dr. dobb's journal of

$1.50

COMPUTER Calisthenics & Orthodontia

Running Light Without Overbyte

August, 1976 | Box 310, Menlo Park CA 94025 | Volume 1, Number 7

A REFERENCE JOURNAL FOR USERS OF HOME COMPUTERS

DON'T KEEP IT A SECRET!

Let us know what exciting new software and systems you are working on. We'll tell everyone else (if you wish). Maybe someone is also working on the same thing. You can work together and get results twice as fast. Or, may be someone else has already done it; no reason for everyone to reinvent the wheel.

DR DOBB'S JOURNAL OF
COMPUTER CALISTHENICS & ORTHODONTIA

Volume 1, Number 7; August, 1976
Box 310, Menlo Park CA 94025

Publisher
People's Computer Company
1010 Doyle, Menlo Park, California
Editor *(415) 323-3111*
Jim C. Warren, Jr.
Contributing Editors
F.J. Greeb
Lichen Wang
Jim Day
Watchdogs
Bob Albrecht
Dennis Allison
Underdog
Rosehips Malloy
Circulation & Subscriptions
Laura Reininger
Bulk Sales
Dan Rosset

POSTMASTER: *Please send Form 3579 to: Box 310, Menlo Park CA 94025. Return postage guaranteed. Application to mail at second-class postage rates is pending at Menlo Park CA. Published 10 times per year; monthly, excluding July & December.*

U.S. subscriptions:

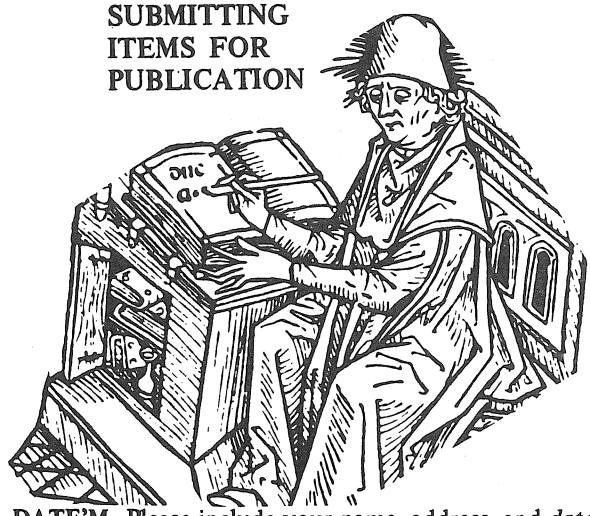
$1.50 for a single issue.
$3 for the first three issues.
$10 per year.

Foreign subscriptions:

Add $4 per year to U.S. rates for surface mail.
Add $12 per year to U.S. rates for air mail.

Discounts available for bulk orders.

SUBMITTING ITEMS FOR PUBLICATION



DATE'M—Please include your name, address, and *date* on all tidbits you send to us.

TYPE'M—If at all possible, items should be typewritten, double-spaced, on standard, 8½ x 11 inch, white paper. If we can't read it; we can't publish it. Remember that we will be retyping all natural language (as opposed to computer languages) communications that we publish.

PROGRAM LISTINGS—We will accept hand-written programs *only as a very last resort*. Too often, they tend to say something that the computer would find indigestible. On the other hand, if the computer typed it, the computer would probably accept it—particularly if it is a listing pass from an assembler or other translator.

It is significantly helpful for program listings to be on continuous paper; either white, or very light blue, roll paper, or fan-folded paper. Since we reduce the copy in size, submitting it on individual pages forces us to do a significant amount of extra cutting and pasting. For the same reason, we prefer that you *exclude* pagination or page headings from any listings.

Please, please, please put a new ribbon on your printer before you run off a listing for publication.

In any natural language documentation accompanying a program listing, please refer to portions of code by their address or line number or label, rather than by page number.

DRAWINGS & SCHEMATICS—Please draw them significantly larger than the size you expect them to be when they are published. Take your time and make them as neat as possible. We do not have the staff to retouch or re-draw illustrations. Use a black-ink pen on white paper.

LETTERS FOR PUBLICATION—We are always interested in hearing your praise, complaints, opinions, daydreams, etc. In letters of opinion for publication, however, please back up any opinions that you present with as much factual information as possible.

We are quite interested in publishing well-founded, responsible evaluations and critiques of anything concerning hobbyist hardware or software, home computers, or computers and people.

We may withhold your name from a published letter if you so request. We will not publish correspondence, however, which is sent to us anonymously.

We reserve the right to edit letters for purposes of clarity and brevity.

ADVERTISING—As long as we can afford to do so, we will not accept commercial advertising. This "keeps us honest" when we pursue the role of consumer advocate.

Personal Computing '76 Looks Like It's Going to be a *Great* Show

by Jim C. Warren, Jr., Editor, *DDJ*

Up until July 23rd, I wasn't really all that enthusiastic about Personal Computing '76, the hobby convention and exposition being held in Atlantic City on August 28th and 29th. I was unexcited, partially because it was way back there on the East Coast, and mostly because I had received no information about explicit programs or events, other than an exhibition of vendors' goodies. Well, after one lengthy conversation with the Chairman of the convention, John Dilks, all of that has changed. It sounds like it's going to be great!

There are going to be a multitude of seminars. To mention a few:

Hal Chamberlin will be discussing computer music.
A Rockwell-type will be talking about computers in outer space and the space shuttle.
Dave Ahl will offer a seminar on computer games.
Lloyd Rice will be discussing computer speech, and demonstrating his talking computer.
DEC will be presenting a computer-controlled train system.
There will be three Z-80 seminars.
Carl Helmers will be running a session on hobby standards.
I will be discussing the immediate and foreseeable future of home computing.
Dan Falystra is coming in from Holland to give a software seminar.
Ted Nelson will be running a seminar of unknown (to me) content.
There will be a medical computing seminar.
A special session will be offered for officers of computer clubs to exchange ideas.
The ham crowd will be there discussing such things as the ham satellite, a microprocessor-controlled ham radio repeater station, hamateur uses for micros, etc.
And, a whole mess of product-specific seminars will be offered by vendors.

There will be a mass of give-aways including: a Lear-Siegler ADM-3 kit, a Sphere 310 kit, a MITS 680 kit, a SWTPC 6800 kit, a MOS Technology KIM-1, an E&L MMD-1 kit, an F-8 evaluation kit from Mostek, a TV Dazzler from CroMemCo, a Z-80 kit from TDL, and 8800V board from Vector, several floppy disc systems, a MCEM8080 from Hal, a 6502 Familiarizer from Ebka, a software self-teaching course from Logical Services, . . . and on and on and on. There will also be thousands of copies of *Dr. Dobb's Journal*, and *Byte* being given away.

As far as the vendor exhibition is concerned, there will be over eighty exhibitors including such unknowns as IBM, and DEC, as well as the many better-known vendors whose names have been dropped in the preceeding paragraph, and many others. Among other things, over 25 computer stores will be present to tell you about their offerings.

If you are a West Coaster, even if you are not in Southern California, you can probably get in on the travel-package deal that is being offered to SCCS members. It's $390 for round-trip air fare, 4-day/3-night hotel accommodations, Phily-Atlantic City transportation, admission to the exposition, Sunday night banquet, and other nickel and dime items. Furthermore, you don't have to come right back—you can hang around the East Coast for a while before you return. For instance, you could run down to COMPCON Fall '76 in Washington, DC, Sept. 7th-10th. Phone Leda Alpert (an SCCS member who also happens to be a travel agent) at (213) 655-0650.

Admission to the whole thing is $5, if paid in advance, or $7.50 at the door. Incidentally, the profit (if any) doesn't go into someone's private pocket. It will go to a nonprofit amateur radio club that is sponsoring the whole thing, the Southern Counties Radio Association of New Jersey. The folks who are doing most of the organizing, John Dilks, (609 927-3873; Davey Jones, (609) 927-6950; and James Main, are club members and computer phreaques who are entirely unpaid volunteers.

HAMATEURS TO HOLD A COMPUTERFEST

The AMRAD COMPUTERFEST will be an exposition of microcomputers for computer amateurs, radio amateurs, and the general public. It is being sponsored by the Amateur Radio Research and Development Corporation (AMRAD), a non-profit scientific and educational organization.

The AMRAD COMPUTERFEST will be held on October 24, 1976 at the Vienna Community Center, 120 Cherry St., Vienna, Virginia, near Exit 11S of the Washington, DC, Beltway.

The exposition will be almost entirely devoted to small computers of the type suitable for home use. There will be displays of microcomputer systems by various manufacturers' representatives as well as tables for used or surplus equipment, circuit boards, and parts. Peripheral devices including video terminals, teletypewriters, and RTTY equipment will be shown. Forums will run throughout the day on subjects of interest to the serious hobbyist, students, and the general public. There will also be an opportunity to talk to representatives of various computer clubs and magazines.

Admission will be $4 at the door ($3.50 advance registration by mail for pickup at the door). Make checks payable to AMRAD. Write: COMPUTERFEST, Box 682, McLean VA 22101.

For reservations, contact any of these nearby motels directly: Vienna Wolf Trap Motel, 430 Maple Ave. E., Vienna VA 22180, (703) 281-2330; Tysons Corner Holiday Inn, 1960 Chain Bridge Rd, McLean VA 22101, (703) 893-2100; or Tysons Corner Ramada Inn, 7801 Leesburg Pike, Falls Church VA 22043, (703) 893-1340.

PATENT LIBRARIES ORGANIZED BY SUBJECT AREA

Wondering whether your hot hardware homebrew is patentable? Looking for homebrewing ideas?

If you live in the San Francisco or Washington, D.C. areas, you can go check a local Patent Library. It is our understanding that there are only two such patent libraries where the information is organized by subject area. The one in the San Francisco Bay Area is in the Sunnyvale City Library. We understand it is sponsored by local industries. (If some Washingtonite will give us more details about the location of their subject-organized patent library, we will be pleased to publish it.)

1977 NCC Includes Significant Personal Computing Events

The National Computer Conference (NCC) is the largest yearly computer conference and exposition in the U.S. (and probably in the world). It is sponsored by the American Federation of Information Processing Societies (AFIPS), a group group that includes all of the major organizations of U.S. computer professionals. The 1977 NCC will be held in Dallas, June 13-16, 1977.

Mr. Mauch (below) states, "A Milestone for personal computing! Personal computing is being seriously recognized by the computing industry." An alternative viewpoint is that this recognition is a *milestone for the computing industry.* Either way, it is absolutely certain that the '77 NCC will be an exciting and enlightening event, for pro and amateur alike. *We urge you to join the fun.*

The 1977 National Computer Conference will feature several events for personal computing enthusiasts including the Personal Computing Fair, exhibits of personal computing equipment by manufacturers, seminars, and social events in addition to paper presentation.

Two days of Personal Computing paper and panel presentations are being planned. Papers and panels in any subject of interest to personal computing enthusiasts are sought including:

- **† personal computer software**
- **† hardware designs and trends for personal computing**
- **† innovative applications of personal computing systems**
- **† the influence of the personal computing movement on the computer industry and computer science education**
- **† standards for personal computing products**
- **† predictions of trends in personal computing**

The '77 NCC will be the year's largest gathering of data processing users and computer professionals. Approximately 30,000 people are expected to gather for the conference program of over 100 sessions plus the year's largest display of computer hardware, software, systems, and services featuring over 250 exhibitors.

The '77 NCC Steering Committee welcomes your comments and suggestions and participation in this event. Please contact:

Dr. Portia Isaacson
Conference Chairperson
Mathematical Sciences
University of Texas
Richardson TX 75080
(214) 690-2172

Dr. Robert Korfhage
Program Chairperson
Computer Science Dept.
Southern Methodist Univ.
Dallas TX 75275
(214) 692-3082

Dear Mr. Albrecht: July 26, 1976

A Milestone for personal computing! Personal computing is being seriously recognized by the computing industry.

It is important that we take maximum advantage of this opportunity to present the worth and purpose of personal computing to the computing industry. As the leading publication serving the personal computing enthusiast, *People's Computer Company* can do much to inform the enthusiast of the importance of his or her participation and involvement in this event.

I would like to enlist your support in promoting the personal computing events in NCC '77, and invite any comments or suggestions you may have.

One of the features of NCC '77 will be presentation of papers pertaining to personal computing. Enclosed is a copy of the call for such papers. Since the submittal deadline is December 1, 1976, this announcement should appear in the earliest possible issue to permit contributors time to prepare their work.

Plans for the conference are still being formalized. We are planning many interesting and, I feel, innovative events for personal computing enthusiasts. We will keep you advised of developments.

Very truly yours,

Harold A. Mauch
Personal Computing Chairperson
1977 National Computer Conference

PerCom Data Company
4021 Windsor
Garland TX 75042

1977 NCC Paper Guidelines

Previously unpublished papers are solicited. Papers submitted for consideration must be in final form with all figures and tables, ready for typesetting. All papers will be refereed. Refereed and approved papers will be sent immediately to the printer, with no opportunity for author changes. The Conference Proceedings Editors reserve the right to edit all papers prior to publication, or to request that the authors change them to meet AFIPS publication requirements.

The material submitted should include:

1) Six copies of the paper. The paper should be a maximum of 5,000 words. The submitted paper should be the final version—cleanly types, double spaced on one side of the paper, ready for typesetting. Each page should be numbered and have the principal author's name on it. Submission of a paper implies guarantee by the author that all necessary approvals and clearances have been obtained.

2) Six copies of a page containing a 150-word abstract, the Computing Reviews Classification, and four to six keywords descriptive of the content of the paper.

3) Three copies of a short biography of the presentor to be used in conference publicity and for introductions by the session chairperson.

Deadline for all submissions is December 1, 1976. Authors will be notified before March 1, 1977 regarding the acceptance of their papers.

Please send all submissions to the '77 NCC Program Chairperson, Dr. Robert Korfhage.

CONSULTANT'S REFERRAL SERVICE

There is now a national consultant's referal group in operation. It is operated by J. Hugo Gottlich and is called the National Software Consultants Referal Service, 8 Gates St., Danvers, MA 01923.

SOFTWARE EXCHANGE FOR $$$

You may be able to peddle well documented, commercially useful software by registering it (for a fee) with the Computer Software Exchange, P.O. Box 27193, San Francisco, CA 94127. [Of course, placing your software in this exchange need not preclude your sharing it with hobbyists via publication in Dr. Dobb's Journal.]

EVIDENTLY OIL AND COMPUTERS DO MIX

Quick notes: Exxon already has a major interest in Zilog. Now, Sun Oil is negotiating to buy DATRAN, one of the largest special carriers in the nation, expecially set up for high-speed, high-reliability digital data communications. The figure involved in the negotiations is $30 million.

COBOL FOR THE 8080

RRC Consulting, Los Altos, CA, has a COBOL Cross Compiler for the Data Point 2200 (which is almost an 8008). They only want a one-time use fee in the neighborhood of $50,000 (yes...$50K!)

The Time for Floppy's is Just About *Now!*

by the Editor

Things—most notably, prices—are coming down, fast, in the world of rotating mass storage appropriate for home computers. Prior to this, floppy disc subsystems have either been unavailable for hobby machines, or they have been priced for the industrial consumer (e.g., around $3,000 for a dual-drive system). Things have changed: A hobbyist can now reasonably expect to obtain a complete, assembled, single-drive subsystem for a price in the neighborhood of $1K. Here is the latest information we have:

The best system we know of—and the least expensive—is available from Digital Systems, 1154 Dunsmuir Pl., Livermore CA 94550; (415) 443-4078 (ask for Dr. John Torode). This is the same crowd that built Gary Kildall's original floppy interface over two years ago [see "First Word on a Floppy-Disc Operating System," in the April issue of the *Journal*]. Gary has yet to have problems with the system. Digital Systems also is marketing a low-cost, floppy-based development system to the industrial market. They know what they are doing. What is much more important is that Dr. Kildall's fancy, DECSystem-10-like operating system—called CP/M—will run on DS's hardware. CP/M has been in use for *over two years* in a production and instructional environment. It is well debugged, well documented, and has some significant software subsystems available with it.

Back to DS's floppy subsystem: The controller is already-assembled, burned-in, and tested, and has been on the market for some time. The controller can handle up to four drives. The drive is a Shugart 800, a highly reliable drive of excellent reputation. The single-system price is $1095, and drops to $995 in quantities of only 10. This price includes the controller, the drive, all cabling between controller and drive, a manual, shipping within the continental U.S., and a 90-day warranty. The manual includes a reproduction of the Shugart tech manual on the drive, a complete manual on the controller, and complete design details for a 20-chip interface from the controller to a hobbyist-standard (Altair/IMSAI) bus. The manual, alone, is available for $5, creditable towards purchase of any system.

Though the controller is not available in kit form (too much potential for subtle, hairy problems), the interface is. If you wish to buy your own parts and build the interface, you can. If you want the parts from DS, they will ship them to you for $50 (Vector PC board, sockets, chips, connectors, cabling . . . the works). If a dual-drive system is desired, the second drive is $600, unit quantity.

As with most of the floppy systems being offered to hobbyists, the power supply (if you don't already have one) is extra. In this case, it costs about $150, and the manual gives complete details if you wish to homebrew your own.

Final notes: DS is a relatively small company. We know Dr. Torode both personally and by reputation, and have been very impressed by everything we have seen and heard. He maintains quite high standards for his product, technically, and for his operations, businesswise. We have complete faith in his competence and in his integrity and recommend him without restriction. He builds a good product, and he backs his work. Incidentally, the controllers, drives, and parts are in-stock, off-the-shelf items.

If you have technical questions, Dr. Torode is the person with whom to talk. Unfortunately, he will be in Europe (teaching microprocessor courses for Phillips) from August 15th to September 15th. Thereafter, however, he will be happy to answer any questions you may have.

Dr. Kildall's CP/M is available from Digital Research, Box 579, Pacific Grove CA 93950; (408) 373-3403, for $35-$70 depending on the level of documentation desired. The software comes in the form of a "loaded" disc, including an editor or two, an assembler or two, a PIP (DECese for a file-transfer program), a debugger, and who knows what else (it grows from month to month).

A second, interesting tidbit is a letter we received, dated June 30th, offering brand new CalComp 110 drives for $395; that's just about half their list price. The letter states that the drives carry CalComp's 90-day unconditional warranty. The drive does *not* use the IBM format (a poor format, but the only "industry standard" available for the time being), and could *not* be easily interfaced to CP/M. Also, CalComp drives appear to have a mixed reputation; some are good, and some are very poorly regarded.

The company making this offer is Martin J. O'Boyle & Associates, Box 9094, Pittsburgh PA 15224; (412) 361-1602. We know nothing about them, but the offer appears to be well worth checking out. Incidentally, their letter indicates they are also offering IMSAI computers at $100 below list; mentions two designers of very low-cost floppy interfaces (Dr. Ken Welles, and Hal Chamberlin); and notes that they have only 95 of their CalComp 110 drives available for the price indicated.

Financially, the next most interesting floppy system for hobbyists appears to be the one available from iCOM, 6741 Variel Ave., Canoga Park CA 91303; (213) 348-1391. It's single-unit price is $1195. The price doesn't drop to $995 (which appears in all their bold-face advertising) until one purchases them in 100-unit quantities. If you get in on an SCCS group buy, however, you can get them for about $1100 which includes tax, handling, and the FDOS-II operating system. As we understand it, for these prices, one gets a Pertec drive, a controller capable of handling more than one drive, and cabling from the drive to the controller, and the controller to an interface board. The interface board is *not* included, but we have been told that a 3P+S will do the necessary job.

The FDOS-II was done by Art Childs, the recent editor of *Interface*, and a great hobbyist supporter. It requires about 1K for the resident I/O drivers, 1K for the monitor, and 4K for the Executive. To be really useful, it requires about an 8 kilobyte system. This is noticeably less than is required by Kildall's CP/M, which needs at least 12K and prefers to dine on about 16K. However, FDOS-II was completed only last Spring (1976), and thus may be assumed to have significantly more bugs than the more mature CP/M. (No criticism of Art's work; it's just an inherent characteristic of large systems software.) Also, CP/M has a comparatively extensive library of systems software that runs under it.

Unanswered questions: We don't know how reliable the Pertec drives are. They generally have a much poorer reputation than the Shugart drives. We have an estimate that the power supply for the system will cost in the neighborhood of $210, but that estimate is 3 months old. We don't know if the indicated prices include shipping or manuals, and we don't know how well-done the manuals may be (the hobby move-

ment has a widespread characteristic of *awful* documentation). We also don't know what warranties are offered, and have no information on how well iCOM treats its customers.

Since it is noticeably overpriced in comparison to the preceeding excellent alternatives, we hesitate to even bother mentioning the IMSAI floppy system . . . but, we will, since they are trying to market it to hobbyists. The controller has its own processor and is being touted as being "intelligent." It will handle up to four drives. It interfaces to a hobbyist-standard bus, of course, and the interface is included. In kit form with a single drive, it costs $1,449, and goes for $1649 in assembled form. Extra drives cost $925. The Disk Operating System is available for $40. 12K Extended BASIC, including disc access, is supposed to be available by now. Ho hum.

We think IMS manufacturers some excellent microcomputers and m-c kits. However, we find their floppy offerings to be badly out-of-line—at least for the hobbyist—with some other products, at least as far as price is concerned. Incidentally, we hear that Kildall's CP/M was made to run on the IMSAI floppy system . . . and run very, very slowly.

Finally, some immediate-future thoughts concerning floppies:

PerSci is now offering a *dual*-drive floppy box for $1K. To our knowledge, no one has yet built a controller and interfaced it to a hobbyist-standard bus, but we are certain that it will happen very soon. This is deserving of attention, not only due to the low price for two drives, but also because PerSci drives use a voice-coil head positioner that really wails; its track access time is something like 10 times as fast as most of the other drives. Also, the feedback we have from industrial users of PerSci products implies that they manufacture excellent, solid, reliable floppy disc drives.

Don't discard the idea of moving to floppies because you can't afford a dual drive system, and you think you can't back up your discs if you have only a single drive system. It's slow (and also cheap), but you *can* back up (save for safety's sake) discs, and even copy disc contents from one to another without a second drive . . . by using the cassette tape storage that you probably already have. Not what one would do in an industrial or business environment where $500 or $1,000 is trivia, but at home . . .

Watch for prices for controllers to drop even lower. We are all waiting for Hal Chamberlin to publish the rest of his series of articles on the super-inexpensive floppy controller/interface in *The Computer Hobbyist*. The first article was published last Spring. Someone in Kansas told us of designing a very simple disc interface that he recently sold to a company, back there. Presumably, they will be announcing a product, shortly. The Palo Alto Byte Shop has mentioned a very inexpensive floppy subsystem that they expect to place on the market, shortly. Shugart has announced its mini-floppy drive, which stores less than a "standard" floppy, but is priced significantly less than their "standard" drive. Gary Kildall is still hoping to be able to offer an interface/controller for somewhere in the neighborhood of $350, and presumably a single-drive system for about $800.

But, there's a problem with watching for prices to come down. We all know they will. (We predict that floppy prices will probably bottom out at around $600 for a single drive, ready to plug in to your hobbyist-standard bus. We think that cost of manufacture will keep the price from dropping much below that . . . at least, for reliable units.) The question is, how long do you want to wait until you have high-speed, mass storage . . . and can do *really* interesting things with your home computer?

HEATHKIT TO INTRODUCE A COMPUTER KIT

We have information from a source we consider to be reliable (but not from Heathkit) that Heathkit will present its first computer kit at the Personal Computing'76 Consumer Trade Fair in Atlantic City, August 28th-29th. The kit is rumored to be using the Signetics 2650 microprocessor. No other information at the moment, but...watch out McDonald's!

DESIGN FOR BINARY INPUT TO HEX DISPLAYS

The July 8th issue of ELECTRONICS Magazine has two quickie design articles detailing how to convert binary input to a HEX-coded, 7-segment display output. One method uses a PROM; the other uses some decoders and diodes.

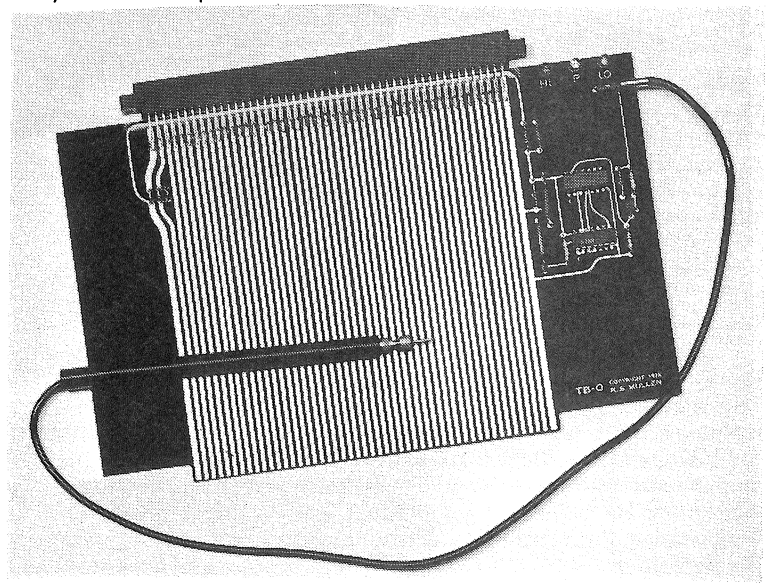
300 NANOSECOND 16x16 MULTIPLIER

TRW Electronics Systems Division has a TTL-compatible, monolithic-bipolar multiplier that can perform 16-bit by 16-bit multiplication in half a microsecond. Designated the MPY-16, sample quantities can be delivered in under 30 days.

