings (ASCII 0 to 9, +, -) to or from signed 16 bit integers.

Before we start, let us examine what the computer sees when a character is read from the keyboard, assuming that the keyboard speaks ASCII. Examining table 1, notice that each character is assigned a unique binary value. Not only are the numeric characters 0 thru 9 grouped together; but, if the left hand four bits were dropped, there would be a direct correspondence to the binary equivalents of 0 thru 9. As shown below, this is a fairly simple task:

Algorithm:

'ASCII char' (AND) (0000 1111) = result

Examples

'0': (0011 0000) (AND) (0000 1111) =
0000 0000
'1': (0011 0001) (AND) (0000 1111) =
0000 0001
'9': (0011 1001) (AND) (0000 1111) =
0000 1001

In each case, the result is a binary number in the low order of the byte after the AND operation has masked the high order bits.

Binary Conversions

Converting the ASCII character codes for 1 and 0 into a true binary value is perhaps the simplest to actually implement, and is a good starting point in understanding how number conversions work. All of the other

routines follow the basic plan presented here.

In the preceding, zapping the left four bits to get a binary value has one fatal flaw; it only works for one character. In developing something to handle a two character string such as '10', it might as well accept ASCII strings with any length, as long as the result can be contained in eight bits (an arbitrary choice).

The simplest way of doing this is to perform the conversion one character at a time as they are entered and develop the result as each character of the string ('1' or '0') is processed. Clearly the first step is to read the character and convert it into the binary value 1 or 0, using the masking technique shown earlier.

Since most computers have some type of shift instruction (see note 1), this is an effective way of moving the new bit into the result which is being calculated. Specifically, we must shift the result left one bit and then OR the new converted value to it. This is mathematically equivalent to multiplying by 2 and adding. For example, the four character binary string '1011' is entered and converted to the binary number 1011. This is equivalent to the expression:

$$1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 11$$

(base 10)

and could be accomplished by the following sequence:

1. answer: = 0
2. INPUT character
3. character: = character (AND) 01 (hex)
4. answer: = answer (SHIFT LEFT) 1
5. answer: = answer (OR) character
6. GO TO 2.

If those four characters were all I wanted to enter, I now need to tell the computer to stop looping, since there is a possibility of entering as many as eight characters. The

	<u>Rel. Addr.</u>	<u>Code</u>	<u>Label</u>	<u>Op.</u>	<u>Operand</u>	<u>Commentary</u>
	0000	0E 08	BOT:	MVI	C, 8	CNT := 8;
	0002	78	BOTLOOP:	MOV	A, B	A := ANSWER;
	0003	07		RLC		CARRY := A ₇ ; rotate A Left;
	0004	47		MOV	B, A	ANSWER := A;
	0005	3E 18		MVI	A, 18H	A := b'00011000';
	0007	17		RAL		rotate A left; A ₀ = CARRY;
Note 1	0008	CD xx xx		CALL	PUT	OUTPUT := A;
	000B	0D		DCR	C	CNT := CNT - 1;
Note 2	000C	C2 xx xx		JNZ	BOTLOOP	if CNT NE 0 then repeat;
	000F	C9		RET		else return;

Note 1: address of PUT should replace "xx xx".

Note 2: "xx xx" should be the address of BOTLOOP.

Listing 1b: The BOT Routine Specified for an 8080. This routine writes out a string of 8 binary encoded ASCII digits, taken from the B register.

Figure 1a: The BIN Routine Flowchart. This routine treats successive ASCII '0' and '1' characters of input as the digits of a binary string. The digits are shifted into ANSWER until an illegal character or overflow returns control. In the 8080 code of listing 1a, ANSWER is register B.

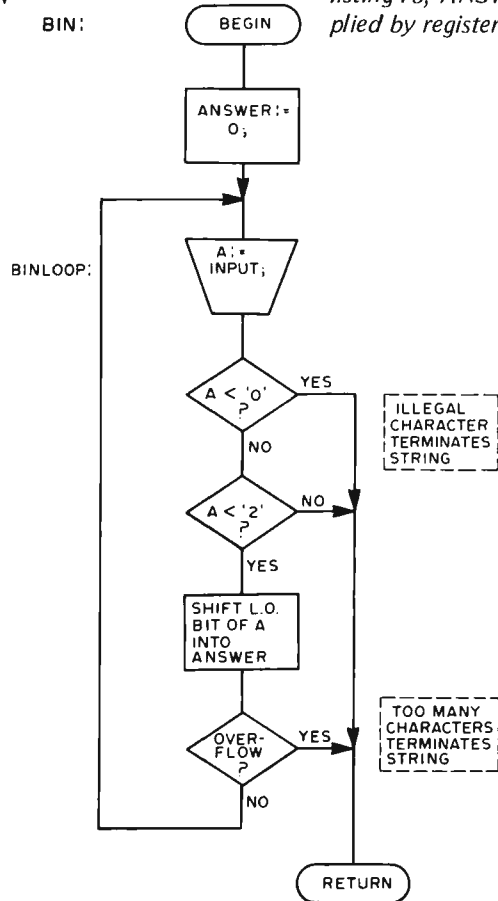
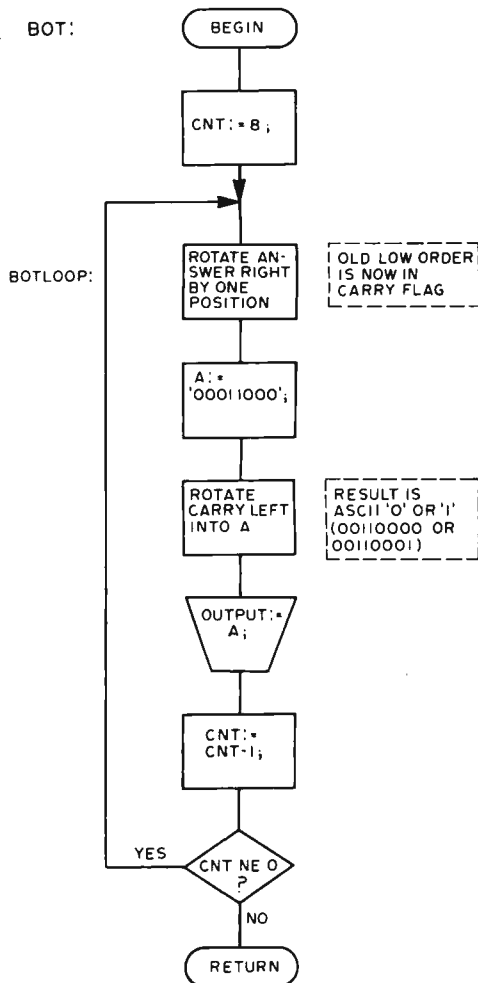


Figure 1b: The BOT Routine Flowchart. BOT is a binary output routine which writes an 8 digit ASCII binary string converted from ANSWER. The digits are printed high order first in a loop which shifts out the bits one by one. In the 8080 code of listing 1b, ANSWER is supplied by register B.



Listing 2a: The OIN Routine Specified for an 8080. This routine accepts an input string of ASCII octal characters and collects the results in ANSWER (CPU register B). Conversion ends with invalid characters or an overflow.

	Rel. Addr.	Code	Label	Op.	Operand	Commentary
	0000	06 00	OIN:	MVI	B, 0	ANSWER := 0;
Note 1	0002	CD xx xx	OINLOOP:	CALL	GET	A := INPUT [character];
	0005	FE 30		CPI	'0'	is A LT '0'?
	0007	D8		RC		if so then return;
	0008	FE 38		CPI	'8'	is A LT '8'?
	000A	D0		RNC		if not then return;
	000B	E6 07		ANI	7	A := A & b'00000111' [mask low order];
	000D	4F		MOV	C, A	C := A;
	000E	78		MOV	A, B	A := ANSWER;
	000F	07		RLC		rotate A left three
	0010	D8		RC		bit positions
	0011	07		RLC		and check for
	0012	D8		RC		overflow into
	0013	07		RLC		CARRY after
	0014	D8		RC		each operation;
	0015	B1		ORA	C	A := A OR ANSWER;
0016	47		MOV	B, A	ANSWER := A;	
Note 2	0017	C3 xx xx		JMP	OINLOOP	reiterate for next digit;

Note 1: address of GET should replace "xx xx".

Note 2: "xx xx" should be the address of OINLOOP.

simplest way of doing this is to have the routine recognize some sort of delimiter (ie: some character other than '0' or '1'). Looking, once again, at table 1, the characters space, period, comma, carriage return, line feed, are all less than the character '0', when considered as binary values. This condition is rather handy, since the same set of machine instructions could recognize a variety of delimiters without rewriting if I want to change what delimiter means. Looking further, if the special characters between the 1 and A are excluded as delimiters, the following pair of tests checks for both delimiters and invalid characters.

- If the character is less than a '0' then finished.
- If the character is greater than a '1' then illegal character.

There is one further consideration that this routine should take into account. The routine should check for a string of characters whose value would exceed the maximum value which could be contained in 8 bits (anything over 255 decimal). Notice that the routine really cannot count the number of characters entered since nine zeros and a one are still the value one, even though 10 characters were processed. Most computers have something called a carry bit or overflow flag. During a shift left this carry bit usually receives the most significant bit from the register being shifted. Thus, as soon as the carry bit becomes a one, then the result has overflowed 8 bits; and the number being entered is too big. Figure 1a shows the detailed flow of the binary input procedure; listing 1a shows the 8080 assembly code of this procedure.

	Rel. Addr.	Code	Label	Op.	Operand	Commentary
	0000	0E 03	OOT:	MVI	C, 3	CNT := 3;
	0002	AF		XRA	A	Clear A; Clear CARRY;
	0003	78		MOV	A, B	A := ANSWER;
Note 1	0004	C3 xx xx		JMP	OOTSKIP	skip around POP first time;
	0007	F1	OOTLOOP:	POP	PSW	restore (A, flags);
	0008	17	OOTSKIP:	RAL		rotate A left
	0009	17		RAL		by three
	000A	17		RAL		bit positions;
	000B	F5		PUSH	PSW	save (A, flags);
	000C	E6 07		ANI	7	A := A & b'00000111'
						[mask low order];
	000E	F6 30		ORI	'0'	A := A OR b'00110000'
						[add hexadecimal 30];
Note 2	0010	CD xx xx		CALL	PUT	OUTPUT := A;
	0013	0D		DCR	C	CNT := CNT - 1;
Note 3	0014	C2 xx xx		JNZ	OOTLOOP	if CNT NE 0 then repeat;
	0017	F1		POP	PSW	flush garbage from stack;
	0018	C9		RET		return to caller;

Note 1: "xx xx" should be the address of OOTSKIP.

Note 2: address of PUT should replace "xx xx".

Note 3: "xx xx" should be the address of OOTLOOP.

Output is simply the reverse process but has error checking omitted. Since the input was left to right, the output should be the same. (It is extremely frustrating to enter the character string '1100' and have the string '0011' printed out.) Fortunately most computers have a rotate left instruction (note 1). If I choose to always print 8 characters per 8 bit value (after all, the computer is working, not me), the output routine should perform the following steps:

1. value = value (ROTATE LEFT) 1
2. character = value (AND) 1
3. character = character (OR) '0' (ASCII character code for '0' is hex 30)
4. OUTPUT character
5. GO TO 1.

Figure 1b contains the flow diagram for this procedure, and listing 1b shows typical code for an 8080 computer.

Octal Conversions

For octal input from strings with ASCII characters '0' thru '7', the binary input routine can be used with some modifications. As shown in figure 2a, the illegal character check now looks for something greater than a '7', the shift left is now three bits instead of one, and the mask used on the character during the logical AND operation is now an octal 7.

The octal output routine was a bit of a problem because the value is an 8 bit quantity. Hence, the routine must process the first two bits, then the next three, then the next three, left to right, as indicated on the flow chart. In my implementation, the 8080 had a rotate which would flow through the carry flag. Thus the bits as they are

Listing 2b: The OOT Routine Specified for an 8080. This routine converts the contents of ANSWER (CPU register B) into a 3 digit ASCII string of octal characters, transferring the result to the output device during the conversion.

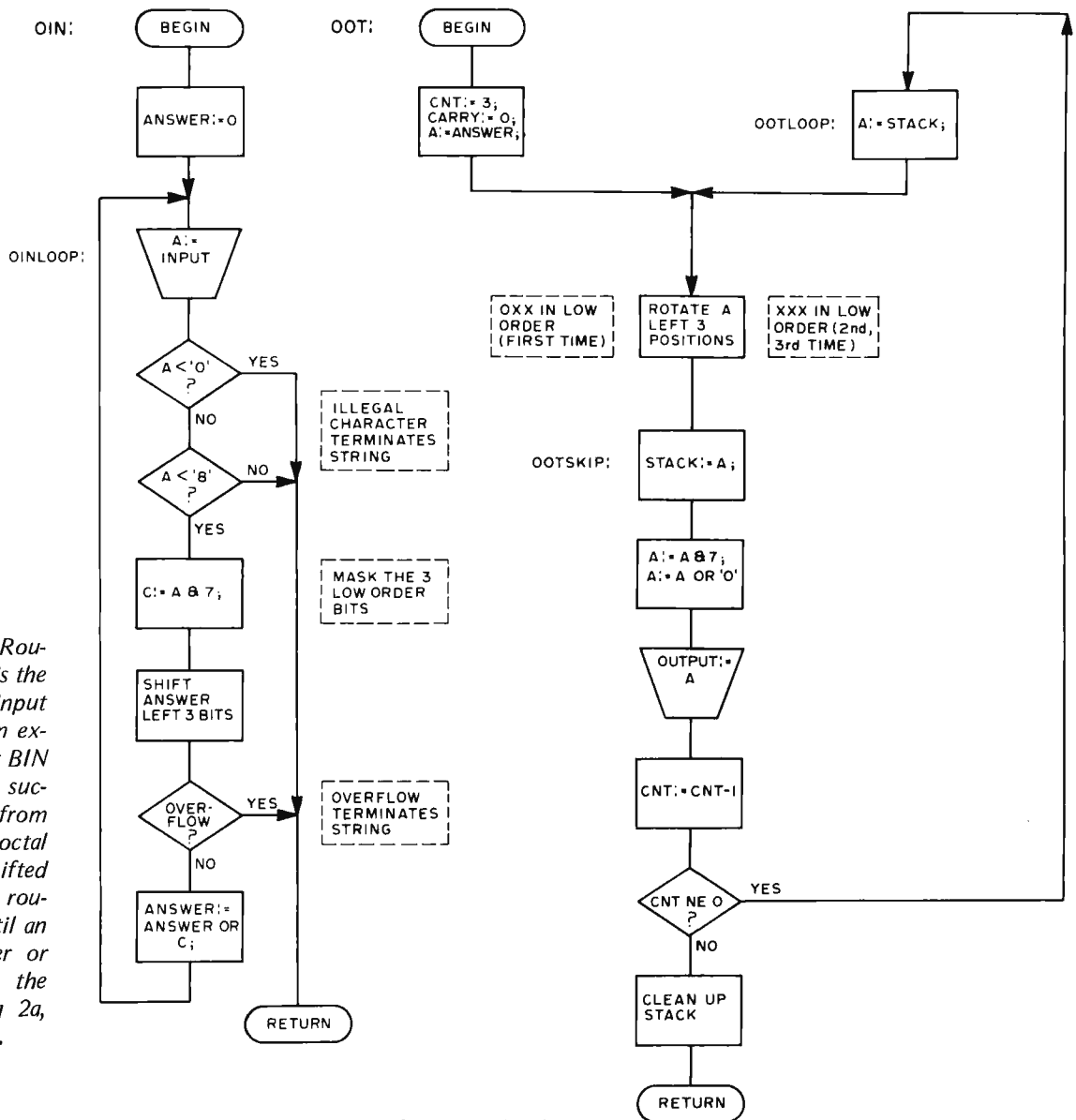


Figure 2a: The OIN Routine Flowchart. OIN is the octal version of an input routine; its logic is an extension of the simpler BIN routine. OIN treats successive characters from ASCII '0' to '7' as octal digits which are shifted into ANSWER. The routine accepts input until an illegal octal character or overflow occurs. In the 8080 code of listing 2a, ANSWER is register B.

handled are shown below, after the value is loaded into the A register and carry reset to zero.

```

Carry A Register
  0 bb bbb bbb
RAL : b bb bbb bb0
RAL : b bb bbb b0b
RAL : b bb bbb 0bb
  
```

At this point carry and the A register are saved and a character put out. Processing then continues at the first rotate, after the saved information is restored. The A register plus carry, in effect, operates as if the machine has a 9 bit register.

Hexadecimal

Input and output of hexadecimals employs logic similar to the preceding routines, with the following differences:

1. ASCII '0' through '9' and 'A' through 'F' are legal numbers.

Figure 2b: The OOT Routine Flowchart. OOT is the octal version of an output routine for character string conversion. Its logic is complicated by the fact that 8 bits is not an even multiple of 3 bits. Thus there is a special case which treats the carry flag as a ninth bit for the first (high order) output digit. Then the basic logic consists of shifting 3 places, extracting 3 bits and creating an ASCII character from '0' to '7'. This routine in its 8080 implementation uses the stack as a temporary data area, as shown in listing 2b.

	Rel. Addr.	Code	Label	Op.	Operand	Commentary	
	0000	21 00 00	XIN:	LXI	H, 0	ANSWER := 0;	
Note 1	0003	CD xx xx	XINLOOP:	CALL	GET	A := INPUT [character];	
	0006	FE 30		CPI	'0'	is A LT '0'?	
	0008	D8		RC		if so then return;	
	0009	FE 3A		CPI	':'	is A LT ':' [numerics]?	
Note 2	000B	DA xx xx		JC	XINSHIFT	if so then go shift it in;	
	000E	FE 41		CPI	'A'	is A LT 'A'?	
	0010	D8		RC		if so then return;	
	0011	FE 47		CPI	'G'	is A LT 'G' [alphabetic A to F]?	
	0013	D0		RNC		if not then return;	
	0014	C6 09		ADI	9	A := A + 9 [convert to hexadecimal];	
	0016	E6 0F	XINSHIFT:	ANI	15	A := A & b'00001111' [mask low order];	
	0018	29		DAD	H	shift ANSWER register pair	
	0019	D8		RC		left four bit	
	001A	29		DAD	H	positions using	
	001B	D8		RC		double byte addition	
	001C	29		DAD	H	and test each	
	001D	D8		RC		operation for	
	001E	29		DAD	H	an overflow error	
	001F	D8		RC		return condition;	
	0020	B5		ORA	L	A := A OR L [add new code to lower order];	
	0021	6F		MOV	L, A	restore low order to ANSWER;	
	Note 3	0022	C3 xx xx		JMP	XINLOOP	reiterate for next nybble;

Note 1: address of GET should replace "xx xx".

Note 2: "xx xx" should be the address of XINSHIFT.

Note 3: "xx xx" should be the address of XINLOOP.

	Rel. Addr.	Code	Label	Op.	Operand	Commentary
	0000	0E 04	XOT:	MVI	C, 4	CNT := 4;
	0002	AF	XOTLOOP:	XRA	A	CARRY := 0; A := 0 [clear A, CARRY];
	0003	29		DAD	H	Shift four bits of ANSWER
	0004	17		RAL		into A using
	0005	29		DAD	H	two byte addition
	0006	17		RAL		with CARRY
	0007	29		DAD	H	receiving each
	0008	17		RAL		bit from the high
	0009	29		DAD	H	order due to overflow;
	000A	17		RAL		
	000B	FE 0A		CPI	10	is A LT 10 [test for numeric digit]?
Note 1	000D	DA xx xx		JC	XOTASCII	if so then go form ASCII character code;
	0010	C6 07		ADI	7	if not then A := A + 7 [adjust to alpha];
Note 2	0012	C6 30	XOTASCII	ADI	'0'	A := A + '0' [convert to ASCII code];
	0014	CD xx xx		CALL	PUT	OUTPUT := A;
Note 3	0017	0D		DCR	C	CNT := CNT - 1;
	0018	C2 xx xx		JNZ	XOTLOOP	if CNT NE 0 then repeat;
	001B	C9		RET		else return to caller;

Note 1: "xx xx" should be the address of XOTASCII.

Note 2: address of PUT should replace "xx xx".

Note 3: "xx xx" should be the address of XOTLOOP.

Listing 3a: The XIN Routine Specified for an 8080. This routine accepts an input string of ASCII hexadecimal characters and collects the results as a 16 bit number in ANSWER (CPU register pair H and L).

Listing 3b: The XOT Routine Specified for an 8080. This routine converts the contents of ANSWER (CPU register pair H and L) into a 4 digit ASCII string of hexadecimal characters, transferring the results to the output device with PUT.

- The shift left is now four bits.
- On input if the character is ASCII 'A' through 'F', then a binary 9 is added to generate a correct value in the low order 4 bits which are then masked as usual:
ASCII A = hexadecimal 41 + 09 = 4A (and) 0F = 0A
- On output if a 4 bit binary value is greater than a 9, then a 7 is added to the value. The conversion is then

completed by adding hexadecimal 30, the ASCII code for 0 (zero).
For example:

$$\begin{aligned}
 00 + 30 &= 30 \text{ or ASCII '0'} \\
 09 + 30 &= 39 \text{ or ASCII '9'} \\
 0A + 07 &= 11 + 30 = 41 \text{ or ASCII 'A'} \\
 0F + 07 &= 16 + 30 = 46 \text{ or ASCII 'F'}
 \end{aligned}$$

The software of 16 bit unsigned hexadecimal input and output conversion is

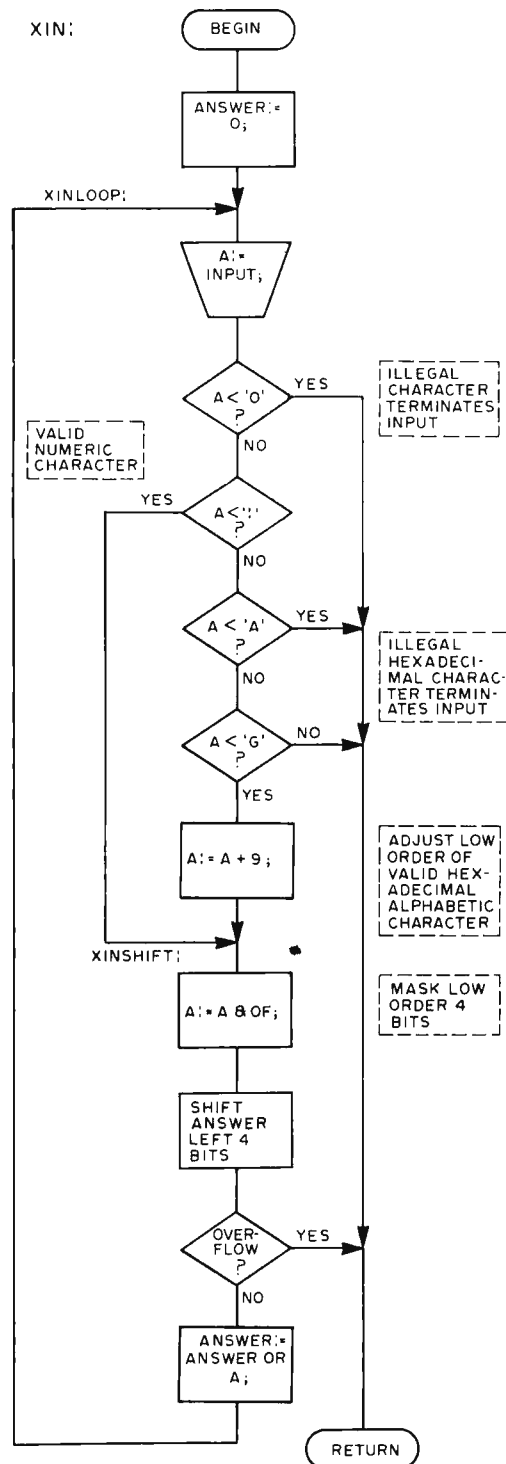


Figure 3a: The XIN Routine Flowchart. XIN is the hexadecimal version of the input algorithm, with the extension of accepting 16 bit values. The XIN routine tests for the validity of the hexadecimal digits, then converts the low order bits to a binary version of the digit. This value is then shifted into the ANSWER being prepared. In the 8080 version of this routine (listing 3a), ANSWER becomes the HL index register pair, and the 8080's double precision addition operation is utilized. Conversion terminates with an invalid character or when overflow occurs.

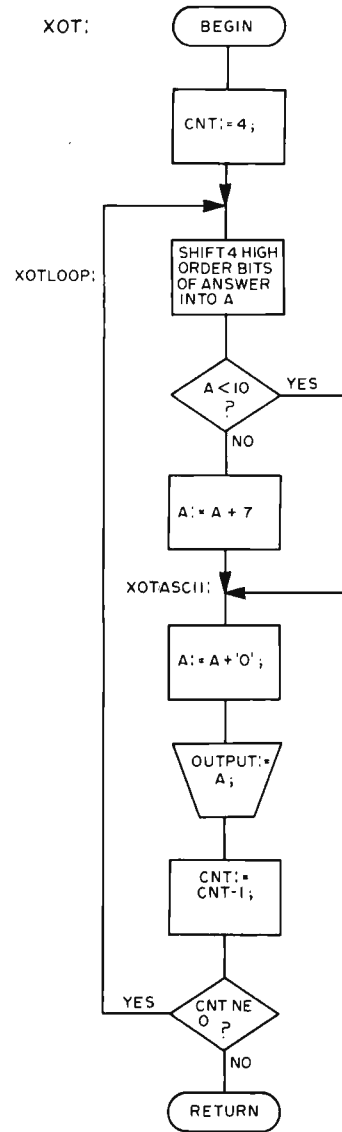


Figure 3b: The XOT Routine Flowchart. XOT converts a 16 bit quantity in ANSWER into a series of ASCII hexadecimal characters, starting with the high order digit. The logic shifts out 4 bits at a time into the accumulator, adjusts the value if alphabetic codes are present then prints the ASCII version obtained by adding '0' to the value. Four digits are created and printed prior to return.

shown in listings 3a and 3b as implemented for an 8080 computer. The flow charts of figures 3a and 3b outline the logic for adaptation to other computers. When this was implemented, an arbitrary choice was made to use 16 bit values instead of 8 bit. This can lead to some inconvenience on an 8 bit microprocessor without 16 bit operations. However, certain instructions were

available on the 8080 to perform double register operations (two 8 bit registers treated as a single unit). The 8080 DAD instruction performs 16 bit addition on the (H,L) register pair using another specified register pair. When the 8080 instruction DAD H is encountered, the value in (H,L) is doubled, thus in effect shifting that pair of registers left one bit. For input shifting, it

	Rel. Addr.	Code	Label	Op.	Operand	Commentary
	0000	21 00 00	DIN:	LXI	H, 0	ANSWER := 0;
	0003	01 00 00		LXI	B, 0	SIGN := 0; NSIGN := 0;
Note 1	0006	CD xx xx		CALL	GET	A := INPUT [character];
	0009	FE 2B		CPI	'+'	is A = '+'?
Note 2	000B	CA xx xx		JZ	DINSIGN	if so then go save sign
	000E	FE 2D		CPI	'-'	is A = '-'?
Note 3	0010	C2 xx xx		JNZ	DINNUMB	if not then go to numeric tests;
	0013	0D		DCR	C	SIGN := -1;
	0014	41	DINSIGN:	MOV	B, C	NSIGN := SIGN;
Note 1	0015	CD xx xx		CALL	GET	A := INPUT [character];
	0018	FE 30	DINNUMB:	CPI	'0'	is A LT '0'?
	001A	D8		RC		if so then return [not numeric];
	001B	FE 3A		CPI	':'	is A LT ':'?
	001D	D0		RNC		if not then return [not numeric];
	001E	E6 0F		ANI	15	A := A & b'00001111' [mask low order];
	0020	4F		MOV	C, A	VALUE := A [save input, low order];
	0021	78		MOV	A, B	A := NSIGN;
	0022	06 09		MVI	B, 9	CNT := 9;
	0024	54		MOV	D, H	MULTPLR := ANSWER [high order];
	0025	5D		MOV	E, L	MULTPLR := ANSWER [low order];
	0026	17		RAL		is SIGN positive? [uses copy in A];
Note 4	0027	D2 xx xx		JNC	DINMPYP	if not then go to positive multiply;
	002A	AF		XRA	A	A := 0; CARRY := 0;
	002B	91		SUB	C	A := A - VALUE [negate VALUE];
	002C	4F		MOV	C, A	C := A [save negated value];
	002D	7C		MOV	A, H	A := ANSWER [high order];
	002E	17		RAL		is ANSWER negative?
Note 5	002F	DA xx xx		JC	DINMPYN	if so then proceed [not first time];
	0032	06		MVI	B, 0	CNT := 0 [so sign extension at DINEGATE works];
Note 6	0033	C3 xx xx		JMP	DINEGATE	first time add VALUE to ANSWER [initialized to zero];
	0036	19	DINMPYN:	DAD	D	ANSWER := ANSWER + MULTPLR [both are negative];
	0037	D0		RNC		if no overflow then return;
	0038	05		DCR	B	CNT := CNT - 1;
Note 5	0039	C2 xx xx		JNZ	DINMPYN	if CNT NE 0 then reiterate;
	003C	05	DINEGATE	DCR	B	CNT := CNT - 1 [now CNT := -1];
	003D	09		DAD	B	ANSWER := ANSWER + (- VALUE) [16 bit ops];
Note 2	003E	C3 xx xx		JMP	DINSIGN	reiterate with next numeric character;
	0041	19	DINMPYP	DAD	D	ANSWER := ANSWER + MULTPLR;
	0042	D8		RC		if CARRY := 1 then return [overflow];
	0043	05		DCR	B	CNT := CNT - 1;
Note 4	0044	C2 xx xx		JNZ	DINMPYP	if CNT NE 0 then reiterate;
	0047	09		DAD	B	ANSWER := ANSWER + VALUE;
Note 2	0046	C3 xx xx		JMP	DINSIGN	reiterate with next numeric character;

Note 1: address of GET should replace "xx xx".

Note 2: "xx xx" should be the address of DINSIGN.

Note 3: "xx xx" should be the address of DINNUMB.

Note 4: "xx xx" should be the address of DINMPYP.

Note 5: "xx xx" should be the address of DINMPYN.

Note 6: "xx xx" should be the address of DINEGATE.

was a simple matter of performing four of these and then using an OR to the low order 8 bits from the value generated as a result of step 3 above. Output necessitated four groups of DAD H and RAL operations to shift a bit into carry, then rotate it into the A register before step 4 was performed (see listing 3b).

Decimal Integer Conversions

Purely out of habit, I choose to use leading minus sign to indicate negative, ASCII '-', with '+' or nothing to indicate positive integers. Again I felt that a 16 bit routine would be more useful than an 8 bit one, allowing two's complement binary

values for 32767 to -32768 instead of 127 to -128 (see note 2).

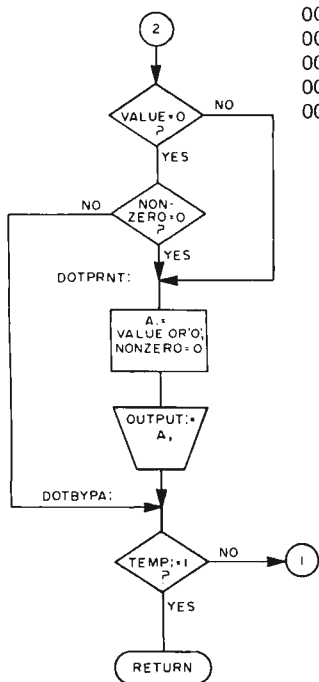
Input was fairly straightforward, as shown by listing 4a and figure 4a. If the first character read is a '-', set the minus flag. Then for all numbers read, if the minus flag is set, the value is negated. The developing answer is multiplied by 10 and the new value read added to it. The implementation shown performs multiplication by repeated addition for simplicity, although a hardware multiply instruction would certainly improve performance if it were available.

Decimal output, unfortunately, could not

Listing 4a: The DIN Routine Specified for an 8080. This routine converts an ASCII decimal string of the form 'SXXXXX' into a signed 16 bit quantity in ANSWER (the CPU's H and L register pair). The 'S' can be either '+', '-' or a null string (''); the 'X' can be a decimal digit '0' to '9' or a null string. (Thus a successful conversion can involve from 1 to 6 characters.) Conversion is terminated by an overflow or an invalid character.

Text continued on page 60

	Rel. Addr.	Code	Label	Op.	Operand	Commentary
Note 1	0000	11 xx xx	DOT:	LXI	D, TENSTABL	POINTER := addr (TENSTABL);
	0003	D5		PUSH	D	STACK := POINTER;
	0004	E0 01		MVI	C, 1	NONZERO := 1;
	0006	7C		MOV	A, H	A := ANSWER;
	0007	17		RAL		is ANSWER negative?
Note 2	0008	D2 xx xx		JNC	DOTPOSIT	if not then go to positive routine;
	000B	7D		MOV	A, L	} ANSWER := -ANSWER - 1 [low order];
	000C	2F		CMA		
	000D	6F		MOV	L, A	
	000E	7C		MOV	A, H	} ANSWER := -ANSWER - 1 [high order];
000F	2F		CMA			
	0010	67		MOV	H, A	
	0011	23		INX	H	ANSWER := (-ANSWER - 1) + 1;
	0012	3E 2D		MVI	A, '-'	A := '-' [ASCII leading minus];
Note 3	0014	CD xx xx		CALL	PUT	OUTPUT := A [display minus sign];
	0017	E3	DOTPOSIT:	XTHL		exchange POINTER and ANSWER;
	0018	5E		MOV	E, M	TEMP := M(POINTER) [low order];
	0019	23		INX	H	POINTER := POINTER + 1;
	001A	56		MOV	D, M	TEMP := M(POINTER) [high order];
	001B	23		INX	H	POINTER := POINTER + 1;
	001C	E3		XTHL		exchange ANSWER and POINTER;
	001D	06 00		MVI	B, 0	VALUE := 0;
	001F	7D	DOTDIVID:	MOV	A, L	} ANSWER := ANSWER - TEMP [low order];
	0020	93		SUB	E	
0021	7C		MOV	A, H		
	0022	7C		MOV	A, H	} ANSWER := ANSWER - TEMP [high order];
0023	9A		SBB	D		
0024	67		MOV	H, A		
Note 4	0025	FA xx xx		JM	DOTOUT	if ANSWER LT 0 then go put character;
	0028	04		INR	B	VALUE := VALUE + 1;
Note 5	0029	C3 xx xx		JMP	DOTDIVID	reiterate, counting in VALUE;
	002C	19	DOTOUT:	DAD	D	ANSWER := ANSWER + TEMP;
	002D	AF		XRA	A	A := 0; CARRY := 0;
	002E	B0		ORA	B	is VALUE = 0?
Note 6	002F	C2 xx xx		JNZ	DOTPRNT	if not then go print it;
	0032	B1		ORA	C	is NONZERO = 0 [leading zero test];
Note 7	0033	C2 xx xx		JNZ	DOTBYPA	if not then bypass leading zero print;
	0036	F6 30	DOTPRNT:	ORI	'0'	A := A OR '0' [convert VALUE to ASCII];
	0038	0E 00		MVI	C, 0	NONZERO := 0 [reset zero flag];
Note 3	003A	CD xx xx		CALL	PUT	OUTPUT := A [display ASCII digit];
	003D	7B	DOTBYPA:	MOV	A, E	A := TEMP [low order];
	003E	FE 01		CPI	1	is TEMP = 1 [low order]?
Note 2	0040	C2 xx xx		JNZ	DOTPOSIT	if not then reiterate;
	0043	D1		POP	D	else flush stack
	0044	C9		RET		and return;
	0045	10 27	TENSTABL:	DW	10000	} define constants for the decimal division routine (note: low order at low memory address for 8080);
	0047	E8 03		DW	1000	
	0049	64 00		DW	100	
	004B	0A 00		DW	10	
	004D	01 00		DW	1	



TENSTABLE:	LOCATION	VALUE (DECIMAL)	(HEX)
	0	10 000	2710
	2	1 000	03E8
	4	100	0064
	6	10	000A
	8	1	0001

NOTE:
INTEL FORMAT IN LISTING 4b REQUIRES LOW ORDER HEXADEDECIMAL BYTE AT FIRST (LOW) ADDRESS.

Listing 4b: The DOT Routine Specified for an 8080. This routine converts the signed two's complement number in ANSWER (register pair H and L) into an ASCII signed decimal string with leading zero suppression. The result is sent to the output device during the conversion.

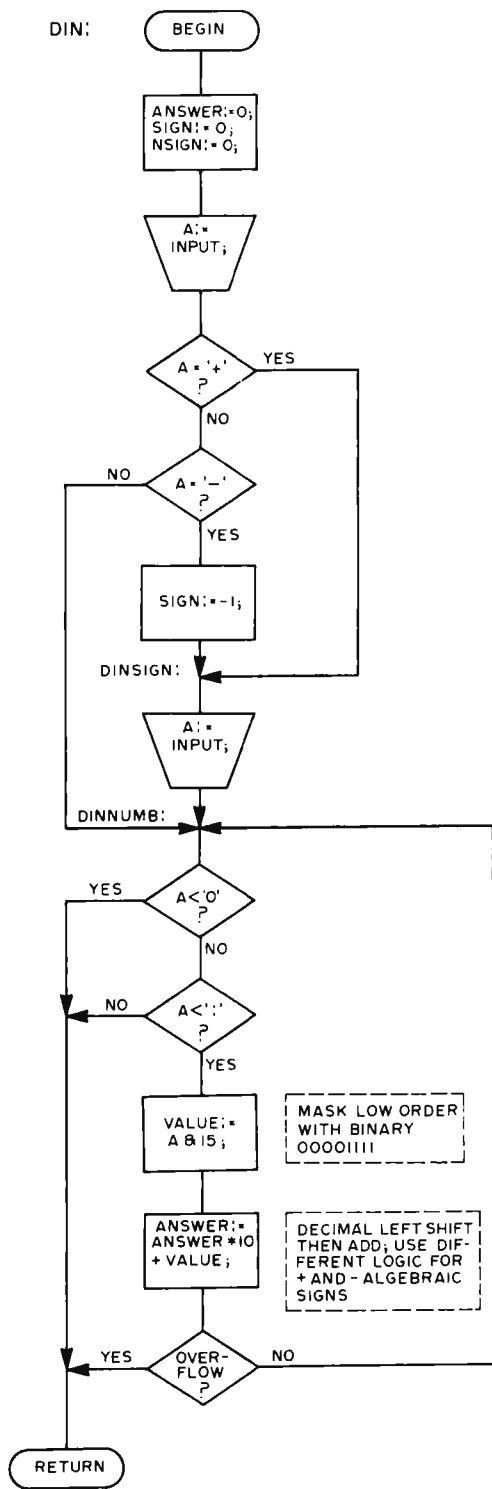


Figure 4a: The DIN Routine Flowchart. With decimal arithmetic values, the shifting involved is no longer an integer multiple of one bit. The DIN routine uses the decimal version of binary shifting: multiplying the value by the base of the number system, then adding in the new low order value. DIN also includes sign decoding logic for the ASCII '+' and '-' characters. In the 8080 version of DIN, the result is a signed two's complement number in ANSWER, a 16 bit quantity in the HL index register pair.

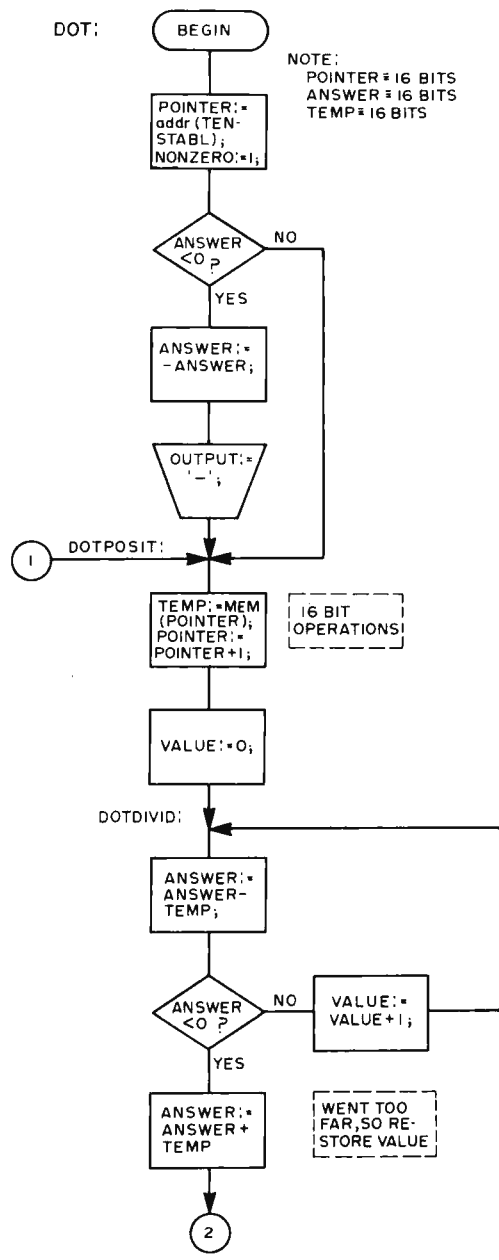


Figure 4b: The DOT Routine Flowchart. The decimal equivalent of the shifting used in the base 2ⁿ output routines is division by the base of 10. This routine also includes leading zero suppression and logic to print a sign digit. Division is performed by repeated subtraction using values stored in TENS-STABL. In the 8080 version of listing 4b, the ANSWER to be output is a 16 bit signed two's complement number in the HL index register pair.

be made quite so simple, primarily because there really exists no decimal (base 10) left shift. This left two alternatives, either repetitively divide by 10 stacking the remainders, or perform a succession of pseudo divisions by subtracting appropriate constants. The latter technique was chosen due to the complexity of multi register division. The code of such a routine for an 8080 is shown in listing 4b, and the corresponding flow chart is figure 4b.

The output routine checks the initial value to determine if it is negative, and if so, output the ASCII character '-'. If the value is negative, it is negated (making it positive) so that positive and negative numbers can be handled the same way. A table containing powers of 10 (10,000; 1,000; 100; 10; 1) was then utilized to perform pseudo divisions by successive subtraction. This is outlined in the flow diagram in figure 4b. For the 8080 implementation, there is no 16 bit subtraction, hence a multiple precision subtract operation is coded.

The handling of signed numbers is optional, as well as the zero suppression. They were included because it is easier to take them out than to try to divine where they go and how to do it.

Many microprocessors have an instruction which maintains decimal numbers. Given the 8 bit quantity hexadecimal 79, assume a hexadecimal 02 is added to it, giving the hexadecimal value 7b. This instruction then can be used to adjust this result back to two decimal digits, 4 bits each. The value then would appear as hexadecimal 81, which can be thought of as adding the decimal numbers $79 + 2$, giving 81. If computations are to be made in this packed decimal mode, then the

Assumptions

The assumptions for the procedures of this article are:

1. An input and output subroutine exists (GET and PUT) which preserve CPU registers except A.
2. The conversion process is itself a subroutine.
3. The conversion process need not save any registers.
4. Validating characters is done (though not necessary).
5. Overflow checking is done (again not necessary and in some instances not desirable).
6. All values are treated as unsigned integers (except the decimal routines).
7. Non significant leading zeros are not required on input.
8. Leading zeros are printed on output (except for decimal).

hexadecimal routine presented could be used to input and output these values.