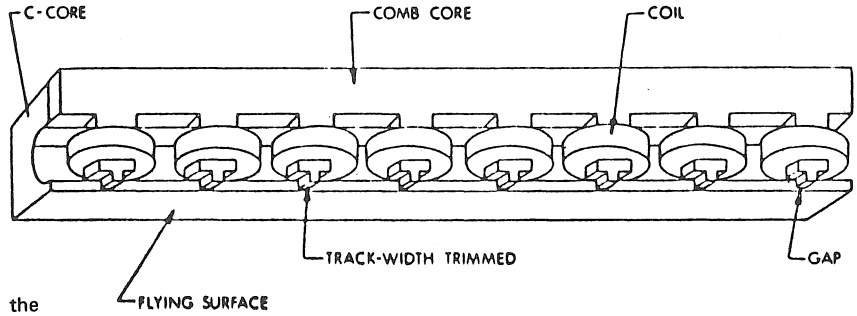
A FLASHY EXTENDER BOARD WITH LOGIC PROBE

This new extender board/logic probe kit is designed to fit either the Altair or IMSAI. It features Logic Level Lights: red $\geq$ 2.4 volts, green $\leq$ 0.8 volts, and yellow $\simeq$ 0.2 second pulse HI/LO transition. A phono-needle probe is provided to ensure good and non-skid contact. Eyelets and jumpers in the power circuits allow convenient current measurement. The leads of the dual 50 read-out edge connector are specially formed to allow convenient access to bus signals. The PC board material is 1 oz. Cu on 0.0625-inch, black FR4 board, which is solder-plated and has 50 μinch gold on the connector edge. The holes are plated through with 0.0015-inch Cu.

This product is available now for $35, from Mullen Computer Boards, BlastMasters Inc., Box 31, Loma Mar CA 94021. Or check your local computer store.



Design Contest Offers Head-Per-Track Hard Discs as Prizes



Gentlepersons: July, 1976

At the request of many computer hobbyists, unhappy with the availability of low-cost, mass memory for permanent program and data storage, we are considering the design of a fast-access, memory kit.

In order to offer you the type of storage that serves your needs, we give you the opportunity to participate in the design. So, open your mind and let us know your ideas.

The major cost item is the engineering time required to design the controller which interfaces the memory and the microprocessor. In order to keep the cost down and to make it interesting and rewarding, we decided to have two contests. One, incorporating hardware, and one for software.

To keep the contest proper we propose that the various computer clubs submit proposals on the rules, and judges (i.e., editors of stated computer hobbyist magazines.).

The start up time for the memory is about 120 days. The interface controller requires approximately 60 days from the closing of the contest.

It would be helpful for us to know how many of these fast-access memory kits will be sold. This is because we need to buy parts for the kits. The more parts we buy the lesser the cost to you. So please, if you are seriously considering the purchase of our memory, let us know—the bigger the response, the lower the production costs and selling price.

The following is what we think you need. Take an close look and see if it is OK, if not, speak up. Let's make it right the first time.

The proposed memory is designed for optimum performance and reliability at the lowest possible production cost.

It features high rotational disc speed and head-per-track operation, thus it sports a ten-times-faster access time as compared to a floppy disc system. The head-per-track design eliminates the time-consuming, mechanical positioning of one head on every track as well as simplifying the interface controller.

A large, 14-inch-diameter disc is chosen to store as many bits with as few heads as possible. The disc is permanently mounted and rotates at 1725 rpm.

A nine-head (Track) per comb-bar assembly is utilized. Eight heads are active, accessing 125,000 unformatted bits per track at a data rate of 3.6MHz. The ninth track is a spare. Therefore each comb-bar assembly (8 Tracks) can store 1 million gross bits or at least 100,000 8-bit words. The memory can be expanded to 12 comb-bar assemblies or 96 active tracks storing 12 million bits or 10 million 8-bit words.

For systems unable to handle the 3.6MHz data rate, the frequency can be reduced, however fewer bits per track can be stored, or more heads must be used for the same data capacity.

A delay modulation recording system is featured. It requires an 8-bit preamble containing a 101 code to synchronize the phase-locked, read decoder. (10101011 at the beginning of each sector.)

Preliminary specifications: Disc memory models

	HTM 8-1 kit Minimum	to HTM 96-12 kit Maximum
Data capacity, bits	1,000,000	12,000,000
Data per track	125,000	
Data transfer rate	3.6 MHz	
Data access time average	17.4 milliseconds	
Size	17"W x 17"D x 10"H	
Power	115V 60Hz 200VA	

Electronics: One read-write amplifier, one index amplifier, and one sector amplifier for 64 sectors.

Interface: TTL twisted pair

Preliminary pricing based on a 100-unit production run:

HTM 8-1 kit with interface controller	$600
Each additional 9-track, comb-bar assy., or 1 million bits	$50
HTM 8-1 kit without interface controller	$450

First Prize: a) The best controller (price-performance) gets the grand prize of a memory with 300,000 8-bit words free. b) So does the best software for the controller, hopefully a program which makes the memory transparent (virtual) to the microprocessor.
Second Prize: One each, 200,000 8-bit word memory for the second best controller and software design.
Third Prize: One each, 100,000 8-bit word memory for the third best hardware and software design.
Fourth Prize: $200 credit each, for the purchase of our memory for the fourth best controller and software design.
Fifth Prize: $100 credit each, for the purchase of our memory for the fifth runner-up for hardware and software design.

THE LOW-COST COMB-HEAD*

Consisting of three basic parts—C-CORE, COMB-CORE, and COIL—the assemblies may have from 1 to 12 heads. Assembly of a typical 9-track COMB-HEAD requires very little time and is accomplished by simply slipping the self-supporting coils over the legs of the COMB-CORE and then pressing C-CORE and COMB-CORE together and applying adhesive. A conductive gap shim is vacuum-deposited (sputtered) over the entire inside of the C-CORE to minimize cross-talk.

That's it! Let's get started, and good luck to you.
Sincerely yours,

E. Mike Engel
E. & U. Engel Consulting
1719 S. Carmelina Ave.
Los Angeles CA 90025
(213) 820-4231

*Developed and licensed by E. & U. Engel Consulting

Editor's note: We hope that the contest judges will be chosen, announced, and publicized, prior to this company soliciting/receiving contest entries.

COMPUTER COMPONENTS OPENS VAN NUYS STORE

Computer Components opened its first retail store to the general public on June 17th. The address is 5848 Sepulveda Blvd., Van Nuys CA 91411. They are currently authorized IMSAI dealers and expect to expand their microprocessor line to other manufacturers as they demonstrate viability. They also carry a complete line of interface and memory kits, hobbyist supplies, etc.

SAN DIEGO COMPUTER STORE

The Computer Center (8205 Ronson Rd., San Diego CA 92111, (714) 292-5302) opened its doors on May 22nd. It carries a variety of computer hobbyist equipment and publications.

Proposed Hobbyist-Standard* Bus Structure

by D. Denney, & J. Broom
BISI
Box 1197, Station A
Vancouver, British Columbia
Canada
(604) 683-5246

The most important and often overlooked part of an Altair/IMSAI computer is the 100 wire bus structure. This bus structure as defined by MITS has become a "de facto" standard that has encouraged numerous companies to build and market Altair/IMSAI compatible boards, knowing that there are thousands of machines in which to use them. The bus standard has stimulated competition and allowed product specialization; and the result has been to raise the Altair/IMSAI from just a 'hobbyist' tag to a full-fledged computer that is rapidly finding its way into industrial, commercial, and educational applications. For what other computer on the market has such a broad base of hardware manufacturers?

To date, most of the second sources have been of RAMs, TV typewriters, etc., but there is a second generation of products just beginning to surface, which will vastly increase the power and longevity of the 100 wire bus. These products are of two basic types:

1. NEW CPU BOARDS using different 8 and 16 bit microprocessor chips that will plug into the bus. (You can already buy an M6800 CPU board for your Altair and work is in progress with CPU chips from Data General, and Texas Instruments 8800 series.)

2. MASS STORAGE—memory capacity inside the Altair chassis in excess of the 65K directly addressable bytes, made possible by the new CCD (charge coupled device), and magnetic bubble technologies. Mass storage will typically be configures as a set of storage boards (from 1 to 20) and controller board. The controller board must furnish to the storage boards several critical timing and control signals (all TTL levels), and because of the number of boards involved, these lines should be placed on the 100-line Altair bus for storage board simplicity, lessened cost, and ease of trouble-shooting.

While opening up the future of the Altair to many new applications, these boards are putting a strain on the bus structure—they use more bus wires than so far have been defined, so the time has come to define more of the unused bus lines.

Altair/IMSAI Bus

O = letter O
Ø = zero

1. +8V Unregulated input to +5V regulators
2. +16V Unregulated input to +12V regulators
3. XRDY Anded with PRDY and goes to 8080 RDY
4. VIØ Vectored interrupt request Ø
5. VI1 Vectored interrupt request 1
6. VI2 Vectored interrupt request 2
7. VI3 Vectored interrupt request 3
8. VI4 Vectored interrupt request 4
9. VI5 Vectored interrupt request 5
10. VI6 Vectored interrupt request 6
11. VI7 Vectored interrupt request 7
12. XRDY2

	CCDS	BORAMS (TENTATIVE)	BUBBLES (TENTATIVE)
13.	CK3 Phase 3 shift clock	13.	13. CK2 Phase 2 clock for coil drivers.
14.	CK1 Phase 1 shift clock	14. ADRSEL Address multiplex signal	14. OUTEN Data output buffer enable signal
15.	$\overline{BWE}$ Write erable-anded with WRITE	15. LDL Load to address byte	15. XFER Transfer gate signal
16.	LDM Load medium address byte	16. LDM Load med address byte	16. ADINX Increment signal for board address
17.	BDSEL Acknowledge signal from addressed storage bd.	17. BD SEL Acknowledge signal	17. BDSEL Acknowledge signal from addr. storage bd.

18. $\overline{STA\ DSB}$ Status buffer disable
19. $\overline{C/C\ DSB}$ Command/control buffer disable
20. UNPROT Input to memory protect circuitry on memory bd.
21. SS Indicates machine is in single step mode
22. $\overline{ADD\ DSB}$ Address buffer disable
23. $\overline{DO\ DSB}$ Data out (from CPU) buffer disable
24. Ø2 Phase two clock TTL levels
25. 01 Phase one clock TTL levels
26. PHLDA Hold acknowledge, buffered 8080 output
27. PWAIT Wait acknowledge, buffered 8080 output
28. PINTE Interrupt enable, buffered 8080 output
29. A5 Buffered address line 5 (32)
30. A4 Buffered address line 4 (16)
31. A3 Buffered address line 3 (8)
32. A15 Buffered address line 15 (32768)
33. A12 Buffered address line 12 (4096)
34. A9 Buffered address line 1 (2)
35. DO1 Buffered data out line 1
36. DOØ Buffered data out line Ø
37. AIØ Buffered address line 1Ø (1024)
38. DO4 Buffered data out line 4
39. DO5 Buffered data out line 5
40. DO6 Buffered data out line 6
41. DI2 Data input line 2
42. DI3 Data input line 3
43. DI7 Data input line 7
44. SM1 Latched 8080 M1 status
45. SOUT Latched 8080 OUT status
46. SINP Latched 8080 INP status
47. SMEMR Latched 8080 MEMR status
48. SHLTA Latched 8080 HLTA status
49. CLOCK 2 mhz clock, crystal controlled
50. GND Logic and power ground return

(A) 51. +8V Unregulated input to +5V regulators
(B) 52. -16V Unregulated input to negative regulators
(C) 53. $\overline{SSW\ DSB}$ Sense switch disable

(D) 54. $\overline{\text{EXT CLR}}$ Clear signal for I/O devices

(E) 55. RTC Real time clock

(F) 56. $\overline{\text{STSTB}}$ Strobe signal (by 8224 clock chip 8800B d/c board)

(H) 57. DIGI Enable signal for CPU DI drivers 8800B

(J) 58. FRDY 8800B Front panel ready signal

(K) 59.

(L) 60.

(M) 61.

	CCDS	BORAMS (tentative)	BUBBLES (tentative)
(N)	62. CK4 Phase 4 shift clock	62. RAS Row address select	62. CK1 Phase 1 clock for coil drivers
(P)	63. CK2 Phase 2 shift clock	63. CAS Column address select	63. R/A Replicate/ annihilate gate signal
(R)	64. BCE Chip enable	64. REF Refresh signal	64. READ Signal for bubble defect circuitry
(S)	65. $\overline{\text{WRITE}}$ Write signal	65. WRITE Write signal	65. Gen Generate gate signal for writing
(T)	66. LDH Load high address byte	66. LDH Load high address byte	66. LD HI Load hi address byte

(U) 67. see below +

(V) 68. MWRT Write enable signal for memory

(W) 69. $\overline{\text{PS}}$ Indicates if addressed memory is protected

(X) 70. PROT Input to memory protect circuitry on memory bd.

(Y) 71. RUN Indicates machine is in run mode

(Z) 72. PRDY Anded with XRDY and goes to 8080 RDY

(a) 73. $\overline{\text{PINT}}$ Input to 8080 interrupt request

(b) 74. $\overline{\text{PHOLD}}$ Input to 8080 hold request

(c) 75. $\overline{\text{PRESET}}$ Clear signal for CPU

(d) 76. PSYNC Buffered 8080 SYNC signal

(e) 77. $\overline{\text{PWR}}$ Buffered 8080 write enable signal

(f) 78. PDBIN Buffered 8080 BDIN signal

(h) 79. AØ Buffered address line Ø (1)

(j) 80. A1 Buffered address line 1 (2)

(k) 81. A2 Buffered address line 2 (4)

(l) 82. A6 Buffered address line 6 (64)

(m) 83. A7 Buffered address line 7 (128)

(n) 84. A8 Buffered address line 8 (256)

(p) 85. A13 Buffered address line 13 (8192)

(r) 86. A14 Buffered address line 14 (16384)

(s) 87. A11 Buffered address line 11 (2048)

(t) 88. DO2 Buffered data out line 2

(u) 89. DO3 Buffered data out line 3

(v) 90. DO7 Buffered data out line 7

(w) 91. DI4 Data input line 4

(x) 92. DI5 Data input line 5

(y) 93. DI6 Data input line 6

(z) 94. DI1 Data input line 1

(AA) 95. DIØ Data input line Ø

(AB) 96. SINTA Latched 8080 INTA status

(AC) 97. $\overline{\text{SWO}}$ Latched 8080 WO status

(AD) 98. SSTACK Latched 8080 STACK status

(AE) 99. $\overline{\text{POC}}$ Clear signal during power up

(AF) 100. GND Logic and power ground return

+ A conflict already exists on line 67. Processor Technology Inc., has defined this line as PHANTOM to disable RAM boards when ROM is addressed. Cromenco Inc., has defined line 67 as REFR DSBL which prevents dynamic RAM refresh cycles during the Dazzler DMA (direct memory access).

***Editor's Note: Up to this point, this bus has been referred to as the "Altair/IMSAI bus." Given the number of manufacturers now making the bus, or making products for it, we think it's time to start calling it what it has become—"The Hobbyist-Standard" bus.**

FIRST RUMORS OF A WEST COAST COMPUTER COMPUTER FAIRE

There are some tentative rumors being passed about concerning a hobbyist Computer Faire on the San Francisco peninsula. It is envisioned to be a two-day orgy, including a multitude of seminars, and a massive exhibition of hobby goodies. There is some vacillation about whether to schedule it just before or after Thanksgiving, or wait until after the first of next year.

Let us know if the prospect of such a Faire interests you, and indicate your preferences and dislikes of the proposed dates. Send responses to:

Bob Reiling
(Editor, *Homebrew Computer Club Newsletter*)
193 Thompson Square
Mountain View CA 94043
(415) 967-6754

or

Jim C. Warren, Jr.
(Editor, *Dr. Dobb's Journal*)
PCC
Box 310
Menlo Park CA 94025
(415) 323-3111 851-7664

The PCM-12: A PDP-8 Look-Alike Well Worth Looking At

by Jim Warren, Jr., Editor

PCM, Box 215, San Ramon CA 94583; (415) 837-5400, is marketing a microcomputer kit based on the Intersil 6100 microprocessor. It is a system well worth giving serious consideration. It's only selling point is that there is a *massive* amount of systems and applications software in existence for it. And, if you want to use it, instead of simply tinker with it, that's a *major* point. The 6100 executes all of the Digital Equipment Corp. PDP-8/E instruction set, except the I/O instructions. Since the "8" has been around for about fifteen years, and is still one of the most popular minicomputers (due almost totally to its software)—there are around 70,000 of them in use—anything that executes a PDP-8 instruction set should be given serious attention.

If you are a hardware builder type, the system should satisfy your desires. It comes in a kit form and can adequately pacify those who love to fondly caress a soldering iron and dikes.

If you are a software type, with some nominal scrounging, you can come up with all sorts of *glorious* software, including several editors, assemblers with macro capabilities, BASIC's, FOCAL, several FORTRAN's, a number of debuggers, business software, and on and on.

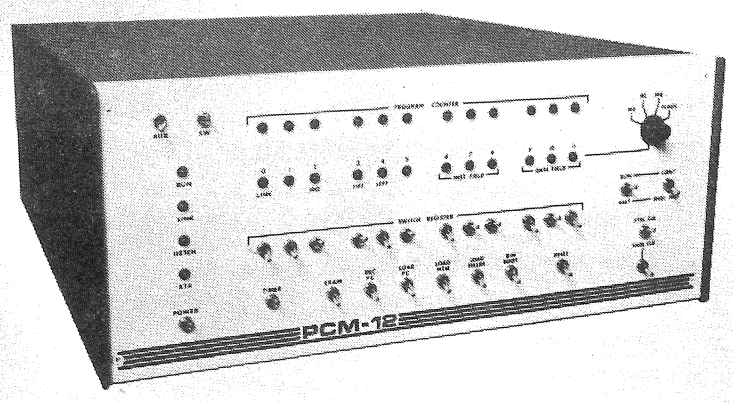
If you are a computer architecture type . . . forget it. The PDP-8 was designed in the early 1960's. It is a single accumulator CPU with no index registers, fixed-block addressing, a 12-bit word length, and doesn't even have a SUBTRACT instruction (you have to do a 2's Complement, then Add). Architecturally, there are far more exciting microprocessors/microcomputers on the market.

And, if you are financially frugal concerning your computer hardware (regardless of the potential expense for the software), then you should also forget the PCM-12. The system with only 1K words is priced at $799 (kit), or $1224 (assembled). To expand it to 4K words costs another $139 (kit form), and to add memory beyond that costs $135 (memory extender kit), plus $245 per 4K memory module. This means that a complete system in kit form, including 8K 12-bit words of memory would cost $1415.

Incidentally, Digital Systems Design in Berkeley has a complete, assembled floppy disc system including two Shugart drives that plug into the PCM-12. It costs $2998. We understand that DEC's *great* PDP-8 Operating System, OS-8, is up and running on the PCM/DSD floppy system, however its direct availability to a purchaser is unclear.

To sum it up, if you have access to PDP-8 software that you find significantly useful, you should investigate the PCM-12. (You should also investigate the PDP-8/A from DEC; it's more expensive, but avoids any bootleggery problems concerning software). If you just want to play, or you want to learn about current computer architecture, you should look elsewhere.

Some personal notes: For much of a decade, I made most of a very comfortable living, programming PDP-8's. Their architecture was infuriating, but their systems software and documentation was significantly better than any other mini around at the time. Even now, DEC software for the "8" remains better than most systems software for most other machines, and is generally better documented. Even when there is functionally equivalent software for other machines, it usually requires more memory than it does on the PDP-8. For example, OS/8 requires only 8K 12-bit words on the PDP-8. It's PDP-11 counterpart, RT-11, requires a minimum of 16K 8-bit bytes. Operating systems for Data General's Nova machines exhibit a similar hunger for memory, and the development systems from the micro manufacturers absolutely gobble memory—though the comparison is only partially valid. If I didn't already have my own PDP-8/I (elderly, but definitely "burned in" and reliable), I would probably get a PCM-12—even though I consider them somewhat overpriced. Incidentally, I consider DSD's floppy system to be grossly overpriced by about $1300.



VRAM-VIDEO DISPLAY RAM

Matrox Electronic Systems, P.O. Box 56, Ahontsic Stn. Montreal, Quebec H3L 3N5, (514) 481-6838 has a number of very interesting "Black Boxes" available for 4-12 week delivery. To the computer, those appear to be "write only" RAM. Their output, however, is a standard (American or European) video signal (75 ohm, crystal controlled). This will drive video monitors up to 3000 feet away. Quick tidbits:

Model	Function	Price (single unit)
MTX-816	8 by 16 large-character; no lower-case character display	$194
MTX-1632	16 by 32 caps & lower-case character display	$285
MTX-1632 SL	Externally synchronized 16 by 32 display	?
MTX-1664*	16 by 64 character display	$364
MTX-3264*	32 by 64 display	$440
MTX-2480*	24 by 80 display	$480
MTX-256**2	a 256 by 256 point arbitrary graphics display	$630

*The 1664 and 3264 require a 5.2 Mhz bandwidth monitor. The 2480 requires 6.5 Mhz.

Whoops! There's a ringer: They say a minimum order is 100 units, and there is a setup charge. Just goes to show—you gotta read *all* the small print.

Z-80 . . . Coming on Strong
Three Kits Under $300--One Includes 2K Bytes, Another Runs at 4 MegaHertz

by Jim C. Warren, Jr., Editor

Three groups are (or will be) offering processor boards or microcomputers to hobbyists, based on Zilog's hot, new μp, the Z-80. They are the Digital Group, TDL, and CroMemCo. The Z-80 has an exciting instruction set that is almost completely upwards compatible with the Intel 8080 (which is, of course, upwards compatible with the Intel 8008). Note: the Z-80 is *not* pin-for-pin compatible with the 8080; the almost-compatibility only concerns instruction sets. Also note: The Z-80 is being second-sourced by Mostek.

The one instruction incompatibility we have heard about, so far, concerns the handling of the parity flag. As explained to us, the Z-80 sets the parity flag only on logical operations, as one would expect, whereas the 8080 sets the parity flag on both logical and arithmetic operations. Incidentally, since Bill Gate's Altair BASIC uses this 8080 eccentricity in three or four places, Altair BASIC will not run on a Z-80 unless a few minor modifications are first made to the code. We understand that listings of those mods are floating about. If someone will send us a good, clear, documented copy, we will be happy to publish them (with or without an author's credit line). Note: We consider publishing such mods to be roughly analogous to an auto magazine publshing details about modifying a patented Ford engine design for the purpose of obtaining more useful or efficient operation; patent infringement is not involved.

The Digital Group, Box 6528, Denver CO 80206; (303) 861-1686, is offering a Z-80 CPU board for $295 (kit), or $395 (assembled). The last we heard, they were planning on initial deliveries around the middle of July. The CPU board includes 2K bytes of half-microsecond static RAM, a 256-byte EPROM bootstrap, two DMA channels, a 400ns-rated Z-80, and some other interesting goodies. The board is interchangeable with previous Digital Group CPU boards, but is *not* plug-compatible to a "hobbyist-standard" (Altair/IMSAI) bus.

Also, the Digital Group is doing something that we broadly applaud: They are offering a Z-80 CPU upgrade kit for $185 to those who have previously purchased systems from them based on some less exciting microprocessor, and wish to upgrade to the Z-80. We have consistently been very favorably impressed by Digital Group's "up front" stance, candidness, apparent integrity, and consistent, soft-sell approach to marketing their goods. A comment, however: We have been getting an increasing number of letters, here at People's Computer Company, complaining about poor service or responsiveness from the Digital Group. We have published such complaints, as well as some compliments, and will continue to do so. Our assumptions are: 1. Customer interest in Digital Group products has exceeded their initial capacity to respond in the manner they would like. 2. They are aware that they have a problem and are doing everything they can to expand their customer service department and procedures. 3. It is their wish to maintain an operation of unusually high integrity and responsiveness to their customers.

TDL (Technical Design Laboratories), 342 Columbus Ave., Trenton NJ 08629; (609) 392-7070, is the second company offering a Z-80 CPU board to the hobbyist community. Their kit price is $269, and includes IC sockets, but excludes the two kilobytes of static RAM found in the Digital Group's subsystem. As of the middle of July, their prototype had been running for about a month, and their PC version (production version) had been running, glitch-free for about half a month. The system has been demonstrated to members of the Amateur Computer Group of New Jersey, several of whom have commented very favorably about it, to us.

TDL is a new entry into the rapidly growing mob of computer hobbyist companies. We have received several favorable comments about them, however, from people who are independent of TDL, and whose judgement we trust. The comments we hear are that the TDL principals are technically competent, and are dedicated to turning out an excellent product, and backing it with first-rate customer service. We are most interested in receiving comments, positive or negative, from their customers.

They are also offering one other rather flashy item: Their Z8K RAM board is an 8 kilobyte static RAM for $295 with *215ns access time*. Furthermore, it only draws about 150ms from a 5V power supply. We are particularly interested in hearing customer comments regarding this RAM. The rumor we hear is that this system uses RAM made by EMM, the same as is in MITS 16K boards . . . and EMM is the only semiconductor house that has been able to make them. We want to spread the word, if they are reliable. And, we have no reason to question their reliability, other than the fact that none of the larger chip houses are marketing a competitor. Presumably, they will, soon.

CroMemCo, 2432 Charleston, Mountain View CA 94043; (415) 964-7400, is the third company to jump onto the Z-80 bandwagon. [This is pre-announcement information, obtained from Dr. Harry Garland, one of the principals in CroMemCo.] They are planning a Z-80/4 CPU card that will plug into a hobbyist-standard bus that sounds like a *super* subsystem. For a starter, the Z-80/4 is a *4 megahertz processor*; that's a 250ns cycle time. The board can be switched, at any time, back and forth between 2MHz and 4MHz. It has a power-on JUMP to any 4K memory blank, jumper selectable. There is a wait-state generator on-board that can place wait-states on M-1 cycles if desired, or on any other cycle. A jumper can be used to select between 8080-mode I/O and Z-80 mode. The subsystem price is expected to be $295 (kit), or $395 (built and burned-in). Dr Garland specified October 1st as being a firm availability date.

CroMemCo is another of those companies that we feel is really "doing right" by its hobbyist customers. As well as this writer can remember, PCC has yet to receive a single complaint about CroMemCo products or service. We know several of the principals in the company, and believe that they act with first-rate integrity; a level of honesty that one would more expect from a close friend than from an "unknown businessman." Their technical competence is also top-notch. Minor tidbit: Aside from being a principal in CroMemCo, Dr. Garland is also Assistant Chairman of the Department of Electrical Engineering at Stanford University.

Remember those secret codes you played with as a kid? Well. . .

By Mal Stiefel

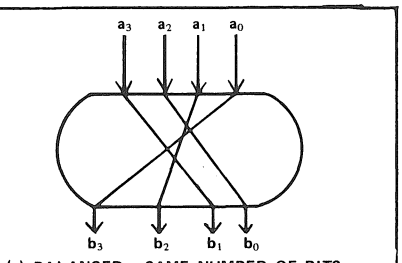
The National Bureau of Standards (NBS) published a proposed standard in the *Federal Register* of Aug. 1, 1975 specifying "a mathematical algorithm for encrypting (enciphering) and decrypting (deciphering) binary-coded information" for federal use.