In conclusion, these routines are not presented as the final answer in number conversions. In order to implement any or all of these routines on your own personal computer, the flow diagrams may be more useful than the sample 8080 implementation. That implementation is targeted for Intel's 8080 microprocessor, one of the most widely used hobby computers at the time of this writing. All the routines made full use of certain special features and strange quirks of the 8080 microprocessor. Whatever your particular machine, the time spent in understanding these routines should save you a few headaches in your next program. ■

NOTES

Note 1:

During a left shift, as the high order bit leaves the register, it enters the carry bit and the vacated low order bit receives a zero.

For example: Before : Carry=0 A=1001 0111
After : Carry=1 A=0010 1110

During a rotate left, as a bit leaves the high order bit position, that value is shifted into the vacated low order bit position. On the Intel 8080, two types of rotate are available:

1. RRL : rotate accumulator copying swapped bit to carry.

before: Carry=0 A=1001 0111
after: Carry=1 A=0010 1111

2. RAL : rotate accumulator thru carry

before: Carry=0 A=1001 0111
after: Carry=1 A=0010 1110

On computers with a rotate through the carry bit, new bits can be shifted into the accumulator while old bits are shifted out.

Note 2:

Two's complement arithmetic uses the high order bit of a value to indicate sign; 1 is negative and 0 is positive. A negative value is formed by complementing all bits of the value (1 to 0 and 0 to 1) and adding one. Thus, the largest positive value for a 16 bit quantity is a hexadecimal 7FFF, and the smallest negative value is a hexadecimal 8000, or decimal 32767 to -32768. The 8 bit values are 7F to 80 or 127 to -128.

For example: given the value 1, create the value -1.

0000 0001 = 1	Start with 1
1111 1110	Complement all 16 bits
+1	Add 1
1111 1111 = -1	Giving the value -1.

Software Bug of the Month 4

Even when a program has been exhaustively tested, bugs can still occur. This month's tale concerns an overconfident programmer who wrote a program, tested it extensively, and then bragged about it, to his ultimate regret.

The program was supposed to test whether the number N was prime. If N was prime, it was supposed to set K = 1; otherwise, it would set K = 0. The idea was to test whether N is a multiple of 2, then 3, then 4, and so on. A trick was used, in that if N is not prime — that is, $N = I * J$ — then either I or J must be less than, or equal to, the square root of N. Therefore we only need to test multiples of numbers up to the square root of N.

The FORTRAN version of the program was as follows:

```

SUBROUTINE PRIME(N, K)
  K = 1
  I = 2
1  IF (MOD(N, I).NE.0) GO TO 2
  K = 0
  RETURN
2  I = I + 1
  IF (I * I .LE. N) GO TO 1
  RETURN
END

```

Not satisfied with his ability to write a program that works the first time, our programmer tried out this one on a wide variety of test cases. All checked out perfectly. Great was his despair, then, when the programmer down the hall said to him one day, "Hey, you know that bug we've been working on for about a month? You know what we just traced it to? Your little old prime subroutine!" (Please don't ask what a prime number testing subroutine was doing in a larger system.)

What was the bug?

[NOTE: The MOD(N,I) function returns the integer remainder of the division N/I.]

Answer in Next Month's BYTE ■

SOLUTION TO BUG OF THE MONTH 3

What happened first was that the recognizer for a digit was called; it found the *first* digit in the unsigned integer, and quit at that point. Thus the rest of the unsigned integer was never found.

The programmer tried to fix this by rearranging the BNF rule as

```
<unsigned integer> ::= <unsigned integer> <digit> / <digit>
```

(that is, putting the second case first). Unfortunately, this time, the first thing the recognizer did was to call itself; this made it call itself again, and so on, producing an endless loop. So the BNF rule was rearranged again:

```
<unsigned integer> ::= <digit> <unsigned integer> / <digit>
```

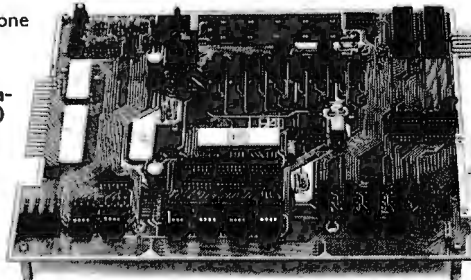
(that is, rearranging the order in the first case). This gave him his second endless loop. His last bug really should have been thought of first: he was working in PL/I, which allows subroutines to be recursive — but they have to be declared RECURSIVE, and this he had forgotten. ■

W Douglas Maurer
University Library Room 634
George Washington University
Washington DC 20052

If you want a microcomputer with all of these standard features...

- 8080 MPU (The one with growing software support)
- 1024 Byte ROM (With maximum capacity of 4K Bytes)
- 1024 Byte RAM (With maximum capacity of 2K Bytes)
- TTY Serial I/O
- EIA Serial I/O
- 3 parallel I/O's
- ASCII/Baudot terminal compatibility with TTY machines or video units

- Monitor having load, dump, display, insert and go functions



- Complete with card connectors
- Comprehensive User's Manual, plus Intel 8080 User's Manual
- Completely factory assembled and tested — not a kit
- Optional accessories: Keyboard/video display, audio cassette modem

interface, power supply, ROM programmer and attractive cabinetry... plus more options to follow. **The HAL MCEM-8080. \$375**

...then let us send you our card.

HAL Communications Corp. has been a leader in digital communications for over half a decade. The MCEM-8080 microcomputer shows just how far this leadership has taken us... and how far it can take you in your applications. That's why we'd like to send you our card — one PC board that we feel is the best-valued, most complete

microcomputer you can buy. For details on the MCEM-8080, write today. We'll also include comprehensive information on the HAL DS-3000 KSR microprocessor-based terminal, the terminal that gives you multi-code compatibility, flexibility for future changes, editing, and a convenient, large video display format.



HAL Communications Corp.
Box 365, 807 E. Green Street, Urbana, Illinois 61801
Telephone (217) 367-7373

The Circuit for Z-80s

Dr Robert Suding
Research Director, The Digital Group Inc
PO Box 6528
Denver CO 80206

The microprocessor integrated circuit is the real engine for your system. Now you can replace that old engine with a real power house, the new Z-80 (the Z-80 was described in Burt Hashizume's Microprocessor Update on page 34 of August 1976 BYTE). After initially reading about this integrated circuit in early '76, I just had to get one to see how many of the blurbs were true (I give sales advertisements a 1% credibility on the first pass).

Aside from a few typos, promised support chips that didn't show, and several mistakes in the software documentation, it was fabulous. The software flexibility added by this chip was a great addition to the 8080/6502/6800 Digital Group stable. The relative branch was very helpful for machine language programming, and the ability to test, set, and clear individual bits in a byte has opened a new world of control applications. I saw a 20% savings in memory requirements even though I was still new to much of the Z-80's special software.

The Z-80's hardware made good sense too. Getting rid of the 18 MHz crystal requirement of the 8224/8080 system and using a 2.5 MHz crystal with a simple single phase TTL clock made me happy. The interrupt and DMA system has some neat features. Sure gonna be hard to justify using the old 8080/6502 or 6800 CPU boards any more, thought I, as I set out to design the circuit for Z-80s.

The circuit for Z-80s presented in this article is the actual wiring used in the Digital Group's Z-80 processor card. Not too unbelievably, we would just love to sell you the whole system. The circuit is being published in complete detail for your information, whether you choose to purchase it as part of your system, or use it as a starting point for your own custom design. The systems ap-

proach to microprocessors which I described in the June 1976 BYTE [*page 32*] is reflected in the design of this central processor circuit.

This Z-80 circuit is shown in figures 1 and 2. In figure 1 you'll find the central processor integrated circuit (IC43, a Z-80 made by Zilog or second source Mostek), and miscellaneous drivers, decoders and gates. In figure 2 you'll find the wiring of 2 K bytes of programmable memory along with a 256 byte 1702A erasable read only memory which can be used to store the bootstrap programs for your system.

Full direct memory access (DMA) is used in this design. What's DMA to you? Well for one thing, DMA permits hand loading of the memory from a front panel which is completely independent of a particular processor. It permits future processor upgrading by replacing a single board. High speed data devices, such as some tape, disk, and video systems which may operate too fast for most processors, can directly load memory using DMA. Finally, for the truly gigantic among you, multiple processors can share common memory with the addition of control logic.

Buffering is included on this processor board design to permit driving a full memory system (64 K bytes) and up to 256 IO ports. Miscellaneous logical functions such as power on reset and single stepping are provided.

The EROM bootstrap provides a convenient way to initialize the system at power on, by using a low cost cassette [*page 46, July 1976 BYTE*]. We use an EROM in the design in order to allow customized initialization by sophisticated users able to program their own EROMs. Circuitry to inhibit EROM selection is included in order to permit full use of "0 page" programmable memory for user software.

Two K bytes of programmable random access memory give sufficient storage for a small operating system. The Digital Group Z-80 system includes a cassette which loads this area of programmable memory with a system monitor which permits reading and

When inserting large integrated circuits into sockets, avoid uneven stresses. In extreme cases of uneven insertion pressure, it is possible to crack the case of a 24 or 40 pin integrated circuit, rendering it useless.

writing other cassettes, keyboard entry of data and programs, and TV display of memory data, all 14 registers, indices, and flags (in octal or hexadecimal).

The system used to interface this processor to memory and IO exemplifies the "processor independence" ideal mentioned in my article in the June BYTE. Two sets of 16 address lines are brought out from each Digital Group processor card. The 16 lines labeled "memory address" in figure 1 lead to the memory boards; the 16 lines labeled "port address" in figure 2 go to the IO port selecting card(s). Similarly, memory data to and from the processor is separated, as is the peripheral IO data to and from the processor.

The Z-80 DMA read, write and IO lines are brought to decoding logic to derive your universal control lines, ie: memory read (MRD), memory write (MWR), IO read (IORD), and IO write (IOWR).

The major objective of processor independency is supported by providing this common set of 32 address lines, 32 data lines, and 4 control lines for each processor. It is the responsibility of the processor board to provide the logical derivation of these 68 lines. The complete list of backplane connections for the system includes all 68 logic lines and is summarized in table 1. The rest of the system is interfaced to this common 68 line system. Processor interchange is thus particularly simple: It is achieved by plugging in a different processor card.

Z-80 Processor Circuit

The logic of this Digital Group Z-80 processor circuit may be logically divided into six interrelated sections. They are the processor itself and immediate "house-keeping" logic, run control, DMA, interrupt, buffering, and memory. The processor and immediate housekeeping consists of the Z-80, a 7400 single phase crystal controlled clock generator, and decoders for read, write, memory and IO operations. These are all found in figure 1.

A power on reset function is provided by IC38d, one section of a 4010 CMOS buffer. An external switch is attached to the backplane assembly for a remote "reset and go" operation after power has been applied.

A 7442, IC48, decodes IO states of the processor: memory reading, memory writing, input port reading, and output port writing. Each of these signals occurs at the proper time as determined by the processor.

Run control logic permits single stepping through a program if a front panel readout is provided for viewing the resulting instruction sequencing. In addition, wait states for slow external memory and the EROM access delay are provided. The wait line input of the Z-80 is utilized to control execution. A feature of this Z-80 circuit is the ability to jumper select either "single step" or "step on instruction." The jumpering for "single steps" permits stepping within an instruction cycle in the same manner as the 8080. "Step on instruction" will display only the first byte of each single or multibyte instruction. Normal processor running mode is unaffected by which stepping mode is selected.

Two sections of a 7402, IC28a and IC28b, are used as a run latch. When the step switch is activated, the run latch is reset, and the one shot (74123, IC37b) fires a 50 ms pulse to debounce the switch. The resultant pulse is held in a 7474 latch section, IC29a, for a very short time until synchronized by the Z-80 and acknowledged through the second oneshot section of IC37. The 7402 NOR gate IC28c passes either the continuous run or the step pulse depending on the mode selected. IC28d will then drop the ready line if either no run command exists (continuous or step), or the "wait" command line goes high. If no "single step" operation is to be used, pin 43 of the backplane is tied to +5 V externally.

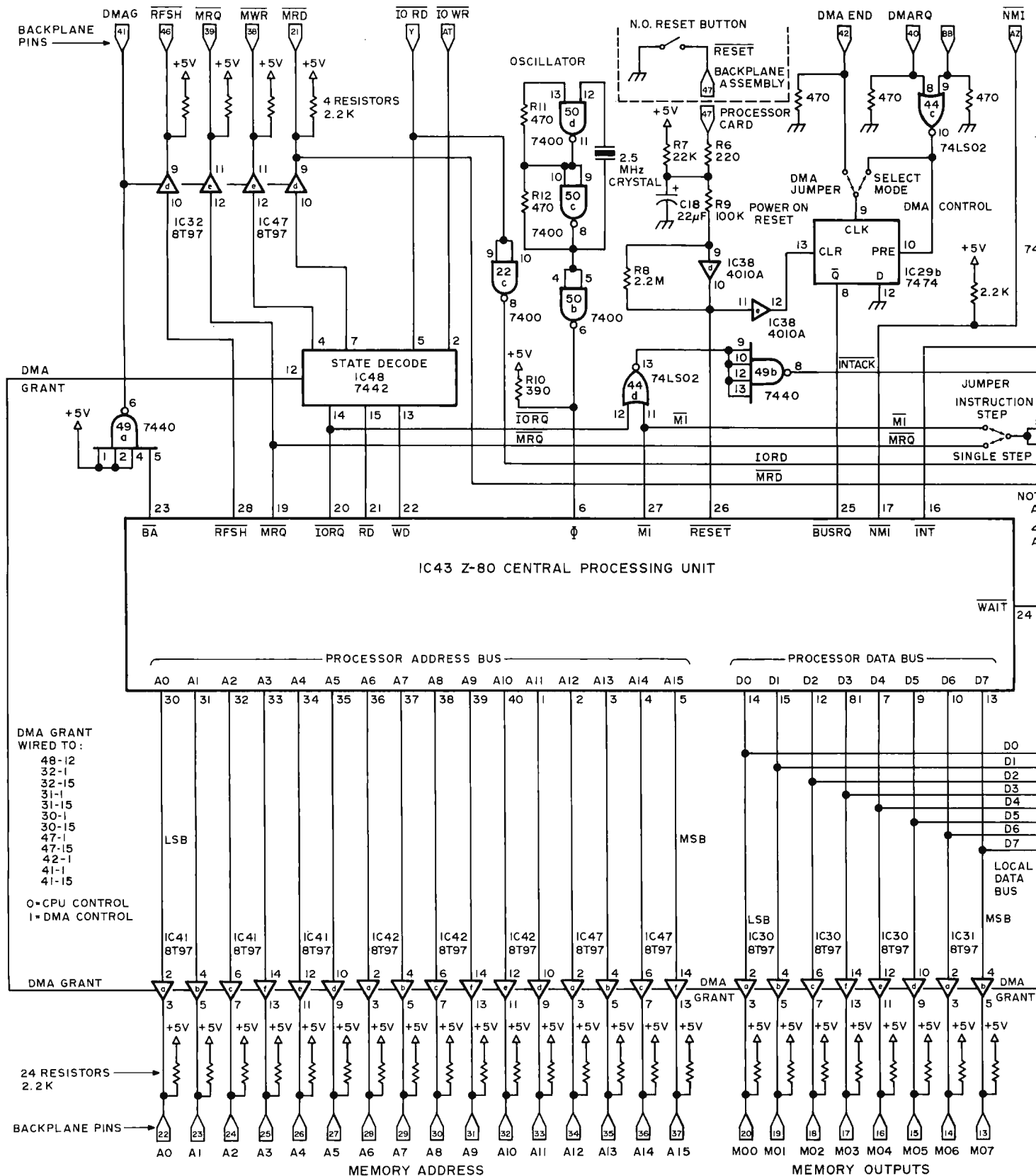
Direct Memory Access

The Z-80 has built-in features for direct memory access. The DMA logic supporting the processor consists of sections of IC44, IC29 and IC49. DMA is designed as an external request for control of memory and the granting of this request as soon as the processor can safely suspend its operations without losing current data. A DMA request is entered whenever either pin 8 or 9 of IC44c goes high. This will set a latch, IC29b, bringing down the Z-80's bus request line.

Contrary to some grapevine rumors, you can't simply unplug your 8080 integrated circuit and plug in a Z-80. A glance at figure 1 and comparison of IC43's Z-80 pinouts with an 8080 specification will shoot that rumor down. Once you have a Z-80 wired, however, the instruction set is a superset of the 8080 instruction set which provides a better general purpose processing architecture.

Text continued on page 68

Figure 1: The central processor of the Z-80 circuit. See also figure 2 for the balance of the logic found in the Digital Group Z-80 central processor card. This figure contains the processor integrated circuit, IC43, and ancillary logic of the system clock, buffers, run control, interrupts and direct memory access control. A summary of back plane connections is found in table 1 accompanying



this article. The complete list of power connections for both figures 1 and 2 is found in table 2. This schematic was redrawn to fit the constraints of the magazine page. A complete schematic in its original form, drawn on one page, is included with the documentation of the Digital Group Z-80 central processor kit.

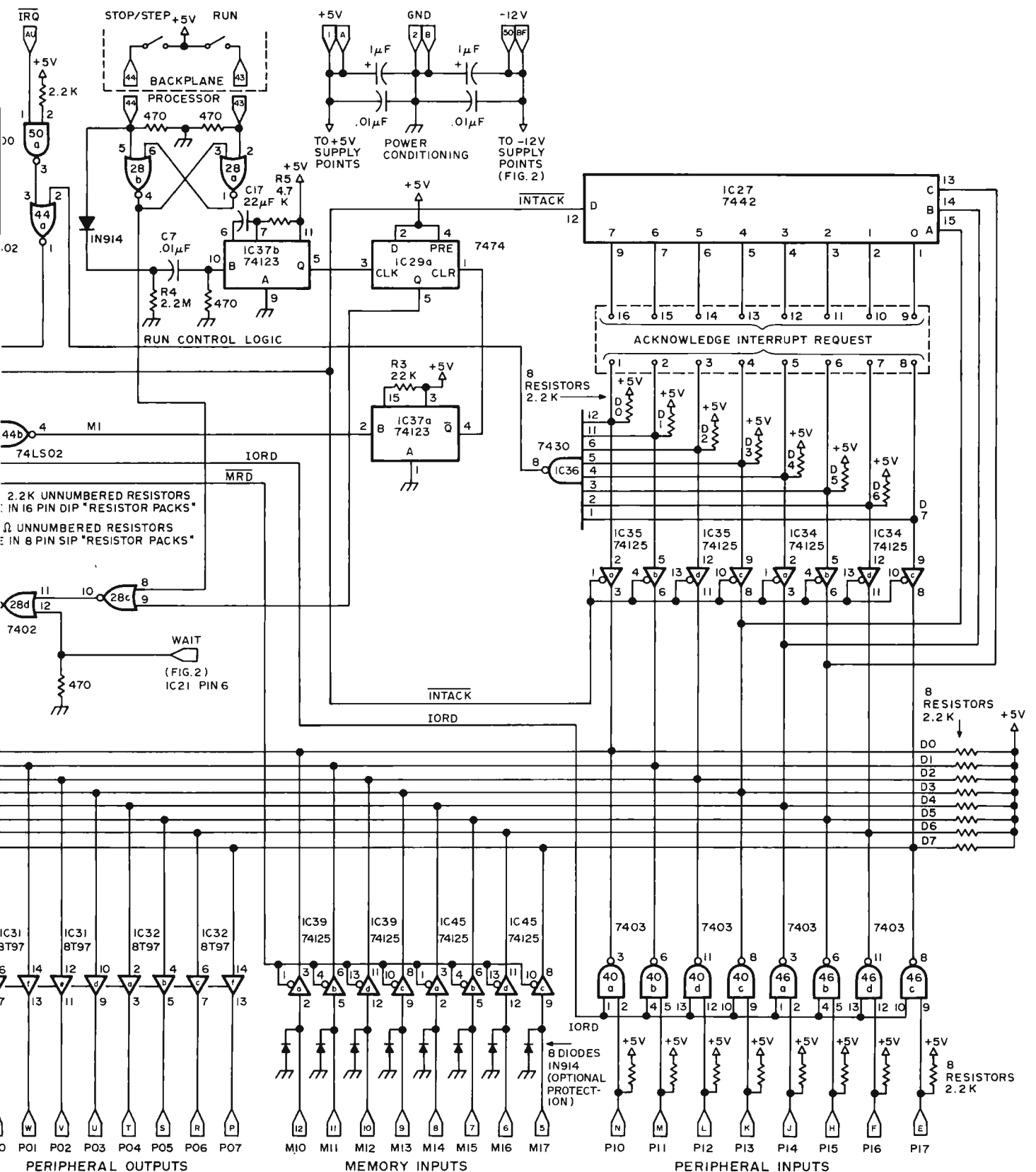


Table 1: A Generalized Processor Independent Bus Structure. This table lists connector pin identification, signal name, DMA access properties, primary signal direction relative to the processor card, and description. This is the bus definition used in the Digital Group systems.

Pin	Name	DMA G ?	In or Out?	Description	Pin	Name	DMA G ?	In or Out?	Description	
1	—	—	—	+5 V power bus	A	—	—	—	+5 V power bus	
2	—	—	—	System ground bus	B	—	—	—	System ground bus	
3	—	—	—	Spare voltage bus	C	—	—	—	Spare voltage bus	
4	—	—	—	−5 V power bus (not used by Z-80)	D	—	—	—	−5 V power bus (not used by Z-80)	
5	M17		IN	Input data from memory	E	PI7		IN	Input data from peripherals	
6	M16		IN		F	PI6		IN		
7	M15		IN		H	PI5		IN		
8	M14		IN		J	PI4		IN		
9	M13		IN		K	PI3		IN		
10	M12		IN		L	PI2		IN		
11	M11		IN		M	PI1		IN		
12	M10		IN		N	PI0		IN		
13	MO7	G	OUT	Output data to memory	P	PO7	G	OUT	Output data to peripherals	
14	MO6	G	OUT		R	PO6	G	OUT		
15	MO5	G	OUT		S	PO5	G	OUT		
16	MO4	G	OUT		T	PO4	G	OUT		
17	MO3	G	OUT		U	PO3	G	OUT		
18	MO2	G	OUT		V	PO2	G	OUT		
19	MO1	G	OUT		W	PO1	G	OUT		
20	MO0	G	OUT		X	PO0	G	OUT		
21	MRD-	G	OUT	Memory read data strobe	Y	IORD-		OUT	Peripheral read data strobe	
22	A0	G	OUT	Memory address lines	Z	PA0	G	OUT	Peripheral address, low order, identical to A0 through A7 in Z-80 processor.	
23	A1	G	OUT		AA	PA1	G	OUT		
24	A2	G	OUT		AB	PA2	G	OUT		
25	A3	G	OUT		AC	PA3	G	OUT		
26	A4	G	OUT		AD	PA4	G	OUT		
27	A5	G	OUT		AE	PA5	G	OUT		
28	A6	G	OUT		AF	PA6	G	OUT		
29	A7	G	OUT		AH	PA7	G	OUT		
30	A8	G	OUT		AJ	PA8		OUT		Peripheral address, high order, wired to ground (logical 0) in Z-80 processor.
31	A9	G	OUT		AK	PA9		OUT		
32	A10	G	OUT		AL	PA10		OUT		
33	A11	G	OUT		AM	PA11		OUT		
34	A12	G	OUT	AN	PA12		OUT			
35	A13	G	OUT	AP	PA13		OUT	Peripheral write data strobe		
36	A14	G	OUT	AR	PA14		OUT			
37	A15	G	OUT	AS	PA15		OUT	Peripheral write data strobe		
38	MWR-	G	OUT	AT	IOWR-		OUT			
39	RFSH-	G	OUT	Refresh line (Z-80) for dynamic memories	AU	IRO-		IN	Interrupt request line	
40	DMARQ		IN	DMA Request #1	AV	*		OUT	Cassette bootstrap: Data output	
41	DMAG		OUT	DMA Grant	AW	*		OUT	Output port 1 bit 0	
42	DMAEND		IN	DMA end signal	AX	*		IN	Cassette bootstrap: Data input	
43	RUN		IN	Run if logic 1, stop or step if 0	AY	*		IN	Input port 1 bit 0	
44	STEP		IN	Stop if 0 and RUN = 0; single step each 1 pulse.	AZ	NMI-		IN	Non maskable interrupt input	
45	WRQ-		IN	Wait request, from external slow memories	BA	ROMDIS		IN	Bootstrap ROM disable	
46	MRQ-	G	OUT	Memory request	BB	DMARQ		IN	DMA Request #2	
47	RESET-		IN	Reset signal	BC	—	—	—	unused	
48	ROMCE-		OUT	ROM on processor board is enabled; do not decode page 0.	BD	*		OUT	Valid memory address (6800, 6502 systems)	
49	—	—	—	+12 V power bus	BE	—	—	—	+12 V power bus	
50	—	—	—	−12 V power bus	BF	—	—	—	−12 V power bus	

NOTES:

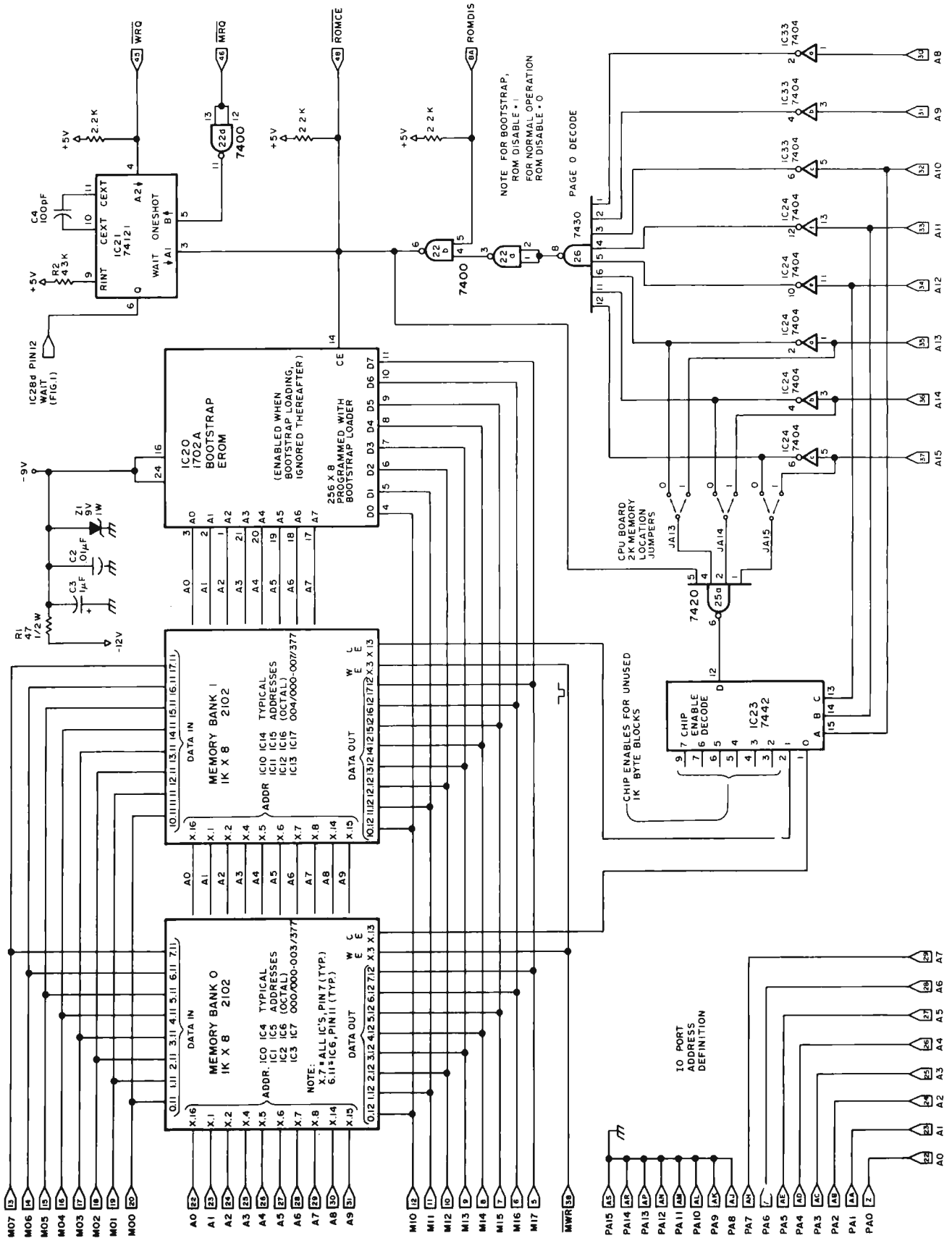
“G” in the “DMA G?” column indicates that the signal is in a high impedance state when the DMAG signal is logical 1. This means that the line in question can be driven by an alternate three state driver during a DMA operation. If the signal is not disabled by DMAG, then this column is blank.

In the “Name” column, if the name is followed by a minus sign as in “MRD-”, then the signal is active low. This is indicated in the logic diagram by a bar over the name in question.

An “*” in the name column indicates a signal which is not defined by the processor circuit of figures 1 and 2 in this article.

“In or Out?” is relative to the central processor card.

Figure 2: The Digital Group Z-80 processor card also includes this memory subsystem. Memory banks 0 and 1 are programmable user memory typically decoded to addresses at split octal locations 000/000 to 007/377, hexadecimal 0000 to 07FF. The programmable jumpers JA13, JA14 and JA15 in this diagram are used to pick the base address for these memory banks, and allow the lower two 1 K blocks of any of the eight 8 K blocks in the Z-80’s 64 K memory address space. The read only memory, IC20, is enabled during bootstrap. During bootstrap, since the ROM addresses overlap the programmable memory addresses at locations 0 to 377 octal (0 to FF hexadecimal) the ROMCE line is used to disable any programmable memory references to page 0. After bootstrapping the programmable memory exclusive of page 0, the ROM becomes invisible to the system when the ROMDIS line is in a high state. (This line should be controlled by a manual switch.)



When the Z-80 is finished with any needed housekeeping, it issues the bus acknowledge signal, granting the request. Further Z-80 operations are suspended and the various buffers, IC31, IC32, IC33, IC41, IC42 and IC47, go to a high impedance state, and the external circuitry making the request is allowed full control over memory using the backplane bus.

DMA request and grant is ended by any of three methods. A reset operation will always end any current DMA operation. A jumper at pin 9 of IC29b allows selecting one of the other two DMA ending operations. If the jumper is connected from pin 9 to pin 10 of IC29b, then the DMA operation will be ended whenever both DMA request lines return low. If the jumper is connected from pin 9 of IC29b to the line labeled DMA end, then a latched DMA operation results. One or more positive going pulses at either DMA Request line will initiate DMA. One or more positive going pulses at the DMA end line will end the DMA.

Interrupts

The Z-80 has extended interrupt processing capabilities, and sufficient hardware is included on the Digital Group Z-80 board to support the three Z-80 interrupt modes. Mode 0 is the same as the 8080A, generally considered as the eight restart instructions which are placed on the data bus upon an interrupt acknowledge signal from the processor. Mode 1 is an automatic interrupt to address 000070. Mode 2 is an extremely powerful vectored interrupt system which is new with the Z-80. A new register, called the I register, is used as a high order portion of the vector address. When an interrupt is encountered and acknowledged, the data placed on the data bus becomes the low order portion of the interrupt vector address. Interrupt processing thus starts at an arbitrary 16 bit address formed from the I register and a variable input. Another interrupt system provided by the Z-80 is called non maskable interrupt (NMI). This interrupt will occur anytime the Z-80's pin 17 is brought low, and is intended for highest priority operations like responding to a power failure before the power supply capacitors bleed down.

IC50, IC44, IC36, IC35, IC34 and IC27 provide the needed interrupt processing interfaces. The 74125s of IC34 and IC35 provide three state buffering for the interrupt address vectoring required by Z-80 interrupt modes 0 and 2. The 7442, IC27,

produces an interrupt honored acknowledgement signal (if required) for use in mode 0. The INT input at the Z-80 pin 16 will be forced low whenever any interrupt input, except NMI, is brought low. Interrupts are interfaced using a 16 pin DIP socket.

Buffering

The Digital Group processor circuits are designed to drive a full complement of memory and IO. In addition, the processors are designed to operate under direct memory access as mentioned previously, and three state buffers permit isolating the processor card from its own (see figure 2) and auxiliary memory.

Sections of 8T97s IC41, IC42 and IC47 provide buffered address outputs from the Z-80 processor with each section capable of each driving 30 standard TTL loads. These drivers handle both memory and IO port addressing. DMA grant is connected to these drivers so that when a DMA is in process, the external device is given full control of the address lines since the processor's drivers are in a high impedance state.

The 8T97 sections used for data output, IC31 and IC32, provide the ability to drive as many as seven Digital Group IO boards (28 ports) without further buffering.

Data input to the processor is placed onto the internal bidirectional bus by two types of circuits. A pair of 74125s provides a three state noninverted buffering of memory input from a backplane bus (pins 5 to 12) which has noninverted data. A pair of open collector 7403s, IC40 and IC46, provide an inverted open collector drive of the same bus, a requirement since the Digital Group peripherals put data onto the backplane in inverted form. Notice, however, that the pin connections of the 7403 are compatible with the 74126 circuit, so if you desire to use this design with noninverting peripherals simply replace the 7403s with 74126s to change the sense of the data on the outputs of the receivers.

Memory (see figure 2) in this Z-80 processor circuit is of two types, EROM and programmable memory. The EROM is a single chip preprogrammed by the Digital Group to simplify system operation of our kits. If you roll your own software, a customized bootstrap EROM could also be used. When power is applied to the system, a "power on reset" function results, which starts the processor running at address 000 000. IC29 and IC25 decode the lowest 256 bytes of memory, resulting in a EROM chip enable condition. The EROM proceeds through its programming to clear the screen, display a message, initialize some program-

One way to test out a newly constructed circuit (not necessarily the best way) is the traditional "smoke test": Turn on power and see if the circuit burns up. A far better method is to do a little thinking and careful inspection first.

mable memory addresses, and control initial cassette reading.

Two K of programmable memory allows an extensive operating system to be entered from cassette. Sixteen 2102s are arranged as two banks of 8 integrated circuits. Which of the two banks selected (if either) is a function of decoding by IC23, IC24 and IC25, as well as the three jumper settings. The 7442 will assign the two banks of 2102s as the bottom 2 K of any one of eight 8 K blocks in memory address space.

The three jumpers permit assigning the processor's 2 K programmable memory to addresses other than the bottom 2 K. When a user wishes to add one or more Digital Group 8 K boards to his or her system, the processor's 2 K may be moved to fall above the highest address of the supplemental 8 K board. Example: A user has two Digital Group 8 K memory boards on his system. By assigning the processor circuit's 2 K to the address range of 16 K to 18 K, one memory board to 0 to 8 K, and the other to 8 K to 16 K, an 18 K system results, with all active memory in the low address range.

The EROM used for bootstrapping is a relatively slow device, so the processor must be forced to wait for its data access. A 74121 provides a 475 ns delaying pulse to the processor when either the processor EROM is accessed or an external slow memory access is required. Since the Digital Group programmable memory cards are built using 500 ns access time (or faster) 2102 static memories, the processor normally runs at full speed.

Some Notes on Construction

While the circuit diagrams of figures 1 and 2 provide the information needed to wire wrap or hand wire your own Z-80 processor, I'll bet you'll find the Digital Group processor board in our kit to be a

worthwhile time saver. This Z-80 processor card is manufactured using two sided FR-10 printed circuit board material and measures 12 inches wide by 5 inches high (30.5 cm wide by 12.7 cm high). It has a dual 50 pin (100 terminals in all) connector to the backplane assembly. The definition of signals at the connector is provided in table 1.

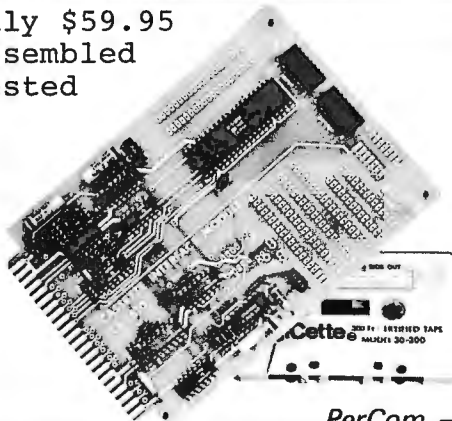
The Digital Group board is not "Altair compatible" due to two major system constraints: processor independency and use of a single fully protected external power supply. These design goals ruled out the bus structure supported by MITS and independent suppliers of peripherals for MITS systems. Experienced designers will undoubtedly interface the Z-80 to the "Altair bus" but the processor dependency problem will remain. Some experimenters may wish to custom design this Z-80 into their own system. The circuit of figures 1 and 2 should provide sufficient details of the Z-80's operation to assist you and provide a starting point. Further detailed information on the Z-80 chip and its specifications is of course available from its manufacturer, Zilog Inc.

Testing

After building the processor circuit, but before inserting any of your (socketed) integrated circuits, try a little preliminary testing with an ohmmeter. Check for a short between backplane terminals 1 and 2, 2 and 50, and 1 and 50. 1 and 2 should show an initial momentary low resistance and then approach infinity as power supply bypass capacitors charge up. 2 and 50 will show some resistance due to the zener, and to ohmmeter polarity, but not a short.

Two techniques are possible at this point. One way (referred to in the fine print of traditional literature as the "smoke test") is to plug in all integrated circuits and insert the card in a backplane assembly wired for

Only \$59.95
Assembled
Tested



FINALLY - A CASSETTE INTERFACE THAT WORKS! The **PerCom** CI-810

- Easily connected to any computer
- An 8-bit parallel interface
- 'Kansas City' Standard
- Load a 1k byte program in 40 seconds
- Little or no software required
- Easily upgraded to 218 byte/sec
- Operate 2 tape units simultaneously
- 18 page Instruction Manual

PerCom Data Co.

4021 Windsor, Garland, Texas 75042

(214) 276-1968

PerCom - 'peripherals for personal computing'



Table 2: Power connections for the Z-80 processor circuit shown in figures 1 and 2. Note that IC8 and IC9, IC18 and IC19 are omitted from the numbering sequence.

Number	Type	+5 V	GND	-9 V
IC0	2102	10	9	—
IC1	2102	10	9	—
IC2	2102	10	9	—
IC3	2102	10	9	—
IC4	2102	10	9	—
IC5	2102	10	9	—
IC6	2102	10	9	—
IC7	2102	10	9	—
IC10	2102	10	9	—
IC11	2102	10	9	—
IC12	2102	10	9	—
IC13	2102	10	9	—
IC14	2102	10	9	—
IC15	2102	10	9	—
IC16	2102	10	9	—
IC17	2102	10	9	—
IC20	1702A	12,13, 15,22, 23	—	16,24
IC21	74121	14	7	—
IC22	7400	14	7	—
IC23	7442	16	8	—
IC24	7404	14	7	—
IC25	7420	14	7	—
IC26	7430	14	7	—
IC27	7442	16	8	—
IC28	7402	14	7	—
IC29	7474	14	7	—
IC30	8T97	16	8	—
IC31	8T97	16	8	—
IC32	8T97	16	8	—
IC33	7404	14	7	—
IC34	74125	14	7	—
IC35	74125	14	7	—
IC36	7430	14	7	—
IC37	74123	16	8	—
IC38	4010	16,1	8	—
IC39	74125	14	7	—
IC40	7403	14	7	—
IC41	8T97	16	8	—
IC42	8T97	16	8	—
IC43	Z-80	11	29	—
IC44	74LS02	14	7	—
IC45	74125	14	7	—
IC46	7403	14	7	—
IC47	8T97	16	8	—
IC48	7442	16	8	—
IC49	7440	14	7	—
IC50	7400	14	7	—

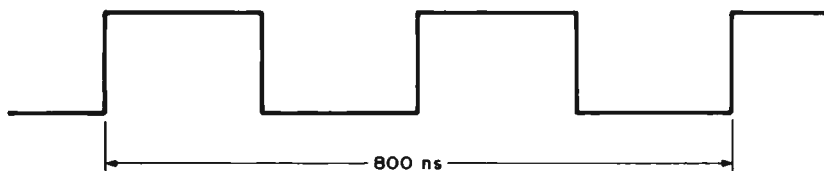


Figure 3: Central processor clock timing waveform. To verify the frequency of oscillation with a calibrated oscilloscope, measure the total time interval for two cycles of the clock waveform. This interval should be 800 ns if the correct crystal is used and it is oscillating at its fundamental frequency. A frequency counter would show 2.5 MHz as the frequency.

power. Another way is to insert only one or two integrated circuits at a time, function by function, and test as you go. The Digital Group has found a compromise which seems to work best when building kits, namely to plug in all but most critical or expensive integrated circuits, then test. This approach is optimal when using printed circuit wiring since the probability of a disastrous wiring error is in general low, assuming a fully debugged printed circuit board. Then if OK so far, plug them in and go ahead.

So, proceeding with this approach, insert all integrated circuits *except* the Z-80, the 1702A, and the 2102s. Note that all integrated circuits except 2102s in the Digital Group Z-80 board have their keyway or dot indicating the pin 1 end oriented away from the connector.

Measure the resistance at the backplane voltage supply pins again. In particular, note the lower resistance value between backplane pins 1 and 2. Reverse the ohmmeter and remeasure. A shorted reading now indicates a bad integrated circuit, and near equal readings indicate a reversed integrated circuit somewhere. Now insert the crystal into its holder. In our Digital Group kits this is done by snapping in the body of the crystal (gently), then pushing forward to contact the pins.

Before inserting the processor card into its backplane connector, measure the voltages at the connector. A single wrong voltage may cost you a board's worth of ICs.

Measure these backplane pins against ground:

- Pin 1 — +5 V $\pm 5\%$
- Pin 2 — 0 V
- Pin 50 — -12 V $\pm 10\%$

(The backplane pin 1 end is marked on the Digital Group Z-80 processor card. If you use a homebrew assembly, use the equivalent test before proceeding.)

Make a final inspection of the processor. Check for shorts between components on the top and lines running underneath. In kit systems, look for any solder bridges. Check the proper pin 1 orientation of all your integrated circuits. If you use the printed circuit, sight down the rows of pins for missing solder points. Missed solder points typically seem to occur at the end pins of integrated circuit sockets, and one side of resistors or capacitors.

After all this preliminary checking you can insert the processor board into its connector.

Apply power to the system and again measure voltages at the processor card as noted previously.

Checking Your Waveforms

Connect a calibrated triggered sweep oscilloscope to pin 6 of the 7400 IC50b. Set the triggering to occur on the positive edge, and the sweep setting to 100 ns per division. Look for a two cycle time of 800 ns seconds as shown in figure 3. If your oscilloscope does not sweep as fast as 100 ns/div, then a slower sweep can be used; but be absolutely sure that the two cycle time is exactly 800 nanoseconds as shown in figure 3.

A frequency counter may also be attached to pin 6 of IC50b. The desired frequency is 2.5 MHz. Any appreciable error indicates either a defective crystal, a bad 7400, or an overtone oscillation (one way to correct this last case is by using 74L00 for IC50).

Measure the voltage at the following pins (before expensive integrated circuits have been inserted). Correct any discrepancy.

Z-80 (IC43) : pin 29 = 0 V
pin 11 = +5 V

1702A (IC20) : pins 24 & 16 = -9 V
pins 12, 13, 15, 22
and 23 = +5 V

Any 2102 RAM: pin 9 = 0 V
pin 10 = +5 V

Carefully insert the Z-80, the 1702A, and the 2102s. With the large Z-80 and 1702 circuits, insertion should be done evenly without allowing excessive stress. Packages have been known to crack into two parts during insertion. Make sure that pin 1 (indicated by either a dot or a 1 on these circuits) is properly oriented. Recheck the processor circuit assembly for orientation, lead shorts, solder shorts, and missing solder joints. Think courageous thoughts. Plug in the processor board. Bravely turn on power.

Using the Z-80 Processor Card

Several operational systems structures (see my June 1976 BYTE article) are consistent with this processor circuit design. This Z-80 circuit can be used with a minimal amount of additional hardware (a PIA and UART, a Teletype machine, and a suitably programmed EROM) as if it were an "evaluation board" that maintains system dependency so that different processor integrated circuits may be compared.

Preferably, this board becomes the key component in a much larger general purpose system. A special EROM is provided in the Digital Group Z-80 kit which interfaces this Z-80 board to our audio cassette and TV based system structure. A cassette of programming is provided with our kit version, which loads programmable memory with an

operating system for reading and writing cassettes, and building and displaying programs.

Conclusion

The Z-80 is a neat chip to use. Contrary to some grapevine rumors, you can't simply unplug your 8080 integrated circuit and plug in the Z-80; but it is an architecturally simple chip to design with. I hope this design excites you as much as the Z-80 excited me. Enjoy. ■

S.T.M. SYSTEMS

Not a Kit

Presents

Fully Tested

BABY!

A complete microcomputer in an attache case.

The unit uses the MCS 6502 8 Bit Microprocessor.

Up to 4K RAM fully buffered * Slot for 4K ROM (2708 type)

DMA, Video Interface (composite video) sixteen 32 character lines.

Audio cassette Interface (data rate approximately 1200 BPS load & dump).

I/O ports with 1 PIA 6820, 6520 type.

Typewriter type 63 key keyboard, (upper and lower case plus Greek with control key).

Power supply 120 VAC to 5 volt 3 amp fully regulated.

Speaker, two (2) LEDs, DMA, 60 Hz real time clock, video on and off keyboard and audio cassette dump and load format all under program control.

The first 200 systems sold will have a frosted Plexiglas case! Standard unit will have molded plastic case, Plexiglas case will become an option.

Audio cassette tape supplied with dump program, text editor, games of Shooting Stars, Life and Ticktack Toe, Music Program (self generated computer music and user generated from keyboard).

***Basic unit with 2K RAM and 512 Byte bootstrap loader and monitor in firmware (PROM) . . . \$ 850.00**

Unit with 4K RAM \$1000.00

Remember it's not a kit, it's fully tested and ready to go. Just plug BABY! in hook up your video monitor, load your auto cassette with the programs we supply and you're off and running.

Optional Video Monitor \$150.00

Be the first person on your block to have this unique, completely portable system.

ORDER TODAY:

S. T. M. SYSTEMS

P.O. Box 248

Mont Vernon, N.H. 03057

BankAmericard Exp. _____ **Cashier's Check**
 Master Charge No. _____ **Money Order**

Personal Check (allow 6-8 weeks for personal check to clear.)
Delivery 60 to 90 days after Receipt Of Order

Name _____

Address _____

City _____ State _____ Zip _____

Ask for our OEM discounts on customized version.

THE Z-80 REVOLUTION

TDL IS PROUD TO ANNOUNCE THE REVOLUTIONARY Z-80 CPU CARD, AN ALTAIR/IMSAI COMPATIBLE CPU CARD FEATURING THE POWERFUL Z-80 uP PRODUCED BY ZILOG INCORPORATED. WHAT'S SO REVOLUTIONARY ABOUT THE Z-80? A LOOK AT THE FOLLOWING COMPARISONS WILL SHOW YOU:

As you can see, the Z-80 is a very powerful and fast uP - in fact its a NEXT generation microprocessor. And its available to you in a totally compatible format, NOW. Just unplug your current CPU card, plug in the Z-80 CPU, load a program, and you're up and running - with a NEXT generation uP. The power and versatility of the Z-80 is unequalled in the uP field, and it opens the door to tremendous developments in the state of the art. More powerful, faster, and less memory consuming versions of your current 8080 software are just a part of the possibilities the Z-80 provides. (TDL's own 8-K BASIC for the Z-80 will be available in September.)

Each Z-80 CPU kit comes complete with:

- Prime commercial quality boards, IC sockets etc.
- easy to follow instructions
- Zilog's Z-80 Manual
- Schematics
- An easy to understand and apply user's guide
- TDL's Z-MONITOR on paper tape (soon to be available in deluxe PROM version)
- And membership in the Z-80 user's group.

Move up to the Z-80. Only \$269.

THE FASTEST RAM? The high speed capability of the Z-80 demands an extra-fast RAM to back it up, and TDL's new Z8K RAM board fills the bill. The Z8K is an 8K by 8 static RAM with the fastest access time in personal computing - 215ns. Its the only RAM in personal computing fast enough to let the Z-80 run at full speed with no wait states. If that isn't enough, it also happens to be one of the lowest powered RAMs around as well. Only 150 ma typical current load on the 5V supply. That makes the Z8K run cool -and perfect for battery standby operation as well. Other so-called "low power" 4K RAM boards can't compete with these specs. Its the perfect match for the Z-80, and its features and low cost make it a perfect match for ANY uP. (It's fully Altair bus compatible of course...) Price: Only \$295

WHAT ABOUT QUALITY? All TDL products share one thing in common - exceptionally high quality. The quality starts with engineering that is dedicated to keep your system state of the art at the lowest possible cost. Consider also the "Qual Division" whose ONLY purpose is

Comparison of the Zilog Z-80, Intel 8080, and Motorola 6800 CPU chips			
	Z80	8080	6800
NUMBER OF:			
Instructions	158*	78	72
Internal Registers	17	7	6
Addressing Modes	10	7	8
Voltage Required	+5	+5, -5, +12	+5
Standard Clock Rate	DC-3MHz	0.5-2MHz	0.1-1MHz
Clock Phases	1	2	2
Clock Voltage	4.2	8.4	4.8
Dynamic RAM refresh and timing signals without slowing down CPU or requiring additional circuitry	Yes	No	No
Single instruction memory to memory and memory to I/O BLOCK TRANSFERS	Yes	No	No
Single instruction SET, RESET, or TEST of any bit in accumulator, any general purpose register, or any external memory location	Yes	No	No
Single instruction BLOCK SEARCH of any desired length of external memory for any 8-bit character	Yes	No	No
Non-Maskable Interrupt and TTL compatible inputs	Yes	No	Yes
Internal sync of inputs and direct strobe of outputs	Yes	No	No

* Includes all 78 machine code instructions of the 8080A and is therefore capable of running any standard 8080A software without modification.

ADDITIONAL FEATURES OF THE Z-80:

- Up to 500% more throughput than the 8080A
- Requires 25% to 50% less memory space than the 8080A CPU
- Three modes of fast interrupt response plus a non-maskable interrupt
- Outperforms any other microcomputer in 4-, 8-, 16-bit applications

to keep TDL's products the best in the industry. And our products use only the finest boards available, prime components, sockets for all ICs, gold plated edge contacts and other earmarks of a commercial grade product. And its backed by a solid 90 day guarantee on parts and materials.

SAVE MONEY NOW Order both a Z-80 CPU card, and one or more Z8K RAM boards before September 1st, 1976, and you can deduct 10% on the total cost. Act now while this special offer lasts. (Does

not apply to COD orders.)

HOW TO ORDER Just send check or money order, or use your BankAmericard or Mastercharge, and your orders will be shipped to you postpaid within 30 days. COD orders must be accompanied by a 25% deposit. Your credit card order must include the serial # of the card, expiration date, and your order must be signed. New Jersey residents add 5% state sales tax. For more information, send for our free catalog.

Dealer Inquiries Invited

TDL

(609) 392-7070

TECHNICAL DESIGN LABS INC.
342 COLUMBUS AVENUE
TRENTON, NEW JERSEY 08629

BYTE'S BITS

Attention: Southern California Readers, Educators

Here is a bulletin board listing of a new course which is probably worth taking if you're a novice, or emulating if you're an instructor.

The prospect of a computer in every home, shop and classroom is no idle "campaign promise" to one professor at California State University, Long Beach.

"If you can't buy one, build one," is one of several approaches taken in three computer courses to be offered on Saturdays beginning September 4 through the CSULB School of Education. All three courses are designed for non-technical people: teachers, librarians, businesspeople; hobbyists or homemakers.

The instructor, Richard C McLaughlin, associate professor of instructional media, says that "some years ago, as a junior high school science teacher, I realized that my role in life was not developing future scientists but rather promoting an appreciation of science and technology among our entire population." His background includes a bachelor's degree cum laude in physics from the State University of New York at Albany and a PhD in instructional communications from Syracuse University. He has recently been active in the Southern California Computer Society, the California Educational Computing Consortium and the North Orange County Computer Club.

While some attention will be paid to traditional computers and minicomputers in these courses, by far the greatest emphasis will be placed upon low cost general purpose computers. These are now available as do-it-

yourself kits (about \$1000) or already assembled and waiting to be plugged in. Prof McLaughlin's courses can be of great use to people having little or no background in computer technology but willing to learn.

The purpose of the courses will be to acquire a functional understanding of computers resulting in practical applications. The first five Saturdays will constitute a course on the building of a microcomputer. No actual construction will be required, but the class should be of immense value to anyone using a microcomputer (or a larger minicomputer) or planning to build one from a kit.

The second course of five Saturdays will cover programming any type of computer (large timesharing service, minicomputer or personal computer system) in the conversational BASIC language now used in many schools and businesses throughout the nation.

The last five Saturdays will be devoted to a course on using computer terminals and setting up work stations tailored to the end user's special needs, be they in the classroom, library, shop or home.

The three courses begin on September 4, October 9 and November 13, running from 8:30 AM to 2:30 PM. Each course is worth two credit units and may be taken independently according to the student's own needs. Classes are open to all high school graduates, college students and adults. Persons not formally admitted to CSULB may enroll at \$66 per course through the Office of Continuing Education, 1250 Bellflower Blvd, Long Beach CA 90840. Telephone: (213) 498-5561.■

Microcomputer Interfacing Workshop

September 23, 24, 25, 1976, a three-day workshop based on the popular 8080 microprocessor, sponsored by the VPI and SU Extension Division of the Continuing Education Center in Blacksburg VA, will include many hours of experience in programming and interface construction with over 12 operating microcomputers for participant use. For more information contact Dr Norris Bell, VPI and SU Continuing Education Center, Blacksburg VA 24061, (703) 951-6328.■

Functional Specification: Altair Bus Driver

A question which has recurred in several letters is "How do I interface my simple 8 bit bidirectional bus to an Altair compatible peripheral?" What is needed is an article which defines the signals of the Altair back plane and gives an interface plan and design for making an Altair compatible extension bus to an arbitrary 8 bit processor such as the 6800, 6502, 8080, Z-80, 2650, etc. Such an article must include a table of pinouts, power and logic requirements, photographs of a prototype and a rough description of the processor and system in which it is used.■



Altair, Imsai, Polymorphic, compatible products

Video Terminal Interface: Connects to standard TV monitor or modified receiver to display 16 lines of 32 or 64 characters. Characters are formed in a 7 x 9 matrix for easy readability. Character set includes 128 upper and lower case ASCII characters and 64 graphic characters for plotting on a 48 x 64 (48 x 128 with memory option) array. An 8-bit input port is provided for the keyboard. Characters are stored in the onboard memory, which may be read out of or written in to by the computer. Cursor control, text editing, and graphics software is included. \$185 (32 char.) kit. \$210 (64 char.) kit.

Poly I/O Idea Board: This will save you a lot of time in making prototype circuits. I/O port address is selectable with dip switch, and inputs and outputs are fully buffered. \$55 kit.

Analog Interface: Good for interfacing your computer to an analog world. Ten bits of resolution in and out. \$145 for one channel and \$195 for two channels (kit).

Ask about how to get a free POLY I/O Idea Board or Analog Board.

8K RAM on a single board. Connection for battery backup. \$300 kit.

Special Offer

Video Terminal Interface
(32 character) and 8K RAM, \$450 kit.
Expires - September 30th, 1976.

You've probably been hearing about the POLY 88 microcomputer system that uses keyboard and video. We don't have the space here to describe all the features. See it at your local computer store.

Support your local computer store.

All prices and specifications subject to change without notice. Prices are USA only. Calif. residents add 6% sales tax. All non-paid orders add 5% USA shipping, handling, and insurance. (Outside USA add 10%) BankAmericard and Master Charge accepted.

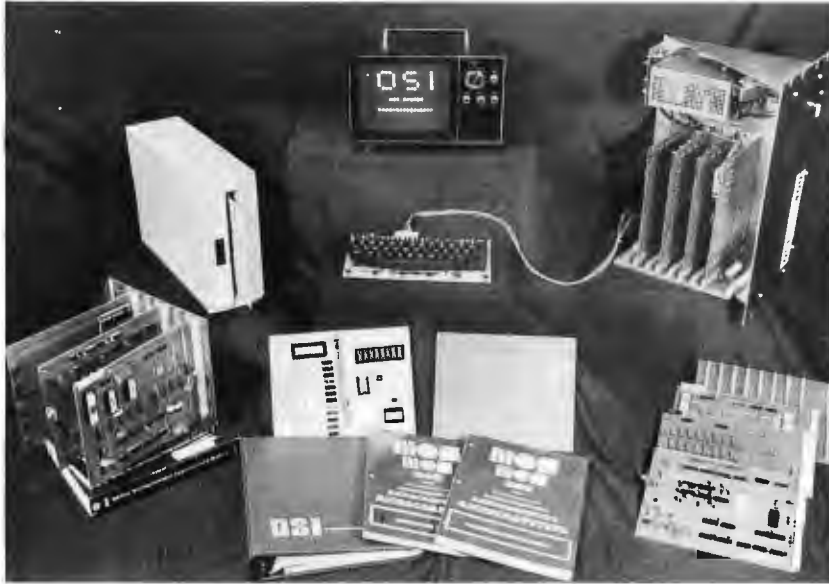
PolyMorphic Systems

737 S. Kellogg, Goleta, CA 93017
(805) 967-2351

What's New?

OSI 400 System

Ohio Scientific Instruments, 11679 Hayden St, Hiram OH 44234, has announced the "Model 400 Superboard" single board computer. The board itself, minus components, sells for \$29 and will work with either the MOS Technology 6502 or the Motorola 6800 central processor circuits. The board has slots for 1 K bytes of memory, 1 6820 PIA, one 6850 PIA, current loop (Teletype)



and RS-232 serial interfaces. A complete kit for a 6502 version with monitor PROM and parts for a Teletype current loop interface is \$139, and the same kit for a 6800 processor is \$159. Also available is the Model 470 floppy disk, the Model 420 memory expansion board, the Model 430 IO board, and the Model 440 video graphics board. Write for the OSI catalog brochure about their boards, kits and assembled products. ■

Advance Information

Lloyd Rice of Computalker has forwarded to BYTE a copy of the "advance announcement" brochure on the Computalker CT-1 Speech Synthesizer. The price of this unit will be \$395 in Altair/IMSAI/Polymorphic compatible plug-in board form. Target date for hardware delivery is September 1 1976. All Computalker CT-1 customers will be supplied with the CSR1 software driver package which features "a sophisticated synthesis by rule system incorporating the latest research findings on human speech.... Versions are planned for the 8080, 6800 and 6502 CPU chips." Write Computalker, 821 Pacific St No. 4, Santa Monica CA 90405, for the brochure, which also is accompanied by a reprint of Lloyd's article in the April 1976 issue of Dr Dobbs' *Journal of Computer Calisthenics and Orthodontia*. ■

Catalog for Electronics Designing and Testing

59 Ways to Save Time and Money Designing and Testing in Electronics, a full color 32 page catalog of electronic prototype breadboarding and test equipment, has been introduced by Continental Specialties Corporation. The catalog is said to have a lot more utility than simply displaying CSC products and prices; it is billed as "a handy problem-solver for electronics hobbyists as well as working designers, technicians and production people who want to save time and money in every aspect of design, production and QC testing." The catalog, which includes a list of domestic and foreign distributors, is available free from Continental Specialties Corporation, 44 Kendall St, POB 1942, New Haven CT 06509. ■



Right from the Source

Intersil, manufacturer of the IM6100 PDP-8 compatible microcomputer, has just announced their version of the prototyping and evaluation board needed to try out the microcomputer. The "Intercept Jr" system is a tutorial trainer utilizing Intersil's IM6100 and related CMOS devices. According to the manufacturer, "the system provides students, hobbyists and designers with practical low cost exposure to microprocessors, RAMs, PROMs and input output interfacing."

The Intercept Jr product recognizes the instruction set of Digital Equipment Corporation's PDP-8/ETTM minicomputer and provides an operating CMOS microcomputer on a 10 by 11 inch (25.4 by 27.9 cm) double sided printed circuit board. Also on the

board are a keyboard, two 4 digit LED displays, a "resident microinterpreter," and a battery power supply. The extra boards shown in the photo include a CMOS 1 K by 12 bit programmable memory module with its own battery backup for non-volatility, a 2 K by 12 bit PROM module and a serial IO interface which has both RS-232 and 20 mA current loop capabilities.

The Intercept Jr system comes completely assembled and factory tested with batteries. Power terminals are also provided to enable running the system from a 5 or 10 V power source. The owner's handbook contains full details of the system's operation, a hardware description and basic programming techniques. The prices are well within the range of the individual who wants a PDP-8 compatible machine: The basic computer module is \$281, the programmable random access memory module is \$145, the PROM module is \$74.65, and the IO module is \$81.70. A computer store could easily buy these modules, put them inside an attractive case and have a commercial finished product with a well known instruction set. According to the Intersil people, all modules are in stock for immediate delivery.

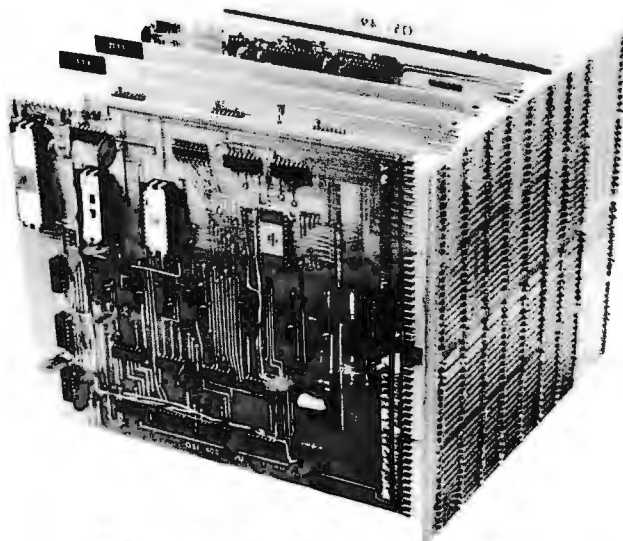
For information and ordering contact Intersil Inc, 10900 No Tantau Av, Cupertino CA 95014. ■

Software New Product: 8080 TRACE Program

The 8080 TRACE Program is a valuable software development tool which speeds and simplifies 8080 program debugging. TRACE performs its function by executing the problem program's instructions under TRACE control and provides the programmer a visual display of the program counter (PC), contents of the status word (SW) and registers A through L for each executed instruction. Thus, deviations from expected performance are readily recognized and corrected with minimal programmer participation.

TRACE uses an IO terminal such as an octal keyboard and display device for entering TRACE run parameters and displaying register contents dynamically. The instruction location is displayed for each instruction executed within the address limits specified by the programmer at TRACE initiation time. The sense switches can be used optionally to select registers for display during the program's run time.

The TRACE program listing and description are available for \$7.50; an Altair ACR-compatible cassette tape is included for \$10. For additional information, send SASE to R E Rydel, 1411 Northgate Sq, Apt 21B, Reston VA 22090. ■



Meet the new OSI 400 Computer System.

Now more performance and more flexibility actually cost you less.

Ask yourself how much system you need. Or how little. Whatever the answer, even if you want to change it later, you get more system for less money with the OSI 400.

Start with the OSI Superboard. Add your choice (!) of a 6502, 6512 or 6800 microprocessor; eight 2102s for 1024 bytes of RAM; and an external front panel. Power it up and you have a working CPU. Or populate the board with a processor, system clock, 512 bytes of PROM, 1024 of RAM, an ACIA with RS-232 or 20 ma loop interface, a PIA with 16 I/O lines and full buffering to as many as 250 system boards for system expansion.

Even fully populated, Superboard costs less than \$140 with a 6502, less than \$160 with a 6800.

But take a look at what you can have for \$29. Our special offer includes a plated-through-hole G-10 epoxy Superboard, bare, plus a 50-page theory of operation and construction manual including complete chip documentation in an attractive OSI binder.

And Superboard is just the beginning of the OSI 400 system. You can expand its memory; interface to many I/Os

including plotters, cassettes, FSK, ASCII, Baudot and more; go video, including graphics; even add floppy disk. And bare boards are just \$29 each, complete with in-depth manuals.

But first things first.



400 me now! SPECIAL — \$29 postpaid with this coupon only.

Name _____

Address _____ City _____

State _____ Phone _____



Enclose check or money order or supply Bank Card information below.

Card No. (include all digits) _____ Good thru _____

Interbank No. (Master Charge) _____ Sign your name _____

Ohio residents, please add tax.

OSI

Ohio Scientific Instruments
11679 Hayden Street
Hiram, Ohio 44234 Dept. B

Microprocessor Update:

SC/MP Fills a Gap

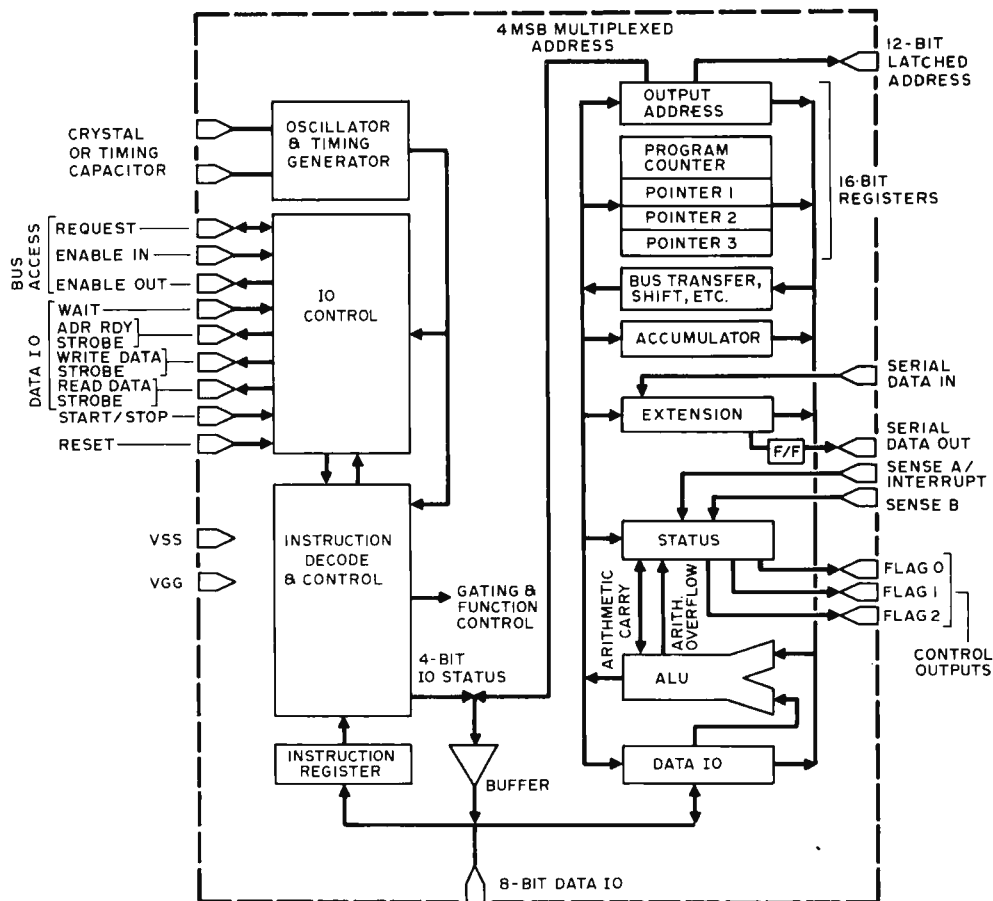
Robert Baker
15 Windsor Dr
Atco NJ 08004

The new National Semiconductor microprocessor SC/MP, commonly called SCAMP, was designed to fill a gap between clumsy 4 bit microprocessors and the currently available 8 bit microprocessors. According to the manufacturer, it is simple to use, requiring very few support chips for a basic system and is upgradable as the need arises. Only a single +10 to +14 V power supply is needed for the 40 pin dual inline processor chip. A block diagram of the processor chip is shown in figure 1.

Microprocessor

The processor provides simple interfacing with an 8 bit data bus that has TTL or CMOS compatible options. There are four serial data output ports and three serial data input ports along with two sense inputs for simple IO hardware. Three software controlled, user accessible output control flags may be used for these direct control output applications. A separate bus access control provides Direct Memory

Figure 1: Internal block diagram of the National Semiconductor SC/MP. In addition to a fairly typical 8 bit bus oriented processor design, the SC/MP includes some features intended for ultra low cost system designs. These include three programmable output flags, a serial input and output port, and two sense inputs, one of which can be used for interrupts. This is one of the reasons it is possible to make an inexpensive minimal system such as the \$99 kit shown in photo 1.



Access (DMA), multiprocessor, and single instruction step capabilities. The direct memory access feature allows fast direct data transfers from memory to peripherals, peripherals to memory, and peripheral to peripheral. Asynchronous control signals are generated on chip for direct interfacing to memories of any speed. Multiple mode, 16 bit addressing allows addressing of up to 64 K bytes of memory with peripherals addressed in the same manner as memory. The available addressing modes include program counter relative, pointer relative, immediate data, and auto-indexing. There are 46 instructions available as listed in table 1.

The typical microcycle time is 2 μ s, so instruction times range from 10 to 46 μ s. Four 16 bit address pointer registers are available as stack pointers to external memory for unlimited subroutine nesting.

Applications

A first level or basic SC/MP system is shown in figure 2. This is a typical dedicated intelligent controller. The only requirements are a power supply (+10 to +14 V), a clock timing capacitor or crystal, and the desired memory. In this configuration, only 4096 bytes of memory are addressable. By adding a hex D flip flop (National MM74C174), an 8 channel digital multiplexer (National MM74C151), and a 1 to 8 demultiplexer (National MM74C42), the system can be expanded to a second level system as shown in figure 3. This system provides 8 input and 8 output serial data ports along with four general purpose, latched, control flags. There is no change in the memory addressing capability; only 4096 bytes can still be addressed.

Expanding the system (further) requires the addition of interface latch elements



Photo 1: National Semiconductor markets an evaluation kit for SC/MP which is probably one of the least expensive ways to try out a working processor. The kit includes the circuit board, all electronic components including processor and firmware ROM, crystal for 1 MHz clock speed and power supply regulator. The assembly of this board, using ample documentation as a guide, results in a computer with 256 bytes of programmable memory, a 512 byte firmware operating system called "KITBUG" and a 20 mA Teletype current loop interface for operator control. A single 12 V power supply is required.

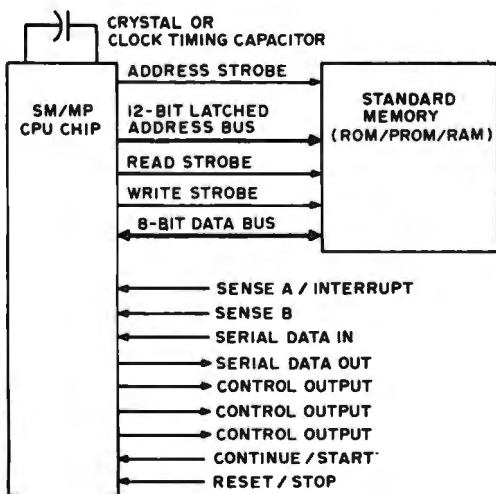


Figure 2: What National Semiconductor calls a "first level" system is illustrated in this block diagram. This block diagram roughly corresponds to what the SC/MP kit produces after assembly. The block labelled "Standard Memory" contains 256 bytes of programmable memory and 512 bytes of read only memory in the kit.

Table 1: SC/MP instruction summary (Typical execution time, 2 μ s per microcycle).

Double-byte instructions:		Execution time in microcycles	Single-byte instructions:		Execution time in microcycles
Memory Reference	Load	18	Extension Register	Load AC from extension	6
	Store	18		AND extension	6
	AND	18		OR extension	6
	OR	18		EXCLUSIVE OR extension	6
	EXCLUSIVE OR	18		Decimal ADD extension	11
	Decimal ADD	23		ADD extension	7
	ADD	19		Complement and ADD extension	8
Transfer	Complement and ADD	20	Pointer Register Move	Exchange pointer low	8
	Jump	11		Exchange pointer high	8
	Jump if positive	9,11		Exchange pointer with PC	7
	Jump if zero	9,11	Shift, Rotate, Serial I/O	Serial I/O	5
Jump if not zero	9,11	Shift right		5	
Memory increment/Decrement	Increment and load	22		Shift right with link	5
	Decrement and load	22		Rotate right	5
Immediate	Load	10		Miscellaneous	Rotate right with link
	AND	10	Halt		8
	OR	10	Exchange AC and extension		7
	EXCLUSIVE OR	10	Clear carry/link		5
	Decimal ADD	15	Set carry/link		5
	ADD	11	Disable interrupts		6
	Complement and ADD	12	Enable interrupts		6
Miscellaneous	Delay	3 to 132,096	Copy status to AC	5	
			Copy AC to status	6	
			No operation	5-10	

(ISP-8A/543) and buffering (such as the ISP-8A/551) to the processor. This expanded system, as shown in figure 4, provides a full capability system which can now address up to 64 K bytes of memory.

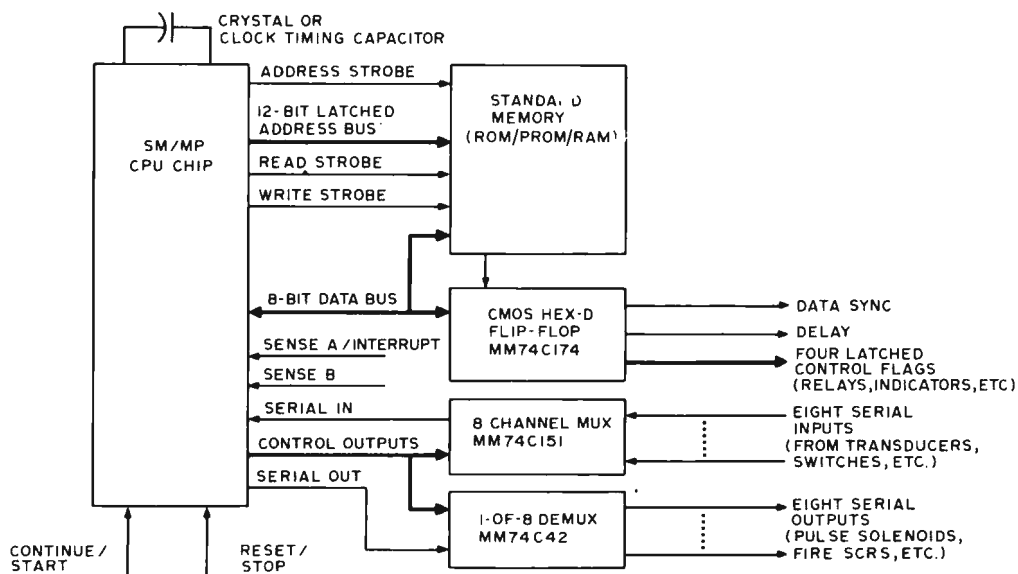
There is a complete collection of hardware and software support for the SC/MP system including a debug system, application cards, assembler, editor, system diagnostics, cross assembler, and application routines. The National Semiconductor users group, COMPUTE, is also available to SC/MP users

as well as their software library which makes programs available for the cost of reproduction.

In Conclusion

The SC/MP processor is approximately an order of magnitude slower than other 8 bit processors such as the 6800, 8080, Z-80 or 6502. For example, the SC/MP addition time for an 8 bit quantity in memory is 19 microcycles or 38 μ s at its rated speed, as opposed to the 4 processor cycles or 4 μ s

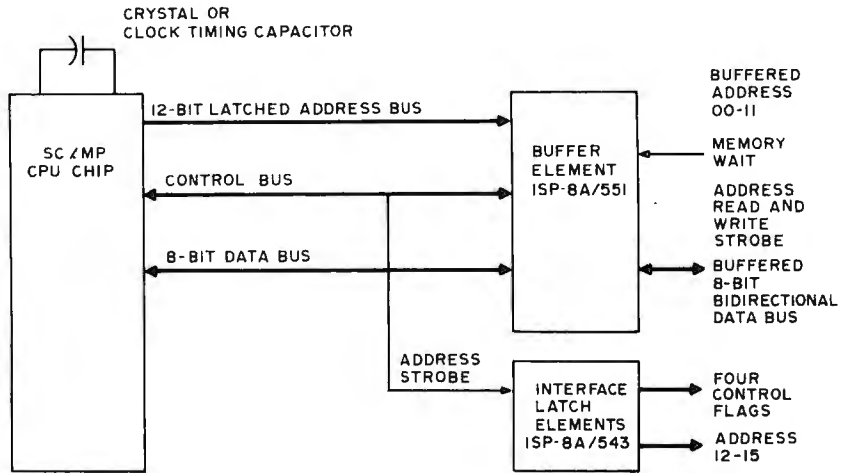
Figure 3: The SC/MP "second level" system is illustrated by this diagram. Here, the use of additional CMOS integrated circuits provides more serial inputs and outputs for use in a dedicated control situation.



required for an MOS Technology 6502 to add two 8 bit numbers at its rated 1 MHz clock speed.

As with any 8 bit processor, the instruction set of the SC/MP is adequate to accomplish general purpose functions. The set includes one very interesting function which reflects the SC/MP's intended use in dedicated low cost systems: in such systems timing loops are a frequent requirement, so National put in a hardware delay instruction which provides a programmable wait period of from 3 to 132,096 microcycles with a single instruction. This eliminates the need for programmed timing loops while allowing a very flexible resolution of from 6 microseconds to 0.264192 seconds. While this is not a true real time clock integrated into an interrupt structure, the delay instruction should provide programming convenience in many time dependent situations.

Thus, balancing its low cost against a relatively slow execution time, the SC/MP will most likely be implemented as a "smart" control element buried into peripherals and accessories for the personal systems market. Adapting it as a general purpose processor is quite possible; however, in larger systems where the processor is a small part



of the total cost, the price advantage relative to a faster processor is minimal. In small systems, such as the prototyping kit illustrated in photo 1, the price advantage relative to faster processors is obvious in the \$99 price of this simple processor available off the shelf from several distributors. If you're building a small computer for a specialized application, then SC/MP will be a likely prospect when combined with some read only memory and custom IO devices. ■

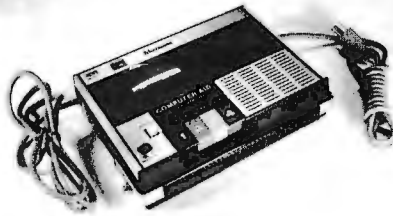
Figure 4: National calls this the "expanded SC/MP" system. Here, the use of a buffer element and interface latch element results in a conventional 16 bit address bus along with an 8 bit bidirectional data bus, for use in larger systems.

MODEL CC-7 SPECIFICATIONS:

- A. Recording Mode: Tape saturation binary. This is not an FSK or Home type recorder. No voice capability. No Modem. (NRZ)
- B. Two channels (1) Clock, (2) Data. OR, Two data channels providing four (4) tracks on the cassette. Can also be used for Bi-Phase, Manchester codes etc.
- C. Inputs: Two (2). Will accept TTY, TTL or RS 232 digital.
- D. Outputs: Two (2). Board changeable from RS 232 to TTY or TTL digital.
- E. Runs at 2400 baud or less. Synchronous or Asynchronous. Runs at 4800 baud or less. Synchronous or Asynchronous. Runs at 3.1"/sec. Speed regulation $\pm .5\%$
- F. Compatibility: Will interface any computer or terminal with a serial I/O. (Altair, Sphere, M6800, PDP8, LSI 11, IMSAI, etc.)
- G. Other Data: (110-220 V), (50-60 Hz); 3 Watts total; UL listed 955D; three wire line cord; on/off switch; audio, meter and light operation monitors. Remote control of motor optional. Four foot, seven conductor remotng cable provided. Uses high grade audio cassettes.
- H. Warrantee: 90 days. All units tested at 300 and 2400 baud before shipment. Test cassette with 8080 software program included. This cassette was recorded and played back during quality control.

ALSO AVAILABLE: MODEL CC-7A with variable speed motor. Uses electronic speed control at 4"/sec. or less. Regulation $\pm .2\%$ Runs at 4800 baud Synchronous or Asynchronous without external circuitry. Recommended for quantity users who exchange tapes. Comes with speed adjusting tape to set exact speed.

DIGITAL DATA RECORDER \$149.95 FOR COMPUTER or TELETYPE USE Any baud rate up to 4800



Uses the industry standard tape saturation method to beat all FSK systems ten to one. No modems or FSK decoders required. Loads 8K of memory in 17 seconds. This recorder, using high grade audio cassettes, enables you to back up your computer by loading and dumping programs and data fast as you go, thus enabling you to get by with less memory. Can be software controlled.

Model CC7... \$149.95
Model CC7A... \$169.95

NATIONAL multiplex
CORPORATION

NEW - 8080 I/O BOARD with ROM. Permanent Relief from "Bootstrap Chafing"

This is our new "turnkey" board. Turn on your Altair or Imsai and go (No Bootstrap-ping). Controls one terminal (CRT or TTY) and one or two cassettes with all programs in ROM. Enables you to turn on and just type in what you want done. Loads, Dumps, Examines, Modifies from the keyboard in Hex. Loads Octal. For the cassettes, it is a fully software controlled Load and Dump at the touch of a key. Even loads MITS Basic. Ends "Bootstrap Chafe" forever. Uses 512 bytes of ROM, one UART for the terminal and one USART for the Cassettes. Our orders are backing up on this one. No. 2SIO (R)

Kit form \$140. - Fully assembled and tested \$170.00

Send Two Dollars for Cassette Operating and Maintenance Manual with Schematics and Software control data for 8080 and 6800. Includes Manual on I/O board above. Postpaid

Master Charge & BankAmericard accepted.

On orders for Recorders and Kits please add \$2.00 for Shipping & Handling. (N.J. Residents add 5% Sales Tax)

3474 Rand Avenue, Box 288
South Plainfield, New Jersey 07080
(201) 561-3600

You'll Want to Nybble at these Byte Books



- **The TTL Cookbook** by Don Lancaster, published by Howard W Sams, Indianapolis, Indiana. Start your quest for data here with Don's tutorial explanations of what makes a TTL logic design tick. 335 pages, \$8.95.

- **The TTL Data Book for Design Engineers**, by Texas Instruments Incorporated. How does an engineer find out about the TTL circuits? He reads the manufacturer's literature. This 640 page beauty covers the detailed specs of most of the 7400 series TTL logic devices. No experimenter working with TTL has a complete library without **The TTL Data Book for Design Engineers**. Order yours today, only \$3.95.

- **The Supplement to The TTL Data Book for Design Engineers**, by Texas Instruments Incorporated. What happens when you can't find a 7400 series device listed in **The Data Book for Design Engineers**? Before you start screaming and tearing your hair out in frustration, turn to the Supplement. The Supplement has 400 pages of additional information including a comprehensive index to both TTL Data Book volumes. To be complete (and keep your hair in place and vocal cords intact) you'd best order the supplement at \$1.95 to accompany the main volume.

- **The Linear and Interface Circuits Data Book for Design Engineers**, by Texas Instruments Incorporated. When you run across one of those weird numbers like 75365 the immediate frustration problem occurs again. What kind of gate could that be? We won't tell in this ad, but you can find out by reading the specifications in **The Linear and Interface Circuits Data Book for Design Engineers**. You can interface your brain to the 72xxx (linear) and 75xxx (interface) series of functions by ordering your copy of this 688 page manual at only \$3.95.

- **The Semiconductor Memory Data Book for Design Engineers**, by Texas Instruments Incorporated. Don't forget the importance of memories to your systems. Refer to this 272 page manual to find out about the TI versions of many of the popular random access memories and read only memories. Order your personal copy today, only \$2.95.

Where does the editor of a computer magazine turn when he must verify some author's hardware design? Information on a 75450 interface gate, or a 74147 priority encoder circuit does not spring forth by magic. Checking the information supplied by authors is part of BYTE's quality control program.

When you build a project, you need this same sort of information. All you find in the advertisements for parts are mysterious numbers identifying the little beasts . . . hardly the sort of information which can be used to design a custom logic circuit. You can find out about many of the numbers by using the information found in these books. No laboratory bench is complete without an accompanying library shelf filled with references — and this set of Texas Instruments engineering manuals plus Don Lancaster's *TTL Cookbook* will provide an excellent starting point or addition to your personal library.

- **The Transistor and Diode Data Book for Design Engineers**, by Texas Instruments Incorporated. You'd expect a big fat data book and a wide line of diodes and transistors from a company which has been around from the start of semiconductors. Well, it's available in the form of this 1248 page manual from TI which describes the characteristics of over 800 types of transistors and over 500 types of silicon diodes. This book covers the TI line of low power semiconductors (1 Watt or less). You won't find every type of transistor or diode in existence here, but you'll find most of the numbers used in switching and amplifying circuits. Order your copy today, only \$4.95.

- **The Power Semiconductor Handbook for Design Engineers** by Texas Instruments Incorporated. To complement the low power transistor handbook, TI supplies this 800 page tome on high power transistors and related switching devices. Here is where you find data on the brute force monsters which are used to control many Watts electronically. Fill out your library with this book, available for only \$3.95.

- **Understanding Solid State Electronics** by Texas Instruments Incorporated. This is an excellent tutorial introduction to the subject of transistor and diode circuitry. The book was created for the reader who wants or needs to understand electronics, but can't devote years to the study. This 242 page softbound book is a must addition to the beginner's library at only \$2.95.

- **The Optoelectronics Data Book for Design Engineers** by Texas Instruments Incorporated. This 366 page book is a compendium of information on TI phototransistors, LEDs and related devices. Order yours at \$2.95.