The process is intended to be implemented "in a special-purpose electronic device. . .designed in such a way that it may be embedded in a computer system or network and provide cryptographic protection to binary-coded data.

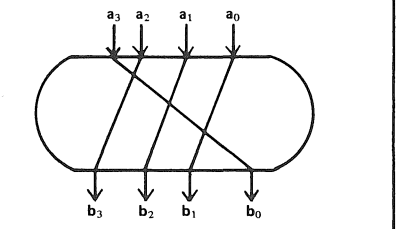
"The method of implementation, the control of the cryptographic device and the interface of the device to its associated equipment will depend on the application and environment. . . Certification of compliance with this standard is the responsibility of the designer and manufacturer of the device."

Basic Principles

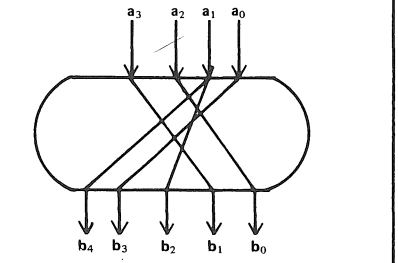
Figures 1 and 2 illustrate the various data manipulation functions found in the algorithm. The permutations in Figure 1 involve rearrangements of bits without changing their value, while the substitutions (also called ciphers) do not generally preserve bit values between input and output.



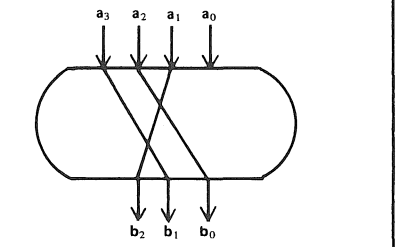
(a) BALANCED – SAME NUMBER OF BITS IN INPUT AND OUTPUT. THIS PERMUTATION IS INVERTIBLE, BECAUSE EACH INPUT BIT IS TIED TO ONE OUTPUT BIT.



(b) CYCLIC – CYCLE LEFT 1 BIT IS ILLUSTRATED

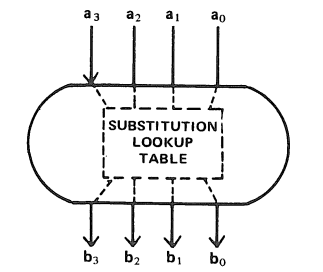


(c) EXPANDED – OUTPUT HAS MORE BITS THAN INPUT

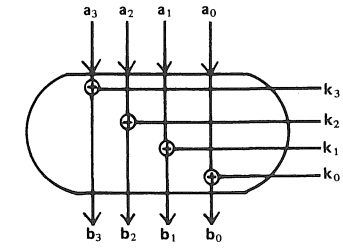


(d) CONTRACTED – OUTPUT HAS FEWER BITS THAN INPUT

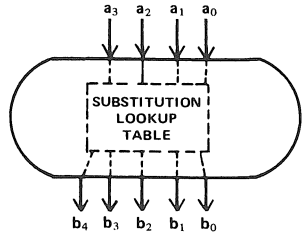
FIGURE 1
PERMUTATIONS WITH 4-BIT INPUTS



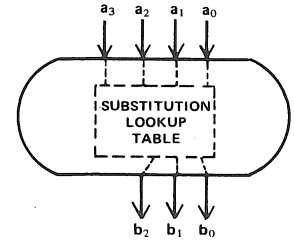
(a) BALANCED



(b) EXCLUSIVE OR (MODULO 2 BIT-BY-BIT ADDITION)



(c) EXPANDED



(d) CONTRACTED

FIGURE 2
SUBSTITUTIONS WITH 4-BIT INPUTS

In fact, an invertible balanced permutation (Figure 1a) may be regarded as a balanced substitution (Figure 2a) in which the number of output 1s is equal to the number of input 1s and the number of output 0s is equal to the number of input 0s for every input bit combination.

The distinction between an invertible process and a noninvertible process may be seen in figures 3a, 3b and 3c. In Figure 3a, every output bit combination (000, 001 . . . 111) occurs only once, and every input bit configuration has a corresponding output.

With these characteristics, the inverse of the process can be constructed, as shown in Figure 3b.

By contrast, the process shown in Figure 3c is not invertible because the output configurations 110 and 101 occur twice and the configurations 000 and 010 do not occur at all.

Thus, if an attempt were made to construct the inverse of Figure 3c, the outputs would be undefined for input values of 000, 010, 101 and 110.

The invertibility property may also be applied to permutations. A permutation is invertible if each input bit is tied to one and only one output bit.

The cyclic permutation, Figure 1c, is a special case of an invertible balanced permutation. The inverse of Figure 1c would be a "cycle right 1 bit" process.

As indicated in Figure 2, substitution may be regarded as a table lookup procedure which could be used if the function were performed by a microprocessor or could be realized by a network of logic gates.

As indicated in the standard, the method of implementation is up to the individual hardware designer.

Taking this one step further, there's no

reason why such a function can't be executed by software on a general-purpose computer, although dedicated custom-tailored hardware will accomplish the task most efficiently.

The most intriguing class of substitution, the exclusive OR process, is shown in Figure 2b; a typical truth table is given in Figure 3d.

The process is not only invertible but self-invertible. Suppose an input bit sequence of:

$a_n, a_{n-1}, \ldots, a_1, a_0$

produces a given output sequence of:

$b_n, b_{n-1}, \ldots, b_1, b_0$

If, instead,

$b_n, \ldots b_0$

is applied as the input, the output will be

$a_n \ldots a_0$

The process may be generated by a circuit which has two inputs – data and a key with equal numbers of bits. Then the output is obtained by a bit-by-bit modulo-2 addition (exclusive OR) of the key bits and data bits.

The bit-by-bit transfer functions is given by:

$0 \oplus 0 = 0;$
$0 \oplus 1 = 1;$
$1 \oplus 0 = 1;$
and
$1 \oplus 1 = 0,$

where the $\oplus$ sign symbolizes the exclusive OR operation.

In the encryption algorithm, as we shall see, the remarkable properties of the exclusive OR circuit permit a common logic to be used for encryption at one end of a transmission line and decryption at the other end.

To look at the exclusive OR process in another light, it may be noted that, if Figure 3a represented the truth table of an exclusive OR function, then Figures 3a and 3b would have been identical.

The expansion and contraction operations, where the number of output bits is not equal to the number of input bits, are used in several stages in the algorithm, along with balanced operations.

Two-Part Algorithm

The algorithm is composed of two distinct subfunctions: key generation and encryption/decryption.

Key generation (Figure 4) is the process of transforming a user-defined 64-bit key into a set of 16 different 48-bit keys, each to be used in a different iteration of the encryption/decryption procedure.

The proposed standard defines the input/output relationships of all the con-

a_2 a_1 a_0	0	1	2	3	4	5	6	7
b_2 b_1 b_0	6	5	1	4	7	0	3	2

(a) BALANCED, INVERTIBLE

a_2 a_1 a_0	0	1	2	3	4	5	6	7
b_2 b_1 b_0	5	2	7	6	3	1	0	4

(b) INVERSE OF (a) ABOVE

a_2 a_1 a_0	0	1	2	3	4	5	6	7
b_2 b_1 b_0	6	5	1	4	7	6	3	5

(c) BALANCED, NOT INVERTIBLE

a_2 a_1 a_0	0	1	2	3	4	5	6	7
b_2 b_1 b_0	5	4	7	6	1	0	3	2

(d) EXCLUSIVE OR, USING A KEY VALUE OF $k_2\ k_1\ k_0 = (101)_2 = (5)_8$
(NOTE SYMMETRY OF INPUT-OUTPUT PAIRS. THE INVERSE OF THIS TRUTH TABLE IS EQUAL TO THE ORIGINAL.)

a_2 a_1 a_0	0	1	2	3	4	5	6	7
b_3 b_2 b_1 b_0	C	5	2	4	E	6	A	0

(e) EXPANDED (4-BIT OUTPUT VALUES SHOWN IN HEXIDECIMAL NOTATION)

a_2 a_1 a_0	0	1	2	3	4	5	6	7
b_1 b_0	3	1	1	0	3	2	3	1

(f) CONTRACTED (2-BIT OUTPUT)

FIGURE 3
TYPICAL TRUTH TABLES FOR THE CLASSES OF SUBSTITUTIONS ILLUSTRATED IN FIGURE 2, WITH 3-BIT INPUTS. INPUT AND OUTPUT VALUES ARE GIVEN OCTAL NOTATION UNLESS OTHERWISE NOTED.

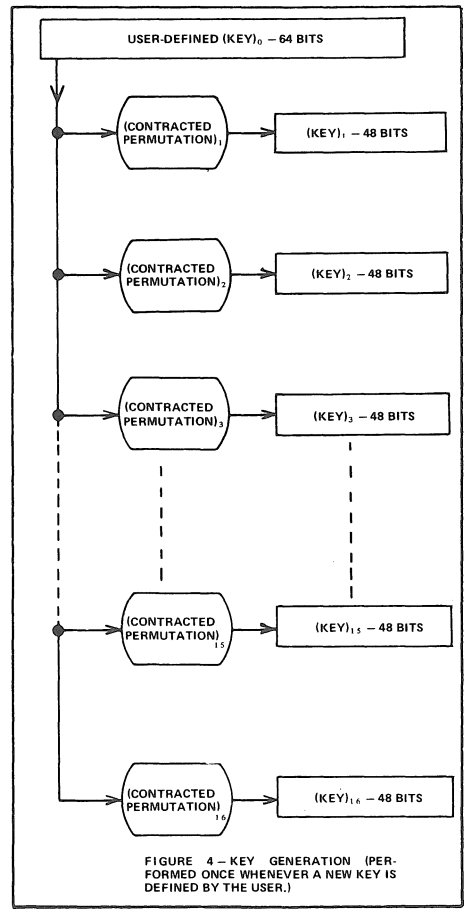


FIGURE 4 – KEY GENERATION (PERFORMED ONCE WHENEVER A NEW KEY IS DEFINED BY THE USER.)

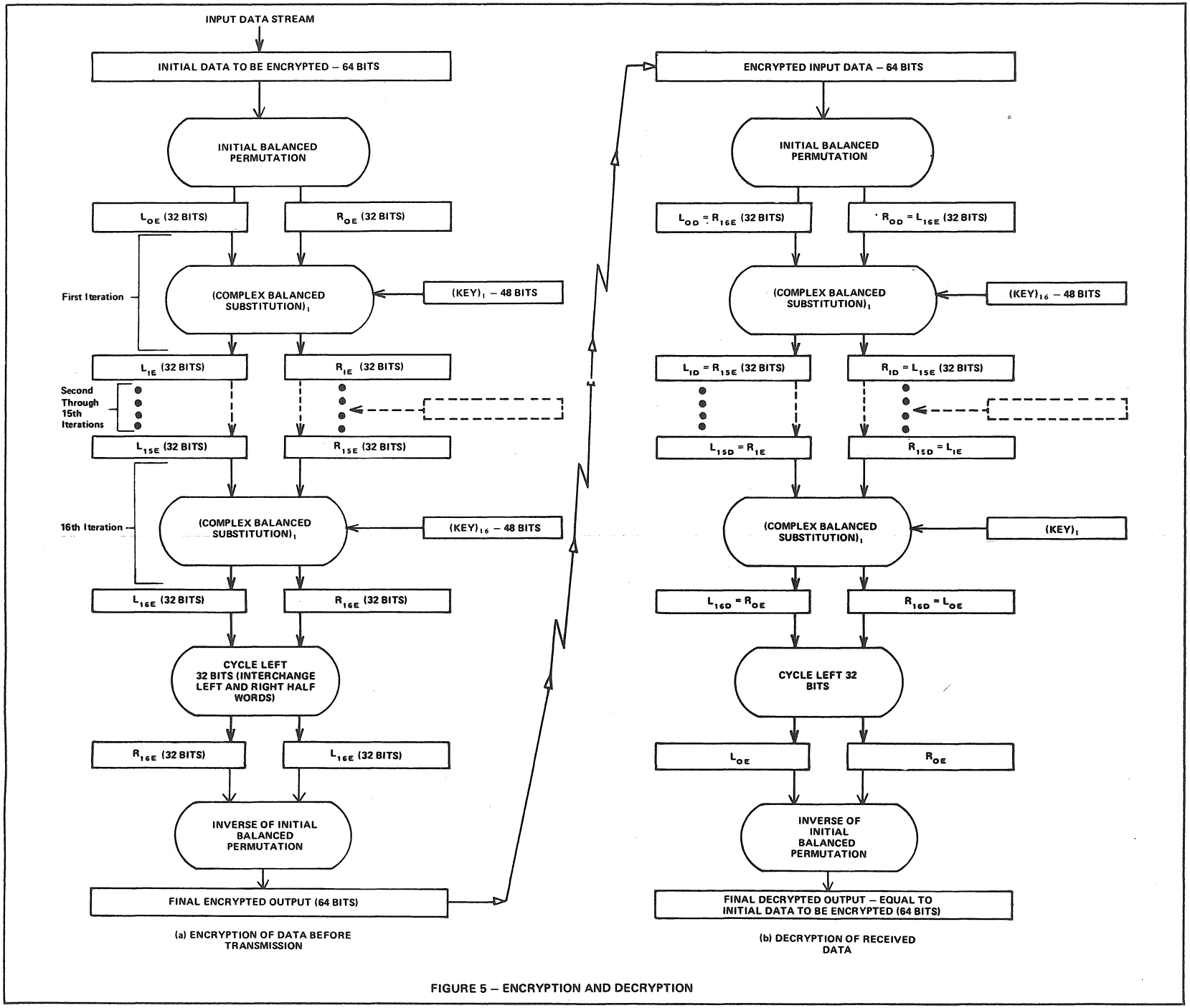


FIGURE 5 – ENCRYPTION AND DECRYPTION

tracted permutations given in Figure 4, as well as all of the other permutations and substitutions of the algorithm. The substitutions (other than exclusive OR) are defined in truth tables, as in Figure 3, and the permutations are delineated by showing tables of input/output bit "connections," illustrated symbolically in Figure 1.

Each contracted permutation in Figure 4 is actually composed of a balanced permutation followed by a cycle left operation and a contracted permutation. The result is logically equivalent to a series of contracted permutations, as indicated in the figure.

Strictly speaking, a user could supply $(KEY)_1$ through $(KEY)_{16}$ directly, without building the hardware to perform the permutations, since all of the 48-bit keys can be predicted for any given 64-bit master key.

For example, an off-line program could easily print a series of keys that could be loaded simultaneously into the 48-bit registers, for each of a limited number of master keys.

Such an approach may be reasonable because the intent of the algorithm appears to be to use a given master key over a long period of time – a matter of hours, days or weeks, for instance.

This conclusion is inferred from the language of the proposed standard, which states, "Data may be protected against unauthorized disclosure by generating a random key and issuing it to the authorized users of the data."

This indicates it isn't necessary to supply a new 64-bit key for every 64-bit data block. That would require synchronization of the encrypting and decrypting key generators so the same key could be used to encipher and decipher each data block.

Instead, a federal agency with a time-sharing system or a remote batch operation can select, for example, a daily key to be used at each transmitting site and the central processing facility. Obviously, any private organization choosing to adhere to the standard can use the same technique for its own operations.

The encryption/decryption process itself is shown in Figure 5. It consists of several steps to be performed on each 64-bit input data block: an initial balanced permutation; a series of 16 iterations of a complex cipher, each iteration using a separate 48-bit key; a cycle left 32 bits (nothing more than an interchange of the left and right half-word); and a final balanced permutation which is chosen to be the inverse of the initial balanced permutation.

The decryption process uses precisely the same bit manipulation stages as the encryption algorithm. The only difference is that the keys are used in the reverse order, beginning with $(KEY)_{16}$ and ending with $(KEY)_1$.

Figure 6 shows the n^{th} iteration in the encryption process, a complex function that encompasses permutation, substitution, expansion and contraction.

If the same process is used for decryption, it can be shown that

$$L_{OD}\ (\text{decryption}) = R_{16E}\ (\text{encryption})$$
$$R_{OD} = L_{16E},\ L_{ID} = R_{15E}$$

and so on. Thus, following the final exchange of left and right half-words and the final permutation, the output of the decryption process is equal to the input to the encryption process, where it all began.

It is worth noting the nature of the contracted substitution in Figure 6, which is composed of a set of eight so-called "primitive functions," each with a 6-bit input and a 4-bit output.

These functions are recommended for inclusion in the algorithm, so there's an implication the user can choose his own primitive functions or his own contracted substitution if he wishes.

This contrasts with the practice (in the proposed standard) of dictating all of the other bit manipulation functions, without any hint of freedom of choice for the user.

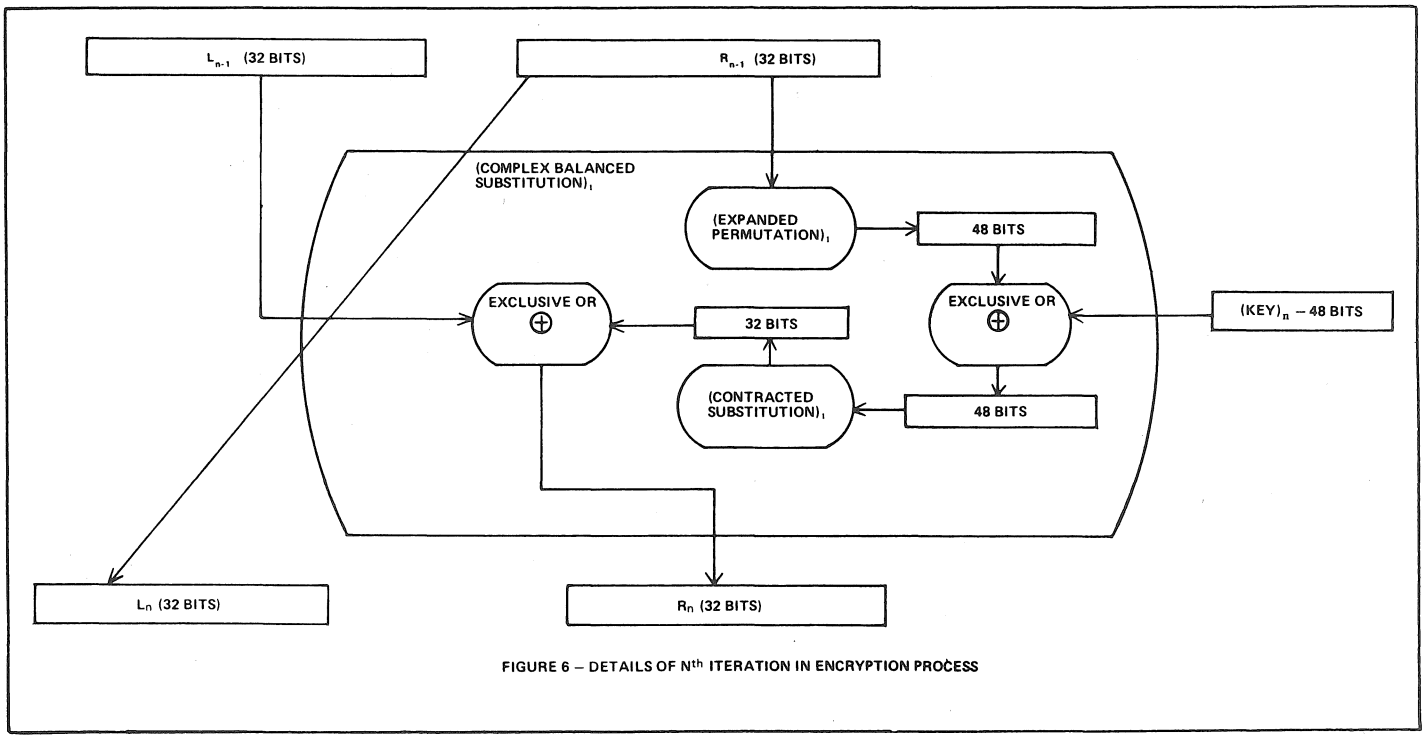


FIGURE 6 – DETAILS OF N^{th} ITERATION IN ENCRYPTION PROCESS

8080 TECO and FLOATING POINT PACK

Dear Jim, June 23, 1976

Congratulations on the *Journal*! This approach to software development and to Tiny BASIC in particular is most stimulating.

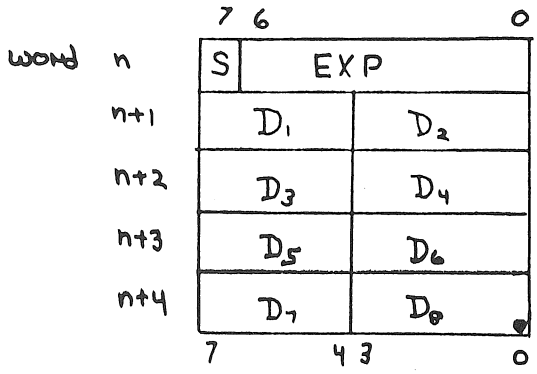
I would, however, be interested in seeing more support of the 8008-based systems. Presently, I'm running a modified Mark 8 with 15K RAM, 1K ROM operating system, 256 level hardware stack, TVT-I, Suding Calculator interface, ASCII keyboard, National Multiplex Digital Data Recorder (2400 baud) driven by serial-parallel interface of my own design.

Recently, I've been working on and have nearly debugged a 4K emulation of most of the commands of DEC's TECO, a powerful text editor. Unfortunately, I have neither an assembler or hardcopy printer, and hand documentation for publication is too time consuming. Any suggestions as to methods of placing TEXTED (as I call it) before the public eye would be appreciated. **[Suggestions, anyone?]**

Tiny BASIC is intriguing, but a bit too tiny. Floating point capabilities are a must. Hence, for the last week, I've been working on a software package for the Suding Calculator. Up and running (with only one or two known bugs) are routines to read/write from/to the Calculator, input to RAM storage a floating point number from keyboard, output from RAM storage a floating point number to TVT.

To assist anyone working with the Calculator and to invite comments from readers, I offer these notes:

1) Storage format for FP numbers. After long consideration, I have decided upon a 5 (8-bit) word convention as follows:



In the format above, the 8 significant digits from the Calculator are stored two BCD (0-9) digits per word as represented by D_1-D_8 with the sign of the entire number indicated by the sign bit "S" (1 if negative). The exponent or scale factor "EXP" is stored in 7 bit, two's complement binary with the assumed decimal point falling after the last digit, D_8. Thus, the useful range is from $\pm 1. \times 10^{-64}$ to $\pm 99999999. \times 10^{+63}$. Although this format is not as compact as a signed magnitude or two's complement binary, it results in considerable savings of time and programming effort in reading/writing from/to the Calculator.

2) Output printing format for FP numbers. Numbers are printed with a leading negative sign (if appropriate) as floating point in the range of ±0.00000001 to ±99999999, and as scientific notation in the range of ±9.9999999E-09 to ±E-64 and of ±1E+08 to ±9.9999999E+70.

3) FP Package calling sequences. It is planned to have Register A contain an operation code (i.e., output, input, multiply, divide, etc.), Registers H and L a pointer to the 5 word FP accumulator in which the result is formed on return, and Registers D and E a pointer to the 5 word FP operand (where applicable). This use of two word address pointers would simplify linkage of the FP package to existing Tiny BASICs now using two word integer arithmetic and operation stacks. Other planned operations (besides the functions provided by the Calculator) would include FP compare, clear, move, and scale.

4) Bugs. Suding's suggested procedure of outputting a Digit 9 request and waiting for the Data Available MSB line to go true to indicate that the Calculator's internal calculations are completed does not allow sufficient time for long calculations such as y^x or N!. Any suggestions would be appreciated. Calculator functions DGR/RAD and Σ do not seem to work or else I'm using them incorrectly.

I'll keep plugging away at a Calculator FP package and if time permits, write up the results for *DDJ*. **[Do! Do!]** I've also had considerable experience developing expression handlers using Polish notation and operation stacks. So I'll also be thinking of developing multi-parameter function call and return sequences. Meanwhile, keep those *Journals* coming!!!!

Sincerely,

William E. Severence, Jr. Center Lovell ME 04016
William E. Severance Realtors (207) 925-2271

P.S. Our DEC Datasystem 310W should be arriving in another 1½ months. If I decide to purchase the scientific operating system with assembler, I'll hopefully be able to write an 8008 cross-assembler.

RANDOM DATA

by Robert Reiling
(reprinted with permission from
Homebrew Computer Club Newsletter, Vol. 2, No. 5)

Kenneth Young wrote in with a report of the last **Southern California Computer Society** (SCCS) meeting and inlcuded his observations about the **Altair 8800B**. The following exerpt is from Ken's letter: *"Pat Ward and David Bunnell of MITs were one of the vendors at the meeting and they were showing off the Altair 8800B. I liked it. They cleaned up their computer a lot. I feel that this machine is comparable to the IMSAI 8080 with a 22-slot motherboard and fan. Unfortunately, the Altair 8800B will cost about $840 in kit form. The IMSAI 8080 with a 28-Amp power supply, 22-slot motherboard and fan costs $680 (IMSAI does not charge that insulting 5% handling fee anymore, so I have been told). The two computers are a little different, but I do feel they are comparable to one another. However, I do not feel that the Altair 8800B should cost more than the IMSAI. My conclusion is that the IMSAI is still a better deal than the Altair."*

Jef Raskin's discussion of FLOW, an instructional computer programming language, at the May 26th meeting of the **Homebrew Computer Club** interested quite a few members. For a brief discussion of the language, have your librarian get a copy of *Computers And The Humanities, Vol. 8, pp 231-237,* Pergamon Press, 1974.

Floating Point Routines for the 6502

by Roy Rankin, Department of Mechanical Engineering,
Stanford University, Stanford, CA 94305
(415) 497-1822

and

Steve Wozniak, Apple Computer Company
770 Welch Road, Suite 154
Palo Alto, CA 94304
(415) 326-4248

Editor's Note: Although these routines are for the 6502, it would appear that one could generate equivalent routines for most of the "traditional" microprocessors, relatively easily, by following the flow of the algorithms given in the excellent comments included in the program listing. This is particularly true of the transcendental functions which were directly modeled after well-known and proven algorithms, and for which, the comments are relatively machine-independent.