- ___ TTL Cookbook @ \$8.95
- ___ TTL Data Book @ \$3.95
- ___ Supplement to TTL Data Book @ \$1.95
- ___ Linear and Interface Circuits @ \$3.95
- ___ Semiconductor Memory Data @ \$2.95
- ___ Transistor and Diode Data Book @ \$4.95
- ___ Power Semiconductor Handbook @ \$3.95
- ___ Understanding Solid State Electronics @ \$2.95
- ___ Optoelectronics Data Book @ \$2.95

Please add 75 cents for postage and handling.

Please allow six weeks for delivery.

Send to:

Name _____

Address _____

City _____ State _____ Zip _____

Check enclosed

Bill MC # _____ Exp. Date _____

Bill BA # _____ Exp. Date _____

Signature _____

BYTE PETERBOROUGH, NH 03458

IF YOU CAN'T FIND IT OFF THE SHELF TRY THE DATA DOMAIN

We are proud to announce we are now dealers for the Digital Group.

IMS Cromemco CSC Vector OSI Processor Tech

The

DATA DOMAIN

111 S. College Av
Bloomington, Ind. 47401
Phone (812) 334-3607

What's New?

OEMs and Kit Makers Take Note:

Bowmar Instrument Corporation, 8000 Bluffton Rd, Fort Wayne IN 46809, has introduced a new thermal printer, called the TP-3120, which can be integrated into products for the consumer markets. The printer is said to be highly reliable due to minimization of moving parts and evidence of a mean time between failure in excess of 3 million characters for the thermal print head and an overall operating life of more than 1 million lines of printing.

The TP-3120 operates at a speed of 29.4 characters per second and prints up to 1.07 lines per second. The printer produces alphanumeric outputs, has low power consumption and quiet operation, and thus should



prove attractive in small systems. The design goal was a printer for use in microprocessor based terminals, medical electronics, point of sale cash register devices, test equipment and other instances where hard copy is a desirable feature if the cost is low enough. ■



A 6800 Evaluation Board – Plus EROM Programmer and Tiny BASIC!

AMI (American Microsystems Inc) has introduced a microprocessor prototyping board for hardware and software evaluation of the 6800 based microcomputer systems family in specific applications.

The AMI 6800 Microprocessor Evaluation Board (EVK300) features a built-in programmer for the S6834 EROM circuit. This feature gives the AMI board an additional capability not usually found in evaluation kits. Using the board, designers can become proficient with the 6800 processor, and system development can take place quickly and painlessly.

The board can also serve as a general purpose computer for low volume systems by the utilization of up to 58 input/output lines and expansion up to 56 K bytes of programmable or read only memory.

The single board computer measures 10.5

inches (26.7 cm) by 12 inches (30.5 cm) and has two 86 pin edge connectors. The board can be used for evaluating incoming micro-circuits and for programming EROMs on a limited production basis. Communication with the computer is done through a Teletype current loop interface.

A high level interpretive computer language called AMI 6800 Tiny BASIC is furnished to EVK300 board users residing in the EROM at no extra charge, and prototyping operating system program (PROTO), residing in the ROM, is also supplied with the board.

The board is available in three package options: in kit form with the printed circuit board and a minimum of parts (EVK100 — \$295); an expanded kit with 512 byte EROM (EVK200 — \$595) and the expanded kit fully assembled and tested having 2 K bytes EROM with Tiny BASIC (EVK300 — \$950). Contact American Microsystems Inc, 3800 Homestead Rd, Santa Clara CA 95051. ■

More Tiny BASIC — Doctor Dobbs Is Really Moving

Electronic Product Associates Inc, 1157 Vega St, San Diego CA 92110, have announced that Tiny BASIC is now available for the Micro-68 computer development system. This BASIC is a 16 bit integer arithmetic subset of Dartmouth BASIC and includes: LET, IF ... THEN, INPUT, PRINT, GOTO, GOSUB, RETURN, END, REM, CLEAR, LIST, RUN, RND, andUSR. The entire system will fit in only 2 K of

memory and is available for a number of different configurations for input and output. Adding Tiny BASIC for \$10 to the Micro-68 computer provides one of the lowest cost BASIC language systems available today. The Micro-68 is a Motorola/AMI/Hitachi 6800 prototype development system which sells complete with power supply, cabinet, hexadecimal keyboard and 6 digit LED priced at \$430. ■



Get Up and Running Quickly with This Self Contained Package

Electronic Product Associates Inc, 1157 Vega St, San Diego CA 92110, (714)276-8911, has announced a complete microcomputer system for \$1050. Called the Expanded-68, the computer is based on the Motorola 6800 microprocessor chip set. Designed for engineering prototype development use, the Expanded-68 comes with 8 K of memory, power supply, 16 digit keyboard, Teletype interface, hexadecimal LED display, expansion cabinet, application manual and programming manual. It should prove quite usable for the experimenter as well as the industrial designer. Also available for direct interfacing are: dual floppy disk drive, 40 column impact printer, 132 column printer, cassette tape interface, TV interface, general purpose board, and full ASCII keyboard. ■



The IMSAI Floppy Disk Subsystem

IMS Associates, 14860 Wicks Blvd, San Leandro CA 94577, recently announced the availability of a floppy disk drive with an intelligent interface and controller. The system is specifically designed for use with the IMSAI 8080 computer.

The floppy disk has a capacity of 243 K



bytes using the IBM 3740 format. The interface and controller contains its own processor and direct access memory which operate independently but under command of the main processor of the IMSAI 8080. This enables the main processor to perform other tasks while a disk operation is in process. Also, the user can change the program format of the disk by reprogramming the interface EROM chips.

Up to four floppy disk drives can be controlled by one interface and controller. Each disk can be write protected under software control.

The disk drive comes in a cabinet with a power supply and the capacity to accommodate a second drive as shown in the photo. A rack mounted version is also available. All interconnection cables are included. The IMSAI floppy disk drive and interface controller are \$1,649 assembled and \$1,449 unassembled. An additional disk drive without a cabinet is \$925. The interface controller alone is \$799 assembled and \$599 unassembled.

Disk operating system software is available on diskette for \$40. Also, 12 K Extended BASIC with disk access capability was announced in July of this year. ■

A PROM Resident 8080 Assembler

Microcomputer Technique Inc, 1120 Reston International Office Center Building, Reston VA 22091, has announced a resident assembler which runs in the Intel System Design Kit (SDK) microcomputer. The assembler requires 4 K bytes of memory and is available from stock for \$450, delivered in four preprogrammed PROM chips.

The MTI assembler operates in one, two or three passes (user selectable), produces relocatable or absolute object code, contains a relocatable loader, has rudimentary conditional assembly statements and is designed to work with serial media such as magnetic or paper tape. ■

IMSAI announces a unique 4K RAM board for just \$139.

Nobody has a 4K RAM board that gives you so much for your money. It's fully compatible with the Altair 8800.

Through the front panel or under software control, you can write protect or unprotect any 1K group of RAM's. Also under software control you can check the status of any 4K RAM board in 1K blocks to determine whether it's protected or not. The board has LED's that clearly show you the memory protect status of each 1K block and which block is active. And there's a circuit provided that will let you prevent the loss of data in the memory if there's a power failure. This low power board has a guaranteed 450 ns cycle time—no wait cycle required. There's nothing like the IMSAI 4K RAM board around.

Dealer inquiries invited.

IMS

IMS Associates, Inc.

IMS Associates, Inc. Dept. B-9
1922 Republic Ave.
San Leandro, CA 94577
(415) 483-2093

Order Your IMSAI 4K RAM Board For Only \$139. Use BankAmericard, Master Charge, personal check or money order.

- Send _____ 4K RAM boards today
- Charge to my credit card.
- BAC No. _____
- MC No. _____

Signature _____

Name _____

Address _____

City/State/Zip _____

MACHINE LANGUAGE

PROGRAMMING FOR THE "8008"

and similar microcomputers

FUNDAMENTAL PROGRAMMING SKILLS

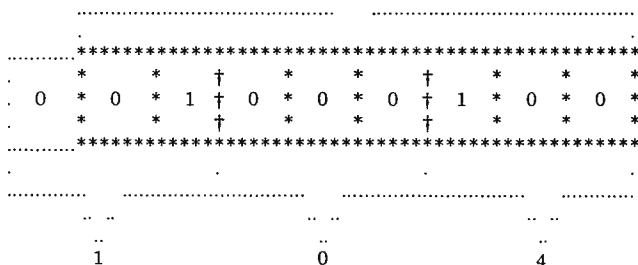
Before one can effectively develop machine language programs for a computer, one must be thoroughly familiar with the instruction set for the machine. It is assumed for the remainder of this manual that the reader has studied the detailed information for the instruction set of the 8008 CPU which was provided in the first chapter. The programmer should become intimately familiar with the mnemonics (pronounced kneemonics) for each type of instruction. Mnemonics are easily remembered symbolic representations of machine language instructions. They are far easier to work with than the actual numeric codes used by the computer when the programmer is developing a program. While the programmer will develop programs and think in terms of the mnemonics, the programmer must eventually convert the mnemonics to the machine codes used by the computer. This, however, is almost purely a look-up procedure. In fact, as will be seen shortly, this task can actually be performed by the computer through the use of an ASSEMBLER program.

Machine language programmers should also be familiar with manipulating numbers in binary and octal form. It is assumed that

readers are familiar with representing numbers as binary values. However, there may be a few readers who are not used to the convention of representing binary numbers by their octal equivalents. The technique is quite simple. It consists merely of grouping binary digits into groups of three and representing their value as an octal number. The octal numbering system only uses the digits 0 through 7. This is exactly the range that a group of three binary digits can represent. The octal numbering system makes it a lot easier to manipulate binary numbers. For instance, most people find it considerably more convenient to remember a three digit octal number such as 104 than the binary equivalent 01000100. An octal number is easily expanded to a binary number by simply placing the octal value in binary form using three binary digits.

The information in an eight bit binary register can be readily converted to an octal number by grouping the bits into groups of three starting with the least significant bits. The two most significant bits in the register which form the last group will only be able to represent the octal numbers 0 to 3. The diagram below illustrates the convention.

EIGHT CELL REGISTER



CONVERTING AN 8 BIT REGISTER FROM BINARY TO OCTAL NUMBERS

Note in the diagram how an imaginary additional binary digit with a value of zero was assigned to the left of the most significant bit so that the octal convention for the two most significant bits could be maintained.

A table illustrating the relationship between the binary and octal systems is provided for reference below.

BINARY PATTERN	REPRESENTATIVE OCTAL NO.
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

A person who desires to develop machine language programs for computers should become familiar with standard conventions used when dealing with closed registers (groups of binary cells of fixed length such as a memory word or CPU register). One very simple point to remember is that when a group of cells in a register is in the all ones condition:

11 111 111

and a count of 1 is added to the register, the register goes to the value:

00 000 000

Or, if a count of: 10 (binary) was added to a register that contained all ones, the new value in the register would be as shown:

```

11 111 111
+00 000 010
-----
00 000 001
    
```

Similarly, going the opposite way, if one subtracts a number such as 100 (binary) from a

Reprinted from MACHINE LANGUAGE PROGRAMMING FOR THE "8008" (and similar microcomputers).

Author: Nat Wadsworth
 Copyright 1975
 Copyright 1976 - Revised
 Scelbi Computer Consulting Inc
 With the permission of the
 copyright owner.

register that contains some lesser value, such as 010 (binary), the register would contain the result shown below:

```

00 000 010
00 000 100
-----
11 111 110
  
```

It may be noted that if one uses all the bits in a fixed length register, one may represent mathematical values with an absolute magnitude from zero to the quantity two to the Nth power, minus one (0 to $(2^{**N} - 1)$) where N is the number of bits in the register. If all the bits in a register are used to represent the magnitude of a number, and it is also desired to represent the magnitude as being either positive or negative in sign, then some additional means must be available to record the sign of the magnitude. Generally, this would require using another register or memory location solely for the purpose of keeping track of the sign of a number.

In many applications it is desirable to establish a convention that will allow one to manipulate positive and negative numbers without having to use an additional register to maintain the sign of a number. One way this may be done is to simply assign the most significant bit in a register to be a sign indicator. The remaining bits represent the magnitude of the number regardless of whether it is positive or negative. When this is done, the magnitude range for an N cell register becomes 0 to $(2^{**N} - 1)$ rather than 0 to $(2^{**N}) - 1$. The convention normally used is that if the most significant bit in the register is a one then the number represented by the remaining bits is negative in sign. If the MSB is zero, then the remaining bits specify the magnitude of a positive number. This convention allows computer programmers to manipulate mathematical quantities in a fashion that makes it easy for the computer to keep track of the sign of a number. Some examples of binary numbers in an eight bit register are shown next.

BINARY REPRESENTATION	OCTAL	DECIMAL
00 001 000	010	+ 8
10 001 000	210	- 8
01 111 111	177	+127
11 111 111	377	-127
00 000 001	001	+ 1
10 000 001	201	- 1

While the signed bit convention allows the sign of a number to be stored in the same re-

gister (or word) as the magnitude, simply using the signed bit convention alone can still be a somewhat clumsy method to use in a computer. This is because of the method in which a computer mathematically adds the contents of two binary registers in the accumulator. Suppose, for example, that a computer was to add together positive and negative numbers that were stored in registers in the signed bit format.

```

          00 001 000 (+ 8 decimal)
PLUS     10 001 000 (- 8 decimal)
-----
EQUAL    10 010 000 (This is not 0!)
  
```

The result of the operation illustrated would not be what the programmer intended! In order for the operation to be performed correctly, it is necessary to establish a method for processing the negative number called the two's complement convention. In the two's complement convention, a negative number is represented by complementing what the value for a positive number would be (complementing is the process of replacing bits that are '0' with a '1,' and those that are '1' with a 0) and then adding the value one (1) to the complemented value. As an example, the number minus eight (-8) decimal would be derived from the number plus eight (+8) by the following operations.

```

00 001 000 (Original + 8)
11 110 111 (Complemented)
00 000 001 (now add +1)
-----
11 111 000 (2's complement
              form of - 8)
  
```

Some examples of numbers expressed in two's complement notation with the signed bit convention are shown below.

BINARY REPRESENTATION	OCTAL	DECIMAL
00 001 000	010	+ 8
11 111 000	370	- 8
01 111 111	177	+127
10 000 001	201	-127
00 000 001	001	+ 1
11 111 111	377	- 1
00 000 000	000	+ 0
10 000 000	200	-128

Note that when using the two's complement method, one may still use the conven-

tion of having the MSB in the register establish the sign. If the MSB = 1, as in the above illustration, the number is assumed to be negative. Since the number is in the two's complement form, the computer can readily add a positive and a negative number and come up with a result that is readily interpreted. Look!

```

          00 001 000 (+ 8 decimal)
ADD     11 111 000 (- 8 dec as 2's comp)
-----
          00 000 000 (Correct answer = 0)
  
```

Another established convention in handling numbers with a computer is to assume that '0' is a positive value. Because of this convention, the magnitude of the largest negative number that can be represented in a fixed length register is one more than that possible for a positive number.

The various means of storing and manipulating the signs of numbers as just discussed have advantages and drawbacks, and the method used depends on the specific application. However, for most user's, the two's complement signed bit convention will be the most convenient, most often used, method. The prospective machine language programmer should make sure that the convention is well understood.

Another area that the machine language programmer must have a thorough knowledge of is the conversion of numbers between the decimal numbering system that most people work with on a daily basis, and the binary and octal numbering system utilized by computer technologists. Programmers working with microcomputers will generally find the octal numbering system most convenient. Because the conversion from octal to binary is simply a matter of grouping binary bits into groups of three as discussed at the start of this chapter, it is easier to remember octal codes than long strings of binary digits. However, most people are used to thinking in decimal terms, which the computer does not use at the machine language level. Thus, it is necessary for programmers to be able to convert back and forth between the various numbering systems as programs are developed.

The conversion process that is generally the most troublesome for people to learn is from decimal to binary, or decimal to octal (and vice-versa)! It is usually a bit easier for people to learn to convert from decimal to octal, and then use the simple octal to binary expansion technique, than to convert directly from decimal to binary. The easier method will be presented here. It is assumed that the reader is already familiar with going from octal to binary (and vice-versa). Only the conversions between decimal and octal (and the reverse) will be presented at this point.

A decimal number may be converted to its octal equivalent by the following technique:

Divide the decimal number by 8. Record the remainder (note that is the REMAINDER!!) as the least significant digit of the octal number being derived. Take the quotient just obtained and use it as the new dividend. Divide the new dividend by 8. The remainder from this operation becomes

the next significant digit of the octal number. The quotient is again used as the new dividend. The process is continued until the quotient becomes '0.' The number obtained from placing all the remainders (from each division) in increasing significant order (first remainder

as the least significant digit, last remainder as the most significant digit) is the octal number equivalent of the original decimal. The process is illustrated below for clarity.

The octal equivalent of 1234 decimal is:

ORIGINAL NUMBER	1234	/	8	=	154	2
LAST QUOTIENT BECOMES NEW DIVIDEND	154	/	8	=	19	2
LAST QUOTIENT BECOMES NEW DIVIDEND	19	/	8	=	2	3
LAST QUOTIENT BECOMES NEW DIVIDEND	2	/	8	=	2	
Thus the octal equivalent of 1234 decimal is:						2 3 2 2

The above method is quite easy and straightforward. Since a majority of the time the user will be interested in conversions of decimal numbers less than 255 (the maximum decimal number that can be expressed in an

eight bit register) only a few divisions are necessary:

The octal equivalent of 255 decimal is:

	QUOTIENT	REMAINDER
ORIGINAL NUMBER	255 / 8 =	31 7
LAST QUOTIENT BECOMES NEW DIVIDEND	31 / 8 =	3 7
LAST QUOTIENT BECOMES NEW DIVIDEND	3 / 8 =	- 3
Thus the octal equivalent of 255 is:		3 7 7

For numbers less than 63 decimal (and such numbers are used frequently to set counters in loop routines) the above method reduces to one division with the remainder being the LSD and the quotient the MSD.

This is a feat most programmers have little difficulty doing in their head!

The octal equivalent of 63 decimal is:

ORIGINAL NUMBER	63 / 8 =	7	7
LAST QUOTIENT BECOMES NEW DIVIDEND	7 / 8 =	-	7
Thus the octal equivalent of 63 is:		7 7	

Going from octal to decimal is quite easy too. The process consists of simply multiplying each octal digit by the number 8 raised to its positional (weighted) power, and then adding up the total of each product for all the octal digits:

2 3 2 2 Octal =

....2	X	(8*0)	=	(2 X 1)	=	2
...2	X	(8*1)	=	(2X8)	=	16
..3	X	(8*2)	=	(3 X 64)	=	192
2	X	(8*3)	=	(2 X 512)	=	1024
Thus the decimal equivalent of 2322 Octal is:						1 2 3 4

Besides the basic mathematical skills involved with using octal and binary numbers, there are some practical bookkeeping considerations that machine language programmers must learn to deal with as they develop pro-

grams. These bookkeeping matters have to do with memory usage and allocation.

As the reader who has read chapter one in this manual knows, each type of instruction used in the 8008 CPU requires one, two, or three words of memory. As a general rule, simple register to register or register to memory commands require but one memory word. Immediate type commands require two memory locations (the instruction code followed immediately by the data or operand). Jump or call instructions require three words of memory storage. One word for the instruction code and two more words for the address of the location specified by the instruction. The fact that different types of instructions require different amounts of memory is important to the programmer.

As programmers write a program it is often necessary for them to keep tabs on how many words of memory the actual operating portion of the program will require (in addition to controlling the areas in memory that will be used for data storage). One reason for maintaining a count of the number of memory words a program requires is simply to ensure that the program will fit into the available memory space.

Often a program that is a little too long to be stored in an available amount of memory when first developed can be rewritten, after some thought, to fit in the available space. Generally, the trade-off between writing compact programs versus not-so-compact routines is simply the programmer's development time. Hastily constructed programs tend to require more memory storage area because the programmer does not take the time to consider memory conserving instruction combinations.

However, even if one is not concerned about conserving the amount of memory used by a particular program, one still often needs to know how much space a group of instructions will consume in memory. This is so that one can tell where another program might be placed without interfering with a previous program.

For these reasons, programmers often find it advantageous to develop the habit of writing down the number of memory words utilized by each instruction as they write the mnemonic sequences for a routine. Additionally, it is often desirable to maintain a column showing the total number of words required for storage of a routine. An example of a work sheet with this practice being followed is illustrated here:

MEMORY WORDS THIS INSTR.	TOTAL WORDS THIS ROUTINE	MNEMONICS	COMMENTS
2	2	LAI 000	Place 000 in accumulator
2	4	LHI 001	Set Register H to 1
2	6	LLI 150	And Regis L to 150
1	7	ADM	Add the contents of memory
1	8	INL	Locations 150 & 151 on page 1
1	9	ADM	Adding second number to first
1	10	RET	End of subroutine

In the example the total number of words used in column was kept using decimal numbers. Many programmers prefer to maintain this column using octal numbers because of the direct correlation between the total number of words used, and the actual memory addresses used by the 8008.

The example just presented can be used to introduce another consideration during program development. That is memory allocation. One must distinguish between program storage areas in memory, and areas used to

hold data that is operated on by the program. Note that the sample subroutine was designed to have the computer add the contents of memory locations 150 and 151 on page 01. Thus, those two locations must be reserved for data. One must ensure that those specific memory locations are not inadvertently used for some other purpose. In a typical program, one may have many locations in memory assigned for holding or manipulating data. It is important that one maintain some sort of system of recording where one plans to store blocks of data and

MEMORY USAGE MAP

PG	LOC	MACHINE CODE		LABELS	MNEMONICS	COMMENTS
01	000			ADD,		Add no's @ 150 & 151
01	010					
01	020					
01	030					
01	040					
01	050					
01	060					
01	070					
01	100					
01	110					
01	120					
01	130					
01	140					
01	150					Number storage
01	151					Number storage
01	152					
01	153					
01	154					
01	155					
01	156					
01	157					
01	160					
01	170					
01	200					

PROGRAM DEVELOPMENT WORK SHEET

PG	LOC	MACHINE CODE		LABELS	MNEMONICS	COMMENTS
01	000	006	000	ADD,	LAI 000	Set ACC = 000
01	002	056	001		LHI 001	Set pntr PG = 1
01	004	066	150		LLI 150	Set pntr LOC = 150
01	006	207			ADM	Add 1'st no. to ACC
01	007	060			INL	Adv pntr to 2'nd no.
01	010	207			ADM	Add 2'nd no. to 1'st
01	011	007			RET	Exit subroutine

where various operating routines will reside as a program is developed. This can be readily accomplished by setting up and using memory usage maps (often commonly referred to as core maps). An example of a memory usage map being started for the subroutine just discussed is shown.

The same type of form may also be used as a program development sheet as shown here. One may observe that the form provides for memory addresses, the actual octal values of the machine codes, labels and mnemonics used by the programmer, and additional information.

Memory usage maps are extremely valuable for keeping large programs organized as they are developed, or for displaying the locations of a variety of different programs that one might desire to have residing in memory at the same time. It is suggested that the person intending to do even a moderate amount of machine language programming make up a supply of such forms (using a ditto or mimeograph machine) to have on hand.

There are some important factors about machine language programming that should be pointed out as they have considerable impact on the total efficiency and speed at which one can develop such programs and get them operating correctly. The factors relate to one simple fact. People developing machine language programs (especially beginners) are very prone to making programming mistakes! Regardless of how carefully one proceeds, it always seems that any fair sized program needs to be revised before a properly operating program is achieved. The impact that changes in a program have on the development (or redevelopment) effort vary according to where in the program such changes must be made. The reason for the seriousness of the problem is because program changes generally result in the addresses of the instructions in memory being altered. Remember, if an instruction is added, or de-

leted, then all the remaining instructions in the routine being altered must be moved to different locations! This can have multiplying effects if the instructions that are moved are referred to by other routines (such as call and jump commands) because then the addresses referred to by those types of commands must be altered too! To illustrate the situation, a change will be made to the sample program presented several pages ago. Suppose it was decided that the subroutine should place the result of the addition calculation in a word in memory before exiting the subroutine, instead of simply having the result in the accumulator. The original program, for example, could have been residing in the locations shown on the program development work sheet on the previous page. Changing the program would result in it occupying the following memory locations:

ASSEMBLER program. However, an important reason for making a copy of the text buffer on paper tape or magnetic cassette tape is because if it is ever necessary to make changes to the source listing, then the old listing can be quickly reloaded back into the computer. Changes may then be rapidly made using the Editor program, and a new clean listing obtained in a fraction of the time that might be required to erase and rewrite a large number of lines using pencil and paper.

Relatively small programs can be developed using manual methods. That is, by writing the source listings with pencil and paper. But, anyone that is planning on doing extensive program development work should obtain an Editor program in order to substantially increase their overall program development efficiency. Besides, an Editor program can be put to a lot of good uses besides just making up source listings! Such as enabling one to edit correspondence or prepare written documents that are nice and neat in a fraction of the time required by conventional methods.

Changes in source listings naturally result in changes to the machine codes (which the mnemonics simply symbolize). Even more important, the addresses associated with instructions often must be changed due to additions or deletions of words of machine code. For instance, in the example routine being used in this section, memory address PAGE 01 LOCATION 011 originally contained the code for a RET (RETURN) instruction which is 007. When the subroutine was changed by adding several more instructions (so the answer could be stored in a memory location), the RET instruction was shifted down to the address PAGE 01 LOCATION 014. The address where it formerly resided was changed to hold the code for the first part of the LLI 160 instruction which is 066. Had changes been made earlier in the routine, then many more memory locations would need to be assigned different machine codes. However, the changes caused by adding on to the sample program previously discussed are not as far reaching as the one presented on the following page. There the changes result in the addresses of subroutines referred to by other routines being changed, so that it is then necessary to go back and modify the machine codes in all of the routines that refer to the subroutine that was changed!

PAGE	LOC	MEMORY CONTENTS	MNEMONICS	COMMENTS	
	01	000	006	LAI 000	Place 000 in accumulator
	01	001	000		
	01	002	056	LHI 001	Set Reg H to 1
	01	003	001		
	01	004	066	LLI 150	Set Reg L to 150
	01	005	150		
	01	006	207	ADM	Add contents of memory
	01	007	060	INL	Locations 150 & 151
	01	010	207	ADM	Add 2nd to 1st
	01	011	066	LLI 160	Set Reg L to 160
**	01	012	160		
**	01	013	370	LMA	Save answer @ 160
**	01	014	007	RET	End of subroutine

The ** locations denote the additional memory locations required by the modified subroutine. If the programmer had already developed a routine that resided in locations 012, 013, or 014, the change would require that it be moved!

If one was using a program development work sheet, one would have had to erase the original RET instruction at the end of the routine and then written in the two new commands, and added the RET instruction at the end. The effects would not be too devastating since the change was inserted at the end of the subroutine. But, suppose a similar change was necessary at the start of a subroutine that had 50 instructions in it? The programmer would have to do a lot of erasing!

The effects of changes in program source listings was recognized early as a problem in developing programs. Because of this people developed programs called EDITORS that would enable the computer to assist people in the task of creating and manipulating source listings for programs. An EDITOR is a program that will allow a person to use a computer as a text buffer. Source listings may be entered from a keyboard or other input device and stored in the computer's memory. Information that is placed in the text buffer is kept in an organized fashion, usually by lines of text. An Editor program generally has a variety of commands available to the operator to allow the information stored in the text buffer to be manipulated. For instance, lines of information in the text buffer may be

added, deleted, moved about or inserted before other lines, and so forth. Naturally, the information in the buffer can be displayed to the operator on an output device such as a cathode ray tube (CRT) or electromechanical printing mechanism. Using this type of program, a programmer can rapidly create a source listing and modify it as necessary. When a permanent copy is desired, the contents of the text buffer may be punched on paper tape or written on a magnetic tape cassette. It turns out that the copy placed on paper tape or a cassette can often be further processed by another program to be discussed shortly which is termed an

PAGE	LOC	MEMORY CONTENTS	MNEMONICS	COMMENTS
00	000	026	OVER, LCI 100	Load reg C with 100
00	001	100		
00	002	106	CAL NEWONE	Call a new subroutine
00	003	013		
00	004	000		
00	005	106	CAL LOAD	And then another
00	006	023		
00	007	000		
00	010	104	JMP OVER	Jump back & repeat
00	011	000		
00	012	000		
00	013	056	NEWONE, LHI 000	Load reg H with zeroes
00	014	000		
00	015	066	LLI 200	And L with 200
00	016	200		
00	017	317	LBM	Fetch mem contents to B
00	020	010	INB	Increment the value in B
00	021	371	LMB	Place B back into memory
00	022	007	RET	End of subroutine

PAGE	LOC	MEMORY CONTENTS	MNEMONICS	COMMENTS
00	023	056	LOAD, LHI 003	Set H to PG 03
00	024	003		
00	025	361	LLB	Place register B into L
00	026	370	LMA	Place ACC into memory
00	027	021	DCC	Decrement value in reg C
00	030	013	RFZ	Return if C is not zero
00	031	000	HLT	Halt when C = zero

Suppose it was decided to insert a single word instruction right after the LCI 100 command in the above program. The new program would appear as shown next.

PAGE	LOC	MEMORY CONTENTS	MNEMONICS	COMMENTS
00	000	026	OVER, LCI 100	Load reg C with 100
00	001	100		
00	002	250	XRA	Clear the accumulator
* 00	003	106	CAL NEWONE	Call a new subroutine
* 00	004	** 014		
* 00	005	000		
* 00	006	106	CAL LOAD	And then another
* 00	007	** 024		
* 00	010	000		
* 00	011	104	JMP OVER	Jump back and repeat
* 00	012	000		
* 00	013	000		
* 00	014	056	NEWONE, LHI 000	Load Reg H with zeroes
* 00	015	000		
* 00	016	066	LLI 200	And L with 200
* 00	017	200		
* 00	020	317	LBM	Fetch mem contents to B
* 00	021	010	INB	Increment the value in B
* 00	022	371	LMB	Place B back into memory
* 00	023	007	RET	Exit subroutine
* 00	024	056	LOAD, LHI 003	Set H to PAGE 03
* 00	025	003		
* 00	026	361	LLB	Place reg B into L
* 00	027	370	LMA	Place ACC into memory
* 00	030	021	DCC	Decrement value in reg C
* 00	031	013	RFZ	Return if C is not zero
* 00	032	000	HLT	Halt when C is zero

of such programs is almost mandatory when programs become large because the manual method becomes highly unwieldy. A primary reason that an Editor and Assembler are so useful is because if a mistake is made in the program, one can use the relatively quick method of utilizing the Editor program to revise the source listing. Then, one may use the Assembler program to reprocess the corrected source listing and produce a new version of the machine code assigned to new addresses if appropriate.

For quite small programs, say less than 100 instructions, the use of Editor and Assembler programs are not mandatory. In fact, even if one uses these aids for small programs, one should know how to manually convert mnemonic listings to object code. This is because it may occasionally be desirable to make minor program changes (patches) without having to go through the process of using an Editor and Assembler. This is particularly true when one is DEBUGGING large programs and wants to ascertain whether a minor correction will correct a problem. The process of converting from a mnemonic listing to actual machine code is not difficult in concept. Many readers will have discerned the process from the examples already provided. However, for any who are in doubt, the process will be explained for the sake of clarity.

Suppose a person desired to produce a small program that would set the contents of all the words in PAGE 01 of memory to 000. The programmer would first develop the algorithm and write it down as a mnemonic (source) listing. Such an algorithm might appear as follows.

Note in the illustration how not only the addresses of all the instructions beyond location 002 (denoted by the *) change, but even more important, that parts of the instructions themselves (the address portion of the CAL instructions, denoted by the **) must now be altered. The essential point being made here is that if the starting address of a routine or subroutine that is referred to by any other part of the program is changed, then each and every reference to that routine must be located and the address portion corrected! This can be an extremely formidable, time consuming, tedious, and downright frustrating task if all the references must be found and corrected by manual means in a large program!

boles) form and translate them into the OBJECT code (actual machine language code) that is utilized directly by the computer. An ASSEMBLER also keeps track of assigning the proper addresses to references to routines and subroutines. This is accomplished through a process initiated by the programmer assigning LABELS to routines in the source listing. One may now see that the combination of an Editor and an Assembler program can greatly ease the task of developing machine language programs over that of the purely manual method. The use

Early computer technologists soon became disgusted with making such program corrections by hand methods after learning that it was almost impossible to develop large programs without making a few errors. They went to work on finding a method to ease the task of making such corrections and came up with a type of program called an ASSEMBLER that could utilize the computer itself to perform such exacting tasks. ASSEMBLER programs are types of programs that are able to process source listings when they have been written in mnemonic (sym-

MNEMONIC	COMMENTS
LHI 001	Set the high address register to PAGE 01.
LLI 000	Set the low address register to the first location on the page assigned by reg. H.
AGAIN, LMI 000	Load the contents of the memory location specified by registers H & L to 000.
INL	Advance register L to the next memory location (but do not change the page).
JFZ AGAIN	If the value of register L is not 000 after it has been incremented then JUMP back to the part of the program denoted by the label AGAIN and repeat the process.
HLT	If the value of register L is 000, then have the computer stop as the program is done!

To convert the source listing to machine (object) code the programmer must first decide where the program is to reside in memory. In this particular case it would certainly not be wise to place the program anywhere on PAGE 01 as the program would self-destruct! The program could safely be placed anywhere else. For the sake of demonstration it will be assumed that it is to reside on PAGE 02 starting at LOCATION 100. To convert the source listing to machine code the programmer would simply make a list of the addresses to be occupied by the program. Then the programmer would simply look up the machine code corresponding to the mnemonic for each instruction and place this number next to the address in which it will reside. (The machine code for each mnemonic used by the '8008' CPU is provided in Chapter ONE of this manual.)

Since some instructions are location dependent in that they require the actual address of referenced routines, it is often necessary to assign the machine code in two processes. The first process consist of assigning the machine codes to specific memory addresses wherever possible. When the machine code requires an address that has not yet been determined, the memory location is left blank. The second process consists of going back and filling in any blanks once the addresses of referenced routines have been determined. In the example being used for illustration, only one process is required because the address specified by the label AGAIN is defined before the label (address) is referenced by the JFZ instruction. The sample program when converted to machine language code would appear as shown next.

It is often helpful to use a utility program known as a MEMORY DUMP program to check the contents of memory locations when testing a new program. A memory dump program is a small utility program that will allow the contents of areas in memory to be displayed on an output device. Naturally, the memory dump program must reside in an area of memory outside that being used by the program being checked. By using this type of program the operator may readily verify the contents of memory locations before and after specific operations occur to see if their contents are as expected. A memory dump program is also a valuable aid in determining whether a program has been properly loaded or that a portion of a program is still intact after a program under test has gone errant.

ORIGINAL MNEMONIC	MEMORY ADDRESS	MEMORY CONTENTS	COMMENTS
LHI 001	02 100	056	Machine code for LHI mnemonic
	02 101	001	Immediate part of LHI mnemonic
LLI 000	02 102	066	Machine code for LLI mnemonic
	02 103	000	Immediate part of LLI mnemonic
AGAIN, LMI 000	02 104	076	Machine code for LMI mnemonic
			Note that the label AGAIN now defines an address of LOCATION 104 on PAGE 02
	02 105	000	Immediate part of LMI mnemonic
INL	02 106	060	Increment low address here
JFZ AGAIN	02 107	110	Machine code for JFZ mnemonic
	02 110	104	Low address portion of the CONDITIONAL JUMP instruction as defined by label AGAIN above
	02 111	002	PAGE address portion of the CONDITIONAL JUMP instruction defined by label AGAIN
HLT	02 112	377	Alternately, the code 000 or 001 could have been used here as the machine code for a HALT command

One will find that having flow charts and memory maps at hand during the DEBUGGING process is also very helpful. They serve as a refresher on where routines are supposed to be in memory and what the routines are supposed to be doing.

If minor corrections are necessary or desired, then one may often make program corrections, or PATCHES as they are commonly referred to by software people, to see if the corrections believed appropriate will work as planned. An easy way to make a PATCH to a program is to replace a CALL or JUMP instruction with a CALL to a new subroutine that contains the desired corrections (plus the original CALL or JUMP instruction if necessary). If a CALL or JUMP instruction is not available in the vicinity of the area where a correction must be made then one can replace three words of instructions with a CALL patch provided that one is very careful not to split up a multi-word instruction. If this cannot be avoided, then the remaining portion of a split-up multi-word instruction must be replaced with a NO-OPERATION instruction such as a LAA command (in an '8008' system). One must also make certain that the instructions displaced by the inserted CALL instruction are placed in the patching subroutine (provided that they are not being removed purposely). An example of several patches being made to the small example program previously discussed will be illustrated next.

Suppose, in the example just presented, that the operator decided not to clear (set to 000) all the words in PAGE 01 of memory, but rather to only clear the locations 000 to 177 (octal) on the page. The program could be modified by replacing the JFZ AGAIN instruction which started at LOCATION 107 on PAGE 02 with the command CAL 000 003 (CALL the subroutine starting at LOCATION 000 on PAGE 03 which will be the PATCH). Now at LOCATION 000 on PAGE 03 one could put:

Once the program has been put in machine language form the actual machine code may be placed in the assigned locations in memory. The programmer may then proceed to verify the algorithm's validity. For small programs such as the example just illustrated the machine code can simply be loaded into the correct memory locations using manual methods typically provided on microcomputer systems. Such small programs can then be easily checked out by stepping through the program one instruction at a time.

If the program is relatively large then a special loader program which is typically provided with an ASSEMBLER program could be used to load in the machine code.

Checking out and DEBUGGING large programs can sometimes be difficult if a few simple rules are not followed. A good rule of thumb is to first test out each subroutine independently. One may choose to STEP through a subroutine, or else to place HALT instructions at the end of each sub-

routine. Then one may verify that data was manipulated properly by a particular subroutine before going on to the next section in a program. The use of strategically located HALT instructions in a program initially being tried out is an important technique for the programmer to remember. When a HALT is encountered the user may check the contents of memory locations and examine the contents of CPU registers to determine if they contain the proper values at that point in the program. (Using the manual operator controls and indicator lamps typically provided with microcomputer development systems.) If all is well at a check point then the programmer may replace the HALT instruction with the actual instruction for that point. One may then continue checking the operation of the program after making certain that any registers that were altered by the examination procedure (typically registers H and L in an '8008' system) have been reset to the desired values if they will effect operation of the program as it continues!

MNEMONIC	MEMORY ADDRESS	MEMORY CONTENTS	COMMENTS
LAI 200	03 000 03 001	006 200	Put value 200 into the ACC Note value of 200 used because contents of register L has been incremented
CPL	03 002	276	Compare contents of the ACC with the contents of register L
JFZ AGAIN	03 003 03 004 03 005	110 104 002	If accumulator and L do not match then continue with the original program
RET	03 006	007	End of PATCH subroutine

Suppose instead of filling every word on PAGE 01 with zeroes the programmer decided to fill every other other word? A patch could be made by replacing the LMI 000

command at LOCATION 106 on PAGE 02 and again inserting a CAL 000 003 command to a patch subroutine that might appear as illustrated below.

MNEMONIC	MEMORY ADDRESS	MEMORY CONTENTS	COMMENTS
LMI 000	03 000 03 001	076 000	Keep the LMI instruction as part of the PATCH
INL	03 002	060	Keep original increment L
INL	03 003	060	And add another increment L to skip every other word
RET	03 004	007	Exit from PATCH subroutine

Finally, to illustrate a patch that splits a multi-word command, consider a hypothetical case where the programmer decided that prior to doing the clearing routine, it would be important to save the contents of register H before setting it to PAGE 01. If a three word CALL command is placed starting at LOCATION 100 on PAGE 02 in the original routine to serve as a PATCH, it may be observed that the second half of the LLI 000 instruction would cause a problem when the program returned from the patch.

(The value of 000 at LOCATION 103 on PAGE 02 in the example program would be interpreted as a HLT command by the computer when it returned from the patch subroutine.) In order to avoid this problem the programmer could place a LAA (effectively a NO-OPERATION command) at LOCATION 103 on PAGE 02 after placing the patch command CAL 000 003 instruction beginning at LOCATION 100 on PAGE 02. The actual patch subroutine might appear as shown below.

MNEMONIC	MEMORY ADDRESS	MEMORY CONTENTS	COMMENTS
LEH	03 000	345	Save register H in register E
LHI 001	03 001 03 002	056 001	Now set register H to point to PAGE 01
LLI 000	03 003 03 004	066 000	And set the low address pointer to LOCATION 000
RET	03 005	007	End of PATCH subroutine

In the balance of this manual numerous techniques for developing machine language programs will be presented and discussed. Many of the examples used will be presented as subroutines that the reader may use when developing customized programs. It is important for the new programmer to learn to think of programs in terms of routines or subroutines and then learn to combine subroutines into larger programs. This practice makes it easier for the programmer to initially develop programs. It is generally much easier to create small algorithms and then combine them, in the form of subroutines, into larger programs. Remember, subroutines are sequences of instructions that can be CALLED by other parts of a program. They are terminated by RETURN or CONDITIONAL RETURN commands. It is also wise when developing programs to leave some room in memory between subroutines so that patches can be inserted or routines lengthened without having to rearrange the contents of a large amount of memory. Finally, while speaking of subroutines, it will be pointed out that the user would be wise to keep a note book of subroutines that the individual develops in order to build up a reference library of pertinent routines. It takes time to think up and check out algorithms. It is very easy to forget just how one had solved a particular problem six months after one initially accomplished the task. Save your accrued efforts. The more routines you have to utilize, the more valuable your machine becomes. The power of the machine is all determined by WHAT YOU PUT IN ITS MEMORY!

1. First, the programmer should clearly define and write down on paper exactly what the program is to accomplish.
2. Next, flow charts to aid in the complex task of writing the mnemonic (source) listings are prepared. They should be as detailed as necessary for the programmer's level of experience and ability.
3. Memory maps should be used to distribute and keep track of program storage areas and data manipulating regions in available memory.
4. Using the flow charts and memory maps as guides, the actual source listings of the algorithms are written using the symbolic representations of the instructions. An Editor program is frequently used to good advantage at this point.
5. The mnemonic source listings are converted into the actual machine language numerical codes assigned to specific addresses in memory. An Assembler program makes this task quite easy and should be used for large programs.
6. The prepared machine code is loaded into the appropriate addresses in the computer's memory and operation of the program is verified. Often the initial check out is done using the STEP mode of operation, or by exercising individual subroutines. The judicious use of inserted HALT instructions at key locations will often be of value during the initial testing phase.
7. If the program is not performing as intended then problem areas must be isolated. Program PATCHES may be utilized to make minor corrections. If serious problems are found it may be necessary to return to step no. 3, or step no. 1! ■

Classified Ads for Individuals and Clubs

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed notice to that effect. To be considered for publication, an advertisement should be clearly non-commercial, typed double spaced on plain white paper, and include complete name and address information. These notices are free of charge and will be printed one time only on a space available basis. Insertions should be limited to 100 words or less. Notices can be accepted from individuals or bona fide computer users clubs only. We can engage in no correspondence on these and your confirmation of placement is appearance in an issue of BYTE. ■

FOR SALE: DEC tape controller model 552 for TU-55 tape drive, two units available, best offer. Edmund Wong 660-44th Av, San Francisco CA 94121 (415) 221-3492.