These floating point routines allow 6502 users to perform most of the more popular and desired floating point and transcendental functions, namely:

Natural Log - LOG
Common Log - LOG10
Exponential - EXP
Floating Add - FADD
Floating Subtract - FSUB
Floating Multiply - FMUL
Floating Divide - FDIV
Convert Floating to Fixed - FIX
Convert Fixed to Floating - FLOAT

They presume a four-byte floating point operand consisting of a one-byte exponent ranging from -218 through +127, and a 24-bit two's complement mantissa between 1.0 and 2.0.

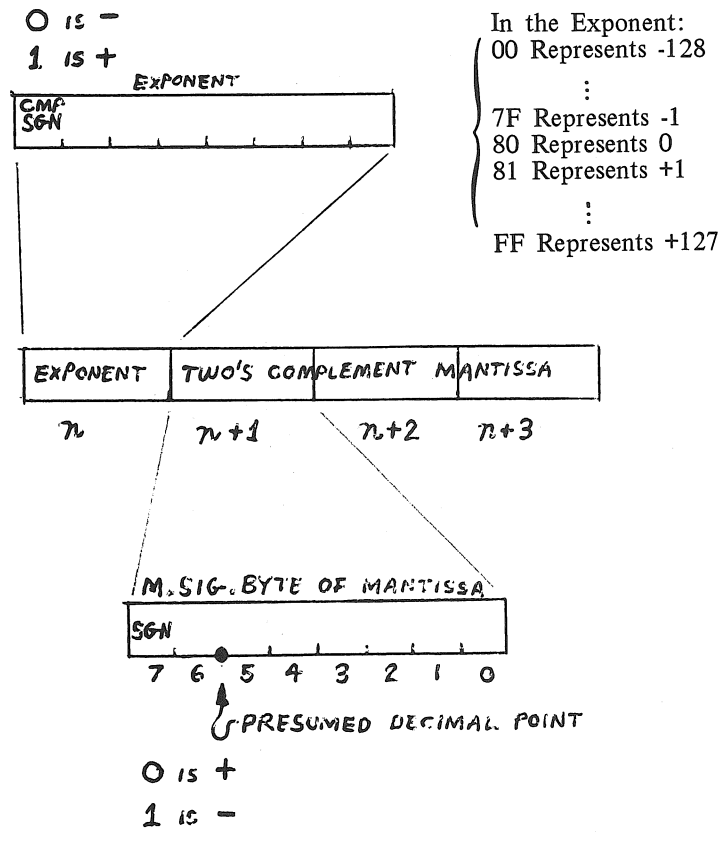
The floating point routines were done by Steve Wozniak, one of the principals in Apple Computer Company. The transcendental functions were patterned after those offered by Hewlett-Packard for their HP2100 minicomputer (with some modifications), and were done by Roy Rankin, a Ph.D. student at Stanford University.

There are three error traps; two for overflow, and one for prohibited logarithm argument. ERROR (1DO6) is the error exit used in event of a non-positive log argument. OVFLW (1E3B) is the error exit for overflow occuring during calculation of e to some power. OVFL (1FE4) is the error exit for overflow in all of the floating point routines. There is no trap for underflow; in such cases, the result is set to 0.0.

All routines are called and exited in a uniform manner: The argument(s) are placed in the specified floating point storage locations (for specifics, see documentation preceeding each routine in the listing), then a JSR is used to enter the desired routine. Upon normal completion, the called routine is exited via a subroutine return instruction (RTS).

Note: The preceeding documentation was written by the Editor, based on phone conversations with Roy and studying the listing. There is a high probability that it is correct. However, since it was not written nor reviewed by the authors of these routines, the preceeding documentation may contain errors in concept or in detail.

-- JCW, Jr.




```
*          JULY 5, 1976
*     BASIC FLOATING POINT ROUTINES
*       FOR 6502 MICROPROCESSOR
*       BY R. RANKIN AND S. WOZNIAK
*
*     CONSISTING OF:
*        NATURAL LOG
*        COMMON LOG
*        EXPONENTIAL (E**X)
*        FLOAT       FIX
*        FADD        FSUB
*        FMUL        FDIV
*
*
*     FLOATING POINT REPRESENTATION (4-BYTES)
*                   EXPONENT BYTE 1
*                   MANTISSA BYTES 2-4
*
*     MANTISSA:    TWO'S COMPLIMENT REPRESENTATION WITH SIGN IN
*       MSB OF HIGH-ORDER BYTE.  MANTISSA IS NORMALIZED WITH AN
*       ASSUMED DECIMAL POINT BETWEEN BITS 5 AND 6 OF THE HIGH-ORDER
*       BYTE.  THUS THE MANTISSA IS IN THE RANGE 1. TO 2. EXCEPT
*       WHEN THE NUMBER IS LESS THAN 2**(-128).
*
*     EXPONENT:    THE EXPONENT REPRESENTS POWERS OF TWO.  THE
*       REPRESENTATION IS 2'S COMPLIMENT EXCEPT THAT THE SIGN
*       BIT (BIT 7) IS COMPLIMENTED.  THIS ALLOWS DIRECT COMPARISION
*       OF EXPONENTS FOR SIZE SINCE THEY ARE STORED IN INCREASING
*       NUMERICAL SEQUENCE RANGING FROM $00 (-128) TO $FF (+127)
*       ($ MEANS NUMBER IS HEXADECIMAL).
*
*     REPRESENTATION OF DECIMAL NUMBERS:    THE PRESENT FLOATING
*       POINT REPRESENTATION ALLOWS DECIMAL NUMBERS IN THE APPROXIMATE
*       RANGE OF 10**(-38) THROUGH 10**(38) WITH 6 TO 7 SIGNIFICANT
*       DIGITS.
*
*

0003                  ORG 3          SET BASE PAGE ADRESSES
0003  EA        SIGN  NOP
0004  EA        X2    NOP            EXPONENT 2
0005  00 00 00  M2    BSS 3          MANTISSA 2
0008  EA        X1    NOP            EXPONENT 1
0009  00 00 00  M1    BSS 3          MANTISSA 1
000C              E     BSS 4          SCRATCH
0010              Z     BSS 4
0014              T     BSS 4
```

```
0018                    SEXP   BSS 4
001C  00                INT    BSS 1
                        *
1D00                            ORG $1D00      STARTING LOCATION FOR LOG
                        *
                        *       NATURAL LOG OF MANT/EXP1 WITH RESULT IN MANT/EXP1
                        *
1D00  A5 09             LOG     LDA M1
1D02  F0 02                     BEQ ERROR
1D04  10 01                     BPL CONT       IF ARG>0 OK
1D06  00                ERROR   BRK            ERROR ARG<=0
                        *
1D07  20 1C 1F          CONT    JSR SWAP       MOVE ARG TO EXP/MANT2
1D0A  A5 04                     LDA X2         HOLD EXPONENT
1D0C  A0 80                     LDY =$80
1D0E  84 04                     STY X2         SET EXPONENT 2 TO 0 ($80)
1D10  49 80                     EOR =$80       COMPLIMENT SIGN BIT OF ORIGINAL EXPONENT
1D12  85 0A                     STA M1+1       SET EXPONENT INTO MANTISSA 1 FOR FLOAT
1D14  A9 00                     LDA =0
1D16  85 09                     STA M1         CLEAR MSB OF MANTISSA 1
1D18  20 2C 1F                  JSR FLOAT      CONVERT TO FLOATING POINT
1D1B  A2 03                     LDX =3         4 BYTE TRANSFERS
1D1D  B5 04             SEXP1   LDA X2,X
1D1F  95 10                     STA Z,X        COPY MANTISSA TO Z
1D21  B5 08                     LDA X1,X
1D23  95 18                     STA SEXP,X     SAVE EXPONENT IN SEXP
1D25  BD D1 1D                  LDA R22,X      LOAD EXP/MANT1 WITH SQRT(2)
1D28  95 08                     STA X1,X
1D2A  CA                        DEX
1D2B  10 F0                     BPL SEXP1
1D2D  20 4A 1F                  JSR FSUB       Z-SQRT(2)
1D30  A2 03                     LDX =3          4 BYTE TRANSFER
1D32  B5 08             SAVET   LDA X1,X       SAVE EXP/MANT1 AS T
1D34  95 14                     STA T,X
1D36  B5 10                     LDA Z,X        LOAD EXP/MANT1 WITH Z
1D38  95 08                     STA X1,X
1D3A  BD D1 1D                  LDA R22,X      LOAD EXP/MANT2 WITH SQRT(2)
1D3D  95 04                     STA X2,X
1D3F  CA                        DEX
1D40  10 F0                     BPL SAVET
1D42  20 50 1F                  JSR FADD       Z+SQRT(2)
1D45  A2 03                     LDX =3         4 BYTE TRANSFER
1D47  B5 14             TM2     LDA T,X
1D49  95 04                     STA X2,X       LOAD T INTO EXP/MANT2
1D4B  CA                        DEX
1D4C  10 F9                     BPL TM2
1D4E  20 9D 1F                  JSR FDIV       T=(Z-SQRT(2))/(Z+SQRT(2))
1D51  A2 03                     LDX =3         4 BYTE TRANSFER
1D53  B5 08             M1T     LDA X1,X
1D55  95 14                     STA T,X        COPY EXP/MANT1 TO T AND
1D57  95 04                     STA X2,X       LOAD EXP/MANT2 WITH T
1D59  CA                        DEX
1D5A  10 F7                     BPL M1T
1D5C  20 77 1F                  JSR FMUL  T*T
1D5F  20 1C 1F                  JSR SWAP       MOVE T*T TO EXP/MANT2
1D62  A2 03                     LDX =3         4 BYTE TRANSFER
1D64  BD E1 1D          M1C     LDA C,X
1D67  95 08                     STA X1,X       LOAD EXP/MANT1 WITH C
1D69  CA                        DEX
1D6A  10 F8                     BPL M1C
1D6C  20 4A 1F                  JSR FSUB       T*T-C
1D6F  A2 03                     LDX =3         4 BYTE TRANSFER
1D71  BD DD 1D          M2MB    LDA MB,X
1D74  95 04                     STA X2,X       LOAD EXP/MANT2 WITH MB
1D76  CA                        DEX
1D77  10 F8                     BPL M2MB
1D79  20 9D 1F                  JSR FDIV       MB/(T*T-C)
1D7C  A2 03                     LDX =3         4 BYTE TRANSFER
1D7E  BD D9 1D          M2A1    LDA A1,X
1D81  95 04                     STA X2,X       LOAD EXP/MANT2 WITH A1
1D83  CA                        DEX
1D84  10 F8                     BPL M2A1
1D86  20 50 1F                  JSR FADD       MB/(T*T-C)+A1
1D89  A2 03                     LDX =3         4 BYTE TRANSFER
1D8B  B5 14             M2T     LDA T,X
1D8D  95 04                     STA X2,X       LOAD EXP/MANT2 WITH T
1D8F  CA                        DEX
1D90  10 F9                     BPL M2T
1D92  20 77 1F                  JSR FMUL       (MB/(T*T-C)+A1)*T
1D95  A2 03                     LDX =3         4 BYTE TRANSFER
1D97  BD E5 1D          M2MHL   LDA MHLF,X
1D9A  95 04                     STA X2,X       LOAD EXP/MANT2 WITH MHLF (.5)
1D9C  CA                        DEX
1D9D  10 F8                     BPL M2MHL
1D9F  20 50 1F                  JSR FADD       +.5
1DA2  A2 03                     LDX =3         4 BYTE TRANSFER
1DA4  B5 18             LDEXP   LDA SEXP,X
1DA6  95 04                     STA X2,X       LOAD EXP/MANT2 WITH ORIGINAL EXPONENT
1DA8  CA                        DEX
1DA9  10 F9                     BPL LDEXP
1DAB  20 50 1F                  JSR FADD       +EXPN
1DAE  A2 03                     LDX =3         4 BYTE TRANSFER
1DB0  BD D5 1D          MLE2    LDA LE2,X
1DB3  95 04                     STA X2,X       LOAD EXP/MANT2 WITH LN(2)
1DB5  CA                        DEX
1DB6  10 F8                     BPL MLE2
1DB8  20 77 1F                  JSR FMUL    *LN(2)
1DBB  60                        RTS            RETURN RESULT IN MANT/EXP1
                        *
                        *       COMMON LOG  OF MANT/EXP1 RESULT IN MANT/EXP1
                        *
1DBC  20 00 1D          LOG10   JSR LOG        COMPUTE NATURAL LOG
1DBF  A2 03                     LDX =3

1DC1  BD CD 1D          L10     LDA LN10,X
1DC4  95 04                     STA X2,X       LOAD EXP/MANT2 WITH 1/LN(10)
1DC6  CA                        DEX
1DC7  10 F8                     BPL L10
1DC9  20 77 1F                  JSR FMUL       LOG10(X)=LN(X)/LN(10)
1DCC  60                        RTS
                        *
1DCD  7E 6F             LN10    DCM 0.4342945
      2D ED
1DD1  80 5A             R22     DCM   1.4142136   SQRT(2)
      82 7A
1DD5  7F 58             LE2     DCM   0.69314718  LOG BASE E OF 2
      B9 0C
1DD9  80 52             A1      DCM   1.2920074
      80 40
1DDD  81 AB             MB      DCM   -2.6398577
      86 49
1DE1  80 6A             C       DCM   1.6567626
      00 66
1DE5  7F 40             MHLF    DCM   0.5
      00 00
                        *
1E00                            ORG $1E00      STARTING LOCATION FOR EXP
                        *
                        *     EXP OF MANT/EXP1 RESULT IN MANT/EXP1
                        *
1E00  A2 03             EXP     LDX =3         4 BYTE TRANSFER
1E02  BD D8 1E                  LDA L2E,X
1E05  95 04                     STA X2,X       LOAD EXP/MANT2 WITH LOG BASE 2 OF E
1E07  CA                        DEX
1E08  10 F8                     BPL EXP+2
1E0A  20 77 1F                  JSR FMUL       LOG2(E)*X
1E0D  A2 03                     LDX =3         4 BYTE TRANSFER
1E0F  B5 08             FSA     LDA X1,X
1E11  95 10                     STA Z,X        STORE EXP/MANT1 IN Z
1E13  CA                        DEX
1E14  10 F9                     BPL FSA        SAVE Z=LN(2)*X
1E16  20 E8 1F                  JSR FIX        CONVERT CONTENTS OF EXP/MANT1 TO AN INTEGER
1E19  A5 0A                     LDA M1+1
1E1B  85 1C                     STA INT        SAVE RESULT AS INT
1E1D  38                        SEC            SET CARRY FOR SUBTRACTION
1E1E  E9 7C                     SBC =124       INT-124
1E20  A5 09                     LDA M1
1E22  E9 00                     SBC =0
1E24  10 15                     BPL OVFLW      OVERFLOW INT>=124
1E26  18                        CLC            CLEAR CARRY FOR ADD
1E27  A5 0A                     LDA M1+1
1E29  69 78                     ADC =120       ADD 120 TO INT
1E2B  A5 09                     LDA M1
1E2D  69 00                     ADC =0
1E2F  10 08                     BPL CONTIN     IF RESULT POSITIVE CONTINUE
1E31  A9 00                     LDA =0         INT<-120 SET RESULT TO ZERO AND RETURN
1E33  A2 03                     LDX =3         4 BYTE MOVE
1E35  95 08             ZERO    STA X1,X       SET EXP/MANT1 TO ZERO
1E37  CA                        DEX
1E38  10 FB                     BPL ZERO
1E3A  60                        RTS            RETURN
                        *
1E3B  00                OVFLW   BRK            OVERFLOW
                        *
1E3C  20 2C 1F          CONTIN  JSR FLOAT      FLOAT INT
1E3F  A2 03                     LDX =3
1E41  B5 10             ENTD    LDA Z,X
1E43  95 04                     STA X2,X       LOAD EXP/MANT2 WITH Z
1E45  CA                        DEX
1E46  10 F9                     BPL ENTD
1E48  20 4A 1F                  JSR FSUB       Z=Z-FLOAT(INT)
1E4B  A2 03                     LDX =3         4 BYTE MOVE
1E4D  B5 08             ZSAV    LDA X1,X
1E4F  95 10                     STA Z,X        SAVE EXP/MANT1 IN Z
1E51  95 04                     STA X2,X       COPY EXP/MANT1 TO EXP/MANT2
1E53  CA                        DEX
1E54  10 F7                     BPL ZSAV
1E56  20 77 1F                  JSR FMUL       Z*Z
1E59  A2 03                     LDX =3         4 BYTE MOVE
1E5B  BD DC 1E          LA2     LDA A2,X
1E5E  95 04                     STA X2,X       LOAD EXP/MANT2 WITH A2
1E60  B5 08                     LDA X1,X
1E62  95 18                     STA SEXP,X     SAVE EXP/MANT1 AS SEXP
1E64  CA                        DEX
1E65  10 F4                     BPL LA2
1E67  20 50 1F                  JSR FADD       Z*Z+A2
1E6A  A2 03                     LDX =3         4 BYTE MOVE
1E6C  BD E0 1E          LB2     LDA B2,X
1E6F  95 04                     STA X2,X       LOAD EXP/MANT2 WITH B2
1E71  CA                        DEX
1E72  10 F8                     BPL LB2
1E74  20 9D 1F                  JSR FDIV       T=B2/(Z*Z+A2)
1E77  A2 03                     LDX =3         4 BYTE MOVE
1E79  B5 08             DLOAD   LDA X1,X
1E7B  95 14                     STA T,X        SAVE EXP/MANT1 AS T
1E7D  BD E4 1E                  LDA C2,X
1E80  95 08                     STA X1,X       LOAD EXP/MANT1 WITH C2
1E82  B5 18                     LDA SEXP,X
1E84  95 04                     STA X2,X       LOAD EXP/MANT2 WITH SEXP
1E86  CA                        DEX
1E87  10 F0                     BPL DLOAD
1E89  20 77 1F                  JSR FMUL       Z*Z*C2
1E8C  20 1C 1F                  JSR SWAP       MOVE EXP/MANT1 TO EXP/MANT2
1E8F  A2 03                     LDX =3         4 BYTE TRANSFER
1E91  B5 14             LTMP    LDA T,X
1E93  95 08                     STA X1,X       LOAD EXP/MANT1 WITH T
1E95  CA                        DEX
```

```
1E96  10 F9           BPL LTMP
1E98  20 4A 1F        JSR FSUB     C2*Z*Z-B2/(Z*Z+A2)
1E9B  A2 03           LDX =3       4 BYTE TRANSFER
1E9D  BD E8 1E  LDD   LDA D,X
1EA0  95 04           STA X2,X     LOAD EXP/MANT2 WITH D
1EA2  CA              DEX
1EA3  10 F8           BPL LDD
1EA5  20 50 1F        JSR FADD     D+C2*Z*Z-B2/(Z*Z+A2)
1EA8  20 1C 1F        JSR SWAP     MOVE EXP/MANT1 TO EXP/MANT2
1EAB  A2 03           LDX =3       4 BYTE TRANSFER
1EAD  B5 10     LFA   LDA Z,X
1EAF  95 08           STA X1,X     LOAD EXP/MANT1 WITH Z
1EB1  CA              DEX
1EB2  10 F9           BPL LFA
1EB4  20 4A 1F        JSR FSUB     -Z+D+C2*Z*Z-B2/(Z*Z+A2)
1EB7  A2 03           LDX =3       4 BYTE TRANSFER
1EB9  B5 10     LF3   LDA Z,X
1EBB  95 04           STA X2,X     LOAD EXP/MANT2 WITH Z
1EBD  CA              DEX
1EBE  10 F9           BPL LF3
1EC0  20 9D 1F        JSR FDIV     Z/(**** )
1EC3  A2 03           LDX =3       4 BYTE TRANSFER
1EC5  BD E5 1D  LD12  LDA MHLF,X
1EC8  95 04           STA X2,X     LOAD EXP/MANT2 WITH .5
1ECA  CA              DEX
1ECB  10 F8           BPL LD12
1ECD  20 50 1F        JSR FADD     +Z/(***)+.5
1ED0  38              SEC          ADD INT TO EXPONENT WITH CARRY SET
1ED1  A5 1C           LDA INT      TO MULTIPLY BY
1ED3  65 08           ADC X1       2**(INT+1)
1ED5  85 08           STA X1       RETURN RESULT TO EXPONENT
1ED7  60              RTS          RETURN ANS=(.5+Z/(-Z+D+C2*Z*Z-B2/(Z*
1ED8  80 5C     L2E   DCM 1.4426950409   LOG BASE 2 OF E
      55 1E
1EDC  86 57     A2    DCM 87.417497202
      6A E1
1EE0  89 4D     B2    DCM 617.9722695                  Z+A2))*2**(INT+1)
      3F 1D
1EE4  7B 46     C2    DCM .03465735903
      FA 70
1EE8  83 4F     D     DCM 9.9545957821
      A3 03
                *
                *
                *       BASIC FLOATING POINT ROUTINES
                *
1F00                  ORG $1F00    START OF BASIC FLOATING POINT ROUTINES
1F00  18        ADD   CLC          CLEAR CARRY
1F01  A2 02           LDX =$02     INDEX FOR 3-BYTE ADD
1F03  B5 09     ADD1  LDA M1,X
1F05  75 05           ADC M2,X     ADD A BYTE OF MANT2 TO MANT1
1F07  95 09           STA M1,X
1F09  CA              DEX          ADVANCE INDEX TO NEXT MORE SIGNIF.BYTE
1F0A  10 F7           BPL ADD1     LOOP UNTIL DONE
1F0C  60              RTS          RETURN
1F0D  06 03     MD1   ASL SIGN     CLEAR LSB OF SIGN
1F0F  20 12 1F        JSR ABSWAP   ABS VAL OF MANT1, THEN SWAP MANT2
1F12  24 09     ABSWAP BIT M1      MANT1 NEG?
1F14  10 05           BPL ABSWP1   NO,SWAP WITH MANT2 AND RETURN
1F16  20 8F 1F        JSR FCOMPL   YES, COMPLIMENT IT.
1F19  E6 03           INC SIGN     INCR SIGN, COMPLEMENTING LSB
1F1B  38        ABSWP1 SEC         SET CARRY FOR RETURN TO MUL/DIV
                *
                *       SWAP EXP/MANT1 WITH EXP/MANT2
                *
1F1C  A2 04     SWAP  LDX =$04     INDEX FOR 4-BYTE SWAP
1F1E  94 0B     SWAP1 STY E-1,X
1F20  B5 07           LDA X1-1,X   SWAP A BYTE OF EXP/MANT1 WITH
1F22  B4 03           LDY X2-1,X   EXP/MANT2 AND LEAVEA COPY OF
1F24  94 07           STY X1-1,X   MANT1 IN E(3BYTES). E+3 USED.
1F26  95 03           STA X2-1,X
1F28  CA              DEX          ADVANCE INDEX TO NEXT BYTE
1F29  D0 F3           BNE SWAP1    LOOP UNTIL DONE.
1F2B  60              RTS
                *
                *
                *
                *     CONVERT 16 BIT INTEGER IN M1(HIGH) AND M1+1(LOW) TO
                *     RESULT IN EXP/MANT1. EXP/MANT2 UNEFFECTED      F.P.
                *
                *
1F2C  A9 8E     FLOAT LDA =$8E
1F2E  85 08           STA X1       SET EXPN TO 14 DEC
1F30  A9 00           LDA =0       CLEAR LOW ORDER BYTE
1F32  85 0B           STA M1+2
1F34  F0 08           BEQ NORM     NORMALIZE RESULT
1F36  C6 08     NORM1 DEC X1       DECREMENT EXP1
1F38  06 0B           ASL M1+2
1F3A  26 0A           ROL M1+1     SHIFT MANT1 (3 BYTES) LEFT
1F3C  26 09           ROL M1
1F3E  A5 09     NORM  LDA M1       HIGH ORDER MANT1 BYTE
1F40  0A              ASL A        UPPER TWO BITS UNEQUAL?
1F41  45 09           EOR M1
1F43  30 04           BMI RTS1     YES,RETURN WITH MANT1 NORMALIZED
1F45  A5 08           LDA X1       EXP1 ZERO?
1F47  D0 ED           BNE NORM1    NO, CONTINUE NORMALIZING
1F49  60        RTS1  RTS          RETURN
                *
                *
                *     EXP/MANT2-EXP/MAN1 RESULT IN EXP/MANT1
                *
1F4A  20 8F 1F  FSUB  JSR FCOMPL   COMPL MANT1 CLEARS CARRY UNLESS ZERO
1F4D  20 5D 1F  SWPALG JSR ALGNSW  RIGHT SHIFT MANT1 OR SWAP WITH
                *                                    MANT2 ON CARRY

                *     ADD EXP/MANT1 AND EXP/MANT2 RESULT IN EXP/MANT1
                *
1F50  A5 04     FADD  LDA X2
1F52  C5 08           CMP X1       COMPARE EXP1 WITH EXP2
1F54  D0 F7           BNE SWPALG   IF UNEQUAL, SWAP ADDENDS OR ALIGN MANTISSAS
1F56  20 00 1F        JSR ADD      ADD ALIGNED MANTISSAS
1F59  50 E3     ADDEND BVC NORM    NO OVERFLOW, NOMALIZE RESULTS
1F5B  70 05           BVS RTLOG    OV: SHIFT MANT1 RIGHT. NOTE CARRY IS CORRECT SIGN
1F5D  90 BD     ALGNSW BCC SWAP    SWAP IF CARRY CLEAR, ELSE SHIFT RIGHT ARITH.
1F5F  A5 09     RTAR  LDA M1       SIGN OF MANT1 INTO CARRY FOR
1F61  0A              ASL A        RIGHT ARITH SHIFT
1F62  E6 08     RTLOG INC X1       INCR EXP1 TO COMPENSATE FOR RT SHIFT
1F64  F0 7E           BEQ OVFL     EXP1 OUT OF RANGE
1F66  A2 FA     RTLOG1 LDX =$FA    INDEX FOR 6 BYTE RIGHT SHIFT
1F68  A9 80     ROR1  LDA =$80
1F6A  B0 01           BCS ROR2
1F6C  0A              ASL A
1F6D  56 0F     ROR2  LSR E+3,X    SIMULATE ROR E+3,X
1F6F  15 0F           ORA E+3,X
1F71  95 0F           STA E+3,X
1F73  E8              INX          NEXT BYTE OF SHIFT
1F74  D0 F2           BNE ROR1     LOOP UNTIL DONE
1F76  60              RTS          RETURN
                *
                *
                *     EXP/MANT1 X EXP/MANT2 RESULT IN EXP/MANT1
                *
1F77  20 0D 1F  FMUL  JSR MD1      ABS. VAL OF THE MANT1, MANT2
1F7A  65 08           ADC X1       ADD EXP1 TO EXP2 FOR PRODUCT EXPONENT
1F7C  20 CD 1F        JSR MD2      CHECK PRODUCT EXP AND PREPARE FOR MUL
1F7F  18              CLC          CLEAR CARRY
1F80  20 66 1F  MUL1  JSR RTLOG1   MANT1 AND E RIGHT.(PRODUCT AND MPLIER)
1F83  90 03           BCC MUL2     IF CARRY CLEAR, SKIP PARTIAL PRODUCT
1F85  20 00 1F        JSR ADD      ADD MULTIPLICAN TO PRODUCT
1F88  88        MUL2  DEY          NEXT MUL ITERATION
1F89  10 F5           BPL MUL1     LOOP UNTIL DONE
1F8B  46 03     MDEND LSR SIGN     TEST SIGN (EVEN/ODD)
1F8D  90 AF     NORMX BCC NORM      IF EXEN, NORMALIZE PRODUCT. ELSE COMPLEMENT
1F8F  38        FCOMPL SEC         SET CARRY FOR SUBTRACT
1F90  A2 03           LDX =$03     INDEX FOR 3-BYTE SUBTRACTION
1F92  A9 00     COMPL1 LDA =$00    CLEAR A
1F94  F5 08           SBC X1,X     SUBTRACT BYTE OF EXP1
1F96  95 08           STA X1,X     RESTORE IT
1F98  CA              DEX          NEXT MORE SIGNFICANT BYTE
1F99  D0 F7           BNE COMPL1   LOOP UNTIL DONE
1F9B  F0 BC           BEQ ADDEND   NORMALIZE (OR SHIFT RIGHT IF OVERFLOW)
                *
                *
                *     EXP/MANT2 / EXP/MANT1 RESULT IN EXP/MANT1
                *
1F9D  20 0D 1F  FDIV  JSR MD1      TAKE ABS VAL OF MANT1, MANT2
1FA0  E5 08           SBC X1       SUBTRACT EXP1 FROM EXP2
1FA2  20 CD 1F        JSR MD2      SAVE AS QUOTIENT EXP
1FA5  38        DIV1  SEC          SET CARRY FOR SUBTRACT
1FA6  A2 02           LDX =$02     INDEX FOR 3-BYTE INSTRUCTION
1FA8  B5 05     DIV2  LDA M2,X
1FAA  F5 0C           SBC E,X      SUBTRACT A BYTE OF E FROM MANT2
1FAC  48              PHA          SAVE ON STACK
1FAD  CA              DEX          NEXT MORE SIGNIF BYTE
1FAE  10 F8           BPL DIV2     LOOP UNTIL DONE
1FB0  A2 FD           LDX =$FD     INDEX FOR 3-BYTE CONDITIONAL MOVE
1FB2  68        DIV3  PLA          PULL A BYTE OF DIFFERENCE OFF STACK
1FB3  90 02           BCC DIV4     IF MANT2<E THEN DONT RESTURE MANT2
1FB5  95 08           STA M2+3,X
1FB7  E8        DIV4  INX          NEXT LESS SIGNIF BYTE
1FB8  D0 F8           BNE DIV3     LOOP UNTIL DONE
1FBA  26 0B           ROL M1+2
1FBC  26 0A           ROL M1+1     ROLL QUOTIENT LEFT. CARRY INTO LSB
1FBE  26 09           ROL M1
1FC0  06 07           ASL M2+2
1FC2  26 06           ROL M2+1     SHIFT DIVIDEND LEFT

1FC4  26 05           ROL M2
1FC6  B0 1C           BCS OVFL     OVERFLOW IS DUE TO UNNORMALIZED DIVISOR
1FC8  88              DEY          NEXT DIVIDE ITERATION
1FC9  D0 DA           BNE DIV1     LOOP UNTIL DONE 23 ITERATIONS
1FCB  F0 BE           BEQ MDEND    NORMALIZE QUOTIENT AND CORRECT SIGN
1FCD  86 0B     MD2   STX M1+2
1FCF  86 0A           STX M1+1     CLR MANT1 (3 BYTES) FOR MUL/DIV
1FD1  86 09           STX M1
1FD3  B0 0D           BCS OVCHK    IF EXP CALC SET CARRY, CHECK FOR OVFL
1FD5  30 04           BMI MD3      IF NEG NO UNDERFLOW
1FD7  68              PLA          POP ONE
1FD8  68              PLA          RETURN LEVEL
1FD9  90 B2           BCC NORMX    CLEAR X1 AND RETURN
1FDB  49 80     MD3   EOR =$80     COMPLIMENT SIGN BIT OF EXP
1FDD  85 08           STA X1       STORE IT
1FDF  A0 17           LDY =$17     COUNT FOR 24 MUL OR 23 DIV ITERATIONS
1FE1  60              RTS          RETURN
1FE2  10 F7     OVCHK BPL MD3      IF POS EXP THEN NO OVERFLOW
1FE4  00        OVFL  BRK
                *
                *
                *     CONVERT EXP/MANT1 TO INTEGER IN M1 (HIGH) AND M1+1(LOW)
                *      EXP/MANT2 UNEFFECTED
                *
1FE5  20 5F 1F        JSR RTAR     SHIFT MANT1 RT AND INCREMENT EXPNT
1FE8  A5 08     FIX   LDA X1       CHECK EXPONENT
1FEA  C9 8E           CMP =$8E     IS EXPONENT 14?
1FEC  D0 F7           BNE FIX-3    NO,SHIFT
1FEE  60        RTRN  RTS          RETURN
                      END
```

Monitor for the 6502

Jack Bradshaw
18 Harbor View Dr.
Hingham, MA 02043

This listing is a Monitor for the 6502, a-la the Ohio Scientific Model 420. The interrupt vector has been set to use John Zeiglers breakpoint routine. [*DDJ*, Vol.1, No.3]. His program needs to have a few addresses changed to use this Monitor, but they should be obvious from the subroutine names.

Ohio Scientific [11679 Hayden St., P.O. Box 269, Hiram, OH 44234] seems to be the cheapest way to go if you roll your own equipment. They sell [albeit slowly] a variety of bare boards at very reasonable prices and good quality. So far I have assembled the CPU board and 4K board, am finishing a homebrew panel, and wrote the Monitor to check things on a borrowed ASR33. The next step calls for revising the Monitor to take the Viatron keyboard and translate to ASCII with a look-up table.

The Viatron keyboard is optically encoded, has parallel output, and is a fantastic bargain at $20.00 from John Meshna [P.O. Box 62, E. Lynn MA 01904] These are new surplus with 73 (I think) keys. Anyway, I will pass along an updated version.

Eventually hope to get TB for the 6502 (if you publish a 6800 version someone will translate it overnight or vice-versa).

Note: I would like to hear from other 6502's in the area.

```
                  $
ADDR/ XX XX XX  LABEL   OP  (MODE) OPER
----  -- -- --  ----    --- (---) --...
FF00/ A9 03     MAIN    LDA (IMM) 03
FF02/ 8D 00 FC          STA (ABS) FC00
FF05/ A9 B1             LDA (IMM) B1
FF07/ 8D 00 FC          STA (ABS) FC00
FF0A/ A2 FF             LDX (IMMF) FF
FF0C/ 9A                TXS (IMP)
FF0D/ 20 5E FF          JSR (ABS) CRLF
FF10/ 20 C6 FF          JSR (ABS) INPT
FF13/ AA                TAX (IMP)
FF14/ 20 4F FF          JSR (ABS) SPAC
FF17/ E0 4C             CPX (IMM) 4C
FF19/ F0 09             BEQ (REL) LOAD
FF1B/ E0 50             CPX (IMM) 50
FF1D/ F0 14             BEQ (REL) PRNT
FF1F/ E0 47             CPX (IMM) 47
FF21/ D0 E7             BNE (REL) FF0A
FF23/ 40                RTI (IMP)
*
FF24/ 20 69 FF  LOAD    JSR (ABS) ADDR
FF27/ 20 77 FF          JSR (ABS) FILL
FF2A/ 91 FE             STA (IND,Y) FE
FF2C/ C8                INY (IMP)
FF2D/ D0 F8             BNE (REL) FF2D
FF2F/ E6 FF             INC (ZER) 00FF
FF31/ D0 F4             BNE (REL) FF27
*
FF33/ 20 69 FF  PRNT    JSR (ABS) ADDR
FF36/ 20 5E FF          JSR (ABS) CRLF
FF39/ A9 F8             LDA (IMM) F8
FF3B/ 85 FD             STA (ZER) 00FD
FF3D/ 20 4C FF          JSR (ABS) MEML
FF40/ E6 FD             INC (ZER) 00FD
FF42/ D0 F9             BNE (REL) FF3D
FF44/ AD 00 FC          LDA (ABS) FC00
FF47/ 4A                LSR (ACC)
FF48/ 90 EC             BCC (REL) FF36
FF4A/ B0 BE                       BCS (REL) FF0A
*
FF4C/ 20 88 FF  MEML    JSR (ABS) EMTY
*
FF4F/ A9 20     SPAC    LDA (IMM) 20
*
FF51/ 48        OUTP    PHA (IMP)
FF52/ AD 00 FC          LDA (ABS) FC00
FF55/ 4A                LSR (ACC)
FF56/ 4A                LSR (ACC)
FF57/ 90 F9             BCC (REL) FF52
FF59/ 68                PLA (IMP)
FF5A/ 8D 01 FC          STA (ABS) FC01
FF5D/ 60                RTS (IMP)
*
*
FF5E/ A9 0D     CRLF    LDA (IMM) 0D
FF60/ 20 51 FF          JSR (ABS) OUTP
FF63/ A9 0A              LDA (IMM) 0A
FF65/ 20 51 FF          JSR (ABS) OUTP
FF68/ 60                RTS (IMP)
*
FF69/ 20 77 FF  ADDR    JSR (ABS) FILL
*$
*
FF69/ 20 77 FF  ADDR    JSR (ABS) FILL
FF6C/ 85 FF             STA (ZER) 00FF
FF6E/ 20 77 FF          JSR (ABS) FILL
FF71/ A8                TAY (IMP)
FF72/ A9 00             LDA (IMM) 00
FF76/ 60                RTS (IMP)
*$
*
FF77/ 20 98 FF  FILL    JSR (ABS) READ
FF7A/ 0A                ASL (ACC)
FF7B/ 0A                ASL (ACC)
FF7C/ 0A                ASL (ACC)
FF7D/ 0A                ASL (ACC)
```

```
FF7E/ 85 FC          STA (ZER) 00FC
FF80/ 20 98 FF       JSR (ABS) READ
FF83/ 29 0F          AND (IMM) 0F
FF74/ 85 FE          STA (ZER) 00FE
FF83/ 29 0F          AND (IMM) 0F
FF85/ 65 FC          ADC (ZER) 00FC
FF87/ 60             RTS (IMP)
*
FF88/ B1 FE    EMTY  LDA (IND,Y) FE
FF8A/ 20 B5 FF       JSR (ABS) UNPK
FF8D/ B1 FE          LDA (IND,Y) FE
FF8F/ 20 B9 FF       JSR (ABS) UNPI
FF92/ C8             INY (IMP)
FF93/ D0 02          BNE (REL) FF97
FF95/ E6 FF          INC (ZER)
FF97/ 60             RTS (IMP)
*
FF98/ 20 C6 FF READ  JSR (ABS) INPT
FF9B/ C9 52          CMP (IMM) 52
FF9D/ D0 03          BNE (REL) FFA2
FF9F/ 4C 00 FF       JMP (ABS) MAIN
FFA2/ C9 30          CMP (IMM) 30
FFA4/ 30 F2          BMI (REL) FF4A
FFA6/ C9 3A          CMP (IMM) 3A
FFA8/ 30 0B          BMI (REL) FFB5
FFAA/ C9 41          CMP (IMM) 41
FFAC/ 30 EA          BMI (REL) FF4A
FFAE/ C9 46          CMP (IMM) 46
FFB0/ 10 E6          BPL (REL) FF4A
FFB2/ E9 07          SBC (IMM) 07
FFB4/ 60             RTI (IMP)
*
FFB5/ 4A       UNPK  LSR (ACC)
FFB6/ 4A             LSR (ACC)
FFB7/ 4A             LSR (ACC)
FFB8/ 4A             LSR (ACC)
*
FFB9/ 29 0F    UNPI  AND (IMM) 0F
FFBB/ 69 30          ADC (IMM) 30
FFBD/ C9 3A          CMP (IMM) 3A
FFBF/ 30 02          BMI (REL) FFC3
FFC1/ 69 07          ADC (IMM) 07
FFC3/ 20 51 FF       JSR (ABS) OUTP
*
FFC6/ AD 00 FC INPT  LDA (ABS) FC00
FFC9/ 4A             LSR (ACC)
FFCA/ 90 FA          BCC (REL)  FFC6
FFCC/ AD 01 FC       LDA (ABS) FC01
FFCF/ 29 7F          AND (IMM) 7F
FFD1/ C9 7F          CMP (IMM) 7F
FFD3/ F0 F1          BEQ (REL) FFC6
FFD5/ 4C 51 FF       JMP (ABS) OUTP
*
FFFC/ 00 FF          RES (RESET) VECTOR
FFFE/ 64 02          IRQ (INTERRUPT) VECTOR
*
```

TINY TIM SEEKS TINY BASIC

Dear Doctor, June 22, 1976

I am an electronics instructor and at school where I teach I have a friend named TIM. Now when I first met TIM he was just a little fella', you know, a chip off the block of silicon. Well, he has put on some pounds with all the bytes he has been consuming. Now there's almost nothing left to feed him. What we need for him is some BASIC nutrition. If you know of even a Tiny BASIC meal for him we would appreciate it.

Thanks, and thanks for TIM.

John Bottoms
United Electronics Inst.
423 Knobloch Ave.
Keffersonville IN 47130

Note: 'TIM' is a Monitor for the 6502.

A GOOD ASSEMBLER PROPOSAL

Dear Mr. Warren: June 27, 1976

All of the implementations of Tiny BASIC have been excellent, and the authors of each version deserve to be congratulated on a job well done. I agree that it may be time to ease up on Tiny for a bit, and devote more space to other equally useful items.

It seems to me that a really good assembler would be of great value to hobbyists. I am thinking primarily of an 8080 assembler, but any machine would benefit from a good assembler. By "good" I mean one with most or all of the following features:

1. Free-form source coding.
2. Symbols up to 15 characters long.
3. Symbol table assignable to tape cassette or floppy disc.
4. Provision for both local and global symbols.
5. Sophisticated macro expansion capability.
6. Operation in either absolute or relocatable mode, assuming use of a suitable link editor for relocation.
7. Optional output of object listing and/or diagnostics.
8. Optional retention of symbol table for use in debugging.
9. Automatic separation of procedural and data areas, to facilitate memory protection.
10. Optional concordance and/or memory map listing.
11. Provision for symbolic patching.
12. Optional inclusion of kitchen sink (for bit bucket overflow).

Sincerely,

Jim Day
17042 Gunther St.
Granada Hills CA 91344

SUBSCRIBERS TO *JOURNAL* SENT A SECOND COPY OF JUNE/JULY ISSUE

Our printers botched the first press run on *DDJ*, Vol.1 Number 6. They used the wrong paper (poorer quality), and they had a bad print blanket on one of the pages.

Before Picky Jim noticed it, Fast Tom (our super-duper mailer male) had processed the subscription issues and given them to the post office.

So, when we received the properly printed copies, we sent our subscribers a second copy. It involved some extra expense and effort for us, but that's the way we do things.

Lunar Lander for the 6502

by Mark Garetz

Editor's Note: Normally, we don't like to publish software listings that have no accompanying user documentation. In this case, however, we feel that (1) the program is simple, (2) the listing is very clearly annotated, (3) it's use is self-documenting during execution, and (4) it's presumption of a TIM or DEMON monitor simplifies I/O problems. Please let us know how you feel about it.

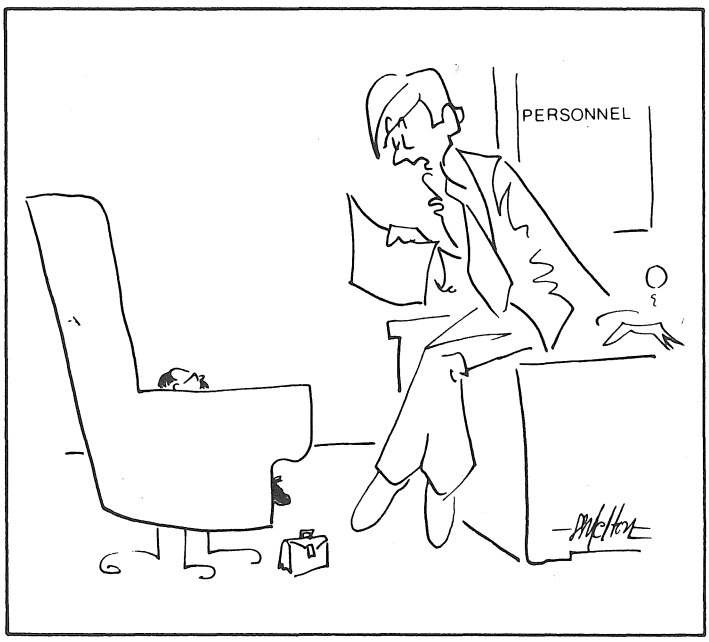
Mark assures me that use of this program is super-simple, and fool-proof. He sez:

"It requires a TIM or DEMON monitor starting at 7000H. TIM is available from MOS Technology. DEMON comes with the JOLT from Microcomputer Associates, Box 1167, Cupertino CA 95014; (408) 247-8940.

After placing the program in memory, simply start execution at 0300H.

ZERO PAGE LOCATIONS

```
00      BYTE THAT CONTAINS TIME
01      BYTE THAT CONTAINS HEIGHT LO VALUE
02      BYTE THAT CONTAINS HEIGHT HI VALUE
03      NOT USED
04      BYTE CONTAINS VELOCITY
05      OUT OF FUEL PRINTED FLAG - TELLS PROGRAM O.O.F. HAS BEEN PRINTED
06      FUEL REMAINING LO BYTE
07      FUEL REMAINING HI BYTE
08      OUT OF FUEL FLAG, IF=00, MEANS FUEL LEFT, =FF=OUT
09      SIGN FLAG, =00 IF VELOCITY IS NEGATIVE, FF IF POSITIVE
0A      CONTAINS AMOUNT OF BURN, SET TO 00 WHEN FUEL OUT
0B      SCRATCH 1
0C      SCRATCH 2
```



"I see that you have quite a lot of experience with microcomputer design, Dr. Featherstone."

Reprinted with permission of Digital Equipment Corporation.

LUNAR LANDER

```
LOC
                                                    0300  JSR  20  8A  72  CRLF SUBROUTINE IN TIM
                                                    0303  JSR  20  8A  72  AGAIN
                                                    0306  JSR  20  71  06  PRINT INSTRUCTIONS
RESTART                                             0309  JSR  20  8A  72  CRLF
                                                    030C  LDA  A9  00       LDA W/00
                                                    030E  STA  85  00       INITIALIZE TIME BYTE
                                                    0310  STA  85  01       INITIALIZE HEIGHT LO BYTE
                                                    0312  STA  85  05       INITIALIZE O.O.F. PRINTED FLAG
                                                    0314  STA  85  08       INITIALIZE OUT OF FUEL FLAG
                                                    0316  STA  85  09       INITIALIZE SIGN FLAG
                                                    0318  LDA  A9  05       LDA W/05
                                                    031A  STA  85  02       INITIALIZE HEIGHT HI BYTE
                                                    031C  LDA  A9  50       LDA W/50
                                                    031E  STA  85  04       INITIALIZE VELOCITY
                                                    0320  LDA  A9  20       LDA W/20
                                                    0322  STA  85  06       INITIALIZE FUEL LO BYTE
                                                    0324  LDA  A9  01       LDA W/01
                                                    0326  STA  85  07       INITIALIZE FUEL HI BYTE
                                                    0328  JSR  20  8A  72  CRLF
                                                    032B  JSR  20  72  07  PRINT HEADINGS
PRINT NEXT                                          032E  JSR  20  8A  72  CRLF
                                                    0331  JSR  20  12  06  PRINT DATA LINE
                                                    0334  BIT  24  08       CHECK OUT OF FUEL FLAG
                                                    0336  BPL  10  10       IF NOT OUT OF FUEL THEN INPUT BURN
                                                    0338  BIT  24  05       CHECK IF O.O.F. HAS BEEN PRINTED
                                                    033A  BMI  30  0F       YES BRANCH TO FUEL SUB
                                                    033C  JSR  20  8A  72  CRLF
                                                    033F  JSR  20  31  08  PRINT OUT OF FUEL
                                                    0342  LDA  A9  00       LDA W/00
                                                    0344  STA  85  0A       SET BURN BYTE TO 00
                                                    0346  BEQ  F0  03       BRANCH TO FUEL SUB, SKIP INPUT BURN
INPUT BURN                                          0348  JSR  20  00  05  INPUT AMOUNT TO BURN
FUEL SUB                                            034B  JSR  20  2E  05  FIGURE FUEL REMAINING
                                                    034E  JSR  20  58  05  FIGURE NEW VELOCITY
                                                    0351  LDA  A5  04       GET NEW VELOCITY
                                                    0353  BNE  D0  04       VELOCITY=00? NO, BRANCH TO HEIGHT
                                                    0355  LDA  A9  00       YES,
                                                    0357  STA  85  09       CLEAR SIGN FLAG
HEIGHT                                              0359  JSR  20  B7  05  FIGURE NEW HEIGHT
                                                    035C  JSR  20  F7  05  INCREMENT TIME BYTE
                                                    035F  LDA  A5  02       GET HEIGHT HI BYTE
                                                    0361  BEQ  F0  03       IF=00 GO TO CHECKLO
                                                    0363  JMP  4C  2E  03  IF NOT THEN GO TO PRINT NEXT
CHECKLO                                             0366  LDA  A5  01       GET HEIGHT LO BYTE
                                                    0368  BEQ  F0  03       IF=00 GO TO DOWN
                                                    036A  JMP  4C  2E  03  IF NOT THEN GO TO PRINT NEXT
DOWN                                                036D  JSR  20  8A  72  CRLF
                                                    0370  JSR  20  53  08  PRINT ON THE MOON
                                                    0373  JSR  20  8A  72  CRLF
                                                    0376  JSR  20  12  06  PRINT FINAL DATA
                                                    0379  JSR  20  8A  72  CRLF
                                                    037C  JSR  20  8A  72  CRLF
                                                    037F  LDA  A5  04       GET VELOCITY BYTE
                                                    0381  BEQ  F0  14       IF=00 BRANCH TO PERFECT
                                                    0383  CMP  C9  03       IF NOT, COMPARE TO 3
                                                    0385  BCC  90  10       IF VLO < 3, BRANCH TO PERFECT
                                                    0387  CMP  C9  07       IF VLO >=3, COMPARE TO 7
                                                    0389  BCC  90  06       IF VLO < 7, BRANCH TO STRANDED
                                                    038B  JSR  20  06  09  IF VLO >=7, PRINT "CREAMED"
                                                    038E  JMP  4C  9A  03  JMP TO TRY AGAIN
STRANDED                                            0391  JSR  20  A9  08  PRINT "STRANDED"
                                                    0394  JMP  4C  9A  03  JUMP TO TRY AGAIN
PERFECT                                             0397  JSR  20  71  08  PRINT "PERFECT"
TRY AGAIN                                           039A  JSR  20  8A  72  CRLF
                                                    039D  JSR  20  8A  72  CRLF
                                                    03A0  JSR  20  43  09  PRINT TRY AGAIN
                                                    03A3  JSR  20  E9  72  INPUT CHARACTER
                                                    03A6  CMP  C9  59       CHECK FOR "Y"
                                                    03A8  BEQ  F0  0C       IF "Y" GO TO JMPBACK
                                                    03AA  JSR  20  8A  72  CRLF
                                                    03AD  JSR  20  8A  72  CRLF
                                                    03B0  JSR  20  9A  09  IF NOT THEN PRINT CONTROL OUT
                                                    03B3  JMP  4C  86  70  RETURN CONTROL TO TIM
JMPBACK                                             03B6  JMP  4C  09  03  JUMP TO RESTART
```

LUNAR LANDER - SUBROUTINES

INPUT BURN

	LOC					
INPUT	0500	JSR	20	E9	72	INPUT CHARACTER
	0503	JSR	20	EB	73	CONVERT TO HEX
	0506	STA	85	0B		STORE AT "0B"
	0508	JSR	20	E9	72	GET NEXT CHAR.
	050B	CMP	C9	0D		CARRIAGE RETURN?
	050D	BEQ	F0	1A		IF CR BRANCH OUT
	050F	JSR	20	EB	73	CONVERT 2ND DIGIT TO HEX
	0512	STA	85	0C		STORE AT "0C"
	0514	LDA	A5	0B		LOAD 1ST DIGIT INTO A
	0516	ASL	0A			SHIFT DIGIT TO
	0517	ASL	0A			LEFT HALF OF
	0518	ASL	0A			BYTE LEAVING RIGHT
	0519	ASL	0A			HALF=00
	051A	ORA	05	0C		PUT RIGHT BYTE IN W/LEFT
	051C	CMP	C9	31		CHECK IF BURN IS OUT OF RANGE
	051E	BCS	B0	03		IF SO, BRANCH TO BURN O.R.
	0520	STA	85	0A		STORE COMPLETED BURN AT "0A"
	0522	RTS	60			RETURN
BURN O.R.	0523	JSR	20	67	09	PRINT "BURN OUT OF RANGE"
	0526	JMP	4C	00	05	GO TO INPUT
OUT	0529	LDA	A5	0B		LDA W/1ST DIGIT
	052B	STA	85	0A		STORE AT BURN LOC. (0A)
	052D	RTS	60			RETURN