FOR SALE: \$500 takes all, or: MIL Mod 8 - CPU, TTY board, buffer, 2 K PROM board, 2 K RAM board, input, output, 4 K ROM/PROM/RAM board, 2 K Monitor-8 ROM, etc, socketed ICs, \$300; Digital Group cassette interface, \$20; Creed TTY: \$100; TVT-1 & KBD-1 \$120. \$500 takes all the above. Altair 8800 new kit, unassembled & untouched \$400. Richard F Schultz, 611 N Dexter, Lansing MI 48910, (517) 393-9438.

FOR SALE: Intel CPU system, Asking \$1,200 or ? Worth over \$5,000. CPU Intel 8008 8 bit parallel 8 K RAM memory 2102 type, expandable in 16 K units, 2 K PROM memory 1702 type SYSCOR dual digital tape cassettes, IBM selectric printer/keyboard model 735, RS232 ASCII 1200 baud modem with cables, 4 heavy duty power supplies. The above rack mounted in or mounted on 30" x 48" x 29" work station table. Software and BASIC available. Call Dave Trimble at (305) 273-9783 after 5 PM or write POB 20401, Orlando FL 32814.

FOR SALE: Digital cassette recorder made by National Multiplex Corp. Same one advertised in BYTE for \$149.95. Like new, less than 10 hours use. First cashier's check or money order for \$100 takes it postpaid in original carton. Charles Packer, 801 Pocahontas Pl, Hampton VA 23661, (804) 722-1364.

FOR SALE: 5262 2 K RAMS - \$2 each, misc core stacks - \$1/Kbyte, PC card edge connectors, 0.156 contact spacing: 1 side x 22 - 15¢, 2 side x 43 - \$2, 2 side x 58 - \$2.50. Have box full of each item - S Wiebking, 919D Magellan Cir, Dallas TX 75218, (214) 328-4035.

FOR SALE: Modem, Model 883A Tele-Data (Singer), 0-300 Baud, Full Duplex, RS-232-B to Bell 103 Series, synchronous or asynchronous, audio and DC loop-back tests, instruction book, like new, Herb Lyon, 2520 Vernon Dr, Greenville TX 75401, (214) 455-9225.

FOR SALE: 1 Tally 311 paper tape system. Unit consists of 1200 baud paper tape reader, 1200 baud paper tape punch, power supply, and control and interface logic for RS232 interface. All parts are mounted in a 5 foot high enclosed 19 inch rack. Maintenance manual, extender board and oiling kit are also included. I also have some assorted teletype equipment. Asking \$300 for the Tally 311 system. Call J Foley at (603) 893-1033 or write to him at 3 Salem St, Salem NH 03079.

FOR SALE: One new Processor Technology VDM-1 display board for ALTAIR (unassembled) \$150; two new TI ASCII keyboards (64 key), \$35 each; J Georgoulis, 504 Ft Drum Dr, Austin TX 78745.

FOR SALE: PORTACOM portable computer terminal, has RS-232C and built-in acoustic coupler, 10 char/sec, \$995. Steve Heffner, 106 So Main St, Pennington NJ 08534, (609) 737-2314 or 924-7086.

WANTED: Teletype Model UPE 800 punch for ASR 33. Quote price and availability in letter. Also need 6800 text editor, assembler, linker. AR Dickinson, 3520 D Pan American NE, Albuquerque NM 87107.

FOR SALE: Model 15 TTY, works good. \$95 plus shipping. Contact C Ascolillo, 892-6130 or POB 1264, Portland ME 04062.

One inch computer tape at bargain prices! Scotch 861 and 871 on 3600' reels in original boxes. List price \$38, my price \$6/reel plus postage. Jim Stitt, 311 N Marshall Rd, Middleton OH 45042.

WANTED: 8008 data book & info - will pay. Did anybody else build a C-MOD 8080 System? I'd like to hear from you. I need some info on driving core memory also. FOR SALE: I have 5 extra brand new 8008-1 CPUs at \$14 each to speed up your system or make a smart terminal. Compact core memory planes 64 X 64 (4 K X 1) at \$5. Money order please. Steve Kelley, 9506 Peach St, Oakland CA 94603.

IMSAI 8080. Assembled and tested by professional EE. 22 slot mother board and all software rights. (RAM, etc, also available) \$850 or offer (reg price \$983 + handling and tax). Also, one in box: \$600 (reg price \$651 + handling and tax). Swap for CRT terminal or other goodies. Contact: Richard Lyon, 265 W Portola Av, Los Altos CA 94022, (415) 941-8159.

WANTED: Maintenance manual for IBM Selectric Model 72. Also looking for printing robot of same. Charles Gelsinger, 4000 Camino Val, Albuquerque NM 87105.

FOR SALE: Computer systems for sale fully assembled and tested equipment for less than similar kit systems. Basic system includes 8080 computer, extended basic software, cassette interface and drive, and much more. For information contact: Glenn Barnas, 85 Strong St, Wallington NJ 07057, (201) 471-5741.

FOR SALE: Data set - Western Electric Model 205B data set and power supply. Will trade or sell for best offer over \$20. Harvey A Sugar, 4301 57th Av #1, Bladensburg MD 20710.

I need manuals, schematics, etc, for Kleinschmit Teletypewriter and reperforator, Army surplus models no. TT 178 & TT 119A/FG. Any help will be appreciated. Please write Philip Wershba, POB 1194, Goleta CA 93017 or call (805) 685-1931. If my wife answers, hang up.

FOR SALE: Sphere System 2, assembled and running. Keyboard, CRT with video and cabinet, 4 K RAM, 2 PIAs (1 port used by KBD), Serial interface including 2 ACIAs, Modem, 2 KC cassettes. Sell for my (kit) cost: \$1187. FOB. Tom Pappas, 8321 E Rose Ln, Scottsdale AZ 85253, (602) 991-9376.

FOR SALE OR ? : Altair 1 K static RAM board assembled with 256 words. Will take cash or trade for 1702As or ? Make offer. Any reasonable offer accepted. Bill Henry, POB 323, Santee CA 92071.

HELP! I got in over my head. Must sell ALTAIR 8800, 8 K dynamic memory, 88-2SIO (serial IO), ACR interface, SWTP TVT-II w/serial IO, KYBD3, custom case, computer controlled cursor, modified TV, 8 K BASIC (on tape). All equipment assembled and operating. \$1500 or any reasonable offer. Donald Bleeden, 661 S Cloverdale #3, Los Angeles CA 90036, (213) 936-1260.

CORE MEMORY PLANES FOR SALE. Two 4 K X 16, 1 µsec core memory with matrix diodes. Brand new surplus in original packing with QC and warranty stickers intact. Documentation included. Best offer over \$50 each plus postage. Ted Becker, 317 158th St SE, Bothell WA 98011 (206) 743-1321.

IMSAI 8080 For Sale: New completely assembled 8080 system. Includes, 12 K RAM, two serial IO ports, cassette recorder interface, eight levels of priority interrupts, and clock board. Only \$2,450 or highest bid. Also ASR-33 Teletype terminal in super condition. just \$875. John Whitney, 2405 Haisley Dr, Ann Arbor MI 48103 (313) 662-2530.

FOR SALE: Burroughs C-3660 programmable calculator with memory and magnetic card reader. Good condition. Best offer over \$500. If no reply to your letter, unit has been sold. Tom French, 909 Society Av, Albany GA 31701.

FOR SALE: System 21 Data management station (VIATRON) as advertised in BYTE March 1976 page 87. This unit works very good and is new status. All modes of operation are in good condition. Complete with instruction manual and schematics. \$400. You pay shipping; if no reply to your letter, unit has been sold. Tom French, 909 Society Av, Albany GA 31701.

FOR SALE: Honeywell high speed line printer (132 characters per line) in excellent condition, complete with Honeywell interface. \$800. Call George Coy, RFD 1, Milton VT 05468, (802) 893-2154.

FOR SALE: MITS Altair 6806 microprocessor kit. Complete documentation; parts still in original packages. First check for \$400 to clear takes it. Wife threatening divorce! Paul Goyer, 1351 W Touhy, Chicago IL 60626. (312) 465-4128 eves.

Before sending your classified ad to BYTE, read it over. Did you include your name, address, phone number (with area code) in the text of the ad? BYTE has received several ads with incomplete phone numbers or missing addresses.

personal discussion with another ham who is familiar with a particular type or band of equipment or program might save many people from costly errors.

Such a net would probably draw a huge following of listeners in addition to the actual participants; remember the first RTTY net on 40? I would be happy to help organize and control such a net; although my actual microprocessor experience is limited at this time, perhaps we could get something going. I will be on 3.820 MHz, plus or minus QRM on Monday evenings at 7 PM CDST. (00:00 GMT, Tuesday.)

If there are any suggestions as to other times, dates, modes, bands, etc, please drop me a line.

Melvon G Hart W0IBZ
936 Dontaos Dr
St Louis MO 63131

Let us know what happens.

KIM ON, NOW

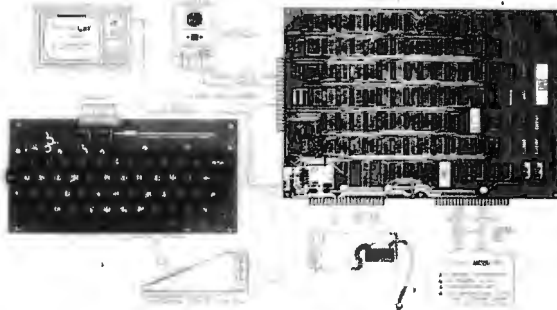
I thought I'd drop you a line to tell you of my experiences with the KIM-1 I've had for 10 days now. I ordered it six weeks ago. It took a little over a month to come — not bad. Everything was pretty much the way BYTE described as far as hardware and software.

I didn't have any power supplies here at home so I read over the *User Manual* the first night. I took the KIM-1 to school where I teach and fired it up with a 5 V supply. I was happy to see the drain only about 0.6 to 0.7 A. The KIM-1 people call for a 1.2 A supply. I have some cute little 5 V 1 A supplies used for the Indiana Instruments Inc logic system so I finally tried one of them and it works fine so far.

The display is very bright and clear. The

keyboard has a nice touch too. Well, I tried the first program example in the *User Manual* which is to add two numbers. I did their example of 2 plus 3 and got 5. I was elated. Then I tried their second example of adding FF to FF to get FE and I got 54. [Hexadecimal notation is used in this letter's arithmetic examples.] Something screwy. So I took an easier pair of numbers, namely 5 plus 5. I was looking for an answer of A of course. But I got 10. That's a funny symptom, I thought. So I tried 6 plus 6. You guessed it. I got 12. I ran around showing all

BURKESHIRE SYSTEMS PRESENTS



EPIC 2

DISCOVER THE EPIC 2: A COMPLETE ONE-BOARD MICRO-COMPUTER SYSTEM AT AN AFFORDABLE PRICE.

EPIC 2 features:

- Easy to use teletype-style fully encoded keyboard.
- Programs loaded and stored via your standard audio tape recorder
- Video interface — displays 16 lines of alpha numeric characters on your TV set or optional video monitor.
- Unified extendable bus.
- 2 K RAM memory.
- 256 byte PROM bootstrap.
- 2 MHz 8080 micro-computer.
- 16 general purpose I/O lines.
- Standard programs included: Monitor, Text Editor, Game of Life, Blackjack Game.
- Memory expansion and our optional Vector Graphics Software Package allows you to create, display, and store graphics.
- Many more options to come!

Please send me _____ EPIC 2 (s). Enclosed is \$775 (California residents include 6% sales tax) for each EPIC 2 ordered. I am allowing 90 days for delivery.

Name _____

Address _____ City _____ State _____ Zip _____

For additional information please write:

BURKESHIRE SYSTEMS P.O. Box 512 Mountain View, California 94040

SOLID STATES MUSIC PRODUCTS

Static Memory Boards (4Kx8)

- MB-1 Mk-8 Board
- Kit\$103 PC Board..... \$22
- MB-2 Altair 8800 board with on board address & wait switching.
- Kit(2102's 1us)\$112
- Kit(91L02A's) \$139 PC Board \$25

Errom Board

- MB-3 EROM Board with on board address and wait switching. 2Kx8 may be expanded to 4Kx8
- 2K Kit \$145 4K Kit\$225
- Kit Less EROMs \$65

I/O Boards

- I/O-1 8 bit parallel in & out ports Common address decoding jumper selected, Altair 8800 Plug compatible.
- Kit\$42 PC Board..... \$25
- I/O-2 I/O for 8800, 2 ports committed pads for 3 more, other pads for EROMs, UART, etc.
- Kit ...\$47.50 PC Board \$25

Misc.

- Altair compatible mother board ... \$45
- 32x32 Video Board\$35
- Altair extended board (less con.) ... \$8

2102's	1usec	.65usec	.5usec
ea.	\$1.95	\$2.25	\$2.50
32	\$59	\$68	\$76

8212	\$5.00	1702A*	\$10.00
MM5320	\$5.95	2101	\$ 4.50
AY5-1013 UART	\$7.95	2111-1	\$ 4.50
	\$7.95	2112	\$ 4.50
74200	\$5.90	8223	\$ 3.00
74L200	\$5.90	91L02A	\$ 2.55
MH0025	\$2.50	32 ea.	\$ 2.40
MH0026	\$2.95	*Programing send	
8T97	\$2.10	Hex List	\$ 5.00

Please send for complete listing of IC's and Xistors at competitive prices.

MIKOS

419 Portofino Dr.
San Carlos, Calif. 94070

Check or money order only. Calif. residents 6% tax. All orders postpaid in US. All devices tested prior to sale. Money back 30 day Guarantee. \$10 min. order. Prices subject to change without notice.

the micro experts this funny result. I knew something was wrong but what?

After a half hour of diddling with the keys I tried my 5 plus 5 again and this time I got A as I should. I added FF to FF and got FE as I should. Whatever was wrong fixed itself. I had to shut down for the night.

The next day I went through the same agony. The processor acted "queer" for about a half an hour then magically was OK. Ah hah. I've got a thermal problem I thought. So I let it cool down for a few minutes by turning it off. When I powered up the problem was back. Now to wait until it gets to that magic temperature where it fixes itself. Well an hour went by and it didn't fix itself. So I started poking the keys again and bingo — it started working correctly. I was in a state of confusion — had visions of sending the thing back after only two days.

I started reading the *Programming Manual*. Then I discovered the processor had a decimal mode and I knew exactly what had happened. When I turn my processor on, the flip flop for decimal mode comes up in that mode rather than the binary mode. I quickly included a CLD instruction at the front of my program and now it clears the decimal mode and does the math in the binary.

Next I tried their program to make notes on a speaker controlled by seven switches. It worked the first time — a beautiful sound.

Then I tried to make a tape recording of the addition program. The recording process went fine. Then I tried to play back. When you have success the display relights. I couldn't get that display to relight to save me.

Back to the manuals. I read and read. Then I recalled 12 V. I had forgotten the 12 V supply in my hurry to get results. You need 12 V to run the 565 PPL for playback. The next day I got hold of a nice little 12 V supply and checked it, for 50 mA load conditions. It was 11.6 V and had about 5 mV ripple. Just right. My problems are over. I hooked it up and went through the record-playback procedure. Nothing. I did it about a dozen times checking and rechecking the procedure. I must have reread that section of the *User Manual* a dozen times. But no luck. Nuts.

Well the processor works anyway and that's the main thing. I tried to write my first program — to multiply two 8 bit numbers and get a 16 bit answer. I learned what "immediate" means the hard way. Also I learned what "relative branch" means the hard way. The single step mode was real handy for debugging my program. I even figured out how I could get the answer to

appear where the 4 address digits normally appear. Hurray!

Meanwhile I tried for several days to get a fix on my recording problem. I tried my Roberts \$200 reel job at 7½ inches per second with no luck. I began checking the waveshapes of the tones because they sounded low frequency to me — not the 3700 and 2400 Hz they were to be. I managed to sync scope just right to measure the tones and they were on the money. It must be the playback circuit. I traced the signal all through it and it was just as it was supposed to be right up to the PB7 port where it enters the interface chip.

I began reading some more. This time I looked at and studied the software listing for the KIM Monitor since I had mastered a few of the op codes, and the Texas Instrument Learning Center lectures told us last year it's "like reading a newspaper." Well it's not like that yet for me but it's coming. I was amazed at all the subroutines. But the great thing is the liberal use of comments. I hope that future contributors of programs to BYTE are as definitive. Then it hit me.

Apparently, to initialize the processor you must have to hit the ST stop key. This is an interrupt command and for it to work properly you must have a special vector set up at location 17FA and 17FB, namely 1C00. I had read this before and knew you needed it for the single step mode but I never connected it to the ST key nor the ST key to the initialization of the processor.

Well I now tried my recording procedures after entering 1C00 at the vector location, and after hitting the ST key a few times and guess what — she worked. That was today and it made my day I'll tell you. I had visions of having to write programs via the keyboard forever and that's a long time. Now the tape recorder part works — even on my \$25 cassette. In fact my cassette works better than the big reel job. I suspect old tape on the reel job. I tried recording from 0000 to 01C0, that's 432 bytes; and it takes about a minute. Playback was perfect twice in a row. The block record has an ID number and you can look for it or not. Also you can relocate the data anywhere in proper memory. Real neat.

Someday I hope to get a TTY or TVT going with this system. But for now, I'm looking for an application for demonstration in classroom. BYTE has been very good to date. Keep up the good work.

George L. Thompson
Associate Professor, EE
Rochester Institute of Technology
One Lomb Memorial Dr
Rochester NY 14623

I had visions of having to write programs via the keyboard forever and that's a long time.

When we are as dependent on the computer as we are now on our electrical distribution network, that is when we will have to worry about being under the control of a computer dictatorship.

CANADIAN SUPPLIERS

In reply to your reader Atwood's letter in the June issue of BYTE, here are three young companies, all with what I believe to be good products:

- (1) 6800 based mini:
Mini-Peripherals Inc
2615 Blackwell St, Unit 112
Ottawa CANADA
- (2) 6800 based intelligent dual floppy disk:
Dynalogic Corp
141 Bentley Av
Ottawa CANADA
- (3) CRT terminals to 19,200 Baud:
Cybernex Ltd
2595 Blackwell St, Unit 111
Ottawa CANADA

In addition, Mini-Peripherals manufactures a dual floppy DMA add-on for 6800 systems.

Norman J McKay, P Eng
Mini-Peripherals Inc
2615 Blackwell St, Unit 112
Ottawa CANADA

ON POWER TO COMPUTERS

"Could a Computer Take Over?" and Mr Carrick have approached the problem of a computer dictatorship from the narrow viewpoint of an engineer. Since when has intelligence been the criterion in determining who will rule? Power has always, and will always, go to those who control the institutions upon which a civilization is dependent. While the argument goes on whether a computer will ever have enough information capacity or reliability to become a world dictator we are using computers to do more and more jobs; ie: traffic control, financial transactions, production line control, etc. When we are as dependent on the computer as we are now on our electrical distribution network, that is when we will have to worry about being under the control of a computer dictatorship. No central master computer will be needed, just a total dependence on a computer network. How will the computer rule? That depends on us.

Gordon R Morrison
33 Maple St
Glastonbury CT 06033

Subscribers: We Need That Label

When corresponding with BYTE about subscriptions, please enclose your LATEST mailing label OR copy exactly the code at the top of the label and include zip code. ■

MERRIMAC PROCESSOR SYSTEMS ANNOUNCES THE CRTV

A complete low cost Teletype® Model 33 KSR replacement using a standard TV as the monitor. Complete, whether purchased as a kit or assembled.

- (1) Full cursor controls - clear, home, cursor up, down, left, right.
- (2) Infinite scrolling.
- (3) Selectable data rate to 600 baud.
- (4) Half/full duplex; local mode.
- (5) "Break" function allows entering many debug routines.
- (6) Model 33 keyboard with added cursor functions, clear, and home keys.
- (7) Housed in attractive enclosure.

Completely assembled and tested . . . **\$495.00**
Complete kit **\$355.00**

For Complete Information write —

Merrimac Processor Systems
P.O. Box 76
Chester, New Hampshire 03036

ALTAIR 8800 OWNERS

We recently received the following letter:

RUN

APRIL 26, 1976

GENTLEMEN:

I JUST WANTED TO TELL YOU THAT I THINK YOUR CLOCK FIX-IT KIT IS REALLY GREAT! I WAS HAVING TROUBLE RUNNING BASIC AND AFTER INSTALLING YOUR KIT FOUR OF MY BITS BOARDS THAT WEREN'T RUNNING CAME BACK TO LIFE AND NOW ARE HELPING ME TO WRITE THIS LETTER ON THE COMPUTER. ENCLOSED IS ANOTHER ORDER FOR A CLOCK KIT. THIS IS FOR THE SECOND ALTAIR THAT I'M NOW IN THE PROCESS OF BUILDING.
AGAIN MANY THANKS FOR SUCH A FINE PRODUCT.

SINCERELY
LUCY L. SMITH

OK:

How well does your Altair run?
A Clock Fix Kit is only \$15 postpaid.

PARASITIC ENGINEERING

PO BOX 6314

ALBANY CA 94706

What's New?

A Lab Computer, Anyone?

Jonathan A Titus, president of Tychon Inc, Blacksburg VA 24060, sent along this picture of the MiniMicro Designer his firm makes and E&L Instruments markets. The



computer is an 8080 processor, and the context in which this system is most appropriate is the experimental laboratory of the electronic engineer, instrumentation engineer, research scientist or educator. The system comes with three modules of programmed learning text and experiments, plus the over 500 pages of text in E&L Instruments' *Bugbook III*.

The single most important feature of this design is the E&L Instruments solderless breadboard section which can be used to breadboard peripherals using the various 8080 system lines which are routed directly to this section of the board. In a laboratory context, this provides a quick and simple way to try out design ideas and test them out in the context of a programmable computer system. This is the type of product one might expect to see in every well equipped teaching laboratory within the next few years.

For detailed information on purchase and delivery, contact E&L Instruments, 61 First St, Derby CT 06418. ■

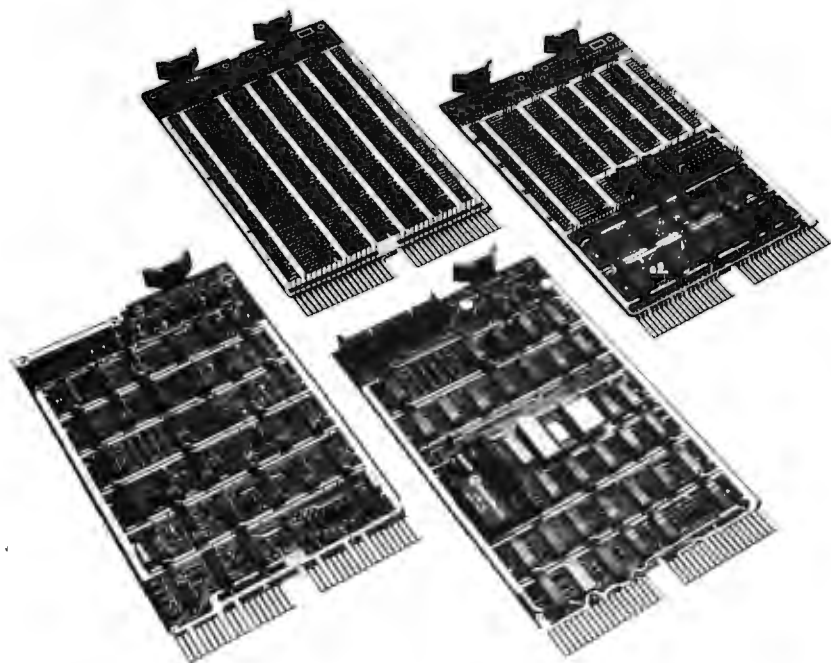
Attention: LSI-11 Hackers

A firm called MDB Systems, 1995 N Batavia St, Orange CA 92665, has announced a line of products compatible with the Digital Equipment Corporation's LSI-11 line. These items include dynamic programmable memory in 4, 8, 12 and 16 K sizes, EROM and PROM modules, a general pur-

pose interface, universal wire wrap modules, peripheral controllers for printers, paper tape and card readers. The firm will also sell you the LSI-11 central processor itself at \$900. Write for a catalog and price list if you are interested in assembling a custom LSI-11 system. ■

Tutorial Computer System

The UC 1800 processor is a new product manufactured by Infinite Incorporated, PO Box 906, Dept CPNR, 151 Center St, Cape Canaveral FL 32920. This neatly packaged processor uses the RCA COSMAC 1802 microprocessor, and comes completely assembled with training aids and a user manual of approximately 200 pages. According to the company, this package is designed to take the individual from computer basics to hands on practice and



experimentation with a live computer. The package price of the unit depicted is \$495. An OEM version of the central processor board for this product is available for industrial and dedicated application, at \$179 in quantities of one.

Specifications of the UC 1800 include a hexadecimal keyboard and display, front panel controls of interrupt, direct memory access and an IO flag, a built in 256 byte programmable memory which is expandable to 64 K bytes with external circuitry and a 120 VAC input power supply. ■

Want to Get Unusual Tools?

A mail order firm called Jensen Tools and Alloys provides an excellent source of unusual specialty tools for the electronics hardware world. No literature shelf is complete without the latest Jensen catalog, which includes tools ranging from vacuum desoldering systems to a zoo of pliers in every shape and form to complete tool kits and power tools from precision lathes to hand held drills. Jensen also carries the OK line of wire wrap tools and carries the Claus "NO NIK" wire strippers so useful in preparing wire wrap wire. Ask for the latest 128 page Jensen catalog by writing them at 4117 No 44th St, Phoenix AZ 85018. ■

CAPTAIN VIDEO



Ideal for communicating with your microprocessor!

RS-232 interface x 32 characters \$400
 TV/TTY kbd./display (16 lines x 64 characters) \$500
 Keyboard/CRT Monitor (24 lines x 80 characters) \$700

----- ORDER FORM -----

Ship To: _____

Subtotal \$ _____


Calif. residents add sales tax _____

Please find a check enclosed in the amount of total order: \$ _____

Bill my BankAmericard exp. _____

Mastercharge exp. _____

My signature as purchaser _____ date _____


MICON
 INDUSTRIES
 Micon Industries
 252 Oak Street
 Oakland, CA 94607
 (415) 763-6033



INTEGRATED COMPUTER SYSTEMS, INC.

MICROCOMPUTER SPECIALIST WANTED

Integrated Computer Systems is a young, rapidly growing company, highly active in Microcomputer education and consulting services in the U.S., Europe and Japan (ICS has already taught Microcomputer courses to more than 4,000 engineers and managers.) We have an immediate requirement for personnel with proven capabilities as:

- Microcomputer Instructors for both Domestic and International courses
- Microcomputer Course Writers
- Microcomputer Hardware Engineers

● Microcomputer Programmers
 The applicants should have direct Microcomputer experience, both in assembly language programming and digital logic design. Teaching experience is obviously also desirable with multilingual (especially French and German) teaching ability preferred.

Salary range is 14-27K commensurate with experience and responsibilities. Mail resume to: Integrated Computer Systems, Inc. P.O. Box 2368, Culver City, Ca 90230.

QUALITY USED TEST EQUIPMT

TEKTRONIX
 Oscilloscopes

Plug-ins
 Sig. gen.s

Carts

HEWLETT-PACKARD

Meters

Probes

Carts

For a complete list of available items, write:

PTI Div. 12

P.O. Box 8699

White Bear Lake, Mn 55110

N. Y.'s Newest Store for Micro and Mini Computer Hardware and Software.

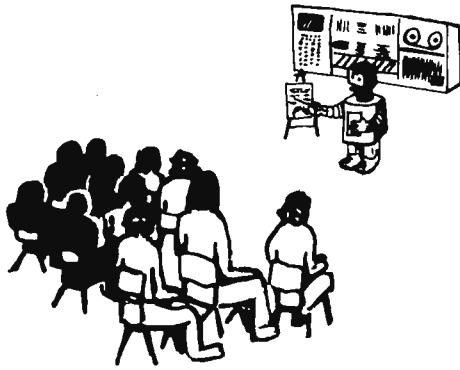
We will be featuring a large selection of: IMSAI (kits and assembled) Processor Technology's full line. Also a selection of choice modules from Cromemco, Solid State Music and others. Plus: full line of TTL logic, discrete components, memory and MPU chips.

Audio Design Electronics

487 Broadway Suite 512

New York NY 10013

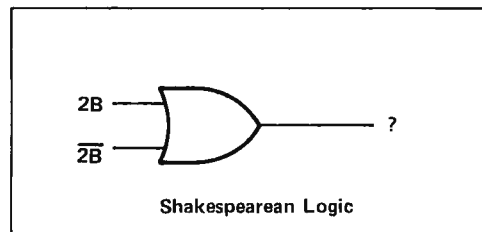
212-226-2038



Clubs and Newsletters

KC Thruput

The Kansas City computer enthusiasts have gotten together to form the "Computer Network of Kansas City" as a result of a BYTE mention in these columns [*May 1976, page 57*]. The first meeting in May led to a fairly quick organization and a three page newsletter. The meeting algorithm was set as "the second Sunday of each month at 7 PM" and meetings for the time being are held at the Midwest Research Institute Library on Volker Blvd near the University of Kansas City. For further information contact Earl Day, president, at 492-9315, Harold Schwartz, vice president, at 371-2616, or George Scheil, secretary, at 363-0814. The newsletter comments by Earl Day closed on the following interesting note



South Florida Activities

The latest issue of the newsletter *I/O* put out by the South Florida Computer Group, dated June 1976, contained eight pages numbered in binary.

The group has split into two chapters based on geography. The Miami chapter meetings are held at 7:30 PM on the first Monday of each month, currently (June 1976) at the Paps Institute, 1155 NW 14th St, Miami FL. Contact Jim Whitmore, 685-1218, or John Lynn, 271-2805, for more information. The Fort Lauderdale chapter meetings are held at 7:30 PM on the

second Thursday of each month at the Florida Power and Light Building, 501 So Andrews Av, Fort Lauderdale FL. Contact Terry Williamson, 752-8395, Lee Hinman, 974-1457, or George Fugate, 522-5358.

Austin TX Club?

RD McCoy, 3501-B Clawson Rd, Austin TX 78704, writes that he is interested in the prospect of meeting other microcomputer enthusiasts in the Austin TX area. He can be reached by phone at 443-0971.

People's Computer Company – Cromemco Contest

Cromemco, maker of the TV Dazzler, and People's Computer Company, POB 310, Menlo Park CA 94025, have gotten together to sponsor a contest with prizes as follows:

First prize: \$500 certificate for Cromemco hardware.

Second prize: \$250 certificate for Cromemco hardware.

Third prizes: 10 certificates for \$35 in Cromemco hardware.

Honorable Mentions: 16 copies of *What To Do After You Hit Return*.

The object of the contest is to develop a program resulting in a new and interesting display using the Cromemco TV Dazzler. Write to PCC for rules, but the deadline is September 30, so you'll have to hurry.

El Paso Computer Group

There is a club now in El Paso TX, as evidenced by the version 0 revision 1 issue of the *El Paso Computer Group Quasi Annual Newsletter*. The club address is El Paso Computer Group, 9716 Saigon Dr, El Paso TX 79925. Current membership is six according to the one page newsletter sheet dated May 1976. A major activity has been designing several club-produced PC boards for the Altair 8800: a vectored interrupt board with real time clock, an extender board for the 100 pin connector, and a conversion board to allow an SWTPC TVTII to talk directly to a Processor Technology 3P+S board. Prices on boards are \$20, \$12 and \$2 respectively, and are available from the club.

Microcomputer Society of Florida

Sandy Meltzer, president of the Microcomputer Society of Florida, sent a progress report on his group's activities. The Tampa Bay chapter meets at Marsh Data Systems on the first and third Saturdays of each month at 2 PM, and presently has over 100 members. The Naples chapter is being organized

FREE!

New CAMBION XQ catalog. Scores of famous CAMBION electronic components for hobbyist or experimenter in small packs or individually at modest cost. Parts available at retail electronic shops or by mail from factory. Coils - chokes - IC sockets - tools - patch cords, and lots more. Circle Readers Service number now.

CAMBION

Cambridge Thermionic Corp.
445 Concord Avenue,
Cambridge, Massachusetts 02138
Phone: 617-491-5400

WANTED

Microprocessors Kit or Assembled
ALTAIR • SPHERE • MOTOROLA
INTEL & OTHERS
Altairs, immediate delivery.

Memories, I/O boards, chassis, power supplies, modems, peripherals, specials, prom programmers, ADCS, DACS, GPIOS, RS232 faces, and any associated software. Ask about our complete Altair system. Available now.

FORWARD INFORMATION AND/OR
INQUIRIES TO:
**AMERICAN USED MICROPROCESSORS,
EQUIPMENT & SUPPLY CORP.**
AT THE CHICAGOLAND AIRPORT
BOX 515
PRAIRIEVIEW, ILLINOIS 60069
TELEPHONE (312) 634-0076

For complete monthly catalog of items available and in stock Enclose \$10. for six issues. Many new unused items. Amazing values for everyone. Tested and guaranteed.

SURPLUS μ P & TERMINAL

- Not a KIT! Ready to use
- 2 ea; 400 line (80 char/line) Cassette Tape decks
- Kbd, μ P, 9" dia Video Display, & Power Supply.
- Modular 'Card' Construction
- MOS-LSI, ASCII, RS-232C
- NEW or like-New, GUARANTEED
- Size: 20"W x 24"L x 33"D
- Everything INCLUDED

This Data-Management-Station (DMS) is a business-oriented, highly versatile, machine especially suited for mailing, addressing, filing, labeling, data storage, and other time & personnel consuming tasks.

DMS SPECIAL PRICE: \$649.00

CFR
Associates 603-382-5179
Box F Newton 9:00-5:00 EST
NH 03858

We also offer many **ELECTRONIC MUSIC SYNTHESIS** kits, plans, parts, and more...

mail us a S.A.S.E. or POSTAGE for FREE INFORMATION

Lear Siegler ADM-3K

**24 X 80 terminal
Kit \$849.95**

**IMSAI 8080 SYSTEM
IN STOCK AT
DISCOUNT PRICES**

Call or write for details & catalog

SYNCHRO-SOUND ENTERPRISES
193-25 JAMAICA AVE.
HOLLIS, N. Y. 11432
(212) 468-7067 (9 AM - 5 PM)

Computer Mart of New York Inc.

Authorized dealer for:
Sphere • IMSAI
Processor Technology
SWTPC 6800 & CT 1024

Featuring the best in microcomputers and books. Competitive prices on systems. Take a byte out of the new Apple-1 computer.

Friendly Service Advice
Problem Solving

Open Monday through Saturday
10-6, Thursday until 9:30

314 Fifth Avenue
New York NY 10001
212 279-1048



Pictured above is the new OP-80A High Speed Paper Tape Reader from OAE. This unit has no moving parts, will read punched tape as fast as you can pull it through (0-5,000 c.p.s.), and costs **only \$74.50 KIT, \$95.00 ASSEMBLED & TESTED.** It includes a precision optical sensor array, high speed data buffers, and all required handshake logic to interface with any μ P parallel I/O port.

To order, send check or money order (include \$2.50 shipping/handling) to Oliver Audio Engineering, 7330 Laurel Canyon Blvd., No. Hollywood, CA 91605, or call our 24 hr. M/C-B/A order line: (213) 874-6463.

by Dr George Haller, 1500 Galleon Dr, Naples FL 33940. The group puts out a newsletter and can be reached by contacting Sandy at Marsh Data Systems, 5405B Southern Comfort Blvd, Tampa FL 33614, or by attending one of the Tampa Bay chapter's meetings.

Texas A&M University Microcomputer Club

Robert R Wier, secretary, sent word of this club's formation. Meetings started in April of this year, and the club has grown to about 55 members with varying backgrounds: from freshman students to senior faculty, from systems analysts to art majors. The meetings are held each Wednesday during the semester at 8 PM in room 333B of the Zachry Engineering Center. Dr Charles Adams is coordinating the activity and may be reached in the evenings at (713) 823-0877. Meetings usually consist of about an hour of general discussion followed by a program or meetings of several smaller groups with specific interests. Currently there are interest groups in the implementation of BASIC, implementation of APL, computer games, and specific user groups for various types of microprocessors.

Robert extends his invitation to anyone passing through College Station TX to stop by at a meeting. The campus is located 100 miles northwest of Houston on state Highway 6. The mailing address is: Texas A&M University Microcomputer Club, POB M-9, College Station TX 77844.

The Carolina Computer and Radio Amateur Association

BYTE received issues 1 and 2 of the "Bits and Dits" newsletter of the CCARAA. This group mixes amateur radio with computing in the form of automated repeater work, talks on logic design and programming, AMSAT operating activities, and applications of computers to amateur radio station activities. For further information contact Boyd S Miner, K4KEP, at the Association's mailing address, POB 341, Laurens SC 29360.

Get RICHC Quick?

Roger C Garrett, 16 Grinnell St, Jamestown RI 02835, wrote a letter noting the formation of the Rhode Island Computer Hobbyist Club. He and members of the club would like to hear from other computer enthusiasts in Rhode Island who would be interested in the prospect. At present, two members are assembling Altairs and two are building home brews based on the MOS

Technology 6502. Roger also reports that he is working on a FORTRAN cross assembler for the 6502 and would like to hear from other 6502 hackers for the purposes of exchanging ideas and programs.

MAPLE Leaves

For small systems enthusiasts interested in the APL language, there is an active group of people beginning the "Microprocessor APL Enthusiasts." They have started a publication, the *MAPLE Leaves* newsletter. On the principle that "n heads are better than 1" when it comes to creating APL software for home computers, the group is interested in promoting APL interpreters in a manner similar to the way Tiny BASIC has evolved.

To find out what's up, contact John Sikorski, president of MAPLE, or Ruth Low, newsletter editor of *MAPLE Leaves*, for information on this grass roots APL awareness project

MAPLE Leaves
Box 574 NUMS
Chicago IL 60611

An Item of Interest

All 6800 owners who have purchased the SWTPC 6800 computer were treated to an excellent issue number 1 (June 1976) of the newsletter which is being sent free to all SWTPC 6800 Computer System owners, as well as to computer hobbyist clubs and publications. (If your club or publication did not get a copy, then it's probably because SWTPC did not know about you, so write and ask for one. Southwest Technical Products is located at 219 W Rhapsody, San Antonio TX 78216. Ask for the *SWTPC 6800 Computer Newsletter*, number 1, June 1976). The newsletter contains some very interesting information, which is summarized here:

- Some editorial copy concerning "SWTPC's attitude on software." Basically, the attitude is one of "bundling" systems software into the price of the computer itself and charging the amount necessary for covering expenses of program duplication and handling. The explicit policy is stated:
"None of the programs available from SWTPC are proprietary. Where available, you may either purchase a tape and instruction manual from us or copy them from a friend. We don't care."
- An announcement of the SWTPC 6800 Editor/Assembler. This is being made

Olympia

MICROCOMPUTER APPLICATIONS — IMSAI DEALER —

In stock now:

°The INTELLIGENT floppy from IMSAI containing an 8080A with its own on-board firmware for distributed processing power

°IMSAI 8080 microcomputers in kit form or custom-assembled to your specifications

°The IMSAI boards and accessories which you need to complete your system

A dozen years experience in all aspects of the computer industry. Experience to meet your digital needs. Call or write for more information:

6009-B 13th Way S.E.
Olympia, Washington 98503
206/456-3924

SAN FRANCISCO

Action Audio Electronics
Westlake Shopping Center
(415) 756-7440

1. Largest repair facility for computer kits in Northern California.

2. Easy access from all major freeways and BART, 4 blocks from Junipero Serra, Skyline, Highway 1, US 280 and Daly City BART.

3. We sell IMSAI, Cromemco, Blast-Master, Byte, etc.

4. Stop by for a free copy of "8800 BUS"; a current list of all 8800 bus compatible boards.

LEARN µCOMPUTER SOFTWARE

\$49.95 BUYS A
COMPLETE COURSE IN
PROGRAMMING FOR THE
SMALL SYSTEM ENGINEER

- PRINCIPLES
- TECHNIQUES
- EXAMPLES
- APPLICATIONS

LOGICAL
SERVICES INCORPORATED

Creative Computer Division
711 Stierlin Road
Mtn View, CA 94043
(415) 965-8365

FREE BROCHURE

WHAT'S A CREED?

A Creed is a compact, modern and reliable English Teletype. You've probably seen references to Creeds and wondered what they are. We don't advertise much, our business is mostly by word of mouth.

Each Creed includes: keyboard, printer, punch, TTL interface parts kit and manual for \$150, F O B Naperville. The Creed is a Baudot TTY, but Baudot to/from ASCII can be done in hardware or software giving you a heavy duty TTY for a fraction of the cost of a light duty 8 bit TTY.

Interested? Send for free data sheet and price sheet or order a 28 page manual on the Creed and hardware and software interfaces for \$1 (credit given on Creed purchase). Lots of other unique things available, e.g. an optical scanner for \$15 postpaid.

Wilcox Enterprises
25W178-39th Street
Naperville, IL 60540
Phone 312-420-8601

HARD COPY A71
~~\$425~~ SOFT PRICE



mp-40

- Completely Assembled
- 64 Character ASCII
- 40 Column Capacity
- 5x7 Impact Dot Matrix
- Ordinary Roll Paper
- Parallel Interface
- Includes Power Supply and Case Work

Utah Residents add 5% Sales Tax
Allow 60 days for delivery

mpi

P. O. Box 22101
Salt Lake City / Ut. 84112

STANDARDIZE!

SPECIAL OFFER...
HIROSE 20 PIN CABLE +
CHASSIS MOUNT CON-
NECTOR

(solder) ONLY \$14.95
REGULARLY \$17.00

Rugged, Reliable, widely
used in video applications.
Qty. discount, no C.O.D.'s,
Details, Specs:

JOHN ANTHONY TELEVISION
DINGMANS FERRY
PA. 18328

available for \$14.95 in either paper tape or "Kansas City Standard" [see *March 1976 BYTE*] magnetic tape form. The assembler requires 8 K bytes of memory to run, and is assembled to work with SWTPC systems. There is no memory purchase requirement. Specify which form you would like (paper tape or magnetic tape cassette) when you order.

- A list of game programs and prices for a commented assembled source listing. All the games except one entitled "Space Voyage" will run in a 2 K SWTPC 6800 (or any 6800 system with MIKBUG and 2 K of memory). Package deals for three different selections from the list are also available.
- Documentation of known SWTPC 6800 problems and bugs. The statement in the newsletter is: "We are happy to say that after having delivered the SWTPC 6800 system for over five months now, we have run across no problems on the system. There are, however, some mistakes in the MP-M and MP-MX instructions on some of the earlier kits" [*The statement*

of the first sentence is confirmed by what we've heard to date at BYTE. For example, Sol Libes of the ACGN/ in conversation May 1 pointed out that of several systems purchased by members in NJ, nearly all were assembled in short order and worked on initial power up.]