FUEL SUBTRACT

	052E	SED	F8			SET DECIMAL MODE
	052F	SEC	38			SET CARRY FLAG
	0530	LDA	A5	06		GET FUEL LO BYTE
	0532	SBC	E5	0A		SUBTRACT BURN
	0534	STA	85	0B		STORE TEMP. RESULT AT 0B
	0536	LDA	A5	07		GET FUEL HI BYTE
	0538	SBC	E9	00		SUBTRACT 00
	053A	STA	85	0C		STORE TEMP. RESULT AT 0C
	053C	LDA	A5	07		GET OLD FUEL HI BYTE
	053F	CMP	C9	01		COMPARE TO 01
	0541	BEQ	F0	0C		IF=BRANCH TO STORE
	0543	LDA	A5	06		GET OLD FUEL LO BYTE
	0545	CMP	C5	0A		CHECK IF > =BURN
	0547	BEQ	F0	02		IF=BRANCH TO FLAGSET
	0549	BCS	B0	04		IF FUEL LO > BURN BRANCH TO STORE
FLAGSET	054B	LDA	A9	FF		LDA W/FF
	054D	STA	85	0B		SET OUT OF FUEL FLAG
STORE	054F	LDA	A5	0B		LOAD A W/NEW FUEL LO BYTE
	0551	STA	85	06		STORE AT FUEL LO
	0553	LDA	A5	0C		LOAD A W/NEW FUEL HI BYTE
	0555	STA	85	07		STORE AT FUEL HI
	0557	RTS	60			

VELOCITY

	0558	SED	F8			SET DECIMAL MODE
	0559	SEC	38			SET CARRY FLAG
	055A	BIT	24	09		CHECK SIGN FLAG
	055C	BPL	10	18		IF NOT SET BRANCH TO GRAVITY ADD
	055E	LDA	A5	04		LOAD A W/VELOCITY (VLO)
	0560	SBC	E9	05		SUBT. 5 COMPENSATE FOR GRAVITY
	0562	STA	85	0B		STORE RESULT AT SCRATCH 1 (NEW VLO)
	0564	CMP	C5	04		COMPARE TO OLD VLO
	0566	BCC	90	14		IF NEW VLO < OLD VLO BRANCH TO STORE 1
	0568	SEC	38			SET CARRY FLAG
	0569	LDA	A9	00		LOAD A W/00
	056B	SBC	E5	0B		SUBTRACT NEW VLO (CONVERT TO NEG. #)
	056D	STA	85	04		STORE NEW VLO
	056F	LDA	A9	00		LOAD A W/00
	0571	STA	85	09		CLEAR SIGN FLAG BYTE
	0573	JMP	4C	7E	05	JMP TO CHECK 1
GRAVITY ADD	0576	SED	F8			SET DECIMAL MODE
	0577	CLC	18			CLEAR CARRY
	0578	LDA	A5	04		LOAD A W/VLO
	057A	ADC	69	05		ADD 5 TO VLO COMPENSATE FOR GRAVITY
STORE 1	057C	STA	85	04		STORE NEW VLO
CHECK 1	057E	BIT	24	09		CHECK SIGN FLAG BYTE
	0580	BPL	10	09		IF CLEAR BRANCH TO BURNSUBT.
	0582	CLC	18			CLEAR CARRY FLAG
	0583	LDA	A5	04		LOAD A W/VLO
	0585	ADC	65	0A		ADD BURN TO VLO
	0587	STA	85	04		STORE NEW VLO
	0589	CLD	D8			CLEAR DECIMAL MODE
	058A	RTS	60			
BURNSUBT.	058B	SEC	38			SET CARRY FLAG
	058C	LDA	A5	04		LOAD A W/VLO
	058E	SBC	E5	0A		SUBTRACT BURN FROM VLO
	0590	STA	85	0B		STORE NEW VLO AT SCRATCH 1
	0592	LDA	A5	0A		LOAD A W/BURN
	0594	CMP	C9	00		CHECK FOR 00
	0596	BEQ	F0	19		IF BURN=00 BRANCH TO OUT 2
	0598	LDA	A5	0B		LDA W/NEW VLO
	059A	CMP	C5	04		COMPARE TO OLD VLO
	059C	BEQ	F0	02		IF = BRANCH TO VELSUBT.
	059E	BCC	90	0D		IF NEW VLO < OLD VLO TO OUT 1
VELSUB	05A0	SEC	38			SET CARRY FLAG
	05A1	LDA	A9	00		LDA W/00
	05A3	SBC	E5	0B		SUBTRACT NEW VLO LEAVING NEG. #
	05A5	STA	85	04		STORE NEW VLO
	05A7	LDA	A9	FF		LDA W/FF
	05A9	STA	85	09		SET SIGN FLAG BYTE
	05AB	CLD	D8			CLEAR DECIMAL MODE
	05AC	RTS	60			
OUT 1	05AD	STA	85	04		STORE NEW VLO
	05AF	CLD	D8			CLEAR DECIMAL MODE
	05B0	RTS	60			
OUT 2	05B1	LDA	A5	0B		LDA W/NEW VLO
	05B3	STA	85	04		STORE NEW VLO
	05B5	CLD	D8			CLEAR DECIMAL MODE
	05B6	RTS	60			

HEIGHT

	05B7	SED	F8			SET DECIMAL MODE
	05B8	SEC	38			SET CARRY FLAG
	05B9	BIT	24	09		CHECK SIGN FLAG
	05BB	BMI	30	25		IF SET BRANCH TO HTADD
	05BD	LDA	A5	01		LOAD A W/HEIGHT LO BYTE (HLO)
	05BF	SBC	E5	04		SUBTRACT VELOCITY (VLO)
	05C1	STA	85	0B		STORE NEW HLO AT SCRATCH 1
	05C3	LDA	A5	02		LDA W/HEIGHT HI BYTE (HHI)
	05C5	BEQ	F0	0B		IF HHI=00 BRANCH TO CMP1
	05C7	SBC	E9	00		SUBTRACT IMAGINARY VHI FROM HHI
	05C9	STA	85	02		STORE NEW HHI
	05CB	JMP	4C	F1	05	JUMP TO OUT 2
	05CE	LDA	A5	0B		LDA W/NEW HLO
	05D0	BEQ	F0	06		IF NEW HLO=00 BRANCH TO ZERO HHI
CMP1	05D2	LDA	A5	01		LDA W/OLD HLO
	05D4	CMP	C5	0B		COMPARE TO NEW HLO
	05D6	BCS	B0	04		IF OLD HLO >= NEW HLO, GO TO RELOAD
ZERO HHI	05D8	LDA	A9	00		LDA W/00
	05DA	BEQ	F0	02		BRANCH TO OUT 1

```
RELOAD  05DC  LDA  A5  0B      LDA W/NEW HLO
OUT 1   05DE  STA  85  01      STORE NEW HLO
        05E0  CLD  D8          CLEAR DECIMAL MODE
        05E1  RTS  60
HTADD   05E2  CLC  18          CLEAR CARRY FLAG
        05E3  LDA  A5  01      LDA W/HLO
        05E5  ADC  65  04      ADD VLO TO HLO
        05E7  STA  85  01      STORE NEW HLO
        05E9  LDA  A5  02      LDA W/HHI
        05EB  ADC  69  00      ADD IMAGINARY VHI TO HHI
        05ED  STA  85  02      STORE NEW HHI
        05EF  CLD  D8          CLEAR DECIMAL MODE
        05F0  RTS  60
OUT 2   05F1  LDA  A5  0B      LOAD A W/NEW HLO
        05F3  STA  85  01      STORE NEW HLO
        05F5  CLD  D8          CLEAR DECIMAL MODE
        05F6  RTS  60
```

INC TIME

```
        05F7  SED  F8          SET DECIMAL MODE
        05F8  CLC  18          CLEAR CARRY FLAG
        05F9  LDA  A5  00      LDA W/TIME
        05FB  ADC  69  01      ADD 1 TO TIME
        05FD  STA  85  00      STORE NEW TIME
        05FF  CLD  D8          CLEAR DECIMAL MODE
        0600  RTS  60
```

PRINT STRING

```
        0601  LDY  A0  00      ZERO Y REG
LOOP    0603  LDA,Y  B9  □  □  VALUES IN BOXES GET PLUGGED IN
                               BY STRING ROUTINES/GET CHARACTER
        0606  JSR  20  C6  72  PRINT CHARACTER
        0609  CPY  C0  □       VALUE IN BOX GETS PLUGGED IN BY
                               STRING ROUTINES/CHECK FOR END
        060B  BEQ  F0  04      ON END GO TO OUT
        060D  INY  C8          INCREMENT Y TO NEXT CHARACTER
        060E  JMP  4C  03  06  JUMP TO LOOP
OUT     0611  RTS  60
```

PRINT DATA

```
        0612  JSR  20  77  73  PRINT A SPACE
        0615  JSR  20  77  73  SPACE
        0618  LDA  A5  00      LOAD A W/TIME
        061A  JSR  20  B1  72  PRINT TIME BYTE
        061D  LDY  A0  0D      LOAD Y REG. W/13 SPACES
LOOP 1  061F  JSR  20  77  73  SPACE
        0622  DEY  88          Y REG NOW HAS 1 LESS SPACE
        0623  BNE  D0  FA      IF Y IS NOT 00 THEN LOOP 1
        0625  LDA  A5  02      LDA W/HEIGHT HI BYTE
        0627  JSR  20  B1  72  PRINT HEIGHT HI
        062A  LDA  A5  01      LDA W/HEIGHT LO BYTE
        062C  JSR  20  B1  72  TYPE HEIGHT LO
        062F  LDY  A0  0A      LOAD Y REG W/10 SPACES
LOOP 2  0631  JSR  20  77  73  SPACE
        0634  DEY  88          DECREMENT Y REG.
        0635  BNE  D0  FA      LOOP UNTIL DONE
        0637  BIT  24  09      CHECK SIGN FLAG
        0639  BMI  30  07      IF SET BRANCH TO PRINT PLUS
        063B  LDA  A9  2D      LDA W/ASCII (-) MINUS
        063D  JSR  20  C6  72  PRINT IT
        0640  BEQ  F0  05      GO TO LOADV (A CAN'T=00)
PRINT PLUS 0642  LDA  A9  2B   LDA W/ASCII + (PLUS)
        0644  JSR  20  C6  72  PRINT IT
LOAD V  0647  LDA  A5  04      LDA W/VELOCITY BYTE
        0649  JSR  20  B1  72  PRINT VELOCITY
        064C  BIT  24  08      CHECK OUT OF FUEL FLAG
        064E  BPL  10  01      IF CLEAR BRANCH TO PRINT FUEL
        0650  RTS  60          GET OUT HERE IF OUT OF FUEL
        0651  LDY  A0  0C      LOAD Y W/12 SPACES
LOOP 3  0653  JSR  20  77  73  SPACE
        0656  DEY  88          DEC. Y REG.
        0657  BNE  D0  FA      LOOP UNTIL DONE
        0659  LDA  A5  07      LDA W/FUEL HI BYTE
        065B  JSR  20  B1  72  PRINT FUEL HI BYTE
        065E  LDA  A5  06      LDA W/FUEL LO BYTE
        0660  JSR  20  B1  72  PRINT FUEL LO BYTE
        0663  LDY  A0  0B      LOAD Y W/11 SPACES
LOOP 4  0665  JSR  20  77  73  SPACE
        0668  DEY  88          DEC. Y REG.
        0669  BNE  D0  FA      LOOP UNTIL DONE
        066B  LDA  A9  3F      LDA W/ASCII ? (QUESTION MARK)
        066D  JSR  20  C6  72  PRINT IT
        0670  RTS  60
```

The Subroutines that follow are those that contain basically only ASCII Data for the messages that the program prints. However the first part of each of these routines plugs values that pertain to it (i.e., length of string & starting addr. of string) into the "PRINT STRING" SUBROUTINE. The "PRINT STRING" subroutine is then called by each of the character string routines.

EXAMPLE OF CHARACTER STRING ROUTINE

- OUT OF FUEL -

```
LOC
0831  LDA  A9  48      LOAD A W/START LO BYTE OF THE
                       CHAR. STRING
0833  STA  8D  04  06  STORE IT IN "PRINT STRING" 0604
0836  LDA  A9  08      LDA W/START HI BYTE OF THE STRING
0838  STA  8D  05  06  STORE IT IN "PRINT STRING" 0605
083B  LDA  A9  0A      LDA W/# OF CHAR. IN STRING
083D  STA  8D  0A  06  STORE IT IN "PRINT STRING" 060A
0840  JSR  20  01  06  PRINT THE STRING
0843  LDA  A9  FF      SET THE "OUT OF FUEL PRINTED"
0845  STA  85  05      FLAG - THIS ROUTINE ONLY, NOT TYPICAL
0847  RTS  60          RETURN
0848       4F          START OF ASCII DATA "O"
0849       55          "U"
084A       54          "T"
084B       20          SPACE
084C       4F          "O"
084D       46          "F"
084E       20          SPACE
084F       46          "F"
0850       55          "U"
0851       45          "E"
0852       4C          "L"
```

All of the rest of the Character String Subroutines are identical, except of course for the initial values and the string data itself. One exception is the Out of Fuel routine which sets location 0005 to FF so that the program knows it has already printed Out of Fuel and the other exception is the "Burn Out of Range"

Subroutine which is preceded by a Carriage Return and Line Feed.

Listed below are the starting addresses of all of the Character String Routines. A complete listing in hex, and examples of what each string says and looks like, follows.

NAME	LOC.
INSTRUCTIONS	0671
HEADINGS	0772
OUT OF FUEL	0831
ON THE MOON	0853
PERFECT	0871
STRANDED	08A9
CREAMED	0906
TRY AGAIN	0943
BURN OR	0967
CONTROL OUT	099A

CHARACTER STRING ROUTINES

Instructions

```
LUNAR LANDER

MINUS VELOCITY (-) MEANS DOWNWARD MOVEMENT
PLUS  VELOCITY (+) MEANS UPWARD MOVEMENT

MAXIMUM BURN IS 30 UNITS/SEC.  (BURN MAY BE ANY INTEGER FROM 0 TO 30)
A BURN OF 5 UNITS/SEC IS REQUIRED TO CANCEL GRAVITY.

GOOD LUCK!
```

Headlines

```
CONTROL TO LUNAR MODULE:
BEGIN LANDING PROCEDURE

 TIME          HEIGHT      VELOCITY      FUEL UNITS      BURN
(SECS)         (FEET)      (FT/SEC)       REMAINING
```

Print Data

```
 00            0500          -50            0120           ?
```

```
OUT OF FUEL
ON THE MOON
PERFECT LANDING!
```

Perfect

```
CONGRATULATIONS
```

Stranded

```
WELL, YOU GOT DOWN ALIVE, BUT DAMAGE TO YOUR CRAFT
HAS STRANDED YOU HERE!
```

Creamed

```
YOU JUST CREAMED A 29 MEGABUCK LANDER!
TRY AGAIN?  (Y/N)

BURN OUT OF RANGE.      BURN ?
CONTROL OUT
```

HEX LISTING OF CHARACTER STRING ROUTINES

```
LIST 0671 09B7
0671  A9 84 8D 04 06 A9 06 8D 05 06 A9 ED 8D 0A 06 20
0681  01 06 60 4C 55 4E 41 52 20 4C 41 4E 44 45 52 0D
0691  0A 0A 4D 49 4E 55 53 20 56 45 4C 4F 43 49 54 59
06A1  20 28 2D 29 20 4D 45 41 4E 53 20 44 4F 57 4E 57
06B1  41 52 44 20 4D 4F 56 45 4D 45 4E 54 0D 0A 50 4C
06C1  55 53 20 20 56 45 4C 4F 43 49 54 59 20 28 2B 29
06D1  20 4D 45 41 4E 53 20 55 50 57 41 52 44 20 4D 4F
06E1  56 45 4D 45 4E 54 0D 0A 0A 4D 41 58 49 4D 55 4D
06F1  20 48 55 52 4E 20 49 53 20 33 30 20 55 4E 49 54
0701  53 2F 53 45 43 2E 20 20 28 48 55 52 4E 20 4D 41
0711  59 20 42 45 20 41 4E 59 20 49 4E 54 45 47 45 52
0721  20 46 52 4F 4D 20 30 20 54 4F 20 33 30 29 0D 0A
0731  41 20 42 55 52 4E 20 4F 46 20 35 20 55 4E 49 54
0741  53 2F 53 45 43 20 49 53 20 52 45 51 55 49 52 45
0751  44 20 54 4F 20 43 41 4E 43 45 4C 20 47 52 41 56
0761  49 54 59 2E 0D 0A 0A 47 4F 4F 44 20 4C 55 43 4B
0771  21 A9 85 8D 04 06 A9 07 8D 05 06 A9 AB 8D 0A 06
0781  20 01 06 60 43 4F 4E 54 52 4F 4C 20 54 4F 20 4C
0791  55 4E 41 52 20 4D 4F 44 55 4C 45 3A 0D 0A 42 45
07A1  47 49 4E 20 4C 41 4E 44 49 4E 47 20 50 52 4F 43
07B1  45 44 55 52 45 0D 0A 0A 0A 20 54 49 4D 45 20 20
07C1  20 20 20 20 20 20 20 20 20 48 45 49 47 48 54 20
07D1  20 20 20 20 20 56 45 4C 4F 43 49 54 59 20 20 20
07E1  20 20 20 20 46 55 45 4C 20 55 4E 49 54 53 20 20
07F1  20 20 20 20 20 42 55 52 4E 0D 0A 28 53 45 43 53
0801  29 20 20 20 20 20 20 20 20 20 20 20 28 46 45 45 54
0811  29 20 20 20 20 20 20 20 28 46 54 2F 53 45 43 29 20
0821  20 20 20 20 20 20 20 20 52 45 4D 41 49 4E 49 4E 47
0831  A9 48 8D 04 06 A9 08 8D 05 06 A9 0A 8D 0A 06 20
0841  01 06 A9 FF 85 05 60 4F 55 54 20 4F 46 20 46 55
0851  45 4C A9 66 8D 04 06 A9 08 8D 05 06 A9 0A 8D 0A
0861  06 20 01 06 60 4F 4E 20 54 48 45 20 4D 4F 4F 4E
0871  A9 84 8D 04 06 A9 08 8D 05 06 A9 24 8D 0A 06 20
0881  01 06 60 50 45 52 46 45 43 54 20 4C 41 4E 44 49
0891  4E 47 21 07 07 07 07 0D 0A 43 4F 4E 47 52 41 54
08A1  55 4C 41 54 49 4F 4E 53 A9 BC 8D 04 06 A9 08 8D
08B1  05 06 A9 49 8D 0A 06 20 01 06 60 57 45 4C 4C 2C
08C1  20 59 4F 55 20 47 4F 54 20 44 4F 57 4E 20 41 4C
08D1  49 56 45 2C 20 42 55 54 20 44 41 4D 41 47 45 20
08E1  54 4F 20 59 4F 55 52 20 43 52 41 46 54 0D 0A 48
08F1  41 53 20 53 54 52 41 4E 44 45 44 20 59 4F 55 20
0901  48 45 52 45 21 A9 19 8D 04 06 A9 09 8D 05 06 A9
0911  28 8D 0A 06 20 01 06 60 59 4F 55 20 4A 55 53 54
0921  20 43 52 45 41 4D 45 44 20 41 20 32 39 20 4D 45
0931  47 41 42 55 43 4B 20 4C 41 4E 44 45 52 21 07 07
0941  07 07 A9 56 8D 04 06 A9 09 8D 05 06 A9 10 8D 0A
0951  06 20 01 06 60 54 52 59 20 41 47 41 49 4E 3F 20
0961  20 28 59 2F 4E 29 20 8A 72 A9 7D 8D 04 06 A9 09
0971  8D 05 06 A9 1C 8D 0A 06 20 01 06 60 42 55 52 4E
0981  20 4F 55 54 20 4F 46 20 52 41 4E 47 45 2E 20 20
0991  20 20 20 42 55 52 4E 20 3F A9 AD 8D 04 06 A9 09
09A1  8D 05 06 A9 0A 8D 0A 06 20 01 06 60 43 4F 4E 54
09B1  52 4F 4C 20 4F 55 54
```

ALARMS & SECURITY STUFF

If you are into alarms, fire sensors, invasion sensors, window foil, under-mat weight sensors, or just general paranoia with an electronic bent, you should request the general catalog (97 pages) from Mountain West Alarm Supply Co., 4215 N. 16th St.,Phoenix, AZ 85016 (602) 263-8831.

LINEAR POSITION & VELOCITY TRANSDUCERS

If you are interested in sensing positions or velocities of equipment that has a linear movement of less than 8 feet, then you should request information on the linear stroke transducers manufactured by G.L.Collins Corp., 5875 Obispo Ave. Long Beach, CA 90805, (213) 636-8141 or 630-3121.

"Applications Software"--
Games in Pittman's 6800 Tiny BASIC

by Carl Kelb

These games—Stars, Acey-Deucy, Trap, and Slot—were adapted from older versions to Tom Pittman's Tiny BASIC for the 6800. Note: A bell (control-G) was imbedded in the print statements for each *.

```
1 PR
1000 PRINT "WHICH GAME DO YOU WANT ?"
1001 PRINT "TYPE 1 FOR STARS"
1002 PRINT "TYPE 2 FOR ACEY-DUCEY"
1003 PRINT "TYPE 3 FOR TRAP"
1004 PRINT "TYPE 4 FOR SLOTS"
1010 INPUT Z
1020 IF Z<1 GOTO 1000
1030 IF Z>4 GOTO 1000
1040 IF Z=1 GOTO 1100
1050 IF Z=2 GOTO 2000
1060 IF Z=3 GOTO 4000
1070 IF Z=4 GOTO 5000
1100 PR
1120 PRINT "WELCOM TO MY GALAXY. I'M IN CHARGE OF THE STARS HERE."
1130 PRINT "PLAY MY GAME *STARS* AND GET SOME STARS FOR YOURSELF!"
1140 PR
1150 PRINT "I WILL THINK OF A WHOLE NUMBER FROM 1 TO 100."
1160 PRINT "TRY TO GUESS MY NUMBER. AFTER YOU GUESS, I"
1170 PRINT "WILL TYPE ONE OR MORE STARS (*). THE CLOSER"
1180 PRINT "YOU ARE TO MY NUMBER, THE MORE STARS I WILL TYPE."
1190 PRINT "ONE STAR (*) MEANS YOU ARE FAR AWAY FROM MY"
1200 PRINT "NUMBER. SEVEN STARS (*******) MEANS YOU ARE VERY,"
1210 PRINT "VERY, VERY CLOSE TO MY NUMBER!!!"
1220 LET X=RND (100)+1
1230 PR
1240 PRINT "OK , STARSEEKER, I AM THINKING OF A NUMBER, START GUESSING."
1250 LET N=1
1260 PR
1270 PRINT "WHAT IS YOUR GUESS ";
1280 INPUT G
1290 IF G=X THEN GOTO 1550
1370 LET D=G-X
1375 IF D < 0 GOTO 1377
1376 GOTO 1380
1377 LET D = D*(-1)
1380 IF D >= 64 GOTO 1500
1390 IF D >= 32 GOTO 1490
1400 IF D >= 16 GOTO 1480
1410 IF D >= 8 GOTO 1470
1420 IF D >= 4 GOTO 1460
1430 IF D >= 2 GOTO 1450
1440 PRINT " *";
1450 PRINT " *";
1460 PRINT " *";
1470 PRINT " *";
1480 PRINT " *";
1490 PRINT " *";
1500 PRINT " *";
1510 PR
1520 LET N=N+1
1530 GOTO 1260
1550 REM
1555 PRINT " ** ** ** ** ** ** ** ** ** ";
1560 PRINT "** ** ** ** ** ** ** ** ** ";
1580 PRINT "!!!"
1590 PRINT "THAT'S IT!!! YOU GUESSED MY COSMIC NUMBER IN ";N;" GUESSES"
1600 PR
1610 PRINT "DO YOU WANT TO PLAY AGAIN ?"
1620 PRINT "TYPE 1 TO PLAY AGAIN OR 2 TO PLAY ANOTHER GAME"
1630 INPUT I
1640 IF I=0 GOTO 1610
1650 IF I=1 GOTO 1220
1670 GOTO 1000
1680 END

2000 REM: ACEY-DUCEY
2100 PR
2101 PRINT "ACEY-DUCEY IS PLAYED IN THE FOLLOWING MANNER:"
2102 PRINT "THE DEALER (COMPUTER) DEALS TWO CARDS FACE UP."
2103 PRINT "YOU HAVE THE OPTION TO BET OR NOT TO BET DEPENDING"
2104 PRINT "ON WHETHER OR NOT YOU FEEL THE NEXT CARD WILL HAVE"
2105 PRINT "A VALUE BETWEEN THE FIRST TWO."
2106 PRINT "IF YOU DO NOT WANT TO BET, INPUT A 0."
2110 PR
2160 N=100
2170 Q=100
2190 PRINT "YOU NOW HAVE ";Q;" DOLLARS."
2195 PR
2200 GOTO 2260
2210 LET Q=Q+M
2220 GOTO 2190
2240 LET Q=Q-M
2250 GOTO 2190
2260 PRINT "HERE ARE YOUR NEXT TWO CARDS...."
2270 LET A=(RND(14))+2
2280 IF A<2 GOTO 2270
2290 IF A>14 GOTO 2270
2300 LET B=(RND(14))+2
2310 IF B<2 GOTO 2300
2320 IF B>14 GOTO 2300
2330 IF A>=B GOTO 2270
2350 IF A<11 GOTO 2400
2360 IF A=11 GOTO 2420
2370 IF A=12 GOTO 2440
2380 IF A=13 GOTO 2460
2390 IF A=14 GOTO 2480
2400 PRINT A
2410 GOTO 2500
2420 PRINT "JACK"
2430 GOTO 2500
2440 PRINT "QUEEN"
2450 GOTO 2500
2460 PRINT "KING"
2470 GOTO 2500
2480 PRINT "ACE"
2500 IF B<11 GOTO 2550
2510 IF B=11 GOTO 2570
2520 IF B=12 GOTO 2590
2530 IF B=13 GOTO 2610
2540 IF B=14 GOTO 2630
2550 PRINT B
2560 GOTO 2650
2570 PRINT "JACK"
2580 GOTO 2650
2590 PRINT "QUEEN"
2600 GOTO 2650
2610 PRINT "KING"
2620 GOTO 2650
2630 PRINT "ACE"
2650 PR
2660 PRINT "WHAT IS YOUR BET ";
2665 INPUT M
2670 IF M<>0 GOTO 2680
2675 PRINT "CHICKEN!!"
2676 PR
2677 GOTO 2260
2680 IF M<=Q GOTO 2730
2690 PRINT "SORRY, MY FRIEND, BUT YOU BET TOO MUCH"
2700 PRINT "YOU HAVE ONLY ";Q;" DOLLARS TO BET"
2710 GOTO 2650
2730 LET C=(RND(14))+2
2740 IF C<2 GOTO 2730
2750 IF C>14 GOTO 2730
2760 IF C<11 GOTO 2810
2770 IF C=11 GOTO 2830
2780 IF C=12 GOTO 2850
2790 IF C=13 GOTO 2870
2800 IF C=14 GOTO 2890
2810 PRINT C
2820 GOTO 2910
2830 PRINT "JACK"
2840 GOTO 2910
2850 PRINT "QUEEN"
2860 GOTO 2910
2870 PRINT "KING"
2880 GOTO 2910
2890 PRINT "ACE"
2910 IF C>A GOTO 2930
2920 GOTO 2970
2930 IF C>=B GOTO 2970
2950 PRINT "YOU WIN!!!"
2960 GOTO 2210
2970 PRINT "SORRY, YOU LOSE."
2980 IF M<Q GOTO 2240
3000 PR
```

```
3010 PRINT "SORRY, FRIEND, BUT YOU BLEW YOUR WAD."
3020 PRINT
3030 PRINT "DO YOU WANT TO TRY AGAIN ?"
3040 PRINT "TYPE 1 TO PLAY AGAIN AND 2 TO PLAY ANOTHER GAME"
3050 INPUT I
3060 IF I=0 GOTO 3040
3070 IF I>2 GOTO 3040
3080 IF I=1 GOTO 2110
3090 IF I=2 GOTO 1000
3100 END

4000 REM: TRAP
4160 PRINT "I WILL THINK OF A NUMBER FROM 1 TO 100."
4170 PRINT "TRY TO GUESS MY NUMBER. ENTER TWO NUMBERS, TRYING"
4180 PRINT "TO TRAP MY NUMBER BY YOUR TWO TRAP NUMBERS. I'LL"
4190 PRINT "TELL YOU IF YOU HAVE TRAPPED MY NUMBER OR IF MY"
4200 PRINT "NUMBER IS SMALLER THAN YOUR TWO TRAP NUMBERS OR"
4210 PRINT "IF MY NUMBER IS LARGER THAN YOUR TWO TRAP NUMBERS."
4220 PRINT "IF I TELL YOU THAT YOU HAVE TRAPPED MY NUMBER, I"
4230 PRINT "MEAN THAT MY NUMBER IS *BETWEEN* YOUR TRAP NUMBERS"
4240 PRINT "OR - PERHAPS MY NUMBER IS THE SAME AS ONE OF YOUR"
4250 PRINT "TRAP NUMBERS."
4260 PR
4270 PRINT "!!!IMPORTANT!!! IF YOU THINK YOU KNOW MY NUMBER, THEN"
4280 PRINT "ENTER YOUR GUESS FOR *BOTH* TRAP NUMBERS."
4290 PR
4300 LET X=(RND(100))+1
4310 PRINT "I'M THINKING...THINKING...AH! I HAVE A NUMBER!"
4320 LET K=1
4330 PR
4340 PRINT "FIRST TRAP NUMBER ";
4350 INPUT A
4360 PRINT "SECOND TRAP NUMBER ";
4370 INPUT B
4371 LET Y=X-A
4372 IF Y<0 THEN Y=-1
4373 IF Y>0 THEN Y=1
4374 LET Z=X-B
4375 IF Z<0 THEN Z=-1
4376 IF Z>0 THEN Z=1
4380 LET T=Y+Z
4390 IF T=-2 GOTO 4430
4391 IF T=-1 GOTO 4410
4392 IF T=0 GOTO 4400
4393 IF T=1 GOTO 4410
4394 IF T=2 GOTO 4450
4400 IF A=B GOTO 4480
4410 PRINT "MY NUMBER IS TRAPPED BY YOUR NUMBERS."
4420 GOTO 4460
4430 PRINT "MY NUMBER IS SMALLER THAN YOUR TRAP NUMBERS."
4440 GOTO 4460
4450 PRINT "MY NUMBER IS LARGER THAN YOUR TRAP NUMBERS."
4460 LET K=K+1
4470 GOTO 4330
4480 PRINT "YOU GOT IT IN ";K;" GUESSES  !  !  !  !  !  !  !"
4485 PR
4490 PRINT "DO YOU WANT TO PLAY AGAIN ?"
4500 PRINT "TYPE 1 IF YOU WANT TO PLAY AGAIN OR 2 IF YOU WANT ANOTHER GAME"
4510 INPUT I
4520 IF I=0 GOTO 4500
4530 IF I>2 GOTO 4500
4540 IF I=1 GOTO 4290
4550 IF I=2 GOTO 1000
4560 END

5000 REM: SLOTS
5100 PR
5120 PRINT "THIS IS A SIMULATION OF A SLOT MACHINE USING A COMPUTER"
5130 PRINT "EACH TIME YOU 'PULL' I WILL ASK YOU IF YOU WISH TO PLAY AGAIN."
5140 PRINT "JUST ANSWER WITH A '1' FOR YES AND A '0' FOR NO."
5150 PRINT "PLEASE PLACE 4 QUARTERS ON MY CPU FOR EACH PLAY."
5160 LET B=0
5170 PR
5180 LET D=(RND(8))+1
5181 LET E=(RND(7))+1
5182 LET F=(RND(7))+1
5201 IF D=1 GOTO 5211
5202 IF D=2 GOTO 5213
5203 IF D=3 GOTO 5215
5204 IF D=4 GOTO 5217
5205 IF D=5 GOTO 5219
5206 IF D=6 GOTO 5221
5207 IF D=7 GOTO 5223
5208 IF D=8 GOTO 5226
5211 PRINT "        BELL ";
5212 GOTO 5231
5213 PRINT "         BAR  ";
5214 GOTO 5231
5215 PRINT "      CHERRY";
5216 GOTO 5231
5217 PRINT "       APPLE ";
5218 GOTO 5231
5219 PRINT "       LEMON ";
5220 GOTO 5231
5221 PRINT "         $    ";
5222 GOTO 5231
5223 PRINT "      CHERRY";
5224 LET D=3
5225 GOTO 5231
5226 PRINT "      CHERRY";
5227 LET D=3
5231 IF E=1 GOTO 5241
5232 IF E=2 GOTO 5243
5233 IF E=3 GOTO 5245
5234 IF E=4 GOTO 5247
5235 IF E=5 GOTO 5249
5236 IF E=6 GOTO 5251
5237 IF E=7 GOTO 5253
5241 PRINT "  BELL ";
5242 GOTO 5261
5243 PRINT "  BAR  ";
5244 GOTO 5261
5245 PRINT " CHERRY";
5246 GOTO 5261
5247 PRINT " APPLE ";
5248 GOTO 5261
5249 PRINT " LEMON ";
5250 GOTO 5261
5251 PRINT "   $    ";
5252 GOTO 5261
5253 PRINT " CHERRY";
5254 LET E=3
5261 IF F=1 GOTO 5271
5262 IF F=2 GOTO 5273
5263 IF F=3 GOTO 5275
5264 IF F=4 GOTO 5277
5265 IF F=5 GOTO 5279
5266 IF F=6 GOTO 5281
5267 IF F=7 GOTO 5283
5271 PRINT "  BELL ";
5272 GOTO 5410
5273 PRINT "  BAR  ";
5274 GOTO 5410
5275 PRINT " CHERRY";
5276 GOTO 5410
5277 PRINT " APPLE ";
5278 GOTO 5410
5279 PRINT " LEMON ";
5280 GOTO 5410
5281 PRINT "   $    ";
5282 GOTO 5410
5283 PRINT " CHERRY";
5284 LET F=3
5410 IF D<>E GOTO 5440
5420 IF E=F GOTO 5530
5430 IF D=E GOTO 5460
5440 IF D<>F GOTO 5490
5450 GOTO 5510
5460 IF D=1 GOTO 5510
5461 IF D=3 GOTO 5510
5462 IF D=5 GOTO 5510
5463 IF D=7 GOTO 5510
5470 LET B=B+5
5475 PRINT " KENO..YOU WIN $5..TOTAL=$";B;
5480 GOTO 5550
5490 LET B=B-1
5495 PRINT " YOU HAVE LOST $1 -- TOTAL=$";B;
5500 GOTO 5550
5510 LET B=B+1
5515 PRINT " YOU HAVE WON $1 --- TOTAL=$";B;
5520 GOTO 5550
5530 IF D=2 GOTO 5540
5531 IF D=6 GOTO 5545
5535 LET B=B+20
5536 PRINT " JACKPOT...$20...TOTAL=$";B;
5537 GOTO 5550
5540 LET B=B+50
5541 PRINT " JACKPOT...$50...TOTAL=$";B;
5542 GOTO 5550
5545 LET B=B+100
5546 PRINT " JACKPOT..$100...TOTAL=$";B;
5550 PRINT " AGAIN ";
5560 INPUT I
5561 IF I>1 GOTO 5550
5562 IF I=1 GOTO 5170
5565 PR
5570 PRINT "IT'S BEEN NICE OPERATING FOR YOU -
5571 PR                              COME BACK SOON!
5575 GOTO 1000
5580 END
```

Low-Cost 6800 Systems Software & Games

by Technical Systems Consultants' staff
Box 2574, W. Lafayette IN 47906

TSC is presently involved in the creation of products which are currently in high demand among computer hobbyists and other micro computer users. Up to this time there has been little or no software available for Motorola 6800 based systems other than Monitor programs in ROM. We have developed many programs both useful and fun which allow the system builder to utilize his creation to its fullest extent. The software listings which we offer implement a variety of user and system type functions. The programs have been written in 6800 assembly language and assembled to run on Motorola and AMI 6800 based systems and utilize I/O routines contained in the MIKBUG* monitor ROM. All references to these external routines are clearly marked, however, facilitating conversion to other I/O routines. The software listings include a fully commented source listing, a hexadecimal machine code dump, sample output, and complete instructions for use. Because software "bugs" are bound to occur regardless of the degree of testing we offer a limited warranty. This 90-day warranty is limited to replacement of the original software listing or providing a patch at the discretion of TSC.

For those requiring the service, all of our routines can be assembled at a custom address or with user supplied I/O routines for an extra charge.

New products are constantly being developed by TSC. These include a Micro BASIC interpreter, a scientific floating point package, a business and accounting system, graphics games, and an 8080 emulator, among others. We also plan to offer some of our programs on "Kansas City" standard cassettes. Hardware items being developed and tested for the 6800 based system include a cassette interface system, A/D and D/A boards, a high speed arithmetic processor and other general purpose items. All of these products will be available when announced in our advertising.

We can only offer what the hobbyists want, so let us know what your needs are in both hardware and software.

*MIKBUG is a registered trademark of Motorola, Inc.

1. HANGMAN: The old word guessing game. Easily modified with your own word list. Requires 640 Bytes. $3.25
2. ACEY-DUCEY: A card game played against the computer. Bet and try to break the bank! Requires 1K Bytes. $3.25
3. CRAPS: A real casino craps game. Match your luck against the computer and try to win money. Requires 1K Bytes. $3.25
4. FLOATING POINT PACKAGE: Full floating point capability. 9 digits of accuracy with exponent range, -99 to +99. Four routines for add, subtract, multiply, and divide are all included. Requires 512 Bytes. $5.00
5. SPACE VOYAGE: Similar to the famous STAR TREK with only a few limitations. Every game is a different adventure! Requires 4K Bytes. $10.00
6. KLINGON CAPTURE: A smaller space simulation game, but, has many of the same surprises. 2K Bytes $4.75
7. STOCKMARKET: Similar to the popular board game of the same name. Simulates real WALL STREET action. Requires 1.5 Bytes. $3.50
8. LINE EDITOR: Allows you to create a file in memory and then completely edit it. Commands are: NEW ADD, INSERT, DELETE, SEARCH, LIST, MOVE, PRINT. Will run in only 512 Bytes! $4.00
9. RANDOM NUMBER GENERATOR: Here is a routine which is an absolute must for writing your own game programs. Requires 60 Bytes. $1.50
10. MASTERMIND: Test your logical abilities; An intricate guessing game requiring both skill and logic. Requires 512 Bytes. $3.00
11. CARD SHUFFLE AND DEAL: Two very useful routines. Includes a driver routine to print out 4 hands of 13 cards each. Requires 512 Bytes. $2.75
12. NUMBER GUESS I: Try to guess the number the computer is thinking of! Requires 256 Bytes. $1.50
13. NUMBER GUESS II: A more advanced number guessing game. Requires 512 Bytes. $2.00
14. HURKLE: Try to find the hiding Hurkle relying upon clues given by the computer. Requires 640 bytes $2.00
15. ROVER: Find and catch the Rover with the aid of hints supplied by the computer. Requires 1K Bytes. $2.50
16. SWITCH: Correctly arrange a random string of digits in the fewest possible moves. Requires 512 Bytes. $2.00
17. CHOMP: A 2 player game which resembles a two dimensional "NIM" game. Requires 512 Bytes. $2.00
18. SUBROUTINE PACKAGE: A special package of very useful subroutines selected by the staff of TSC. This package could save you many hours when writing your own programs. $3.00

**SPECIAL PACKAGE DEALS....Supplied in a 3-ring binder

I. Contains programs 1,2,3,6,9,10 $13.50

II. Contains programs 1,2,3,9,10,11,12,13,14,15, 16,17 $18.95

III. Contains programs in I and II plus 4 and 8 $29.50

**********SPECIAL ANNOUNCEMENT**********

At last there exists a valuable service to the computer hobbyist– THE PROGRAM OF THE MONTH CLUB-- For only $2.00 you will get a one year membership. You will receive a monthly bulletin describing the main selection as well as many alternates. Members will receive a 15% discount on the featured program. THERE IS NO OBLIGATION TO BUY ANYTHING! If you join now, you will receive free our Random Number Generator.

[$1.00 handling charge for orders under $10. Add 5% for First Class Mail. Indiana Residents add 4% sales tax.]

PITTSBURGH CLUB HAS 50 MEMBERS & GROWING FAST

There is a club in Pittsburgh. It has about 50 members and is still growing; all sorts of machines and hardware; anybody in Western Pennsylvania, Eastern Ohio or West Virginia who is interested can write:

Pittsburgh Area Computer Club
400 Smithfield St.
Pittsburgh, PA 15205

Or Call:

Eric Liber (Pres.) (412) 276-6546 Nite
Fred Kitman (Treas.) (412) 391-3800 Day

Shooting Stars for Uiterwyk's 6800 Micro-BASIC

Dear Jim, (received July 29, 1976)

My copy of *DDJ* is arriving right on schedule—and full of good stuff too!. Keep up the good work. To help you and/or PCC with that work I am enclosing a copy of my latest game program. This is a version of Shooting Stars (Alias Teaser). The program is written in Micro-BASIC as supplied in the June SWTPC 6800 newsletter. (This interpreter was written by Robert Uiterwyk and Bill Turner, and has provided me with many hours of enjoyment.) Since the Micro-BASIC allows only arithmetic comparisons, I have had to use a rather unusual method for determining which positions are stars and which are black holes. The interpreter and game will fit in 4K of memory if you remove all the REM statements (doesn't REM mean REMOVE anyway?). The game features a randomization of the universe at the start of the game to keep you from getting too complacent or bored if you have a good memory.

By the way, Corvallis is in the Willimette Valley about 80 miles south of Portland. It is the home of Oregon State University—which has a good computer center (with Super Star Trek!). Unfortunately, our town lacks a computer store at present—know anybody who wants to invest in a small business in a community with an active and growing-technically-oriented population? I'd work for a ridiculous wage.

I am currently work on a text editor for my 6800 system which is very similar to the "classy" 8080 editor in your June/July issue. I still have a few commands let to program, but it is now running with the INSERT, DELETE, APPEND, FIND, LIST, CLEAR, TOP, and NEXT commands active. I still have to get the CHANGE command on line. This is not too much of a problem at present since I use a TVT with a 32-character line—it's easy enough to change a whole line, so I left this one until last. Since I am using the cross-assembler on the computer center's CYBER system to write this program I may try to recoup some of my costs by selling the editor in an article where I can get some money for it. Rest assured that unless I change professions (I'm now a grad student in Oceanography), the source code for this and any other programs I find worth spreading around will be distributed at cost (mailing cost—not my development cost—this is a hobby, right?).

Sincerely,

Mark J. Borgerson
325 NW 9th, No. 3
Corvallis OR 97330

If you find that the other hobby mags are unwilling to publish the complete, annotated source code for your Editor, pass it along to us. We do *publish useful-though-long program listings —Jim W.*

PROGRAMMING NOTES FOR "SHOOTING STARS"

The game of Shooting Stars was originally published in the September, 1974, issue of *People's Computer Company* (under the name of Teaser). It is well described in the May, 1976, *Byte*, so I'm not going to say much about the game itself here. The primary problem in writing the game in Micro-BASIC is finding a way to group the stars and black holes in the universe into the appropriate galaxies. I have solved this problem using an array of nine numbers which are each the product of several prime factors. These are the values of the "F" array in the program. Each point in the universe is assigned a prime number value (the "S" values in the program). A positive value indicates a "Star", and a negative value is a "black hole." The program can then check for stars and black holes with a simple arithmetic comparison (i.e., IF X> 0). Reversing stars and black holes is as simple as changing their signs. The program determines which stars are in a given galaxy by finding all the prime number factors of the "F" value for that position. The appropriate galaxy is then reversed. The integer arithmetic of the Micro-BASIC interpreter allows checking for the factors of a number in the following manner:

Suppose we have A = 3
B = 12
then (B/A)*A = 12/3 * 3 or 12
But if B = 5 then (B/A)* A = 10
because 12/5 = 2.4 and this is truncated to 2.

This is the type of test which is performed in line 780 of the program. If the S value divides evenly into the F value the appropriate point in the galaxy is inverted. A FOR—NEXT loop checks and inverts the appropriate points in the complete universe. The same technique of checking for even division is used to print the carriage returns after each third point to produce a square matrix display (line 890).

The "S" values must be prime numbers to ensure that no extra factors creep into the "F" values which are the product of the appropriate "S" values for each galaxy.

This program should run in any other type of Tiny BASIC which will handle-one-dimensioned arrays. You will have to change the RND statement (380, 390) to fit your random number generator. The RND statement in Micro-BASIC produces a random integer between 1 and 32,762. Line 390 causes the starting universe to be biased toward black holes; roughly two black holes for each star. The program can easily be converted to a scale basic interpreter by using the INT function when checking for factors. For example, line 780 would become: 780 IF(INT(F(X)/S(K))*S(K) = F(X) LET S(K) = – S(K).

If you're not using a TVT with cursor control, change line 840 to a simple "print" or whatever form of page control you need.

Good Luck!

```
00100 REM  SHOOTING STARS IN 6800 MICRO-BASIC.
00110 REM  MARK BORGERSON 7-23-76
00120 REM  DESIGNED FOR SWTPC 6800 WITH TVT-II(CT-1024).
00130 DIM F(9),S(9)
00140 REM  THE FOLLOWING STEPS ENTER THE INITIAL VALUES OF THE
00150 REM F AND S ARRAYS.(MICRO-BASIC HAS NO DATA AND READ STATEMENTS)
00160 S (1)=-23
00170 S(2)=-3
00180 S(3)=-19
00190 S(4)=-11
00200 S(5)=2
00210 S(6)=-5
00220 S(7)=-13
00230 S(8)=-7
00240 S(9)=-17
00250 F(1)=1518
00260 F(2)=1311
00270 F(3)=570
00280 F(4)=3289
00290 F(5)=2310
00300 F(6)=1615
00310 F(7)=2002
00320 F(8)=1547
00330 F(9)=1190
00340 REM INITIALIZE SHOT COUNTER
00350 C=0
00360 REM RANDOMIZE STAR AND BLACK HOLE PATTERN
00370 FOR I=1 TO 9
00380 X=RND
00390 IF X>20000  S(I)=-S(I)
00400 NEXT I
00410  REM PRINT INITIAL PATTERN
00420 GOSUB 840
00430 REM GET FIRST SHOT
00440 PRINT "YOUR SHOT";
00450 INPUT X
00460 REM INCREMENT SHOT COUNTER
00470 C=C+1
00480 REM CHECK FOR VALID SHOT
00490 IF S(X)>0 GO TO 530
00500 PRINT"YOU CAN ONLY SHOOT STARS"
00510 GO TO 440
00520 REM INITIALIZE SCORING COUNTER
00530 B=0
00540 REM INVERT(CHANGE SIGN) OF APPROPRIATE GALAXY.
00550 GOSUB 770
00560 REM CHECK SCORE BY ADDING STAR VALUES
00570 FOR L=1 TO 9
00580 B=B+S(L)
00590 NEXT L
00600 REM PRINT OUT MODIFIED UNIVERSE
00610 GOSUB 840
00620 REM IF B=-100 ALL POINTS ARE BLACK HOLES
```

continued

```
00630 IF B=-100 GO TO 690
00640 REM IF B IS NEITHER -100 NOR 96 GAME CONTINUES
00650 IF B<>96 GO TO 440
00660 PRINT"YOU WIN"
00670 PRINT"YOU FIRED "J;C;" SHOTS."
00680 GO TO 700
00690 PRINT"YOU LOST"
00700 PRINT "TO PLAY AGAIN TYPE 'RUN', CR."
00710 GO TO 950
00720 REM THE FOLLOWING SUBROUTINE CHECKS TO SEE IF THE F VALUE
00730 REM FOR THE SHOT CAN BE EVENLY DIVIDED  BY THE SVALUE
00740 REM FOR EACH POSITION.  IF  S DIVIDES INTO F WITHOUT
00750 REM  A REMAINDER, THE STAR OR BLACK HOLE IS INVERTED
00760 REM (ITS SIGN IS CHANGED)
00770 FOR K=1 TO 9
00780 IF(F(X)/S(K))*S(K)=F(X)  S(K)=-S(K)
00790 NEXT K
00800 RETURN
00810 REM THIS SUBROUTINE PRINTS OUT THE GALAXY
00820 REM STATEMENT 9500 DOES A HOME UP AND ERASE OF SCREEN ON TVT
00830 REM EQUIPPED WITH COMPUTER CURSOR CONTROL.
00840 PRINT"<CNTRL-P><CNTRL-V>"
00850 FOR J=1 TO 9
00860 IF S(J)<0  PRINT"  .";
00870 IF S(J)>0  PRINT"  *";
00880 REM IF J ISN'T DIVISIBLE BY 3, SKIP CARRIAGE RETURNS.
00890 IF J/3*3<>J  GO TO 920
900 PRINT
910 PRINT
920 NEXT J
930 RETURN
940 REM END OF SUBROUTINE AND GAME
950 END
READY.
```

A PL/6800 CROSS COMPILER

Intermetrics, 701 Concord Ave., Cambridge MA 02138, is reported to have a PL/M-type cross compiler for the 6800.

SINGLE-CHIP CONTROLLER FOR 6800

Motorola has scheduled a single-chip controller for the 6800 m-p for delivery around the second quarter of 1977. Designated the 6802, it will have 8K bits of ROM, 256 bits of RAM, and will include I/O capabilities.

(ANOTHER) L.A. STORE: THE DATA CENTER

Gentlepersons: May 13, 1976

We are currently opening "The Data Center," another micro-computer store, in the Los Angeles area.

Sincerely,

Mel Norell
General Manager
Programma

3400 Wilshire Blvd.
Los Angeles CA
90010

SPACE GAMES MARATHON (IN MENLO PARK, CA)

The Community Computer Center, 1919 Menalto, Menlo Park, CA 94025, is having a computer marathon on August 20th and 21st. During the 24-hour period beginning at 9 PM, Friday evening, the Center will offer reduced rates for computer time ($1.20/hour), and will charge a one-time admission fee of 50 cents. There will be 25 door prizes each being an hour of free computer time at the Center.

There will be for-fun (non-prize) competition for such things as most ships destroyed, longest time, etc.

6800 TBX FOR FREE . . . BUT IT'S SOFTCOPY

Dear Folks, 12 June 1976

The receipt yesterday of *DDJ* [Vol. 1, No. 3], and the continued inability or unwillingness of Sphere to deliver a BASIC interpreter has motivated me to write you with some good news and some bad news.

First, the good news:

I am willing to place in the public domain a running version of TBX which I developed for my 6800 (Sphere) system. It contains all of the features of TBX as published in *DDJ* [Vol. 1, No. 2], including the DATA statement; plus REMarks, LOAD & SAVE (tape cassette commands), EDIT & CLEAR (CRT commands), and PRINT formatting with a ':' for concatenation with no space.

Now the bad news:

My listing is in the same sort of shape as the TBX one you published—probably for the same reasons—i.e., hand-written on about 55 IBM assembler coding sheets, hand-assembled into hex.

Additionally, the code jumps about to patch locations, is CRT-oriented, and uses routines in the Sphere-supplied PROM monitor whenever possible.

If you would be interested in publishing this, (and can scrape up the necessary volunteer labor to type it into your format) I will send you a copy of the listing plus a page or so of comments about the significant differences from TBX as you published it. Please let me know.

Sincerely,

Chuck Crayne

734 S. Ardmore Ave
Los Angeles CA 90005

Well, all you 6800 fans who have been yelling for software . . . How 'bout pitching in and gening some publishable=sharable hardcopy and documentation? —JCW, Jr.

AMI 6800 EVALUATION KIT INCLUDES TINY BASIC

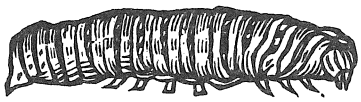
American Microsystems, Inc., 3800 Homestead Rd., Santa Clara, CA 95051, is marketing a 6800 Microprocessor Evaluation Board. It includes the 6800 version of Tiny BASIC, a ROM-resident prototyping operating system, and a built-in EPROM programmer. In kit form with a PC (evidently without memory?), it is $295. With 512 bytes of EPROM, the kit is $595. A fully assembled kit with 2K bytes of EPROM is $950.