- Documentation of additional clock rates for use with the MP-S serial interface board of the system. With this documentation, it is possible to run the terminal with your SWTPC 6800 at rates of up to 9600 baud, since nearly every standard data rate is available from the clock circuit in the system.
- Detailed technical information abounds: modifications of a previously published Tic Tac Toe game, modifying the 6800 for 220 VAC operation, modifying the CT-1024 for European television (625 line, 50 Hz) operation, the complete documentation of the BLKJAK-1 program, used to play the well known game at your terminal (no assembly listing, just a 5F8 byte long hexadecimal dump), a memory dump program (assembly listing), etc.
- The crowning glory of the whole newsletter is the complete documentation of the "Line Numbering Editor and Microbasic Interpreter." These are provided as complete user documentation *plus fully assembled source listings*. The editor program was written by Robert H Uiterwyk, and uses the MIKBUG ROM for utility subroutines. The assembly of the editor results in a module requiring hexadecimal 524 bytes, and is assembled with an origin at location 0020 in memory address space. The Microbasic is written by Robert H Uiterwyk and Bill Turner. It features the following language features:

Line numbers are 16 bit integers (1 to 65535)

Line editing is built into the system along with the following commands:

NEW clears the program space
 LIST lists varying amounts of the BASIC program
 SIZE prints bytes used and bytes remaining exclusive of variables
 RUN executes a BASIC program immediate execution of un-numbered statements

The BASIC subset implemented includes:

Clubs and Newsletters Directory

As a summary of the current state of local, regional, and national organizations, special interest groups, and periodicals of interest to personal computing people, BYTE will publish a directory in a forthcoming issue. We would appreciate it if each such organization would provide a summary of applicable information in the following list:

- *Name of organization* [eg: Silicon Hollow Computer Coven].
- *Mailing address* [eg: PO Box 31, Silicon Hollow, Transylvania 00000].
- *Meeting location* [eg: Third stump past the 11th sinkhole on the old Silicon Hollow game trail].
- *Meeting algorithm* [eg: "First Tuesday after the second Wednesday before the first full moon of leap years"].
- *Name of newsletter or publication* [eg: Silicon Boule].
- *Contact person* [eg: Witch Hazel].
- *Contact phone number.*
- *Dues or subscription fees.*
- *Special interests* [eg: Computer applications: the automated swamp].
- *Other comments.*

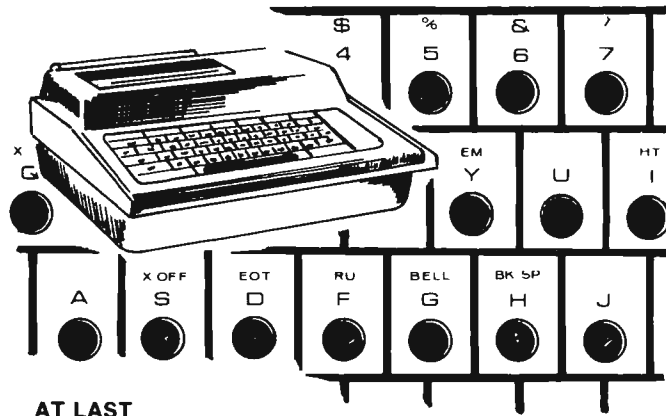
The deadline for the directory information is September 15 1976. If you wish to be certain that the latest information about your club, newsletter or organization is available, be sure to send this information to

*BYTE
 Clubs & Newsletters Directory
 70 Main St
 Peterborough NH 03458*

16 bit signed integer arithmetic
 Variable names A to Z with subscripting
 No string variables
 DIM statements: two dimensions maximum, 255 is largest dimension
 Simple arithmetic expressions with multiply and divide
 LET
 relational tests and IF statement
 GOTO n
 GOSUB n
 RETURN
 FOR . . . NEXT
 PRINT
 INPUT

The entire symbolic assembly listing of Microbasic is given, so that customization is well within the realm of possibility. The memory requirements of this program are hexadecimal C82 (or decimal 3202) bytes, and the origin is hexadecimal location 0020 in memory address space.

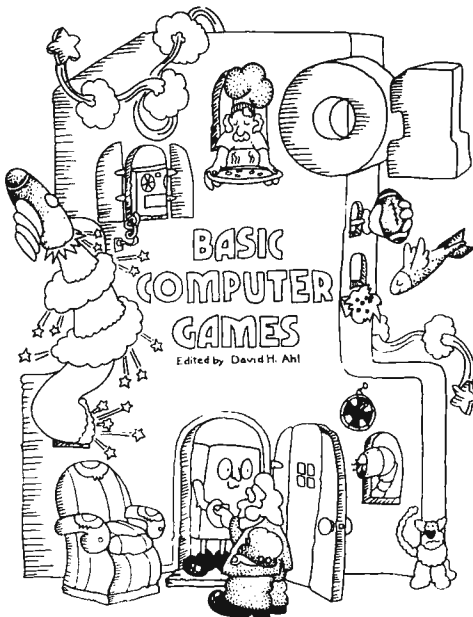
The entire contents of this excellent newsletter from SWTPC take up 50 pages. It is highly recommended reading, both for BASIC hackers and as a source of useful information. ■



**AT LAST
 A HARD COPY TERMINAL AT AN AFFORDABLE PRICE.**

- For personal computing, system applications, or remote I/O.
- The *model iSi-12111* uses a 6800 microprocessor.
- Has "Kansas City" cassette interface.
- The *model iSi-12111* emulates a Teletype* with paper tape reader and punch, but at either 100 or 300 BAUD and with both RS-232 and 20 MA current loop I/O.
- Quiet (Non impact printing)
- \$995 assembled (not a kit)
- Small (fits into a standard brief case).
- *iSi*, providing custom instrumentation for 17 years
- Please write for details. These will be sold on a first ordered first delivered basis.

isi INSTRUMENTATION SERVICES INC.
 857 WINNETKA AVENUE NORTH, MINNEAPOLIS, MN 55427
 PHONE (612) 544-8816 • TELEX 28-0101



101 BASIC Computer Games is the most popular book of computer games in the world. Every program in the book has been thoroughly tested and appears with a complete listing, sample run, and descriptive write-up. All you need add is a BASIC-speaking computer and you're set to go.

101 BASIC Computer Games. Edited by David H. Ahl. 248 pages. 8½x11 paperbound. \$7.50 plus 75¢ postage and handling (\$8.25 total) from Creative Computing, P.O. Box 789-M, Morristown, NJ 07960.

Contents

Game	Brief Description		
ACEYDU	Play acey ducey with the computer	HILO	Try to hit the mystery jackpot
AMAZIN	Computer constructs a maze	HI-Q	Try to remove all the pegs from a board
ANIMAL	Computer guesses animals and learns new ones from you	HMRABI	Govern the ancient city-state of Sumeria
AHARI	Ancient game of rotating beans in pits	HOCKEY	Ice Hockey vs. Cornell
BAGLES	Guess a mystery 3-digit number by logic	HORSES	Off-track betting on a horse race
BANNER	Prints any message on a large banner	HURKLE	Find the Hurkle hiding on a 10 x 10 grid
BASBAL	Baseball game	KINEMA	Drill in simple kinematics
BASKET	Basketball game	KING	Govern a modern island kingdom wisely
BATHUM	Match wits in a battle of numbers vs. the computer	LETTER	Guess a mystery letter - computer gives you clues
BATTLE	Decode a matrix to locate enemy battleship	LIFE	John Conway's Game of Life
BINGO	Computer prints your card and calls the numbers	LIFE-2	Competitive game of life (2 or more players)
BLKJAC	Blackjack (very comprehensive); Las Vegas rules	LITQZ	Children's literature quiz
BLKJAK	Blackjack (standard game)	MATHD1	Children's arithmetic drill using pictures of dice
BOAT	Destroy a gumball from your submarine	MNOPLY	Monopoly for 2 players
BOMBER	Fly World War II bombing missions	MUGWMP	Locate 4 Mugwumps hiding on a 10 x 10 grid
BOUNCE	Pilot a bouncing ball	NICOMA	Computer guesses number you think of
BDWL	Bowling at the neighborhood lanes	NIM	Silly number matching game
BOXING	3 round Olympic boxing match	NUMBER	Challenging game to remove checkers from a board
BUG	Roll dice vs. the computer to draw a bug	OCHECK	Destroy an orbiting germ-laden enemy spaceship
BULCOW	Guess a mystery 5-digit number vs. the computer	ORBIT	Destroy an orbiting germ-laden enemy spaceship
BULEYE	Throw darts	PIZZA	Deliver pizzas successfully
BULL	You're the matador in a championship bullfight	POETRY	Computer composes poetry in 4-part harmony
BUNNY	Computer drawing of the Playboy bunny	POET	Computer composes random poetry
BUZZWD	Compose your speeches with the latest buzzwords	POKER	Poker game
CALNDR	Calendar for any year	QUEEN	3-dimensional tic-tac-toe
CAN-AM	Drive a Group 7 car in a Can-Am road race	REVERSE	Order a series of numbers by reversing
CHANGE	Computer imitates a cashier	ROCKET	Land an Apollo capsule on the moon
CHECKR	Game of checkers	ROCK11	Lunar landing from 500 feet (with plot)
CHEMST	Dilute hydrocyanic acid to make it harmless	ROCK12	Very comprehensive lunar landing
CHIEF	Silly arithmetic drill	ROCKSP	Game of rock, scissors, paper
CHOMP	Eat a cookie avoiding the poison piece (2 or more players)	ROULET	European roulette table
CIVILW	Fight the Civil War	RUSROU	Russian roulette
CRAPS	Play craps (dice), Las Vegas style	SALVO	Destroy an enemy fleet of ships
CUBE	Negotiate a 3-D cube avoiding hidden landmines	SALVO2	Destroy 4 enemy outposts
DIAMND	Prints 1-page diamond patterns	SLOTS	Slot machine (one-arm bandit)
DICE	Summarizes dice rolls	SNOOPY	Pictures of Snoopy
DIGITS	Computer tries to guess digits you select at random	SPACWR	Comprehensive game of spacwar
DOGS	Penny arcade dog race	SPLAT	Open a parachute at the last possible moment
EVEN	Take objects from a pile - try to enc with an even number	STARS	Guess a mystery number - stars give you clues
EVEN1	Same as EVEN - computer improves	STOCK	Stock market simulation
FIPOFP	Solitaire logic game - change a row of Xs to Os	SYNONM	Word synonym drill
FOGTBL	Professional football (very comprehensive)	TARGET	Destroy a target in 3-D space - very tricky
FOTBAL	High School football	3D PLOT	Plot's families of curves - looks 3-dimensional
FURS	Trade furs with the white man	TIC-TAC	Tic-tac-toe
GOLF	Golf game - choose your clubs and swing	TOWER	Towers of Hanoi puzzle
GOMOKO	Ancient board game of logic and strategy	TRAIN	Time speed-distance quiz
GUESS	Guess a mystery number - computer gives you clues	TRAP	Trap a mystery number - computer gives you clues
GUNNER	Fire a cannon at a stationary target	23MICH	Game of 23 matches - try not to take the last one
GUNER1	Fire a cannon at a moving target	UGLY	Silly profile plot of an ugly woman
HANG	Hangman word guessing game	WAR-2	Card game of war
HELLO	Computer becomes your friendly psychiatrist	WEXDAY	Facts about your birthday
HEX	Hexapawn game	WORD	Word guessing game
		YAHITZ	Dice game of Yahitæ
		ZOOP	BASIC programmer's nightmare

A Flameless IC Recycling Trick

Jonathan Bondy
16 W Lancaster Av
Ardmore PA 19333

Ralph Droms
30 N Waterloo Rd
Devon PA 19333

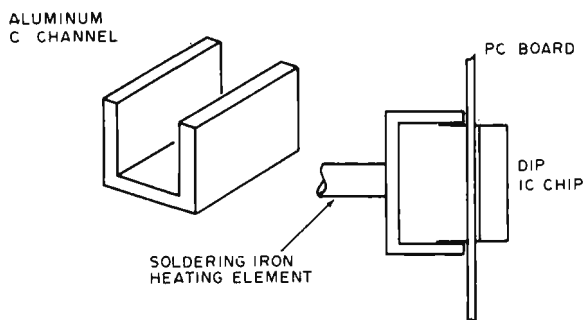


Figure 1: A short section of aluminum C channel attached to the soldering iron heating element provides the method of simultaneously heating all the pins of a 14 or 16 pin integrated circuit flamelessly. The soldering iron should preferably have a 25 W or greater heating element to provide adequate heat to all the IC pins.

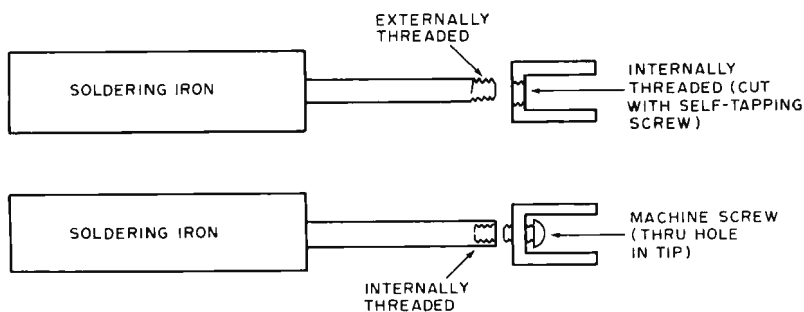


Figure 2: Depending upon the type of soldering iron element, one of these two methods can be used to attach the C channel section to the element.

Recently, some computer boards became available to us at a reasonable price (free). Having stacked them up in a couple of boxes, we decided that storing just the integrated circuits rather than the boards would make much more sense, so we investigated techniques for depopulating the boards. An article in a previous issue of BYTE suggested the use of a blowtorch to heat the pins of the chips. Discovering that the cheapest propane torch kit we could find (\$15 at today's prices) would have purchased a sizable number of surplus chips, we pressed on in search of a more inexpensive device. The final product was a standard soldering iron (which we already had) with a specially designed tip.

A trip to the local hardware store produced a six foot (1.83 m) length of aluminum C channel for about \$2. Cutting off a piece the length of a 16 pin IC chip provided us with a tip that can heat all the pins on a standard 14 or 16 pin dual in line package simultaneously, as shown in figure 1. We found that our two soldering irons employed two different methods of attaching the tip to the heating element: Either the tip was internally threaded and the iron externally threaded, or vice versa. Duplicating both kinds of threading (external and internal) was accomplished by use of two screws which duplicated the thread sizes on the two irons. One was a self tapping screw which cut internal threads in a hole drilled through the aluminum piece; and the other was inserted into the internally threaded iron through the hole drilled in the tip. Figure 2 illustrates the two assembly methods.

The resulting iron-tip combination allowed us to depopulate a 15 chip board in approximately 10 minutes (with less than 1% breakage single handed and 0% when one of us heats and one pulls). Our only problem is that we have a 5 feet 11 inch (1.8 m) section of aluminum C channel left over, so we will make available a do it yourself "kit" consisting of an undrilled 2 inch (5.08 cm) piece of C channel for only 50 cents (shipping and handling included). Send to Ralph Droms. ■

S. D. SALES CO.

P. O. BOX 28810
DALLAS, TEXAS 75228

UP YOUR COMPUTER!

21L02-1 1K LOW POWER

500NS STATIC RAM

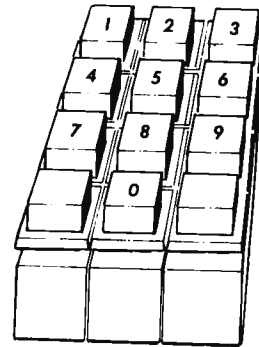
TIME IS OF THE ESSENCE

And so is power. Not only are our RAM's faster than a speeding bullet but they are now very low power. We are pleased to offer prime, new 21L02-1 low power and super fast RAM's. Allows you to **STRETCH** your power supply farther and at the same time keep the **WAIT** light off.

8 for \$17.50

TOUCHTONE KEYBOARD SWITCH SET

By Controls Research. High quality long life switches with keytops. For encoders, combination locks, etc.



12 switches and tops, including 0 thru 9. Switch contacts are independent, allows hook-up to any matrix. Keytops easily removed.

\$2.95 Set
2 for \$5.00

4K LOW POWER RAM BOARD KIT

Imsai and Altair 8080 plug in compatible. Uses low power static 21L02-1 500 ns. RAM's. Fully buffered, drastically reduced power consumption, on board regulated, all sockets and parts included. Premium quality plated thru PC Board.

THE WHOLE WORKS

\$89.95

*Exclusively
Yours*



1 MORE TIME

Call your **BANK AMERICARD** or **MASTER CHARGE**
order in on our
CONTINENTAL UNITED STATES TOLL FREE WATTS:

1-800-527-3460

TEXAS RESIDENTS CALL COLLECT:

214/271-0022

Please call between 8:30 AM and 6:00 PM C.S.T. — Monday through Friday. You may also call to check stock or just ask a question. However, only B.A.C. and M.C. orders will be accepted. We do not ship C.O.D. (See terms of sale on other page.)

S. D. SALES CO.

P. O. BOX 28810 DALLAS, TEXAS 75228

ALARM CLOCK KIT SIX DIGIT LED

Thousands of hobbyists have bought and built our original clock kit and were completely satisfied. But we have received many requests for an alarm clock kit with the same value and quality that you have come to expect from S.D. So, here it is!

THE KIT INCLUDES:

- 1 Mostek 50252 Alarm Clock Chip
- 6 Hewlett Packard .30 in. common cathode readouts.
- 15 NPN Driver Transistors
- 2 Switches for time set
- 2 Slide Switches for alarm set and enable
- 1 Filter Cap
- 4 IN4002 Rectifiers
- 1 IN914 Diode
- 1 .01 Disc Cap
- 15 Resistors
- 1 Speaker for alarm
- 1 LED lamp for PM indicator.

\$9.95

PCB - 3.00
XFMR - 1.50

60 Hz. Crystal Time Base

FOR DIGITAL CLOCKS **\$5.95**
S. D. SALES EXCLUSIVE!

KIT FEATURES:

- A. 60 Hz output with accuracy comparable to a digital watch.
- B. Directly interfaces with all MOS clock chips.
- C. Super low power consumption (1.5 Ma typ.)
- D. Uses latest MOS 17 stage divider IC.
- E. Eliminates forever the problem of AC line glitches.
- F. Perfect for cars, boats, campers, or even for portable clocks at ham field days.
- G. Small size, can be used in existing enclosures.

BUY TWO FOR \$10.00

Kit includes crystal, divider IC, P.C. Board plus all other necessary parts and specs.

MOS 4 DIGIT COUNTER I C

All in one 28 PIN DIP. 4 Decade counters, latches, MUX circuits, display decoders, etc. Features: 5VDC operation, 25 MW power consumption, BOTH 7 segment and BCD outputs. Perfect for DVM's, frequency meters, tach's, etc. Can be cascaded for more digits. #5002 - \$8.95.

8008-1 MICRO PROCESSOR

New Units. High speed 8008. Almost twice as fast as units sold by others. Still a very versatile and widely used MPU. No data book, only pinout data included at this price. \$12.50 LIMITED QTY.

1101 A RAM IC

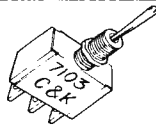
256 X 1 BITS. Perfect for 8008 or small systems such as telephone dials that do not require a lot of memory. Special 59c. 8 FOR \$4.

C & K MINI TOGGLE SWITCH

#7103 SUB MINI SPDT Center OFF. Special - 99c.

TTL INTEGRATED CIRCUITS

7400 - 19c	7430 - 19c	7476 - 35c	74153 - 75c
7402 - 19c	7432 - 34c	7480 - 49c	74154 - 1.00
74L04 - 29c	7437 - 39c	7483 - 95c	74157 - 75c
74S04 - 44c	7438 - 39c	7485 - 95c	74161 - 95c
7404 - 19c	7440 - 19c	7486 - 45c	74164 - 1.10
7406 - 29c	7447 - 85c	7490 - 65c	74165 - 1.10
7408 - 19c	7448 - 85c	7492 - 75c	74174 - 95c
7410 - 19c	7451 - 19c	7495 - 75c	74181 - 2.50
7411 - 29c	7453 - 19c	7496 - 89c	74191 - 1.25
7413 - 50c	7473 - 39c	74121 - 38c	74192 - 1.25
7420 - 19c	7474 - 35c	74123 - 65c	74193 - 1.00
	7475 - 69c	74141 - 75c	74195 - 69c



MOTOROLA POWER DARLINGTON - \$1.99
BACK IN STOCK!
Like MJ3001, NPN 80 V., 10A., HFE 6000 TYP. TO-3 case.
We include a free 723C volt. reg. with schematic for a power supply.
SPECIAL - \$1.99

MOTOROLA RTL IC'S

Brand new, factory prime. Hard to find, but still used in a variety of projects. (See the RTL Cookbook by Howard W. Sams.)

MC724P-59c	MC780P-89c	MC791P-69c
MC725P-59c	MC785P-49c	MC792P-59c
MC764P-49c	MC787P-89c	MC799P-59c
MC767P-69c	MC788P-49c	MC9704P-89c
MC771P-49c	MC789P-59c	MC9709P-69c
MC775P-89c	MC790P-89c	MC9760P-69c

MV-50 TYPE LED'S

by LITRONIX
10 for \$1
Factory Prime!

3 DIGIT LED ARRAY - 75c

by LITRONIX
DL33MMB. 3 MAN-3 Size Readouts in one package. These are factory prime, not retested rejects as sold by others. compare this price! 75c 3 for \$2.

SALE ON CUT LEAD SEMICONDUCTORS

Leads were cut for PCB insertion. Still very useable.

1N914/1N4148	100/\$2
1N4002 1 Amp 100 PIV	40/\$1
1N4745A 16V 1W Zener	20/\$1
EN2222 NPN Transistor	25/\$1
EN2907 PNP Transistor	25/\$1
2N3904 NPN Driver Xstr.	25/\$1
2N3392 GE Pre-amp Xstr.	25/\$1
C103Y SCR. 800MA. 60V.	10/\$1

ALL NEW.
UNUSED.
SOME ARE
HOUSE #

SLIDE SWITCH ASSORTMENT

Our best seller. Includes miniature and standard sizes, single and multi-position units. All new, first quality, name brand switches. Try one package and you'll reorder more. Special - 12 for \$1 (Assortment)



DISC CAP ASSORTMENT

PC leads. At least 10 different values. Includes .001, .01, .05, plus other standard values.
60 FOR \$1



UPRIGHT ELECTROLYTIC CAPS

47 mfd 35 V-10/\$1 68 mfd 25V-8/\$1
Brand new by Sprague. PC leads.

RESISTOR ASSORTMENT

1/4 W 5% and 10%. PC leads.
A good mix of values. 200/\$2



1000 MFD FILTER CAPS

Rated 35 WVDC. Upright style with P.C. leads. Most popular value for hobbyists. Compare at up to \$1.19 each from franchise type electronic parts stores. S.D. Special 4 for \$1



FAIRCHILD BIG LED READOUTS

A big .50 inch easy to read character. Now available in either common anode or common cathode. Take your pick. Super low current drain, only 5 MA per segment typical.

FND - 510	Common Anode	YOUR CHOICE \$1.50 ea. 6 for \$7.50
FND - 503	Common Cathode	

DUAL 741C (5558) OP AMPS

Mini dip. New house numbered units by RAYTHEON.
4 FOR \$1

FET'S BY TEXAS INSTRUMENTS - SPECIAL 5 for \$1
#TIS-75 but with an internal house number. TO-92 plastic case. N. Channel, Junction type FET.

We do not sell junk. Money back guarantee on every item. No C.O.D. Texas Res. add 5% tax. Postage rates went up 30%! Please add 5% of your total order to help cover shipping.

S. D. SALES CO.

P. O. BOX 28810
DALLAS, TEXAS 75228

T - - - K - - -
- - - - -

ORDERS UNDER \$10
ADD 75c HANDLING.

ORDERS OVER \$15 CHOOSE \$1 FREE MERCHANDISE

FOREIGN ORDERS MUST BE PAID IN U.S. FUNDS

What is AMSAT?

AMSAT is the name of one of the most exotic technical activities practiced by amateur radio operators: building and orbiting communications satellites. The Radio Amateur Satellite Corporation is a joint project of amateur radio groups all over the world. The practice is to design and build small satellites which can be carried "piggyback" on a NASA launching of other satellites. These "OSCAR" (Orbiting Satellite Carrying Amateur Radio) vehicles contain facilities for receiving and retransmitting messages originating from amateur radio stations around the world. To find out about AMSAT, contact AMSAT at PO Box 27, Washington DC 20044.

AMSAT 8080 Standard Debug Monitor:

AMS80 Version 2

This monitor or debug package resides in low memory in an 8080 system. It is designed as a minimum system debug package.

The source code (see listing 1) contains a large number of comments, explaining in detail what is being done in each routine, so it is not discussed in great detail, but just summarized in a few words.

The monitor contains interrupt vectors, utility subroutines and a command sequence which allows the contents of memory locations and the 8080 registers to be examined or changed. Commands are listed in table 1. The utility routines are also available for incorporation in user programs.

Changes to memory and registers are made through the system console which might be a Teletype or CRT terminal. As presented, the monitor includes a listing of the Teletype routines used to drive the system for which it was configured. These routines will have to be changed to conform to the hardware of a particular user system.

Utility Subroutines

The utility subroutines are located within the monitor. A jump table is used to interface these routines to the user programs. This jump table is located at hexadecimal location 40. This jump indirect technique adds a very small overhead to the total execution time of a user program, which in practice is hardly ever noticed. It also ensures that in the event of later versions of the monitor being used in a system, programs already developed will still run. This is because when the monitor is changed (to improve it, to add a function such as a cassette interface driver, or just to

Richard C Allen W5SXD
4648 Spruce St
Bellaire TX 77401

Joe Kasser G3ZCZ
11532 Stewart Ln
Silver Spring MD 20904

Table 1: List of AMS80 monitor commands.

A	Address: examine/change the contents of a memory address location.
.	Examine current location.
LF	Examine next location (LF = line feed character).
-	Examine previous location (minus sign).
D	Dump an area of memory on the Teletype.
F	Fill a block of memory with a constant.
G	Go to (and start executing from) a memory location.
M	Move a block of memory.
X	Set up an execution address.
J	Restore all registers, then jump to (and start executing from) a memory location.
R	Examine/change the contents of the registers. Follow R with mnemonic of a particular register if desired.
P	Punch the contents of memory on paper tape via the Teletype in Intel hexadecimal format.
E	Punch an end of file mark on the tape.
L	Load a paper tape (previously punched in Intel hexadecimal format) into memory.
N	Punch 100 nulls on the paper tape.
CR	Terminate sequence of commands (CR = carriage return character).

eliminate a bug) the actual locations of the subroutines may change, but the location of the jump table will not.

For example, a user routine calculating and printing out spacecraft orbits will use the instruction CALL @TYPE in many places. If that user then gets a later version of the monitor in which the location of the TYPE routine has changed, the program will still run, because the location of @TYPE has not changed, but the jump instruction at that location calls the new position of TYPE. If @TYPE had not been used as an interface to the user program, the user would have had to reassemble his program so that his CALL TYPE instructions would locate the subroutine in the new location.

The following utility routines are available:

TYPE types an ASCII character from the A register.

GETCH gets a character from the Teletype (or terminal) to the A register.

CHIN gets a character from the Teletype, echoes it to the Teletype, strips off the parity bit, and ends with the character in the A register.

MSG prints an ASCII message on the Teletype; the message must end with a FF (all ones). The message address is passed in the H and L registers of the 8080 processor.

CRET types a carriage return/line feed on the Teletype.

SPACE types a space character.

THXN types the 4 low order bits of the A register as a hexadecimal ASCII character.

THXB types the contents of the A register in as a two digit ASCII hexadecimal representation.

THXW types the contents of the HL register pair in as a four digit ASCII hexadecimal representation.

GHXN gets a hexadecimal nybble to the A register (1 keystroke).

GHXB gets a hexadecimal byte from the Teletype to the A register (2 keystrokes).

GHXW gets a hexadecimal 2 byte address to the HL register pair (4 keystrokes).

STORE stores a byte of data in memory, with check.

NEGDE negates the contents of the DE register pair.

PWAIT types "PAUSE" and waits for any character from the keyboard.

OK? types "OK?", and waits as in PWAIT. A space character means go ahead or OK; any other character causes an abort and a return to the monitor.

Interrupt Handling

The monitor provides for seven interrupts as well as the initial reset. Each interrupt service routine must be told where the service routine is located in upper memory. The addresses of these routines are stored in a vector table located in programmable memory. These vectors are located and loaded into the HL register pair by the service routines after which the proper branch is executed. The monitor does not reset the 8080 interrupt disable flag or any external interrupt status ports. These operations must be part of the user interrupt service routines. The service routines do not change the contents of any of the 8080 registers, so that they are available and can be saved by the user as needed.

IO Routines

The monitor is written so that any Teletype IO routines can be used as long as they are located at GETCH and TYPE. The print and punch routines also operate the Teletype, and include automatic turn on and

Listing 1: The complete AMS80 monitor listing. This is a photo reproduction of an assembly of the monitor, version 2.0.

```

TITLE 'AMS80 - AMSAT STANDARD 8080 MONITOR, V2.0'
;
; AMS80 - AMSAT STANDARD 8080 MONITOR
; VERSION 1.0, 09NOV75
;
; RICHARD C. ALLEN, W5SXJ
; 4648 SPRUCE STREET
; BELLAIRE, TEXAS 77401
;
;
; MODIFIED VERSION OF AMS80 V1.0
;
; JOE KASSER G3ZCZ
; 11532 STEWART LANE
; SILVER SPRING, MARYLAND 20904
;
; TEL: 301-622-2194
;
;
; THIS MONITOR IS A MINIMUM 8080 SYSTEM MONITOR
; FOR USE BY AMSAT MEMBERS. IT PROVIDES THE BASIC
; STRUCTURE NECESSARY FOR 8080 DEBUG AND ALSO A
; STANDARD BASE FOR AMSAT MEMBERS USING THE 8080.
;
; THIS STANDARD BASE WILL ALLOW PROGRAMS
; TO BE USED BY ALL AMSAT MEMBERS AND AID IN THE
; TRANSMISSION OF PROGRAM MATERIALS VIA THE OSCAR
; SATELLITES.
;
;
; THE ROUTINE ALLOWS FOR MEMORY EXAMINE AND MODIFY.
; USER INTERRUPT/RST VECTORS, AND VARIOUS
; TELETYPE SUPPORT ROUTINES TO LOAD AND DUMP MEMORY
; IN A STANDARD FORMAT ( SAME AS THE INTEL FORMAT )-
;
;
;
; DEFINE THE SYSTEM MEMORY PARAMETERS
;
0000 ROM EQU 0 ;START OF READ-ONLY-MEMORY
0C00 RAM EQU 0C00H ;START OF READ-WRITE MEMORY
0D00 STACK EQU RAM+256 ;TOP OF MONITOR STACK
; AND END OF MONITOR RAM
;
; DEFINE TTY CONTROL CHARS
;
000D CR EQU 0DH ;CARRIAGE RETURN
000A LF EQU 0AH ;LINE FEED
007F RBO EQU 7FH ;RUB-OUT
;
0014 TOFF EQU 14H ;TAPE OFF COMMAND
0018 TON EQU 18H ;TAPE ON COMMAND
;
0013 XOFF EQU 13H ;PUNCH OFF COMMAND
0011 XON EQU 11H ;PUNCH ON COMMAND

```

Text continued
on page 122

```

;
; START OF SYSTEM
0000      ORG      ROM

EXEC:      JMAIN ENTRY INTO EXEC-80
0000 22120C SHLD SVHL      ;SAVE HL
0003 C37000 JMP BEGIN      ; AND BEGIN
0006 7105     DW ENDRM      ;PAD BYTES ONLY

; DEFINE USER INTERRUPT/SUBROUTINE VECTORS
;
UINT      MACRO VECT
          PUSH H      ;SAVE HL
          LHL D VECT  ; FETCH USER VECTOR
          XTHL       ;PUT ONTO STACK RESTORING HL
          RET        ;GO TO USER PROC
          DW 0
          ENDM

RS1:      UINT RST1   ;RST 1
          PUSH H      ;SAVE HL
          LHL RST1   ; FETCH USER VECTOR
          XTHL       ;PUT ONTO STACK RESTORING HL
          RET        ;GO TO USER PROC
          DW 0

0008 E5
0009 2A000C
000C E3
000D C9
000E 0000

RS2:      UINT RST2   ;RST 2
          PUSH H      ;SAVE HL
          LHL RST2   ; FETCH USER VECTOR
          XTHL       ;PUT ONTO STACK RESTORING HL
          RET        ;GO TO USER PROC
          DW 0

0010 E5
0011 2A020C
0014 E3
0015 C9
0016 0000

RS3:      UINT RST3   ;RST 3
          PUSH H      ;SAVE HL
          LHL RST3   ; FETCH USER VECTOR
          XTHL       ;PUT ONTO STACK RESTORING HL
          RET        ;GO TO USER PROC
          DW 0

0018 E5
0019 2A040C
001C E3
001D C9
001E 0000

RS4:      UINT RST4   ;RST 4
          PUSH H      ;SAVE HL
          LHL RST4   ; FETCH USER VECTOR
          XTHL       ;PUT ONTO STACK RESTORING HL
          RET        ;GO TO USER PROC
          DW 0

0020 E5
0021 2A060C
0024 E3
0025 C9
0026 0000

RS5:      UINT RST5   ;RST 5
          PUSH H      ;SAVE HL
          LHL RST5   ; FETCH USER VECTOR
          XTHL       ;PUT ONTO STACK RESTORING HL
          RET        ;GO TO USER PROC
          DW 0

0028 E5
0029 2A080C
002C E3
002D C9
002E 0000

RS6:      UINT RST6   ;RST 6
          PUSH H      ;SAVE HL
          LHL RST6   ; FETCH USER VECTOR
          XTHL       ;PUT ONTO STACK RESTORING HL
          RET        ;GO TO USER PROC
          DW 0

0030 E5
0031 2A0A0C
0034 E3
0035 C9
0036 0000

RS7:      UINT RST7   ;RST 7
          PUSH H      ;SAVE HL
          LHL RST7   ; FETCH USER VECTOR
          XTHL       ;PUT ONTO STACK RESTORING HL
          RET        ;GO TO USER PROC
          DW 0

0038 E5
0039 2A0C0C
003C E3
003D C9
003E 0000

;
; MONITOR SUPPORT SUBROUTINE VECTORS
;
; USER UTILITY SUBROUTINES
;
; THE FOLLOWING SET OF JUMPS ARE PROVIDED SO
; USER PROGRAMS CAN REFERENCE COMMON ENTRY POINTS
; TO THE VARIOUS ROUTINES. THESE LOCATIONS WILL
; REMAIN CONSTANT WHILE THE ACTUAL LOCATION OF EACH
; ROUTINE MAY CHANGE FROM ONE REVISION LEVEL TO THE
; NEXT.
;
; THE CALLING SEQUENCE FOR EACH SUBROUTINE
; REMAINS THE SAME AS DEFINED IN THE LISTING, WITH
; ONLY A SLIGHT EXECUTION TIME OVERHEAD FOR THE
; EXTRA JMP.
;
0040 C35705 @TYPE: JMP TYPE      ;TYPE A CHARACTER FROM 'A'
0043 C36605 @GETCH: JMP GETCH ;GET CHAR TO A (NO ECHO)
0046 C30001 @CHIN: JMP CHIN ;GET CHAR TO 'A' WITH ECHO
; ( PARITY SET OFF )
0049 C31801 @MSG: JMP MSG ;TYPE MSG, POINTER IN HL
; ( MSG TERMINATED BY OFFH )
004C C32D01 @CRET: JMP CRET ;TYPE CR, LF, RUB-OUT
004F C33A01 @SPACE: JMP SPACE ;TYPE A SPACE
0052 C34F01 @THXN: JMP THXN ;TYPE B3-B0 OF 'A' IN HEX
; ( ONE ASCII CHARACTER )
0055 C34801 @THXB: JMP THXB ;TYPE 'A' IN ASCII-HEX 2 CH
0058 C36001 @THXL: JMP THXL ;TYPE 'HL' IN ASCII-HEX 4 CH
005B C36B01 @GHXN: JMP GHXN ;GET HEX NIBBLE TO B3-B0 'A'
005E C38201 @GHXB: JMP GHXB ;GET HEX BYTE FROM TTY1 -> 'A'
0061 C39501 @GHXL: JMP GHXL ;GET HEX WORD TO HL
0064 C3B001 @STORE: JMP STORE ;STORE A BYTE M,A WITH CHECK
0067 C31302 @NEGDE: JMP NEGDE ;NEGATE THE DE REGISTER
006A C30C04 @PWAIT: JMP PWAIT ;TYPE 'PAUSE' AND WAIT FOR
; ANY CHARACTER ON TTY1
006D C38E02 @OK?: JMP OK? ;TYPE 'OK?' AND WAIT FOR
; SPACE IF OK, OTHERS WILL
; PRINT ABOUT MSG AND RETURN
; TO MONITOR.

0070 211DOC LXI H, TMPA ;SET PSEUDO
0073 3600 MVI M, 0 ; CARRY TO 0
0075 D27A00 JNC $+5 ; NO CARRY ON INPUT
0078 3601 MVI M, 1 ;PSEUDO CY TO 1
007A E1 POP H ;POPCALL ADDRESS IF ANY
007B 220E0C SHLD SVPC ; AND SAVE PC
007E 21FEFF LXI H, -2 ;FETCH SP
0081 39 DAD SP ; ADJUSTING FOR POP
0082 22100C SHLD SVSP ;SAVE USER STACK POINTER
0085 311A0C LXI SP, SVA+1 ;SET SP FOR REGISTER SAVE
0088 F5 PUSH PSW ;SAVE A,PSW
0089 C5 PUSH B ;SAVE BC
008A D5 PUSH D ;SAVE DE
008B 21180C LXI H, SVF ;POINT TO SAVED PSB
008E 7E MOV A, M ; ALD FETCHIT
008F E6FE ANI OFEH ;ZERO CY
0091 47 MOV B, A ; AND SAVE
0092 3A1DOC LDA TMPA ;GET INPUT SAVED CY
0095 80 ORA B ; AND INSERT
0096 77 MOV M, A ;RESTORE PSB WITH OK CY
0097 31000D LXI SP, STACK ;SET SP TO EXEC STACK AREA
009A 21EA04 LXI H, MO ;TYPE ENTRY
009D CD1801 CALL MSG ; MESSAGE

;
; BEGIN MONITOR
;
BEGIN:
;
; NEXT MONITOR COMMAND
;
00A0 31000D NEXT: LXI SP, STACK ;RESTORE SP
00A3 21F904 LXI H, M1 ;TYPE
00A6 CD1801 CALL MSG ; PROMPTER
00A9 CD0001 CALL CHIN ;GET COMMAND CHAR
00AC 47 MOV B, A ; AND SAVE COMMAND

;
; SEARCH OPERATION TABLE FOR COMMAND
;
00AD 21CF00 LXI H, OPTAB ;FETCH TABLE VECTOR

SRCH: MOV A, M ;GET TABLE COMMAND BYTE
CPI -1 ;CHECK FOR END OF TABLE
JZ ILLEG ;MUST BE ILLEGAL INPUT
CMP B ;COMPARE TO INPUT
JZ FNDCH ;FOUND COMMAND
INX H ;BUMP TO
INX H ; NEXT
INX H ; COMMAND
JMP SRCH ;AND CONTINUE

;
; UNDEFINED COMMAND, TYPE ERROR MESSAGE
;
00C0 210005 ILLEG: LXI H, M2 ;UNDEFINED
00C3 CD1801 CALL MSG ; MESSAGE
00C6 C3A000 JMP NEXT ;TRY AGAIN

;
; FOUND COMMAND, NOW FETCH ADDRESS AND EXECUTE COMMAND
;
00C9 23 FNDCH: INX H ;BUMP TO LOW ADDRESS BYTE
00CA 5E MOV E, M ; AND FETCH IT
00CB 23 INX H ;GET HIGH
00CC 56 MOV D, M ; ADDRESS BYTE
00CD EB XCHG ;ADDRESS TO HL
00CE E9 PCHL ;GOTO COMMAND PROCESSOR

;
; OPERATION DECODE/DISPATCH TABLE
;
00CF 41 OPTAB: DB 'A' ;COMMAND
00D0 C101 DW GETAD ; TO GET ADDRESS

00D8 0D DB CR ;COMMAND
00D3 A000 DW NEXT ; EFFECTIVE NOP

00D5 2E DB '-' ;COMMAND
00D6 D301 DW LOCAT ; TO EXAMINE CURRENT LOCATION

00D8 0A DB LF ;COMMAND
00D9 F601 DW NKLOC ; TO EXAMINE NEXT LOCATION

00DB 2D DB '-' ;COMMAND
00DC 0C02 DW LSTLC ; TO EXAMINE PREVIOUS LOCATION

00DE 44 DB 'D' ;COMMAND
00DF 1D02 DW DUMP ; TO DUMP MEMORY AREA

00E1 46 DB 'F' ;COMMAND
00E2 CB02 DW FILL ; TO FILL MEMORY

00E4 47 DB 'G' ;COMMAND
00E5 7C02 DW GOTO ; TO GOTO MEMORY LOCATION

00E7 4D DB 'M' ;COMMAND
00E8 AC02 DW MOVE ; TO MOVE AREA OF MEMORY

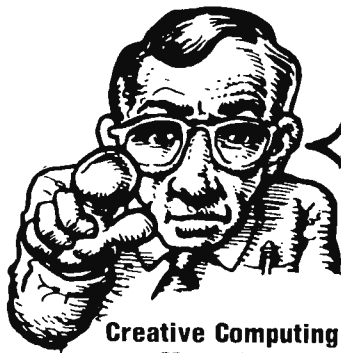
00EA 58 DB 'X' ;COMMAND
00EB 8402 DW GETXA ; TO GET XEA ADDRESS

00ED 4A DB 'J' ;COMMAND
00EE 4202 DW JUMP ; TO JUMP TO MEMORY LOCATION

00F0 52 DB 'R' ;COMMAND
00F1 F202 DW REGEX ; REGISTER EXAMINE

00F3 50 DB 'P' ;COMMAND
00F4 A003 DW PUNCH ; PUNCH MEMORY

```

Creative Computing Magazine

A bi-monthly 88-page magazine for students, hobbyists, and anyone curious about computers. Fiction, articles, humor about computers, cybernetics, careers, building info., etc. Emphasis on games, puzzles, and projects. Contemporary, non-technical approach. Subscription. \$8.00 pp.

Games & Puzzles Issue of Creative Computing

88 pages of games and puzzles for pocket calculators, computers, and humans. "Beating the Game," "Computer Chess," "Hunting a Wumpus in a Cave," building your own computer, reviews of 24 games, books, and much more! \$1.50 pp.

Futures Issue of Creative Computing

Artificial Intelligence (Bertram Raphael, Herbert Dryfus, etc.), Extraterrestrial Intelligence (Isaac Asimov, Martin Harwit, etc.), microprocessors, videodiscs as an ultimate computer input device, 4 new games, and more. 88 big pages! \$1.50 pp.

Artist and Computer

A high-quality, 4-color book edited by Ruth Leavitt which displays the work of 35 internationally-known computer artists. Each artist describes his or her work in non-technical terms. 140 illustrations. \$4.95.

ISN'T IT TIME YOU ORDERED SOME CREATIVE COMPUTING STUFF!! TODAY!

The Best of Creative Computing

A 328-page book featuring stories by Isaac Asimov and others; articles on cybernetics, robots, computer crime, privacy; computer games such as Star Trek, Rabbit Chase, Magic Square, Madlib, and 14 more; super computer graphics; cartoons, reviews; poetry; and more! \$8.95.

Creative Computing T-Shirt

Albert Einstein portrait produced by Blocpix™ process. Scarlet trim, black design. Available in adult sizes: S, M, L, XL. \$4.00 pp.

Mr. Spock Computer Image

Big 17x22" computer scanner image. Heavy stock. Comes in strong mailing tube. \$1.50 pp.

Star Trek People Computer Images

Six 8½x11 computer images on heavy stock of Kirk, Spock, McCoy, Scott, Chekov, and Uhura. \$1.50 pp.

101 BASIC Computer Games

A collection of 101 games in BASIC, each one with a complete listing, sample run, and write-up. 256 pages. \$7.50.

Please send me the following:

Item	Price

Shipping (books only) \$1.00
All orders outside U.S.A. add \$1.00

New Jersey residents add 5% sales tax
Total

Cash, check, M.O. enclosed
 Charge my Bankcard (minimum charge \$15)

BankAmericard Master Charge

Acct. No. _____

Expiration date. Mo. _____ Yr. _____

Name _____

Address _____

City _____

State _____ Zip _____

Creative Computing, P.O. Box 789-M,
Morristown, N.J. 07960, U.S.A.

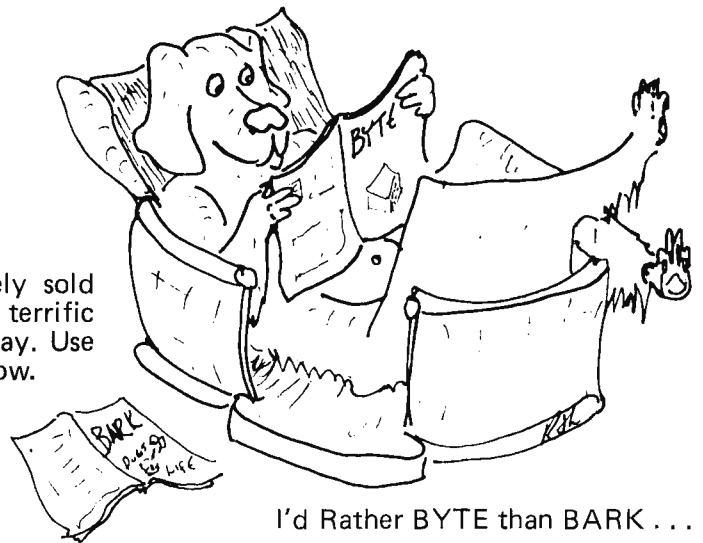
DO IT NOW . . .

SUBSCRIBE TO BYTE

Many issues of BYTE are already completely sold out. Don't take a chance of missing any of the terrific articles planned for future issues. Subscribe today. Use the pullout postcard in this issue or the form below.

Please allow six weeks for processing.

(Artwork of this doodle supplied by Richard Lerseth)



I'd Rather BYTE than BARK . . .

Name _____

Address _____

City _____ State _____ Zip _____

Bill me Bill MC BAC # _____ Exp. Date _____

BYTE 70 Main Street Peterborough, NH 03458

```

00F6 45      DB 'E'          ;COMMAND
00F7 1804    DW PEND          ; PUNCH END-OF-FILE

00F9 4C      DB 'L'          ;COMMAND
00FA 6804    DW LOAD          ; LOAD MEMORY

00FC 4E      DB 'N'          ;COMMAND
00FD 5D04    DW NULL          ; PUNCH NULLS

00FF FF      DB -1          ;END OF TABLE CODE

;
; CHIN - ROUTINE TO INPUT ONE CHARACTER,
; STRIP OFF PARITY, AND ECHO IF ABOVE
; A SPACE (I.E., NOT CR, LF, ETC.)
;
; CALLING SEQUENCE ...
;
;          CALL CHIN          ;CHARACTER IN
;          ...                ;RETURN AFTER ECHO STARTED
;          ;WITH CHAR .AND. 7FH IN 'A'
;

0100 3EFF    CHIN: MVI A,-1      ;SET ECHO
0101 381A0C  STA ECHO          ; FLAG ON
0102 CD6605  CHINN: CALL GETCH      ;GET CHARACTER
0103 E67F    ANI 7FH          ;STRIP PARITY
0104 F5      PUSH PSW          ;SAVE DATA
0105 3A1A0C  LDA ECHO          ; AND CHECK
0106 A7      ANA A           ; ECHO FLAG
0107 C81401  JNZ $+5          ; ECHO SET
0108 F1      POP PSW          ;ECHO NOT SET
0109 C9      RET
010A F1      POP PSW          ;RESTORE DATA AND ECHO
010B FE90    CPI ' '        ;CHECK FOR CONTROL
010C D45705  CMC TYPE         ;TYPE IF != SPACE
010D 11A C9  RET          ;RETURN

;
; MESSAGE PRINT ROUTINE
;
; CALLING SEQUENCE ...
;
;          LXI H,ADDRESS      ;ADDRESS OF MESSAGE
;          CALL MSG           ;CALL ROUTINE
;          ...                ;RETURN HERE AFTER LAST CHAR
;          ; INITIATED, ALL REGISTERS
;          ; PRESERVED
;

0118 F5      MSG:  PUSH PSW          ;SAVE PSW
0119 E5      PUSH H           ;SAVE HL
011A 7E      MNXT:  MOV A,H          ;GET A CHARACTER
011B FEFF    CPI -1          ;CHECK FOR 3778/OFFH/-1 TERMTR
011C CA2A01  JZ MDONE          ;FOUND THE TERMINATOR
011D CDS705  CALL TYPE         ;TYPE THE CHARACTER
011E 83      INX H           ;BUMP MEM VECTOR
011F C31D01  JMP MNXT          ; AND CONTINUE

0120 E1      MDONE: POP H           ;RESTORE HL
0121 F1      POP PSW          ; AND PSW
0122 C9      RET          ;EXIT TO CALLER

;
; ROUTINE TO TYPE CR, LF, RBO
;
; CALLING SEQUENCE ...
;
;          CALL CRET
;          ...                ;RETURN HERE WITH ALL
;          ;REGISTERS PRESERVED
;

0128 E5      CRET:  PUSH H           ;SAVE HL
0129 813601  LXI H,CRMSG       ;ADDRESS OF CR,LF,RBO MSG
012A CD1B01  CALL MSG          ;TYPE IT
012B E1      POP H           ;RESTORE HL
012C C9      RET          ; AND RETURN

0136 0D0A7FFF CRMSG: DB CR,LF,RBO,-1

;
; ROUTINE TO TYPE ONE SPACE
;
; CALLING SEQUENCE ...
;
;          CALL SPACE
;          ...                ;RETURN HERE
;
; ALL REGISTERS PRESERVED
;

013A F5      SPACE: PUSH PSW          ;SAVE A,PSB
013B 3E90    MVI A,' '      ;GET A SPACE
013C CD5705  CALL TYPE         ; AND DO IT
013D F1      POP PSW          ;RESTORE PSW
013E C9      RET          ;AND RETURN

;
; ROUTINE TO TYPE VALUE IN 'A' IN HEX ON TTY
;
; CALLING SEQUENCE ...
;
;          LDA DATA          ;DATA BYTE IN 'A'
;          CALL THXB          ;TYPE IN HEX
;          ...                ;RETURN HERE
;
; ALL REGS PRESERVED
;

0142 F5      THXB:  PUSH PSW          ;SAVE A,PSB
0143 0F      RRC           ;SHIFT
0144 0F      RRC           ; TO
0145 0F      RRC           ; LEFT
0146 0F      RRC           ; NIBBLE
0147 CD4F01  CALL THXN          ;TYPE HEX NIBBLE
0148 F1      POP PSW          ;RESTORE DATA
0149 CD4F01  CALL THXN          ;TYPE RIGHT NIBBLE
014A C9      RET          ; AND EXIT

;
; ROUTINE TO TYPE ONE ASCII CHARACTER REPRESENTING
; BITS 3-0 OF 'A' IN HEX
;
; CALLING SEQUENCE ...
;
;          LDA DATA          ;DATA NIBBLE IN BITS 3-0
;          CALL THXN          ;TYPE NIBBLE IN HEX
;          ...                ;RETURNS HERE
;
; ALL REGS PRESERVED, AND CONTENTS OF 'A' BITS 7-4
; ARE NOT SIGNIFICANT AND ARE IGNORED.
;

014F F5      THXN:  PUSH PSW          ;SAVE PSW
0150 E60F    ANI 0FH          ;ISOLATE NIBBLE B3>B0
0151 FE0A    CPI 10         ;SEE IF > 9
0152 DA5901  JC $+5          ;NIBBLE <= 9
0153 C607    ADI 7          ;ADJUST ALPHA CHAR
0154 C630    ADI '0'        ; ADD IN ASCII 0
0155 CD5705  CALL TYPE         ;AND TYPE THE NIBBLE
0156 F1      POP PSW          ;RESTORE PSW
0157 C9      RET          ; AND RETURN

;
; ROUTINE TO TYPE A WORD IN HEX
;
;          LHL WORD           ;WORD IN HL
;          CALL THXW          ;TYPE IT IN HEX
;          ...                ;RETURN HERE
;
; ALL REGISTERS PRESERVED
;

0160 F5      THXW:  PUSH PSW          ;SAVE PSW
0161 7C      MOV A,H          ;GET HIGH BYTE
0162 CD4801  CALL THXB          ; AND TYPE IT
0163 7D      MOV A,L          ;GET LOW BYTE
0164 CD4801  CALL THXB          ; AND TYPE IT
0165 F1      POP PSW          ;RESTORE PSW
0166 C9      RET          ; AND RETURN

;
; ROUTINE TO GET ONE HEX CHARACTER FROM TTY
;
; CALLING SEQUENCE ...
;
;          CALL GHXN          ;GET HEX NIBBLE
;          JC NONHX          ;CY SET IF NON HEX
;          ...                ;HEX NIBBLE IN 'A' B3-B0
;
; IF THE CHARACTER ENTERED IS 0 TO 9 OR A TO F THEN
; 'A' WILL BE SET TO THE BINARY VALUE 0 TO F AND
; THE CARRY WILL BE RESET.
;
; IF THE CHARACTER ENTERED IS NOT A VALID HEX DIGIT
; THEN THE 'A' REGISTER WILL CONTAIN THE ASCII CHAR
; AND THE CARRY WILL BE SET TO A 1.
;
; ALL REGISTERS EXCEPT PSW PRESERVED
;

0168 F5      GHXN:  CALL CHINN        ;GET CHARACTER IN
0169 FE30    CPI '0'        ;CHINN IN CASE NOT ECHO)
016A D8      RC           ;RETURN IF
016B D8      RC           ; < '0'
016C FE3A    CPI '1'        ;SEE IF NUMERIC
016D DA7F01  JC GHX1        ;CHAR IS 0 TO 9
016E FE41    CPI 'A'        ;SEE IF A TO F
016F D8      RC           ;CHAR '1' TO ' '
0170 FE47    CPI 'G'        ;SEE IF > 'F'
0171 3F      CMC           ;INVERT CY SENSE
0172 D8      RC           ;CHAR > 'F'
0173 D607    SUI 7          ;CHAR IS A TO F SO ADJUST
0174 D630    SUI '0'        ;ADJUST TO BINARY
0175 C9      RET          ; AND EXIT

;
; ROUTINE TO GET ONE HEX BYTE FROM TTYI
;
; CALLING SEQUENCE ...
;
;          CALL GHXB          ;GET HEX BYTE
;          JC NONHX          ;SAME AS GHXN, NON-HEX INPUT
;          ...                ;HEX BYTE IN 'A'
;
; ALL REGS EXCEPT PSW PRESERVED, CY SET AS IN GHXN
;

0182 CD6801  GHXB:  CALL GHXN          ;GET LEFT NIBBLE
0183 D8      RC           ;LEAVE IF NON-HEX
0184 C5      PUSH B          ;SAVE BC
0185 07      RLC           ;SHIFT
0186 07      RLC           ; TO
0187 07      RLC           ; LEFT
0188 07      RLC           ; NIBBLE
0189 47      MOV B,A        ;AND SAVE IN B
018A CD6B01  CALL GHXN          ;GET RIGHT NIBBLE
018B DA9301  JC $+4          ;JMP IF NON-HEX
018C 80      ADD B          ;ADD IN LEFT NIBBLE
018D C1      POP B          ;RESTORE BC
018E C9      RET          ;AND EXIT

0182 CD6801  GHXB:  CALL GHXN          ;GET LEFT NIBBLE
0183 D8      RC           ;LEAVE IF NON-HEX
0184 C5      PUSH B          ;SAVE BC
0185 07      RLC           ;SHIFT
0186 07      RLC           ; TO
0187 07      RLC           ; LEFT
0188 07      RLC           ; NIBBLE
0189 47      MOV B,A        ;AND SAVE IN B
018A CD6B01  CALL GHXN          ;GET RIGHT NIBBLE
018B DA9301  JC $+4          ;JMP IF NON-HEX
018C 80      ADD B          ;ADD IN LEFT NIBBLE
018D C1      POP B          ;RESTORE BC
018E C9      RET          ;AND EXIT

0192 80      ADD B          ;ADD IN LEFT NIBBLE
0193 C1      POP B          ;RESTORE BC
0194 C9      RET          ;AND EXIT

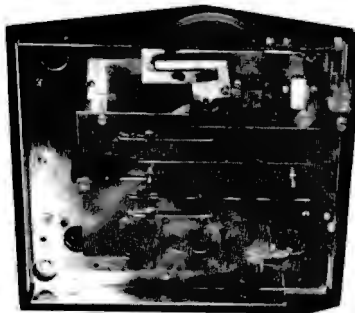
```




DELTA ELECTRONICS

POST OFFICE BOX 2, AMESBURY, MASSACHUSETTS 01913

Phone (617) 388-4705

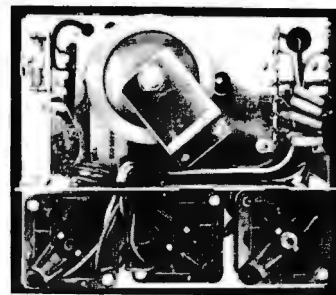


Data Cassette Transport

We have a limited number of these digital tape transports left. They are mechanically complete, but do not include drive electronics or recording heads. Use as the basis for your own design bulk storage unit. Each deck has 3 motors, one driving the capstan and one each for the take up reels. Each motor is 24v AC, split capacitor synchronous type. The transport has a slot loading mechanism for the cassette. 7" x 7 1/2" x 3 3/4", 7 lbs.

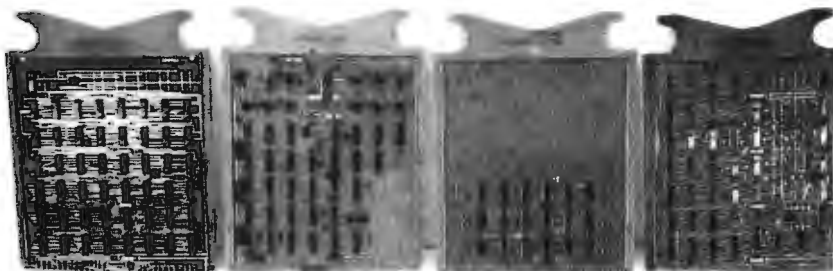
STOCK NO. B5222

\$17.95 ea, 2/35.00



7400 series TTL Parts Special

We have a series of surplus computer boards loaded with 7400 series TTL ICs, which are easily removed for reuse. We select 2 boards (samples shown) and guarantee at least 60 ICs, but most will have more than 70. The ICs are made mostly by T.I., and include gates, counters, flip flops, one shots, shift registers, decoders, etc. Less than 10 cents an IC, many of which sell for over \$1. Shipping weight 1 lb. each.



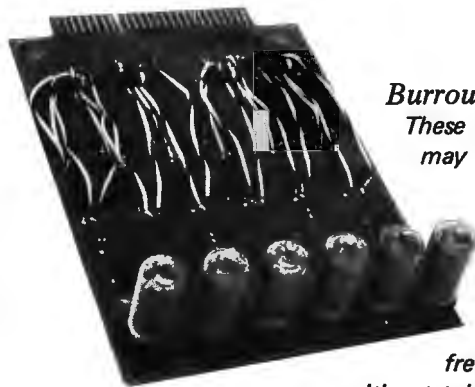
STOCK NO. B5409

2 boards for \$5.00, 5 boards for \$10.00

Burroughs Decade Counters,

Burroughs Series C2506 20 Mhz Decade Counters. 4 to 7 digits

These counters are self contained decade counters with 4, 5, 6, or 7 digits, and may be ganged together for more digits. They are rated at 20 Mhz minimum, 25 Mhz typical. By using one of the new fast decade counter ICs in the first stage, they could be used at 200 Mhz or higher. Each digit has a decade counter, strobed latch, and decoder/driver (with socket) for a B-5750 or similar Nixie tube. The BCD outputs are brought out for external use (or even for driving 7 segment or other displays). All inputs and outputs are TTL, power requirements are +5v and +200v. Ideal for frequency counters, DVMs, digital thermometers, etc. We will sell them with or without tubes. 1973 distributor prices were \$91 for the 4 digit model, \$167 for 7 dig.



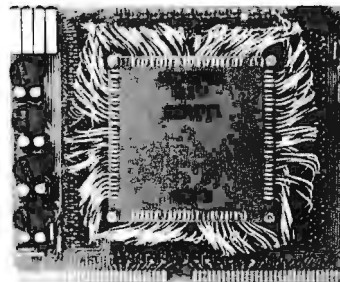
Burroughs C-2506-4	4 digits, without tubes	STOCK NO. B5134	\$18.00
	4 digits, with tubes	STOCK NO. B5134T	\$29.50
Burroughs C-2506-5	5 digits, without tubes	STOCK NO. B5135	\$22.50
	5 digits, with tubes	STOCK NO. B5135T	\$36.00
Burroughs C-2506-6	6 digits, without tubes	STOCK NO. B5136	\$27.00
	6 digits, with tubes	STOCK NO. B5136T	\$42.50
Burroughs C-2506-7	7 digits, without tubes	STOCK NO. B5137	\$31.50
	7 digits, with tubes	STOCK NO. B5137T	\$49.00

10,816 bit Core Memory Board

If you have ever lost a long program because of a power failure, you are aware of the value of non-volatile magnetic core memory. These Burroughs memory boards are organized as a 52 x 52 x 4 bits wide (2704 x 4 bits), for a total of 10,816 bits. Or, 1352 8 bit words. We were able to locate some basic data for this memory. 7 1/2" x 9" x 1 1/4". Less than 1/10 cent per bit! See the July 76 BYTE for an excellent discussion on using core memories.

STOCK NO. B5316 2 lbs.

\$9.95 each, 4/35.00



Send for our latest free catalog. Minimum order \$5, phone orders welcome. Include sufficient postage (2 lbs min.), excess will be refunded. BANKAMERICARD & MASTERCHARGE welcome, ALL numbers needed for processing. Minimum charge \$15.

```

; JUMP TO LOCATION 1234H
;
; IN ADDITION TWO OTHER MODES ARE POSSIBLE.
; THE COMMAND
; ** J (CR)
;
; WILL CAUSE THE ADDRESS SAVED AS A RESULT OF
; A RST 0 TO BE USED FOR THE EXECUTION ADDRESS.
; INSERTING A RST 0 IN A PROGRAM AS A BREAKPOINT
; WILL CAUSE THE ENTIRE MACHINE STATE TO BE SAVED
; AND THE J (CR) WILL RETURN YOU TO THE POINT AFTER
; THE RST 0.
; OF COURSE, IF THE RST 0 REPLACED PART OF AN
; INSTRUCTION YOU MUST REPLACE THE RST 0 AND
; MODIFY THE ADDRESS WITH 'RP=' SO THAT YOU WILL GET
; BACK INTO THE PROGRAM AT THE PROPER PLACE.
;
; ALSO
; THE COMMAND
; ** J (LF)
;
; WILL CAUSE THE ADDRESS ENTERED WITH THE 'X'
; COMMAND TO BE USED AS IF IT WERE TYPED IN.
;
; THE CARRIAGE RETURN AND LINE FEED RESPONSES WILL
; CAUSE THE ADDRESS TO BE TYPED FOR VERIFICATION AND
; AFTER THE ADDRESS THE 'OK?' PROMPT WILL REQUIRE A
; SPACE REPLY FOR EXECUTION TO PROCEED. OTHERWISE
; THE OPERATION WILL BE ABORTED.
;
;
0242 CD3A01 CALL SPACE ;COMMAND 'J'
0245 CD9501 CALL GHXW ;SPACE
0246 D26102 JNC JMP3 ;GET ADDRESS
0248 FEDD CPI CR ;HEX ADDRESS ENTERED
024D C25602 JNZ JMP1 ;SEE IF CR RESPONSE
0250 BA0E0C LHL D SVPC ;NO, GO CHECK FOR LF
0253 C35E02 JMP JMP2 ;GET SAVED PC VALUE
0256 FEDA JMP1 ;AND GO PROCESS
0258 C2CA01 CPI LF ;CHECK FOR LF RESPONSE
025B 2A200C JNZ ILLCH ;ALL OTHERS ILLEGAL
025E CD6001 LHL XEQAD ;GET XEQ ADDRESS FROM 'X'
0261 C08E02 JMP2 ;CALL THXW ;TYPE ADDRESS
0264 221E0C JMP3 ;CALL OK? ;OK?
0267 3EC3 SHLD GOGO+1 ;SET UP FINAL JUMP
0269 321D0C MVI A,0C3H ;JMP COMMAND
026C 31140C STA GOGO ;TO RAM
026F D1 LXI SP,SVPC ;RESTORE REGS
0270 C1 POP D ;TO
0271 F1 POP B ;VALUES
0272 2A100C POP PSW ;IN RAM
0275 F9 LHL SVSP ;SAVED SP
0276 2A120C SHL ;SET NEW SP
0279 C31D0C LHL SVHL ;AND HL
; JMP GOGO ;AND EXECUTE

;
; COMMAND 'G' - DIRECT GOTO ADDRESS
;
027C CD9503 GOTO: CALL PU3 ;GET HEX ADDRESS
027F EB XCHG ;TO HL
0280 C08E02 CALL OK? ;Vfy
0283 E9 PCHL ;THEN JMP ADR

;
; COMMAND 'X' - SET EXECUTION ADDRESS FOR 'J'
;
0284 CD9503 GETXA: CALL PU3 ;GET HEX ADDRESS
0287 EB XCHG ;TO HL
0288 22200C SHLD XEQAD ;SAVE IT
028B C3A000 JMP NEXT ;AND BACK TO NEXT

;
; OK? - ROUTINE TO VERIFY OPERATION
;
; CALLING SEQUENCE ...
;
; CALL OK? ;VERIFY
; ... ;RETURN HERE IF SPACE
; ;ABORT IF NOT
;
; ALL REGISTERS PRESERVED
;
028E F5 OK?: PUSH PSW ;SAVE PSW
028F E5 PUSH H ;AND HL
0290 214605 LXI H,M7 ;ADR OF 'OK?' MSG
0293 CD1B01 CALL M2G ;PRINT IT
0296 214D05 LXI H,M2 ;POSSIBLE ABORT
0299 CD0001 CALL CHIN ;GET ANSWER
029C FE20 CPI ' ' ;SPACE?
029E C2C300 JNZ ILLEG+3 ;NO, GO ABORT
02A1 E1 POP H ;RESTORE HL
02A2 F1 POP PSW ;AND PSW
02A3 C9 RET ;AND LEAVE

;
; ROUTINE TO CHECK FOR LAST OPERATION IN
; DUMP, FILL, MOVE, ETC.
;
;
02A4 E5 LAST: PUSH H ;SAVE MEM VECTOR
02A5 19 DAD D ;ADD NEGATIVE END ADDRESS
02A6 DAA000 JC NEXT ;DONE IF CARRY
02A9 E1 POP H ;RESTORE VECTOR
02AA E3 INX H ;BUMP AND
02AB C9 RET ;EXIT

;
; COMMAND 'M' - MOVE MEMORY BLOCK
;
; FORMAT
; ** M XXXX YYYY ZZZZ OK?
;
; WILL MOVE THE BLOCK OF MEMORY STARTING AT
; XXXX AND ENDING AT AND INCLUDING YYYY TO THE
; BLOCK STARTING AT ZZZZ.
;
; ***** THE FOLLOWING RESTRICTIONS APPLY! *****
;
; EITHER ZZZZ <= XXXX
;
; OR ZZZZ > YYYY
;
; THE ROUTINE MOVES BYTES IN ASCENDING MEMORY ORDER
; SO IF THE HEX ADDRESS VALUES DO NOT SATISFY
; THE ABOVE RULES, MOVED DATA WILL OVERWRITE DATA TO
; BE MOVED.
;
02AC CD9503 MOVE: CALL PU3 ;GET XXXX
02AF D5 PUSH D ;SAVE ON STACK
02B0 CD9503 CALL PU3 ;GET YYYY TO DE
02B3 CD9503 CALL PU3 ;GET ZZZZ TO DE, YYYY TO HL
02B6 EB XCHG ;DE=Y, HL=Z, TOP=X
02B7 E3 XTHL ;DE=Y, TOP=Z, HL=X
02B8 CD1302 CALL NEGDE ;DE=-Y, TOP=Z, HL=X
02BB C08E02 CALL OK? ;
02BE 7E MOV A,M ;GET THRU X
02BF E3 XTHL ;HL=Z, TOP=X
02C0 CDB001 CALL STORE ;CHECKED STORE
02C3 23 INX H ;BUMP Z
02C4 E3 XTHL ;RESTORE
02C5 CDA402 CALL LAST ;CHECK FOR END
02C8 C3BE02 JMP MOVE ;AND CONTINUE

;
; COMMAND 'F' - FILL A BLOCK OF MEMORY WITH VALUE
;
; FORMAT
; ** F XXXX YYYY VV OK?
;
; WILL CAUSE MEMORY LOCATIONS XXXX THRU YYYY
; INCLUSIVE TO BE SET TO THE VALUE VV (HEX).
;
02CB CD9503 FILL: CALL PU3 ;DE=X
02CC CD9503 CALL PU3 ;DE=Y, HL=X
02CD CD1302 CALL NEGDE ;DE=-Y
02CE CD3A01 CALL SPACE
02CF D8801 CALL GHXB ;GET VV -> 'A'
02D0 DAA01 JC ILLCH ;MUST BE VALID HEX
02D1 C08E02 CALL OK?
02D2 CDB001 FILL: CALL STORE ;STUFF IT
02D3 CDA402 CALL LAST ;CHECK IT
02D6 C3BE02 JMP FILL ;AND CONTINUE IT

;
; REGISTER EXAMINE/MODIFY ROUTINE
;
; THE MONITOR COMMAND 'R' FOLLOWED BY A SINGLE
; CHARACTER WILL CAUSE THE ENTRY SAVED
; CONTENTS OF THAT REGISTER TO BE PRINTED AND A
; MODIFICATION ACCEPTED, IF THE 'R' IS FOLLOWED BY
; A CR THEN ALL OF THE REGISTERS WILL BE PRINTED.
;
; RA - ACC
; RF - FLAGS, PSB
; RB - B
; RC - C
; RD - D
; RE - E
; RH - H
; RL - L
; RS - SP
; RP - PROGRAM COUNTER IF MONITOR 'CALLED'
;
; R(CR) - PRINT ALL REGISTERS
;
; REGISTERS S AND P WILL BE PRINTED AS 4 HEX DIGITS
; AND MODIFICATIONS TO THEM MUST BE 4 DIGITS ALSO.
;
02E9 4164243 RXLST: DB 'AFBCDEHL',0 ;REGISTER LIST
02ED 4445484C
02F1 00

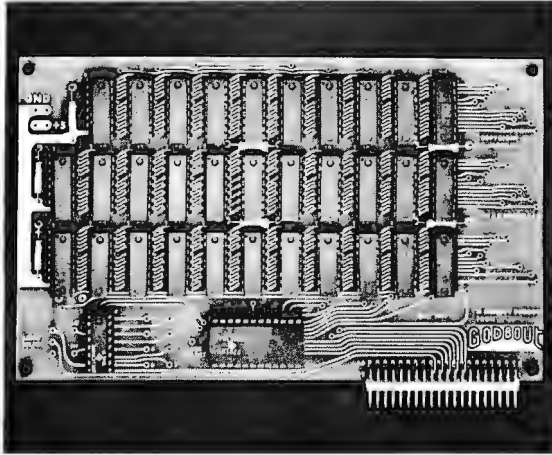
;
; REGET:
; FROM COMMAND 'R'
; GET REGISTER ID
; CHECK FOR CR
; DO ALL IF CR
; SAVE ID
; TYPE
; =
; CALL TYPE
; RESTORE ID
; LXI D,SVPC ;ADDRESS OF PC
; CPI 'P' ;SEE IF
; JZ RX2 ;PRINT PC
; INX D ;POINT TO
; SP ;SP
; CPI 'S' ;CHECK S
; JZ RX2 ;DO SP
; MOV B,A ;SAVE ID
; LXI H,RXLST ;LIST VECTOR
; LXI D,SVA ;ADDRESS OF 'A' STORAGE
; MOV A,M ;GET TABLE ID
; ANA A ;CHECK FOR END
; JZ ILLCH ;NOY IN TABLE

```

GODBOUT has Board kits for you ...

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

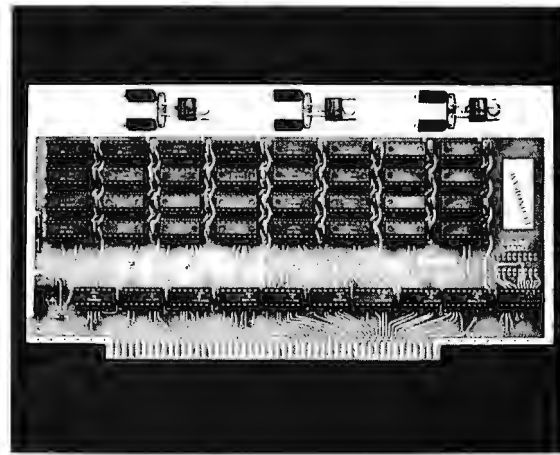
TERMS: Add 1 lb. postage per kit. Californians add tax; no COD orders. For Mastercharge® or BankAmericard® orders, call (415) 357-7007, 24 hours. Have you sent for our flyer? It's free.



4Kx8

This is a multipurpose, simple, and cost-effective memory. Does your 6800, 6502, 8008, SC/MP, or other processor need some memory? The Naked Ram is compatible with any system using a bi-directional data buss. Own a JOLT system? The Naked Ram is directly compatible...it even comes with an appropriate 40 pin connector and is the same size. This kit uses 2112s, and draws 750 ma or less to keep you on good terms with your power supply. Sockets included for all ICs, too.

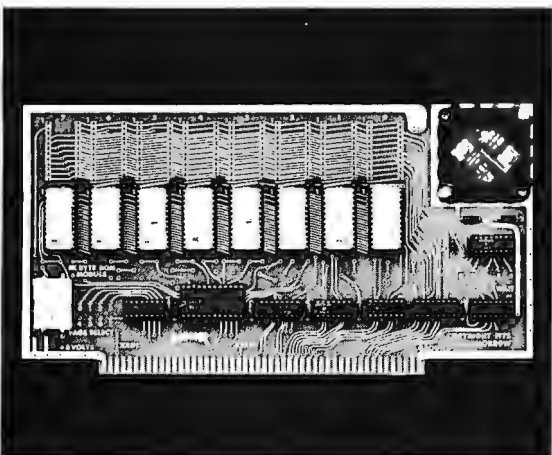
NAKED RAM



ECONORAM™

\$99.95

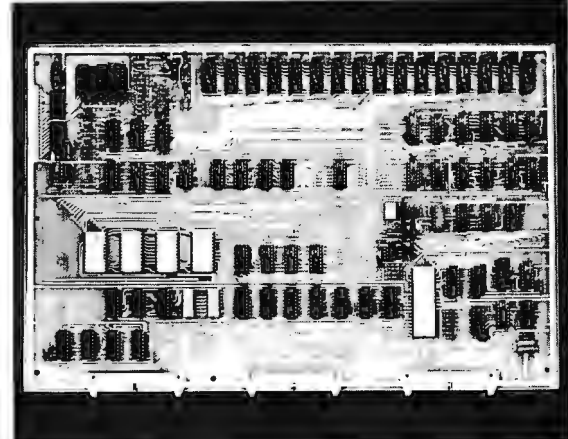
EVERYTHING YOU COULD WANT IN A 4K X 8 MEMORY KIT...Altair 8800/MSAI/Polymorphic plug-in compatible, zero wait states guaranteed over the full temp range of 0 to 55°C, maximum current drain 750 ma (typical 600 ma), buffers on the addresses, data lines, and outputs, enough regulators to give some headroom to your supply, lots of bypass caps, low power Schottky support ICs...with sockets included for all ICs, a really nice PC board, complete instructions, and a warranty against defective **POPPES**.



8080 Software / Econorom

The picture to the left shows our 8080 Software Board (\$159.95). The EROMs are programmed with assembler, editor, and monitor routines for the 8080...a valuable first step to getting away from machine language programming. There's not enough room to describe all the software functions, but we'll send you the software packet and board documentation for \$2.95 (refundable with order for board). Have your own software? Put it in our 4Kx8 ECONOROM board (\$159.95). Program it yourself or have us do the programming for a nominal surcharge. For greater storage, we have an 8K X 8 board for \$249.95; if you're just getting started, there's a 2K version for \$115. All board kits are ALTAIR 8800/MSAI compatible, featuring low power operation, buffered addresses and outputs, sockets for ICs, logic print, and instructions.

C
P
U



16 BITS ... ALIVE & WELL!

You may recall our announcement of a 16 bit machine a few months ago. You may also recall that we didn't solicit any orders, because we frankly didn't know when we'd be able to get our machine out on the market. Constantly changing supply situations, design improvements, and software have kept us occupied for the last year...but the first part is just about wrapped up, so here's a sneak preview of the CPU board. If you look closely you'll see not just the PACE CPU, but a cassette/TTY interface, and on-board RAM and ROM. BUT...we're not soliciting orders (yet), so please don't write us just now. When we have a complete system, available off-the-shelf, you'll see it in our ads. Hang in...

```

031C B8      CMP B      ;CHECK INPUT ID
031D CA8503  JZ RX1      ;FOUND IT
0320 83      INX H      ;NEXT TBL
0321 1B      DCX D      ;NEXT REG
0322 C31703  JMP RX0      ;CONTINUE

0323 1A      RX1:   LDAX D      ;GET THE RGE
0326 CD4801  CALL THXB   ; AND PRINT IT
0329 CD3A01  CALL SPACE  ;SPACE
032C CD8201  CALL GHXB   ;AND WAIT FOR REQUEST
032F DA3603  JC RX1A    ;NON-HEX SO SEE IF CR
0332 12      STAX D      ;STORE INPUT IN RG
0333 C3A000  JMP NEXT   ;AND BACK TO MONITOR
0336 FE0D    RX1A:  CPI CR      ;CR OK
0338 CAA000  JZ NEXT   ;BACK TO MON
033B C3CA01  JMP ILLCH  ;OTHERS ILLEGAL

033E EB      RX2:   XCHG      ;RG ADR TO HL
033F 5E      MOV E,M    ;GET LOW S OR P
0340 23      INX H      ;BUMP VECTOR
0341 56      MOV D,M    ;GET HIGH S OR P
0342 EB      XCHG      ;RG VAL TO HL
0343 CD6001  CALL THXW   ;TYPE WORD
0346 CD3A01  CALL SPACE  ;SPACE
0349 CD9501  CALL GHXW   ;AND GET REQUEST
034C DA3603  JC RX1A    ;IF NON-HEX
034F EB      XCHG      ;STORE RAM VECTOR FOR RG
0350 72      MOV M,D    ;STORE HIGH S OR P
0351 2B      DCX H      ;BUMP VECTOR DOWN
0352 73      MOV M,E    ;STORE LOW S OR P
0353 C3A000  JMP NEXT   ;BACK TO MON

0356 CD3A01  RXTSE:  CALL SPACE  ;SPACE
0359 CD5705  CALL TYPE   ; TYPE ID
035C 3E3D    MVI A,'='   ; THEN
035E CD5705  CALL TYPE   ;
0361 C9      RET      ; AND RETURN

0362 CD8D01  REXAL:  CALL CRET   ;RETURN CARRIAGE FOR ALL REGS
0365 11190C  LXI D,SVA   ;ADDRESS OF 'A'
0368 21E902  LXI H,RXLST ;ID LIST
036B 7E      RXA1:  MOV A,M    ;GET ID
036C A7      ANA A      ;CHECK FOR LAST
036D CA7C03  JZ RXA2    ;DONE SINGLES
0370 CD5603  CALL RTSE   ;TYPE SPACE, ID, AND =
0373 1A      LDAX D      ;GET REG
0374 1B      DCX D      ;BUMP RG PNTR
0375 83      INX H      ; AND LIST PNTR
0376 CD4201  CALL THXB   ;TYPE REGISTER
0379 C36B03  JMP RXA1    ; AND CONTINUE

037C 3E50    RXA2:  MVI A,'P'  ;DO PC
037E CD5603  CALL RTSE   ;SP, ID, =
0381 8A0E0C  LHL D,SVPC  ;GET PC
0384 CD6001  CALL THXW   ; AND PRINT
0387 3E53    MVI A,'S'   ; DO SP
0389 CD5603  CALL RTSE   ;SP,ID,=
038C 8A100C  LHL D,SVSP  ;GET SP
038F CD6001  CALL THXW   ;AND PRINT
0392 C3A000  JMP NEXT   ;BACK TO MON

0395 CD3A01  PU3:   CALL SPACE  ;SPACE
0398 CD9501  CALL GHXW   ;GET HEX WORD
039B DACA01  JC ILLCH   ;IF BAD
039E EB      XCHG      ;SAVE TO DE
039F C9      RET      ; AND RETURN

;
;
; ROUTINES TO PUNCH OR LOAD MEMORY ON TTY
;
; THESE ROUTINES WORK WITH DATA IN THE INTEL
; BINARY FORMAT- THE FORMAT CONSISTS OF A RECORD
; HEADER, UP TO 16 BYTES OF DATA, AND
; A RECORD CHECKSUM.
;
; RECORD FORMAT
;
; HEADER CHARACTER '!'
; HEX-ASCII BYTE COUNT, TWO CHARACTERS
; HEX-ASCII LOAD ADDRESS, FOUR CHARACTERS HLLL
; HEX-ASCII RECORD TYPE, TWO CHARACTERS OO FOR DATA
;                                     OI FOR EOF
; DATA BYTES IN HEX-ASCII, TWO CHARACTERS EACH
;
; HEX-ASCII CHECKSUM, TWO CHARACTERS
;
; THE CHECKSUM IS CALCULATED SUCH THAT THE
; SUM OF ALL OF THE TWO CHARACTER BYTE FIELDS
; WILL BE ZERO.
;
; THE EOF RECORD MAY CONTAIN AN EXECUTION ADDRESS
; IN THE LOAD ADDRESS FIELD. THE LOAD ROUTINE WILL
; TRANSFER CONTROL TO THIS ADDRESS AFTER READING THE
; TAPE IF THE ADDRESS IS NON-ZERO.
;
; HL HAS LOW ADDRESS, DE HAS HIGH ADDRESS
;
03A0 CD9503  PUNCH:  CALL PU3    ;COMMAND 'P'
03A3 CD9503  CALL PU3    ;GET FROM ADDRESS
03A6 CD0C04  CALL PWAIT  ;GET TO ADDRESS
03A9 3E11    MVI A,XON   ;TYPE PROMPT AND WAIT
03AB CD5705  CALL TYPE   ;START
; THE PUNCH
;
; HL HAS LOW ADDRESS, DE HAS HIGH ADDRESS
;
03AE 7D      PMO:   MOV A,L    ;
03AF C610    ADI 16     ;
03B1 4F      MOV C,A    ;
03B2 7C      MOV A,H    ;
;
; ROUTINE TO LOAD HEX-ASCII TAPE
;
LOAD:
;
; COMMAND 'L'
;
0393 CE00    ACI 0      ;
0395 47      MOV B,A    ;
0396 7B      MOV A,E    ;
0397 91      SUB C      ;
0398 4F      MOV C,A    ;
0399 7A      MOV A,D    ;
039A 98      SBB B      ;
039B DAC303  JC PN1     ;RCD LENGTH = 16
039E 3E10    MVI A,16   ;
039F 3E10    JMP PN2     ;
03A0 C3C603  PN1:   MOV A,C    ;LAST RECORD
03A3 79      ADI 17     ;
03A4 C611    PN2:   ORA A      ;
03A6 B7      JZ PDONE   ;
03A7 D5      PUSH D     ;SAVE HIGH
03A8 D5      MOV E,A    ;E=LENGTH
03A9 5F      MVI D,0    ;CLEAR CHECKSUM
03AA D0      CALL CRET   ;PUNCH CR,LF,RBO
03AB 3E10    MVI A,'!'  ;PUNCH HDR
03AC 3E10    CALL TYPE  ;
03AD A2      MOV A,E    ;
03AE D0      MVI D,0    ;
03AF D0      CALL CRET   ;
03B0 3E10    MOV A,L    ;PUNCH LENGTH
03B1 3E10    CALL PBYTE ;
03B2 A2      MOV A,H    ;
03B3 D0      CALL PBYTE ;
03B4 3E10    MOV A,L    ;
03B5 3E10    CALL PBYTE ;
03B6 AF      XRA A      ;
03B7 D0      CALL PBYTE ;PUNCH RECORD TYPE
03B8 D0      CALL PBYTE ;
03B9 7E      PN3:   MOV A,M    ;GET DATA
03BA 23      INX H      ;
03BB D0      CALL PBYTE ;
03BC 1D      DCR E      ;DECR CNT
03BD C2E603  JNZ PN3    ;CONTINUE
03BE AF      XRA A      ;CALCULATE
03BF 92      SUB D      ;CHECKSUM
03C0 CDF803  CALL PBYTE ;AND PUNCH IT
03C1 D1      POP D      ;RESTORE HIGH ADDRESS
03C2 D1      JMP PNO    ;AND CONTINUE
03C3 C3AE03  ;
;
03F8 CD4201  PBYTE:  CALL THXB   ;
03FB 82      ADD D      ;ADD TO SUM
03FC 57      MOV D,A    ;
03FD C9      RET      ;
;
03FE CD2D01  PDONE:  CALL CRET   ;
0401 3E13    MVI A,XOFF ;PUNCH
0403 CD5705  CALL TYPE  ;OFF
0406 CD6605  CALL GETCH ;WAIT FOR GO-AHEAD
0409 C3A000  JMP NEXT   ;BACK TO MON
;
; ROUTINE TO TYPE 'PAUSE' MESSAGE
; AND WAIT FOR TTYI GO-AHEAD
;
040C E5      PWAIT:  PUSH H      ;SAVE H
040D 212A05  LXI H,MS   ;PROMPT
0410 CD1B01  CALL MSG   ;MESSAGE
0413 E1      POP H      ;
0414 CD6605  CALL GETCH ;WAIT FOR GO-AHEAD
0417 C9      RET      ; AND THEN LEAVE
;
; ROUTINE TO PUNCH EOF RECORD
;
0418 CD3A01  PEND:   CALL SPACE  ;
041B CD9501  CALL GHXW   ;GET ADDRESS OR CR
041E D22904  JNC PEND1   ; ADDRESS
0421 210000  LXI H,0     ;SET O ADDRESS
0424 FE0D    CPI CR      ;CHECK FOR CR REPLY
0426 C2CA01  JNZ ILLCH  ; OTHERS ILLEGAL
0429 CD0C04  PEND1:  CALL PWAIT  ;PROMPT PAUSE
042C 3E11    MVI A,XON   ;PUNCH
042E CD5705  CALL TYPE  ; ON
0431 CD2D01  CALL CRET   ;CR,LF,RBO
0434 3E3A    MVI A,'!'  ;
0436 CD5705  CALL TYPE  ;TYPE HDR ;
0439 AF      XRA A      ;
043A 57      MOV D,A    ;ZERO CHECKSUM
043B CDF803  CALL PBYTE ;AND OUTPUT ZERO LENGTH
043E 7C      MOV A,H    ;
043F CDF803  CALL PBYTE ;EXECUTION
0442 7D      MOV A,L    ;
0443 CDF803  CALL PBYTE ; ADDRESS
0446 3E01    MVI A,1     ;RCD TYPE
0448 CDF803  CALL PBYTE ;
044B AF      XRA A      ;
044C 92      SUB D      ;CALCULATE CHECKSUM
044D CDF803  CALL PBYTE ; AND PUNCH IT
;
; PUNCH NULLS
;
0450 0E64    NULLS:  MVI C,100  ;100 NULLS
0452 AF      XRA A      ;
0453 CD5705  CALL TYPE  ;
0456 0D      DCR C      ;
0457 C25304  JNZ $-4     ;CONTINUE
045A C3FE03  JMP PDONE   ;DONE
;
045D CD0C04  NULL:   CALL PWAIT  ;COMMAND 'N'
0460 3E11    MVI A,XON   ;PROMPT PAUSE
0462 CD5705  CALL TYPE  ;PUNCH
0465 C35004  JMP NULLS   ; ON
; GO DO IT
;

```


World's Lowest IC Prices

TTL

7400	14	74151	60
7402	14	74157	60
7404	16	74160	75
7410	14	74161	75
7420	14	74163	75
7427	25	74165	80
7438	25	74173	1.25
7440	14	74174	75
7445	45	74175	75
7447	65	74177	70
7450	14	74180	80
7451	14	74181	1.50
7473	22	74191	1.00
7474	23	74192	70
7493	50	74193	70
7495	49	74198	1.00
74107	29	9602	50
74116	1.00	9300	75
74123	50	9312	70
74150	60		

SCHOTTKY

74S02	25	74S172	3.50
74S37	40	74S175	1.50
74S85	2.00	74S181	3.50
74S139	1.50	74S197	1.50
74S140	50	74S257	1.50
74S153	2.50		

HIGH SPEED

74H00	20	74H51	20
74H01	20	74H52	20
74H04	20	74H74	40
74H10	20	74H103	50
74H11	20	74H106	50
74H40	20		

LOW POWER SCHOTTKY

74LS00	25	74LS164	1.50
74LS02	25	74LS174	1.50
74LS10	25	74LS175	1.50
74LS73	40	74LS193	1.50
74LS75	50	74LS251	1.50
74LS151	85	74LS253	1.50
74LS153	95	74LS257	1.50
74LS157	1.50	74LS258	1.50
74LS163	1.50		

CMOS

4001	16	4027	40
4002	16	4028	60
4006	90	4030	35
4007	16	4040	95
4008	70	4042	60
4011	16	4043	75
4012	16	4044	70
4013	35	4049	38
4015	80	4050	38
4016	35	4066	65
4019	70	4068	35
4020	90	4069	16
4021	95	4071	16
4023	16	4073	16
4024	75	4075	16
4025	20	4516	85
		4585	85

LINEARS

NE555V	43
NE556A	90
741V	30
1458V	52
566V	1.25
567V	1.35
540L	2.00

RAMS

2102	1.50
------	------

PROMS

82S23 S123	1.95
------------	------

Order Minimum \$10.00 Add \$1.00 shipping and handling charge per order California residents add 6% sales tax All orders shipped promptly

Order the famous lasis 6 volume Programmed Learning Course "Microcomputer Design is a Snap" for \$99.50 and receive a special \$10.00 credit on any group of IC's.

Satisfaction 100% guaranteed.

(C.O.D. Orders: Phone (day or night) 408/354-1448

ELTRON

PO Box 2542 B
Sunnyvale, CA 94087

```

0468 CD3A01      CALL SPACE
0468 CD9501      CALL GHXW      JGET BIAS OR CR
046E D27904      JNC LDO       JBIAS ADDRESS ENTERED
0471 210000      LXI H,0       JBIAS 0
0474 FE0D        CPI CR        JCHECK FOR CR
0476 C2CA01      JNZ ILLCH    JOTHERS M-G.
0479 E5          LDQ: PUSH H     JSAVE BIAS
047A AF          XRA A        JKILL
047B 321A0C      STA ECHO     J TTY ECHO
047E 3E1E        MVI A,TOM   JTAPE
0480 CD5705      CALL TYPE    J ON
0483 E1          LDO: POP H     JGET BIAS
0484 E5          PUSH H      JAND RESTORE
0485 CDDC04      CALL RIX    JGET INPUT
0488 063A        SUI B,'!'   JCHCK FOR RCD MARK
048A 90          SUB B
048B C28304      JNZ LDO     JCLEAR CHECKSUM
048E 57          MOV D,A     JGET LENGTH
048F CDE204      CALL BYTE   JZERO ALL DONE
0492 CACA04      JZ LDR      JSAVE LENGTH
0495 5F          MOV E,A     JGET HIGH ADDRESS
0496 CDE204      CALL BYTE   J AND SAVE
0499 F5          PUSH PSW    JROUTINE TO GET A CHARACTER FROM THE TTY
049A CDE204      CALL BYTE   JGET LOW ADDRESS
049D C1          POP B       JFETCH MSBYTE
049E 4F          MOV C,A     JBC HAS ADDRESS
049F C5          PUSH B      JSAVE VECT
04A0 E3          XTHL       J TO HL
04A1 22220C      SHLD BLKAD JSAVE BLOCK ADDRESS
04A4 E3          XTHL       J IN CASE OF ERROR
04A5 C1          POP B       JRESTORE
04A6 09          DAD B       JADD TO BIAS
04A7 CDE204      CALL BYTE   JGET TYPE
04AA CDE204      LD1: CALL BYTE JGET DATA
04AD CDB001      CALL STORE JAND STORE IT
04B0 23          INX H
04B1 1D          DCR E
04B2 C2AA04      JNZ LDI     JCONTINUE
04B5 CDE204      CALL BYTE   JGET CHECKSUM
04B8 CA8304      JZ LDO     JCONTINUE
04BB 213805      LXI H,M6   JCHECKSUM ERROR
04BE CD1B01      CALL MSG    J MSG
04C1 2A280C      LMLD BLKAD JADDRESS OF THIS BLOCK
04C4 CD6001      CALL THXW  J FOR REFERENCE
04C7 C3A000      JMP NEXT   JAND EXIT

04CA CDE204      LD2: CALL BYTE JGET MSB OF XEQAD
04CD 67          MOV H,A
04CE CDE204      CALL BYTE
04D1 6F          MOV L,A
04D2 B4          ORA H
04D3 3E14        MVI A,TOFF JTAPE RDR
04D5 CD5705      CALL TYPE   J OFF
04D8 CAA000      JZ NEXT    JMON IF NO XEQAD
04DB E9          PCHL      JGO TO ROUTINE

04DC CD6605      RIX: CALL GETCH
04DF E67F        ANI 7FH
04E1 C9          RET

04E2 CD8201      BYTE: CALL GHXB JGET TWO CHARS
04E5 4F          MOV C,A
04E6 82          ADD D
04E7 57          MOV D,A
04E8 79          MOV A,C
04E9 C9          RET

J$$$$$$
J
J SYSTEM MESSAGES
J

04EA 0D0A0A41 M0: DB CR,LF,LF,'AMS80 V2-0',LF,-1
04EE 4D533830
04F2 2056322E
04F6 300AFF
04F9 0D0A7F2A M1: DB CR,LF,RBO,'** ',-1
04FD 2A20FF
0500 20495320 M2: DB ' IS UNDEFINED',-1
0504 554EA445
0508 46494E45
050C 44FF
050E 203F3FFF M3: DB '??',-1
0512 140D0A7F M4: DB TOFF,CR,LF,RBO,'MEM WRITE ERROR AT ',-1
0516 4D454D20
051A 57524954
051E 45204552
0522 524F5220
0526 415420FF

052A 20504155 M5: DB ' PAUSE ',-1
052E 534520FF
0532 14204348 M6: DB TOFF,' CHKSM ERR, BLOCK ',-1
0536 4B534D20
053A 4552522C
053E 20424C4F
0542 434B20FF
0546 20204F4B M7: DB ' OK? ',-1
054A 3F20FF
054D 2041424F M8: DB ' ABORTED! ',-1
0551 52544544
0555 21FF

JSYSTEM I/O ROUTINES
JUSER IS TO PATCH HIS OWN TELETYPE
JROUTINES HERE
JROUTINE TO TYPE A CHARACTER
JCALLING SEQUENCE

0557 F5          TYPE: PUSH PSW JSAVE CONTENTS OF 'A'
0558 D601        IN 1 JINPUT TTY STATUS
055A E604        ANI 4 JTEST FOR BUSY
055C C25805      JNZ TYPE+1 JIF BUSY, KEEP TRYING
055F F1          POP PSW JRETRIEVE THE DATA
0560 F5          PUSH PSW JAND SAVE IT AGAIN
0561 2F          CMA JPREPARE THE DATA
0562 D300        OUT 0 JOUTPUT IT
0564 F1          POP PSW JRESTORE 'A'
0565 C9          RET

JTHIS ROUTINE WORKS IN MY SYSTEM
JBUT MAY NOT WORK IN YOURS

JROUTINE TO GET A CHARACTER FROM THE TTY
JCALLING SEQUENCE
J CALL GETCH JGET CHARACTER
J ***** JRETURN HERE WITH CHARACTER
J IN 'A'

JALL REGISTERS PRESERVED EXCEPT 'A' WHICH
JCONTAINS THE INPUT CHARACTER

0566 DB01        GETCH: IN 1 JINPUT TTY STATUS
0568 E601        ANI 1 JTEST FOR READY
056A C26605      JNZ GETCH JKEEP TRYING IF NOT READY
056D DB00        IN 0 JGET THE CHARACTER
056F 2F          CMA JPROCESS IT
0570 C9          RET

JTHIS ROUTINE WORKS IN MY SYSTEM BUT MAY NOT
JWORK IN YOURS

0571          ENDRON EQU $ JBOUNDARY MARKER
J
J SYSTEM RAM AREA DEFINITIONS
J
0571          ORG RAM
J
J USER RESTART VECTORS 1 - 7
J
RST1: DS 2
RST2: DS 2
RST3: DS 2
RST4: DS 2
RST5: DS 2
RST6: DS 2
RST7: DS 2
J
J MONITOR REGISTER SAVE AREA
J

0C00          SVPC:
0C02          SVPCL: DS 1 JSAVED PC LOW
0C04          SVPCH: DS 1 JSAVED PC HIGH
J
0C10          SVSP:
0C11          SVSPL: DS 1 JSAVED SP LOW
0C11          SUSPH: DS 1 JSAVED SP HIGH
SVHL:
J
0C12          SVL: DS 1 JSAVED L
0C13          SVH: DS 1 JSAVED H
J
0C14          SVE: DS 1 JSAVED E
0C15          SVD: DS 1 JSAVED D
0C16          SVC: DS 1 JSAVED C
0C17          SVB: DS 1 JSAVED B
0C18          SVF: DS 1 JSAVED PSB, FLAGS
0C19          SVA: DS 1 JSAVED ACC
J
0C1A          ECHO: DS 1 JCHIN ECHO FLAG, <=>ECHO
J=0 = NO ECHO
J
0C1B          ADR: DS 2 JEXAMINE/MODIFY ADDRESS
J
0C1D          TMPA: JTEMP STORAGE LOCATIONS
GOGO: DS 3 J'JUMP' STORAGE
J
0C20          XEQAD: DS 2 J'X' EXECUTION ADDRESS
J
0C22          BLKAD: DS 2 J'L' BLOCK ADDRESS
J
0C24 00          NOP JPROGRAM BOUNDARY MARKER
J
0000          END
0

```


turn off instructions to the machine. These routines will be different for other users because of the hardware in the interface. This listing incorporates the routines used on the development system as an example only.

Modifications to the Monitor

This monitor was assembled to reside in low memory. Thus when the system is first turned on, the power reset circuit will put it into the monitor. A large area of ROM in

low memory has been saved for future expansion. This expansion area will include drivers for high speed paper tape devices and for a cassette interface.

Summary

Although presented as an AMSAT users monitor, its use is by no means limited to AMSAT members. Anybody who has an 8080 system will be able to use, modify or otherwise operate upon this software. ■

Listing 2: An example of use of the AMS80 monitor.

(All numbers are in hexadecimal notation.)

The "sign on" message is printed whenever the monitor is initiated. A command is entered to examine the contents of memory location 1234. It contains the number 42. The number 01 is typed in, the monitor echoes the 01, and then a line feed character is typed in, advising the monitor to examine the next location. Locations 1235 through 1237 are examined and changed in the same manner. At 1238 the sequence is terminated. The contents of memory locations 1234 through 1238 are then examined without changes. Then at 1238, the address pointer is

backed up by the " - " command to examine the contents of location 1237.

The block of memory from 1234 to 1238 is then displayed.

A command is then entered to fill each location within the block of memory from 1234 to 1238 with the number 76 (the 8080 HLT instruction). After entering parameters, the computer asks "OK?". If a "space" character is typed, the fill command is executed. The contents of the block of memory are then displayed and sure enough the 76s have been entered.

Next the contents of the block of memory locations from 1300 to 1304 are displayed. After this command to move the contents of memory locations 1234 to 1238 to a block starting at location 1300 is given. The monitor again asks "OK?" so you can verify addresses, after which depressing the "space" key causes the move command to be executed. The contents of memory locations 1300 to 1304 are then displayed to verify the execution of the move command. If a move command (or other command of this type) is then entered incorrectly, it can be aborted by depressing the "CR" key after the query "OK?".

The contents of the registers are examined using the "R" command. The contents of the accumulator are changed using the "RA" command after which all registers are again examined using "R".

A punch command is then entered and a "PAUSE" is typed out by the monitor. When the tape punch is deemed to be ready, typing a "space" character causes the command to execute and punch the tape as instructed. The program then pauses, and when the tape punch is deemed to be off, execution continues after another "space" character is depressed. An end of file mark is then punched in a similar manner.

Finally an execution address of 3800 is set up with "X" and the program is entered with a "J" command. The program starting at memory location 3800 begins executing, printing out the message "8080 V 3.0" which ends this example. ■

```

AMS80 V2.0 Sign on message
** A 1234 Examine Memory Location 1234
1234 42 01 01 Change Contents
1235 01 02 02
1236 CD 03 03
1237 A4 04 04
1238 02
** A 1234 Examine without changing contents
1234 01
1235 02
1236 03
1237 04
1238 02 - Back up one location
1237 04
** D 1234 1238
1234 01 02 03 04 02 Display block
** F 1234 1238 76 OK? Fill memory area with constant
** D 1234 1238
1234 76 76 76 76 76 Display it
** D 1300 1304
1300 F1 11 0E 0C FE Display another area
** M 1234 1238 1300 OK? Move block
** D 1300 1304
1300 76 76 76 76 76 Verify that data was moved
** M 1342 1234 1345 OK? ABORTED! Aborted function
** R Examine registers
A=00 F=46 B=0D C=0D D=00 E=03 H=00 L=C3 P=3CD2 S=2FE2
** RA=00 12 Change accumulator [A]
** RA=12
** RA=12 00 Change it back
** R Examine registers
A=00 F=46 B=0D C=0D D=00 E=03 H=00 L=C3 P=3CD2 S=2FE2
** P 1243 1248 PAUSE
:061243003A01CD9501D235 Punch tape

** E PAUSE
:00000001FF Punch end of file mark

** X 3800 Set up location of Intel Monitor
** J 3800 OK? Go to it
8080 V3.0 Program executing
    
```

Note: The program expects a new command after printing asterisks.

Examine registers

MC14412 UNIVERSAL MODEM CHIP

MC14412 contains a complete FSK modulator and de-modulator compatible with foreign and USA communications. (0-600 BPS)

FEATURES:

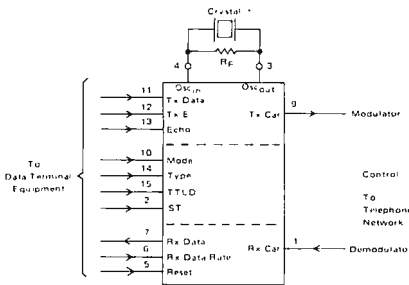
- . On chip crystal oscillator
- . Echo suppressor disable tone generator
- . Originate and answer modes
- . Simplex, half-duplex, and full duplex operation
- . On chip sine wave
- . Modem self test mode
- . Selectable data rates: 0-200
0-300
0-600

- . Single supply
VDD=4.75 to 15VDC - FL suffix
VDD=4.75 to 6 VDC - VL suffix

TYPICAL APPLICATIONS:

- . Stand alone - low speed modems
- . Built - in low speed modems
- . Remote terminals, acoustic couplers

MC14412FL.....	\$28.99
MC14412VL.....	\$21.74
6 pages of data.....	.60
Crystal for the above.....	\$4.95



MC14411 BIT RATE GENERATOR.

Single chip for generating selectable frequencies for equipment in data communications such as TTY, printers, CRT's or microprocessors. Generates 14 different standard bit rates which are multiplied under external control to 1X, 8X, 16X or 64X initial value. Operates from single +5 volt supply. MC14411..... \$11.98
4 pages of data..... .40
Crystal for the above..... \$4.95

FLUID LEVEL DETECTOR LM1830 is used to compare external probe to probe resistance with a reference resistor. Use as water level detector or with any polar fluid. Smoke level detector can be made by substituting a photo cell for the fluid probes...etc. LM1830...\$2.99. Specs .60

PRECISION REFERENCE DIODE. LM399 is a temperature-stabilized monolithic sensor and buffer. Internal heater stabilizes output to .0002%/°C. Buffer reduces zener impedance to .5 ohm. Use in lab standards and calibrators. Initial break down tolerance is 2%. LM399H.....\$5.95. Specs for LM399H...60c

4 DIGIT COUNTER. MM74C926 is a 4 digit counter with 7 segment output. Carry output for cascading and internal display select allows outputting of counter or set of internal latches. 3 to 6V operation. Great for clocks, event and frequency counters. MM74C926 - with spec sheet..... \$12.00

FOUR QUADRANT MULTIPLIER. MC1495L provides output as a linear product of two analog input. Use for frequency doubler, balanced modulator/demodulator, electronics gains control. MC1495L.....\$5.50
6 pages of specs......60
9 pages of applications......90

TELETYPE CODE CONVERSION CHIP
MM5220BL converts 5 level Baudot into 8 level ASCII. Use this chip to make your old TTY talk to your new computer. MM5220BL..... \$18.00
Specs for the above..... .30

MOS TIME BASE KIT.
Only 1" X 1.5". Input 5 to 15 VDC, output is 60HZ square wave for portable or mobile clocks. PC board is drilled! MTBK-60HZ..... \$5.88

78H05 Voltage regulator. Fairchild 5V, 5A, TO-3 regulator. Take care of those heavy current requirements without separate regulator/pass transistor combinations. Use it with the same ease of installation as the 309K(same pin arrangement)...with specs..... \$11.25.

LM317 Voltage Regulator. 1.5A, 3 terminal adjustable regulator in TO-3 case. Adjusts from +1.2V to +37V. Complete overload protection. .1% load regulation, .01%/V line regulation. No need to stock assorted regulators - just stock resistors.....\$4.99
Specs for the above..... .70

DATA BOOKS BY NATIONAL SEMICONDUCTOR
DIGITAL. Covers TTL, DTL, Tri-State, etc. \$3.95
LINEAR. Covers amplifiers, pre-amps, op-amps, .. \$3.95
LINEAR APPLICATIONS. Dozens of application notes and technical briefs covering the use of op-amps, regulators, phase locked loops and audio amps..... \$3.25
CMOS Gates, Flip Flops, registers, functional blocks \$3
VOLTAGE REGULATORS. A must for anyone making a power supply. Complete theory including transformers, filters, heat sinks, regulators, etc..... \$3.00
MEMORY. Information on MOS and Bipolar memories: RAMS, ROMS, PROMS and decoders/encoders. \$3.95
INTERFACE. Covers peripheral drivers, level translators, line driver/receivers, memory and clock drivers, sense amps display driver and opto-couplers..... \$3.95
(Outside U.S., add postage for 1.5lbs)

DATA BOOKS FROM FAIRCHILD.
uA Linear. 776 pages of data and applications for Fairchild linear ICs. Great value..... \$4.25
MOS/CMOS/nMOS/pMOS/CCD. Data and applications on MOS and charge coupled devices including preliminary data on new and future offerings. Want to know about 16K charge coupled line addressable memories?.....\$3.95

AMPL'ANNY

Says

GET YOUR HIGHER EDUCATION AT
GOOD OLD TTI* WITH THE LATEST
IN DATA BOOKS AND MANUALS
*Tri-Tek, Inc.



GOLD CHIP

Linear Integrated Circuits

Brand new process by RCA in which the aluminum metalization has been replaced by gold. The chip is then hermetically sealed. What this means to you is unprecedented reliability and uniformity. Plastic parts that meet mil specs!! Tri-Tek is proud to be the first to bring this new level of performance to you at SURPLUS PRICES. Why buy regrades??

- CA301A..Improved, general purpose op-amp, 8 pin dip..59c
- CA307...Super 741 op-amp. 8 pin dip.....52c
- CA324...Compensated quad op-amp, 14 pin dip.....\$1.80
- CA339A.. Low offset quad comparator. 14 pin dip...\$1.59
- CA741C..Famous general purpose op-amp, 8 pin dip.. 45c
- CA747C..General purpose dual op-amp, 14 pin dip... 82c
- CA748C..Externally compensated 741, 8 pin dip.....49c
- CA1458..General purpose dual op-amp. 8 pin dip.....69c
- CA3401..Quad single supply (5-18V) op amp. 14 pin..89c

Another super buy from RCA. CA555 timer, 8 pin dip. 59c

NEW NATIONAL BOOKS!

- AUDIO HANDBOOK** contains detailed discussions, including complete design particulars, covering many areas of audio with real world design examples...\$3.25
- SPECIAL FUNCTIONS DATA BOOK** contains detailed information for specifying and applying special amplifiers, buffers, clock drivers, analog switches and D/A-A/D converter products.....\$3.25

ULTRASONIC TRANSCIEVER. LM1812 is a special I.C. containing a 12W ultrasonic transmitter, selective receiver, noise rejection circuitry, display driver and keyed modulator. Use in sonars, burglar alarms, liquid level control, direction control for model submarines, etc. LM1812.....\$9.15. Specs and apps.... 60c

PRECISION REFERENCE AMP
LH0070-1H provides a precise 10.0 volts for use in BCD A to D converters or meter calibrators. Typical initial accuracy is .3% (± .03V). Comes in TO-5 can. LH0070-1H.....with specs..... \$5.35

TV CAMERA SYNC GENERATOR. MM5320N I.C. generates all sync signals for T.V. camera. Even color!! MM5320N.....\$18.80.....Specs \$1.00

T.V. CLOCK CHIP SET. MM5318/5841 two chip set forms basis for time of day display on your TV screen. Interesting and convenient way to keep time. MM5318/5841 Set.....\$22.45.....Specs.....\$1.00

D-A CONVERTER BY ZELTEX
8 bit precision hybrid circuit for use in controllers, timers, volt meters, etc. Molded plastic package with P.C. pins. Super buy on this better than usual subsystem. ZELTEX model ZD430. DAC-430.....\$4.95

NEW BOOK!!! "An Introduction to Microcomputers"
This is the book which Fairchild Semiconductor Company called ".....the best darned introduction to the industry to date." Covers everything from basic concepts to a review of real microcomputers. IMC-001.....\$8.00



tri-tek, inc.

6522 NORTH 43RD AVENUE,
GLENDALE, ARIZONA 85301
phone 602 - 931-6949

We pay shipping on all orders over \$10 US, \$15 foreign in US funds. Orders under \$10, please add \$1 handling. Please add insurance, Master Charge and Bank America cards welcome, (\$20 minimum) Telephone orders may be placed 11AM to 5PM daily, Mon thru Fri. Call 602-931-4528. Check reader service card or send stamp for our latest flyers packed with new and surplus electronic components.

Programming Quickies:

8 Bit Fractional Multiplication

Ira Chayut, 3030 Brighton 12th St, Brooklyn NY 11235, submits the following subroutine for a Motorola 6800 which returns the most significant 8 bits of the integer product of two 8 bit operands. The program is shown here beginning at address 0000 hexadecimal. This multiplication is equivalent to treating one number (for example the argument in A) as an integer from 0 to 255 and the second number (for example the argument in B) as a fraction from 1/256 to 255/256.

Ira wrote the routine for use in a digital filtering application where the fractional interpretation was needed. The version submitted was located at addresses 026F to

027F as an artifact of the Motorola Design Evaluation Kit which was used; we've relocated it to location 0000 (but we kept the long form of memory reference to ARG1). To relocate this program at an arbitrary address, the address constants in instructions at locations 0000, 0004, and 000A will have to be changed to reflect the new location of ARG1.

Do you ever spend a spare moment creating a little program or subroutine to explore some of the possibilities of your computer? Write down a symbolic and absolute listing in the language of your computer plus a short paragraph describing the program and its purpose. Then send the result to Quickies, BYTE, 70 Main St, Peterborough NH 03458. Each Programming Quickie published will earn its originator \$20 worth of fame and fortune.

Addr	Hex Code	Label	Op	Operand	Commentary
0000	B7 00 10	MULT	STAA	ARG1	ARG1 := A [save in temporary];
0003	4F		CLRA		A := 0 [initial product sum is zero];
0004	74 00 10	MLOOP	LSR	ARG1	ARG1 := ARG1 / 2;
0007	58		ASLB		CY := MSB(B); B := ASL(B,1);
0008	24 03		BCC	NONADD	if CY = 0 then skip the addition;
000A	BB 00 10		ADDA	ARG1	else A := A + ARG1;
000D	26 F5	NONADD	BNE	MLOOP	if ARG2 NE 0 then reiterate;
000F	39		RTS		else return with result in A;
0010	xx	ARG1			single byte temporary data area, uninitialized;



Computing 1776 Poster

Robert Tinney painted a beautiful oil painting on a bicentennial theme bridging two centuries of America's development. This painting has been reproduced on the cover of this issue, and a full-sized poster in color without the BYTE logo has been printed for you. It will make a perfect wall decoration in your office, home or computer room.

The poster is 20" by 24" (51 cm by 61 cm) large with a white border of 2 inches (5 cm) at all four sides. The image size is 16" by 20" (41 cm by 51 cm), and it is the original size. The price is \$2.95 postpaid, and the poster is shipped in a mailing tube to avoid folding. Only 2,000 copies have been printed on the first run which will be sold on a first come first served basis. So hurry if you want to be among the first to show this beautiful poster to your friends.

BYTE Posters Allow 6 - 8 weeks for processing.
 PO Box 274
 Jaffrey Center NH 03454
 603-924-7217 _____ Posters \$2.95 each

Name _____

Address _____

City _____ State _____ Zip _____

Bill BankAmericard Check Enclosed

Bill Master Charge \$ _____

No. _____ Exp. Date _____

Continued from page 8

for computer people to meet computer people, and for the transfer of technical information through the vehicle of the seminars to be held during the show. And then, of course, there is always the prospect of technological surprises cooked up in the laboratories and workshops of various persons and organizations

A Note About Publicity in BYTE

The planning and execution of an event such as the Personal Computing '76 show must begin a long time in advance of the actual event. When John Dilks first called in January and asked for some publicity for the show, it was eight months in advance, yet in view of the things he had to get done, it felt like the show was to be held the next week. The lead time was nearly nine months for this show, and that might be too short for a large event (which the show has become, thanks to the generous support of interested attendees and manufacturers). For example, another large show mentioned earlier, the the AFIPS National Computer Conference has a cycle of planning which begins more than 18 months ahead of the date of the show.

BYTE has worked quite closely with the

organizers of the Personal Computing '76 show from its inception last January. It is in the interests of those participating in this exciting maturation of computing technology to help call attention to the prospects and utility of personal computing systems. In effect, such events help "sell" the idea and promote a larger market with more options and more products through education. If your club or organization is planning to hold a computer festival, flea market or other event to help publicize the idea of personal computing systems, BYTE will extend a similar hand. We won't promise to give the same coverage to a regional or local show as we would for an event which is clearly national in scope; but the principle is the same in either case: Help bring people with our common interests together for purposes of fun, enjoyment, education and commerce. An important thing to remember is that planning should begin well ahead of the event (three months at least in terms of the magazine production cycle).

I'll be looking forward to meeting many BYTE readers at Personal Computing '76, as has happened in the past at events such as the World Altair Convention in Albuquerque, the Trenton Computer Festival, and the NCC show in New York. ■

At Last!

BYTE T-shirts are here!



At last! No more wardrobe crises! BYTE T-shirts are here! Now you have the perfect garb for computer club meetings, Altair Conventions, playing Shooting Stars and computer chess. (A pair of trousers from your own closet is suggested as an addition to the BYTE T-shirt. The Bytique can't do everything for you.)

BYTE T-shirts are of top quality 100% cotton or cotton-polyester. The original design, by artist Judy Lee Rehling, is silk screened in red on white shirts with blue trim on collars and sleeves, or on blue heather shirts. Each shirt is mailed first class for safe, rapid delivery to you.

The \$5 price includes handling and first class postage.

Send to: In unusual cases, processing may exceed 30 days.

The Bytique
PO Box 274
Jaffrey Center NH 03454

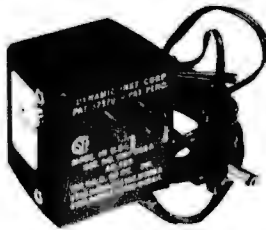
Please send me extra large blue heather
 large white with blue trim and
 medium red letters
 small T-shirts @ \$5 each (includes handling and first class postage)

Total enclosed \$ _____

Bill MasterCharge No. _____ Exp. Date _____
 Bill BankAmericard No. _____ Exp. Date _____

Name _____
 Address _____
 City _____ State _____ Zip _____
 Signature _____

Other colors, styles I'd like to see _____



UNIVERSAL POWER SUPPLY

A unique plug-in supply by Panasonic. Useful for calculators, small radios, charging many & various small NiCad batteries. Adjustment screw plug on the side changes output voltage to 4½, 6, 7½, or 9 volts DC at 100 MA. Output cord with plug, 6 ft long.

No. SP-143C \$4.50 3/\$12

POWER SUPPLY

LAMBDA 5VDC 74 AMP

LV-EE-5-OV \$125.00

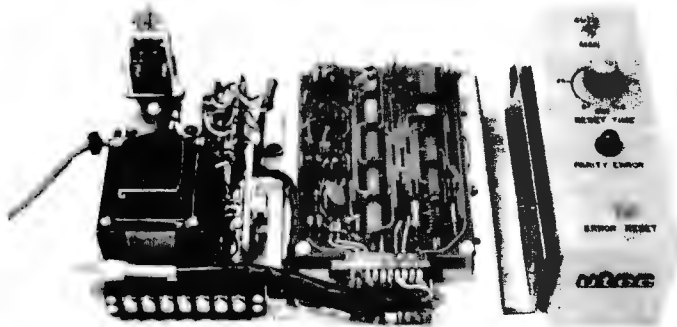
NJE 5/OUP-D5

5 VDC 32 AMP \$75.00

CLOCK KIT \$14.00

Includes all parts with MM5316 chip, etched & drilled PC board, transformer, everything except case.

SP-284 \$14.00 each 2/\$25.00



PARITY DETECTOR

New packaged, made for RCA, detects even or odd parity, baud rate 110, 150 or 134.46. Built-in logic supply for the IC's, operates from standard 115 vac. Control panel allows manual or automatic reset mode of operation. Aluminum enclosure (not shown), covers the electronics. TTY compatible.

Ship wgt. 10 lbs. \$16.50

COMPUTER DISPLAY TUBE

New Sylvania 9 inch CRT, 85 degree deflection, with tinted faceplate. Same as used in Viatron systems (buy a spare). With complete specs.

Ship wgt. 5 lbs. \$15.00

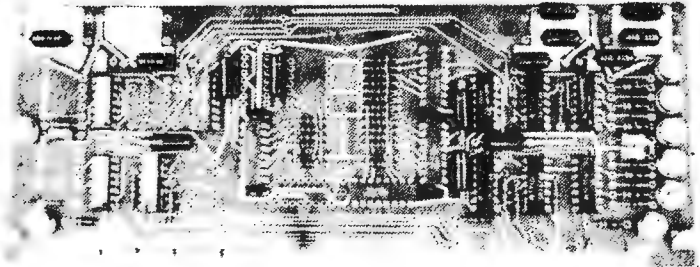
LINEAR by RCA, brand new, gold bond process

301	\$.60	747	\$.82	MM5314	\$3.00
307	.52	748	.50	MM5316	3.00
324	1.80	1458	.96	7001	8.00
339A	1.60	3401	.80		
741	.50	555 timer	.60		

MEMORY SYSTEM \$125.00

New memory system by Honeywell, small ... measures only 9x4x1 inches. 1024 core memory, 1024 words with 8,9,10 bits/word. Random access, with all logic, register, timing, control, core select and sense functions in one package. New, booklet of schematics and data. Looks like a good beginning for a mini-computer. Limited supply on hand.

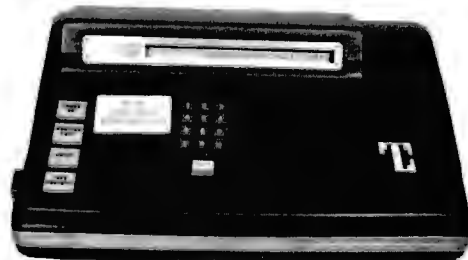
Ship wgt 3 lbs. #SP-79 \$125.00



DATA ENTRY AUDIO TERMINAL

Sends and receives hard copy or audio of touch tone data sent & received. Output 600 ohms for phone lines. ITT touch phone pad with oscillators. Hard copy by strip printer 5 characters per inch, 35 digits visually displayed at once. Prints & receives touch tone codes, digits only. Power supply etc. all in the one case. With two units you can send and receive with monitoring of visual & audio tone at both stations.

Original cost \$1065 Ship wgt. 25 lbs. \$49.00 each



Please add shipping cost on above.

PHONE 617-596-2275 FREE CATALOG SP-8 NOW READY
MESHNA PO Bx 62 E. Lynn Mass. 01904

Meshna

BYTE

reader service

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from this list. Add a 9 cent stamp to the card, then drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by BYTE. This helps us bring you a bigger BYTE.

A116	Action Audio Electronics 101	A 94	Merrimac 95
A 70	American Microprocessors 99	A 18	Meshna 127
A107	Audio Design Electronics 97	A 93	Micon 97
A111	Brigar 105	*	Microcomputer Applications 101
A120	Burkeshire 93	A 71	Micro Peripherals 101
	BYTE's Books 80	A119	Midwest Scientific Inst 48
	BYTE's Poster 124	A 57	Mikos 93
	BYTE's Subs 111	A112	MiniTerm 8
	BYTE's T-Shirts 126	*	MIT'S CIV, 2, 3, 21
A 88	Cambion 99	A 22	National Multiplex 79
A 89	CFR Associates 99	A 40	Ohio Scientific Inst 75
A 83	Computer Mart of NY 99	A 64	Oliver Audio Engineering 99
A 87	Creative Computing 103, 111	A 50	Osborn and Associates Inc 43
A 41	Cromemco 1	A 63	Parasitic Engineering 49, 95
A 97	Data Domain 81	A 85	Per Com Data 69
A 7	Delta 115	A 23	Polymorphic Systems 73
A 78	Digital Group 7	A 24	Processor Tech 24, 25
A 79	Economy Co 49	A110	PTI 97
A102	Eltron 119	A 26	Scelbi 19
A 9	Godbout 117	A 27	SD Sales 106, 107
*	HAL Communications 61	A 59	Solid State Sales 121
*	IMS 13, 83	A 29	Southwest Tech CII
A113	Instrumentation Services 103	A 30	Sphere CIII
*A123	Integrated Computer System 97	A 99	STM Systems 71
A117	Intel 14, 15, 23	A 96	Synchro Sound 99
A118	Intersil 39	A 82	Technical Design Labs 72
A 15	James 113, 125	*	Texas Instruments 34, 35
A122	John Anthony Television 101	A 32	Tri Tek 123
A 90	Logical Services 101	A115	Wilcox 101

*Reader service inquiries not solicited. Correspond directly with company.

BOMB

BYTE's
Ongoing
Monitor
Box

June BOMB Results

Winner of the BOMB bonus for the June issue was Bob Abbott, for his article "Building an M6800 Microcomputer." The three runners-up were Dr Suding's "Systems Approach to a Personal Microprocessor," G H Gable's "Interact with an ELM," "Programming for the Beginner" by Ron Herman.

Feedback is what keeps a linear amplifier in line. Like a linear amplifier, BYTE can use a bit of feedback. The BOMB analysis is done once a month to provide encouragement to authors and some formal feedback on how readers appreciate articles. BYTE pays the winning author a \$50 bonus, so you can encourage the authors you like by voting your preferences. Remember that with few exceptions BYTE authors are just readers who have sat down at their typewriters to tell a story about what they've done or what they know about some aspect of this technology.

PAGE	ARTICLE	LIKED										
		LEAST									BEST	
16	Mooers: Are You an Author?	0	1	2	3	4	5	6	7	8	9	10
26	Guthrie: Mathematical Function Unit—Part 1	0	1	2	3	4	5	6	7	8	9	10
36	Grappel: Randomize Your Programming	0	1	2	3	4	5	6	7	8	9	10
40	Herd: BASIC Star Trek Trainer	0	1	2	3	4	5	6	7	8	9	10
44	Barbier: MSC 8080+ Microcomputer	0	1	2	3	4	5	6	7	8	9	10
50	Brown: How to Do a Number of Conversions	0	1	2	3	4	5	6	7	8	9	10
62	Suding: The Circuit for Z-80s	0	1	2	3	4	5	6	7	8	9	10
76	Baker: SC/MP Fills a Gap	0	1	2	3	4	5	6	7	8	9	10
84	Wadsworth: "8008" Programming—Chapter 3	0	1	2	3	4	5	6	7	8	9	10
108	Allen-Kasser: AMSAT 8080 Standard Debug Monitor	0	1	2	3	4	5	6	7	8	9	10

M 6800

CRT'S PDQ **only \$149**

Compatible with
Motorola Evaluation Kits SWT Kits Cramer Kits

With a CRT interface module from Sphere you can communicate with your computer . . . PDQ! No fooling around with blinking lights, you get up to 32 characters by 16 lines of instant display on your television set or video monitor.

The Sphere CRT module interfaces to your Motorola 6800 product with easy to connect flat ribbon cable bus through 16 lines of address bus, 8 lines of data bus, the VMA, the phase #2 clock and the read/write port.

Mail this coupon today for your CRT PDQ - OK!

Please send me _____ Sphere CRT interface kits.
Enclosed Check/Mastercharge/Bankamericard

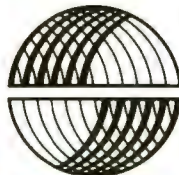
Amount \$ _____ Card No. _____

Name _____

Address _____

City _____ State & Zip _____

Limited offer, prices subject to change • Allow up to 30 days for delivery.



SPHERE
CORPORATION

Distributorships
Available

Dept. 104 P. O. Box 213 Bountiful, Utah 84010 (801) 292-8466

The small wonder



of the micro-world

Measuring just 11" wide x 11" deep x 5" high, and weighing a mere 7 pounds, the Altair™ 680b is a complete, general-purpose computer.

The secret to this revolutionary, small computer is its CPU board. This double-sided board fits along the bottom of the Altair case and plugs directly into the front panel board. It contains the new 6800 microprocessor, 1,024 bytes of RAM memory, a 256 byte PROM monitor, provisions for 768 bytes of additional PROM or ROM, and a single Interface port with a Motorola ACIA serial interface adapter which can be configured either RS-232 or TTY. A five level Baudot interface option is also available.

The Altair 680b can be programmed from front panel switches, or it can be interfaced to a video display terminal, or teletypewriter. Three additional circuit boards can be plugged inside the Altair 680b for further memory and interface expansion. The first of these boards is a 16K static RAM memory board.

Software already developed includes Altair 680 BASIC with all the features of the 8K BASIC previously developed for the Altair 8800. These include Boolean operators, the ability to read or write a byte from any I/O port or memory location, multiple statements per line, and the ability to interrupt program execution and then continue after the examination of variable values. This software takes only 6.8K bytes of memory space and a copy is included free with the purchase of the Altair 680 16K memory board.

Other software includes a resident two pass assembler. The Altair 680b is also compatible with Motorola 6800 software.

The Altair 680b is ideal for hobbyists who want a powerful computer system at an economic price. Altair 680b owners qualify

for membership in the Altair Users Group, and like other Altair owners, they receive a complimentary subscription to **Computer Notes** and complete factory support.

PRICES:

Altair 680b kit with complete, easy-to-understand assembly manual, operator's manual, and programming manual	\$466
Assembled Altair 680b	\$625
Altair 680b Turnkey model kit	\$395
Expander Card 680MB (required to expand 680)	\$ 24
Altair 680BSM 16K static RAM board kit with 680 BASIC	\$685
Altair 680 BASIC when purchased separately	\$200
Baudot option	\$ 42

MAIL THIS COUPON TODAY

Enclosed is a check for \$ _____

Bank Americard # _____ or Master Charge # _____

Altair 680b Kit Assembled Other (specify) _____


enclose \$8 for postage and handling

Please send free information package.

NAME _____

ADDRESS _____

CITY _____ STATE & ZIP _____



2450 Alamo SE/Albuquerque, NM 87106, 505-243-7821

NOTE: Altair is a trademark of MITS, Inc.

Price, specifications subject to change. Please allow up to 60 days for delivery.