SOMEONE COPIES 6800 TINY BASIC...AND PAYS, ANYWAY

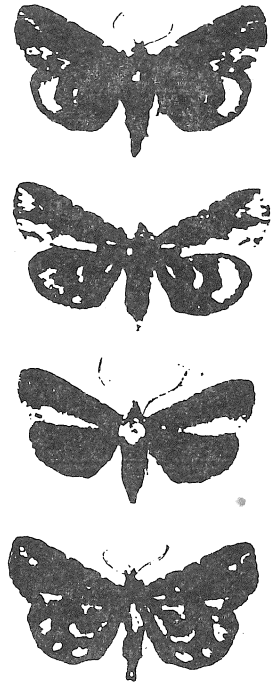
Tom Pittman, [Box 23189, San Jose, CA 95153] who is offering an excellent version of Tiny BASIC for the 6800 for $5, told us of receiving a five dollar payment from someone who said he had already copied the documentation and paper tape (from a friend) and was sending the payment "retroactively".

FAR OUT! (As we used to say in the '60's). We applaud and encourage such actions by hobbyists. We feel Tom is charging a fair price for a good product. This case supports our theory that reasonably priced software will not be "ripped off" by hobbyists.

We actively encourage the continuation of such ethical behavior on the part of hobbyists, relative to fairly-priced software. We believe that it will encourage Tom and other computer pros to continue to develop excellent software and offer it to the hobbyist community at reasonable prices. We applaud such action by vendors such as Processor Technology, Apple Computer, Southwest Texas Products, Digital Research, and others.



BUGS & fixes



MINOL: TINY TREK— MORE DETAILS & A CORRECTION

Dear Jim, July 12, 1976

If you are planning to publish the TINY TREK game, here are some additional details and corrections.

Sincerely,

Erik Mueller

Britton House
Roosevelt NJ 08555

Too late! We have already published it in *DDJ* Vol. 1, No. 6. —JCW

TINY TREK
(written for MINOL)

A Sample TVT screen update looks like this:

```
  1234567
1 ·  K·
2     K     SECTOR      6  5   - Location of the Enter-
                                   prise in quadrant
3 K K       STARDATE    76      - Increases by one each
                                   display
4  ·  ·     ENERGY      231     - Amount of energy
                                   (maximum = 255)
5       B   KLINGONS    12      - Number of Klingons
                                   left in galaxy
6  ·  E     CONDITION   *RED*   - There are Klingons in
                                   the galaxy
7

7 UNIT HIT FROM KLINGON

COMMAND?
```

On the space printout, the "K" represents a Klingon ship; the ".", a star; the "B", a starbase; and the "E", the Enterprise. The purpose of the game is to wipe out all the Klingons without the energy dropping below 60.

COMMANDS

1. Go to new sector (move within quadrant). The amount of energy used depends on the distance traveled.

2. Go to a new quadrant. The galaxy is made up of an infinite amount of quadrants (such as the one above) which are generated when you move to them. You wll never see the same quadrant twice, and you have no control over where you go. The amount of Klingons in a quadrant is always less than or equal to the amount left in the galaxy.

3. Fire phasers. You fire phasers to destroy Klingons. The amount of energy used depends on the distance away from the Enterprise.

Traveling or firing through obstructions is OK, although you may not move to an occupied sector.

Refuelling is accomplished by "orbiting" a starbase: Maneuver yourself to within one unit of it, and then move to adjacent sectors until your energy goes back up to 255.

Corrections to listing previously sent:

```
3 X=1 : A=0 : S=250 : T=250
10 J=J+1 : IF A=W; GOTO12 : IF J<8; GOTO9
16 (delete)
13 ( 12,E- 1*8+f+200 )=3:IF 150<!; GOTO17
39 X=X+1 : IF X 8; GOTO22 : PR : IF L<50 ; GOTO170
```

ERRATA FOR COATS' TVT-2 KEYBOARD LOADER

Bob, 6/18/76

I just received a copy of *Dr. Dobb's Journal* Volume 1, No. 3. I was looking over my program on page 16 ["Keyboard Loader for Octal Code via the TVT-2"], and I noticed an error. The label GO should be at 006, and location 061 should contain 006.

Keep up the good work.

Jack O. Coats, Jr

213 Argonaut, No. 27
El Paso TX 79912

MINOL Errata & Praise

Dear Jim: July 5, 1976

I have just received a letter from Joseph F. Gaffney listing a zillion errors or typos in the MINOL listing. Below is a list of the corrections that should be made. Apparently, the listing has been published. But I still haven't received the issue or any issues after the third. Please check with the subscription department for me. **ERRORS** (Most of them were pointed out by Joseph F. Gaffney, 321 Lyndhurst Ave., Lyndhurst NJ 07071.)

Changes are underlined.

```
GSM     001106
        002345 303 013 001
ACT     003123
        003211              CPI "("
        003206 312 371 003
        003317 320 315 003
        003327 016 000      MVI C 0
        003333 312 345 003
        003375 303 134 003  JMP, NXGT
        004043 303 134 003  JMP, NXGT
NXGT    003134
        004005 176          MOV A,M
INPTXT  004052
        004135 302 144 004
        004155 303 057 004  JMP, INQ
CHEKN   005062
        005256 312 276 005  JZ, INU
        006027 376 243      CPI "#"
        006353 311          RET
        006053 322 013 001  JC, LPBUB
        006056 303 041 001  JMP, EXEC
        006101 302 066 006  JNZ,SHME
```

Sincerely yours,

Erik T. Mueller
36 Homestead Lane
Roosevelt NJ 08555

Thanks for the errata. Your subscription was entered on May 19th. Issues no. 4 and no. 5 were mailed a week and a half apart, about a month prior to your letter. I *encourage* you to complain to your local congressional reps (complaining to the Post Office appears to be useless). I also mailed an extra copy of the issue in which MINOL appeared, separately.
–JCW

Dear Mr. Warren, July 19, 1976

Erik Mueller's MINOL version of Tiny BASIC in the April issue is fantastic, and I'm really enjoying it! I relocated it to fit with my monitor (a modified 'JAMON' [MITS User's Group]), and it's running with a Model 33 Teletype. Some of the MINOL subroutines are useful in other programs as well, and are easily called (particularly useful is PRINTXT). MINOL is fun, certainly, but it is also very amazing (how can it be so smart and yet so small?).

There were a few typographical errors which were easy to correct. Corrections (at the original addresses) are shown below.

Address	Was	Change to
001/350	342	242
002/050	274	273
002/346	OMITTED	013,001
003/207	271	371
003/317	320	302
003/320	OMITTED	315,003
003/327	OMITTED	016,000
003/334	OMITTED	345,003
004/005	OMITTED	176
004/060	OMITTED	107
004/137	OMITTED	004
005/256	OMITTED	312
005/257	DISPLACED	276,005
006/353	OMITTED	311

As the program stands, the processor will enter an endless loop if you try to divide by zero. This doesn't hurt anything, but it does hang it up. To cure this, you might wish to add the following routine to test for division by zero. It adds Error 7.

Change:		003/326	315,000,004	CALL DIV0
DIV0 *	MOVAB	004/000	170	MOVE B TO A
	ORAA	004/001	267	SET STATUS
	MOVAC	004/002	171	MOVE C TO A; STATUS UNAFFECTED
	MVIC	004/003	016,000	CLEAR C
	RNZ	004/005	300	RETURN NOT ZERO
	MVIB	004/006	006,067	ERR '7'
	JMP	004/010	303,226,004	JMP ERR

*This is my 'relocated' code. Any convenient locations will do.

Yours truly,

Phillip L. Hansford
6841 Haywood St.
Tujunga CA 91042

EUGENE STORE: THE REAL OREGON COMPUTER CO.

Dear Bob, 4/26/76

Indeed we are running a store and would love it if you mentioned us. The store opened May 8.

Thanks,

John Montgomery
The Real Oregon Computer Co.
205 W 10th
Eugene OR 97401

NEW CLUB CONTACTS: VENTURA COUNTY COMPUTER SOCIETY

VCCS is a Chapter of the Southern California Computer Society. Its mailing address is P.O. Box 525, Port Hueneme, CA 93041. For more direct responses, contact their Secretary, Fred Moeckel, 4240 Harbor Blvd. No.208, Oxnard, CA 93030.

Computer Music Bibliography

by John Snell, Research Engineer, People's Computer Co.
Box 310, Menlo Park CA 94025; 415 323-3111

Digital circuits may be used for controlling analog synthesizers, direct digital synthesis, composing music, analyzing (or tracking several parameters of) traditional musical instruments and the voice, spacial movement of sounds, and processing of musical sounds (filtering, reverberation, choral effects, etc.). I hope the following list of articles and books will help some of you to develop systems which are capable of making music enjoyable even by master musicians. This list is relatively short, and includes only a "taste" of relevant topics not specifically about digital music. For a more comprehensive, well-organized listing see the bibliography from *Electronotes* (a fine electronic music periodical edited by Bernie Hutchins).

J. Allen, "Computer Architecture for Signal Processing," *IEEE*, Vol. 63, No. 4; April, 1975.

W. Apel, *Havard Dictionary of Music*, Halliday Lithograph, 1969.

J. Backus, *The Acoustical Foundations of Music*, W.W. Norton, 1969.

J.K. Baker, "The DRAGON System—An Overview," *IEEE*, Vol. ASSP-23, No. 1; Feb., 1975.

S.C. Bass, and B.J. Leon, "Designers' Guide to Digital Filters," *EDN*, Jan. - June, 1974 (6 issues).

J.W. Beauchamp & H. Von Foerster, editors, *Music by Computer*, Wiley, 1969.

J.W. Beauchamp, "Analysis and Synthesis of Cornet Tones Using Nonlinear Interharmonic Relationships," *JAES*, Vol. 23, No. 10; 1975.

J.W. Beauchamp, "A Computer System for Time-Variant Harmonic Analysis and Synthesis of Musical Tones," *Music by Computers*, 1969.

J.W. Beauchamp, "Additive Synthesis of Harmonic Musical Tones," *JAES*, Vol. 14, No. 4, p. 332; 1966.

L. Beranek, "Digital Synthesis of Speech and Music," *IEEE Transactions on Audio*, Vol. AU-18, No. 4; Dec., 1970.

S. Bertram, "Frequency Analysis Using the Discrete Fourier Transform," *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-18, No. 4; Dec., 1970.

B.A. Blesser, K. Baeder, & R. Zaorski, "A Real-Time Digital Computer for Simulating Audio Systems," *JAES*, Vol. 23, No. 9, p. 698; 1975.

B.A. Blesser, "An Investigation of Quantization Noise," *JAES*, Vol. 22, No. 1, p. 20; 1974.

B.A. Blesser, & F. Lee, "An Audio Delay System Using Digital Technology," *JAES*, Vol. 19, No. 5, p. 393; May, 1971.

E. Blom, editor, *Grove's Dictionary of Music and Musicians*, St. Martin's Press.

D. Bohn, editor, *Audio Handbook*, National Semiconductor Corp., Santa Clara, Cal.; 1976.

R.W. Burhams, "Digital Tone Synthesis," *JAES*, Vol. 19, No. 8; Sept., 1971.

R.W. Burhams, "Single-Bus Keyboard Control for Digital Musical Instruments," *JAES*, Vol. 19, No. 10, p. 865; Nov., 1971.

R.W. Burhams, "Simplified Educational Music Synthesizer," *JAES*, Vol. 19, No. 2, p. 127; Feb., 1971.

R.P. Ceely, "A Composer's View of MITSYN," *AES Preprint No. 811 (M-1)*, Oct., 1971.

H. Chamberlin, "Fourier Series Waveform Generator, Part 1," *EN*, Vol. 5, No. 39, p. 2; May, 1974.

J.M. Chowning, J.M. Grey, L. Rush, & J.A. Moorer, *Computer Simulation of Music Instrument Tones in Reverberant Environments*, Stanford University Dept. of Music Report No. STAN-M-1; June, 1974.

J.M. Chowning, "The Simulation of Moving Sound Sources," *JAES*, Vol. 19, No. 1, p. 2; Jan., 1971.

J.M. Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *JAES*, Vol. 21, No. 7, p. 526; Sept., 1973.

J.M. Chowning, "Stanford Computer Music Project," *NW*, No. 1; 1972.

W. Cochran, J. Cooley, D. Favin, H. Helms, R. Kaenel, W. Lang, G. Maling Jr., D. Nelson, C. Rader, P. Welch, "What is the Fast Fourier Transform," *IEEE*, Vol. 55, No. 10, p. 1664; Oct., 1967

R. Crochiere, & P. Penfield, Jr., "On the Efficient Design of Bandpass Digital Filter Structures," *IEEE*, Vol. ASSP-23; 1975.

M.G. Crosby, "Carrier and Side-Frequency Relations with Multi-Tone Frequency or Phase Modulation," *RCA Review*, Vol. 3; 1938.

D.R. Curtis, "A Monolithic Voltage-Controlled Amplifier Employing Log-Antilog Techniques," *JAES*, Vol. 24, No. 2; March, 1976.

D. De Kold, "Binary Division Produces Harmonic Frequencies," *Electronics*, Dec., 1972.

J.L. Divilbliss, "The Real-Time Generation of Music with a Digital Computer," *JMT*, No. 8, p. 99; 1964.

J. Dubnowski, R. Schafer, & L.R. Rabiner, "Real-Time Digital Hardware Pitch Detector, *IEEE*, Vol. ASSP-24, No. 1; Feb., 1976.

J.L. Flanagan & L.R. Rabiner, editors, *Speech Synthesis*, Dowden, Hutchinson, & Ross; 1973.

A. Forte, *Tonal Harmony in Concept and Practice*, Holt, Rinehart, & Winston; 1962.

N.V. Franssen & L. Van der Peet, "Digital Tone Generation for a Transposing Keyboard Instrument," *Philips Tech. Rev.*, No. 31, p. 354; 1970.

M.D. Freedman, "Technique for Analysis of Musical Instrument Tones," Ph.D. dissertation, University of Illinois, Urbana; 1965.

M.D. Freedman, "Analysis of Musical Instrument Tones," *JASA*, No. 41, p. 793; 1967.

M.D. Freedman, "A Method for Analyzing Musical Tones," *JAES*, Vol. 16, No. 4, p. 419; Oct., 1968.

S. Freeny, "Special-Purpose Hardware for Digital filtering," *IEEE*, Vol. 63, No. 4; April, 1975.

R. Gabel, "A Parallel Arithmetic Hardware Structure for Recursive Digital Filtering," *IEEE*, Vol. ASSP-22; Aug., 1974.

J. Gabura & G. Ciamaga, "Computer Control of Sound Apparatus for Electronic Music," *AES preprint No. 520;* 1967.

S. Gill, "A Technique for the Composition of Music in a Computer," *Computer Journal*, Vol. 6, No. 2, p. 129; July, 1963.

J.M. Grey, *An Exploration of Musical Timbre*, Stanford University Dept of Music Report No. STAN-M-2; Feb., 1975.

P. Grogono, "MUSYS: Software for an Electronic Music Studio," *Software—Practice and Experience*, No. 3, p. 369; 1973.

H.L.F. Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music*, translation—originally written in German in 1863—and additions by A.J. Ellis, Dover, N.Y.; 1954.

W. Henke, *Multiple Interactive Tone Synthesis System*, Research Laboratory of Electronics, M.I.T.; Oct., 1971.

L. Hiller & L. Isaacson, "Musical Composition with a High-Speed Digital Computer," *JAES*, Vol. 6, p. 154; July, 1958.

L. Hiller, "Computer Music," *Scientific American*, p. 109; Dec., 1959.

L. Hiller, "A Review of Decca Recording DL-9103, 'Music from Mathematics'," *IEEE*, Vol. 51, No. 3, p. 538; March, 1963.

L. Hiller, "Musical Applications of Electronic Digital Computers," *Gravesano Review*, pp. 26-28; Nov., 1965.

C. Hovey & D. Seamans, "A Polyphonic Keyboard for a Voltage-Controlled Music Synthesizer," *JAES*, Vol. 23, No. 6, p. 459; 1975.

K. Huehne, "Programmable ROM's Offer a Digital Approach to Waveform Synthesis," *EDN*, Aug., 1972.

B. Hutchins, "B-004 Sourcelist of Electronic Music and Musical Engineering," *EN*, 1975.

B. Hutchins, *Musical Engineer's Handbook*, (*EN* address) 1975.

B. Hutchins, "The Frequency Modulation Spectrum of an Exponential Voltage-Controlled Oscillator," *JAES*, Vol. 23, No. 3, p. 200; 1975.

B. Hutchins, "Application of a Real-Time Hadamard Transform Network to Sound Synthesis," *JAES*, Vol. 23, No. 7, p. 558; 1975.

B. Hutchins, "Digital Harmonics 1, Waveshaping 2," *EN*, Vol. 2, No. 12, p. 5; Sept., 1972.

B. Hutchins, T. Mintner, & C. Anderton, "Digital Harmonics 2," *EN*, Vol. 2, No. 13, p. 12; Oct. 1972.

B. Hutchins, "Analog Delay for Musical Engineering," *EN*, Vol. 7, No. 56; Aug., 1975.

IEEE, Vol. 63, No. 4, special issue on digital signal processing; April, 1975.

F. Itakura, & S. Saito, "Digital Filtering Techniques for Speech Analysis and Synthesis," in Conf. Rec., 7th Int. Congr. Acoustics, Paper 25C No. 1; 1971.

E.V. Jansson, & J. Sundberg, "Long-Time-Average-Spectra Applied to Analysis of Music," *Acustica,* Vol. 34, No. 1; Nov., 1975.

L.B.W. Jolley, *Summation of Series,* Dover, N.Y.; 1961.

P. Knowlton, "Capture and Display of Keyboard Music," *Datamation,* P. 56; May, 1972.

S. Kriz, "A 16-Bit A-D-A Conversion System for High Fidelity Audio Research," IEEE Symposium on Speech Recognition, p. 278; April, 1974.

P. Lehman, "Harmonic Structure of the Tone of the Bassoon," *JASA,* Vol. 36, No. 9, p. 1649; Sept., 1964.

V. Lesser, D. Fennell, L. Erman, & R. Reddy, "Organization of Hearsay II Speech Understanding System," *IEEE,* Vol. ASSP-23, No. 1; Feb., 1975.

T. Letowski, "Difference Limen for Nonlinear Distortion in Sine Signals and Musical Sounds," *Acustica,* Vol. 34, No. 2; Dec., 1975.

H. Lincoln, editor, *The Computer and Music,* Cornell University Press; 1970.

J. Link, Theory and Tuning: *Aron's Meantone Temperament and Marpsurg's Temperatment "I",* Tuners Supply Co., Boston; 1969.

B. Lin, & A. Peled, "A New Hardware Realization of High Speed Fast Fourier Transformers," *IEEE,* Vol. ASSP-23, No. 6; Dec., 1975.

H.C. Longuet-Higgins, & M.J. Steedman, *On Interpreting Bach in Machine Intelligence VI;* B. Meltzer, & D. Michie, editors, Edinburgh University Press, Edinburgh; p. 221; 1971.

D. Luce, "Description of a Real-Time Multipartial Waveform Analyzer-Synthesizer," AES preprint No. 611; 1968.

D. Luce, "Dynamic Spectrum Changes of Orchestral Instruments," *JAES,* Vol. 23, No. 7; 1975.

D. Luce, & M. Clark, "Physical Correlates of Brass Instrument Tones," *JASA,* Vol. 42, p. 1232; 1967.

M. Mathews, *The Technology of Computer Music,* M.I.T. Press, Boston; 1969.

M. Mathews, et. al., "Computers and Future Music," *Science,* Vol. 183, No. 4122; Jan. 25, 1974.

N. Mathews, "The Digital Computer as a Musical Instrument," *Science,* Vol. 142, p. 553; 1963.

M. Mathews, & J. Kohut, "Electronic Simulation of Violin Resonances," *JASA,* No. 53, p. 1620; 1973.

C.A. McGonegal, L.R. Rabiner, & A.E. Rosenberg, "A Semi-automatic Pitch Detector," *IEEE,* Vol. ASSP-23, No. 6; Dec., 1975.

N.J. Miller, "Filtering of Singing Voice Signal from Noise by Synthesis," Ph.D. thesis; University of Utah, Dept. of Computer Science; May, 1973.

N.J. Miller, "Pitch Detection by Data Reduction," *IEEE,* Vol. ASSP-23, No. 1, pp. 72-78; Feb., 1975.

F.R. Moore, & M. Mathews, "Grove—A Computer Program for Real-Time Music and Sound Synthesis," *Communications of the ACM,* Vol. 13, No. 12; Dec., 1970.

J.A. Moorer, "The Optimum Comb Method of Pitch Period Analysis of Continuous Digitized Speech," *IEEE,* Vol. ASSP-22, No. 5, p. 330; Oct., 1974.

J.A. Moorer, "Music and Computer Composition," *Communications of the ACM,* Vol. 15, No. 2; Feb., 1972.

J.A. Moorer, *On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer,* Stanford University Dept. of Music Report No. STAN-M-3; May, 1975.

J.A. Moorer, *The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulae,* Stanford University Dept. of Music Report No. STAN-M-5; Dec., 1975.

J.A. Moorer, *On the Loudness of Complex Time-Variant Tones,* Stanford University Dept. of Music Report No. STAN-M-4; Feb., 1975.

NUMUS West, No. 4; 1973. (computer music issue)

T.H. O'Beirne, "Computer Program Which Plays Music by Microtones," *Computer Journal,* Vol. 13, No. 4, p. 350; 1970.

A.V. Oppenheim, & R.W. Schafer, *Digital Signal Processing,* Prentice Hall; 1975.

W. Piston, *Counterpoint,* W.W. Norton Publishers.

W. Piston, *Orchestration,* W.W. Norton Publishers.

G. Plitnik, & W. Strong, "Digital Filter Technique for Synthesis of Bassoon Tones," *JASA,* No. 47, p. 131; 1970.

R. Plomp, "Timbre as a Multidimensional Attribute of Complex Tones," *Frequency Analysis and Periodicity Detection in Hearing,* edited by R. Plomp & G.F. Smoorenburg; A.W. Sijthoff, Leiden; 1970.

L. Rabiner & B. Gold, *Theory and Applications of Digital Signal Processing,* Prentice-Hall, N.J.; 1975.

L. Rabiner & C. Rader, editors, *Digital Signal Processing,* IEEE Press; 1972.

J.C. Risset & M. Mathews, "Analysis of Musical Instrument Tones," *Physics Today,* Vol. 22, No. 2, p. 23; 1969.

J.C. Risset, "Computer Study of Trumpet Tones," Bell Telephone Lab, Murray Hill, N.J.; 1966.

P. Samson, "Digital Signal Synthesizer," Systems Concepts, San Francisco, Cal.

P. Samson, "Digital Synthesizer—Brief Functional Description," Systems Concepts, San Francisco, Cal.

P. Samson, "Digital Synthesizer—Programming Specifications," Systems Concepts, San Francisco, Cal.

S. Saunders, "FM The Easy Way," Carnegie Mellon University Computer Science Dept & Xerox PARC; 1974.

R.A. Schaefer, "Digital Generation of Equal Temperament," *IEEE,* Vol. AASP-23, No. 4; Aug., 1975.

M. Schroeder, "Natural Sounding Artificial Reverberation, *JAES,* Vol. 10, p. 219; 1962.

M. Schroeder, "Models of Hearing," *IEEE,* Vol. 63, No. 9; Sept., 1975.

R. Schumacher, "Self-Sustained Musical Oscillators," paper delivered at the 91st meeting of the Acoustical Society of America; April, 1976.

J. Sennott, "Efficient Digital Conversion of Audio Signals," AES preprint No. 970 (M-4); Sept., 1974.

W. Slawson, Syntal II; *A Speech Oriented Computer Synthesizer Revisited,* Dept. of Music, University of Pittsburgh, Pa.

W. Slawson, "Vowel Quality and Musical Timbre as Functions of Spectrum Envelope and Fundamental Frequency," *JASA,* No. 43, p. 87; 1968.

G.W. Small, "Synthesis of the Musical Scale Using Non-integer Frequency Division," *JAES,* Vol. 21, No. 4, p. 261; May, 1973.

G.W. Small, "Rate-Feedback Binary Counters in Musical Scale Generation," *JAES,* Vol. 21, No. 9, p. 702; Nov., 1973.

L. Smith, "Score—A Musician's Approach to Computer Music," *JAES,* Vol. 20, No. 1; Jan./Feb., 1972.

L. Smith, "Editing and Printing Music by Computer," *JMT,* p. 292; Fall, 1973.

S. Smith, "Computer Music in 1972," *Computers and Automation,* Oct., 1972.

G. Steinke, "Experimental Music with the Subharchord Subharmonic Sound Generator," *JAES,* Vol. 14, No. 2, p. 140; April, 1966.

S.S. Stevens & H. Davis, *Hearing—Its Psychology and Physiology,* Wiley, N.Y.; 1938.

T.G. Stockham, Jr., "A-D and D-A Converters: Their Effect on Digital Audio Fidelity," AES Preprint No. 834 (D-1), Oct., 1971.

W. Strong & M. Clark, "Synthesis of Wind-Instrument Tones," *JASA,* Vol. 41, p. 39; 1967.

W. Strong, & M. Clark, "Perturbations of Synthetic Orchestral Wind-Instrument Tones," *JASA,* No. 41, p. 277; 1967.

R. Talambiras, "Digital-to-Analog Converters: Some Problems in Producing High Fidelity Systems," *Computer Design,* Vol. 15, No. 1, p. 63; Jan., 1976.

J.C. Tenny, "Sound Generation by Means of a Digital Computer," *JMT,* No. 7, p. 24; 1963.

H.M. Tremaine, *Audio Cyclopedia,* Sams, N.Y.; 1969.

D.E. Walker, "The SRI Speech Understanding System," *IEEE,* Vol. ASSP-23, No. 5, Oct., 1975.

L. Wedin & G. Goude, "Dimensional Analysis of the Perception of Instrumental Timbre," *Scand. Jour. Psych.,* No. 13, P. 228; 1972.

S. White, "On Mechanization of Vector Multiplication," *IEEE,* Vol. 63, No. 4; April, 1975.

T. Winograd, "Linguistics and the Computer Analysis of Tonal Harmony," *JMT,* Vol. 12, No. 1, p. 2; 1968.

J. Zingheim, "Introduction to Computer Music Techniques," *EN,* Vol. 6, No. 42, p. 1; Aug., 1974.

ABBREVIATIONS USED, AND ADDRESSES

ACM	Association for Computing Machinery
	Acustica, S. Hirzel, Stuttgart 1, Birkenwaldstr, 44, Postfach 347, Germany.
AES	Audio Engineering Society
ASSP	*IEEE* Transactions on Acoustics Speech and Signal Processing

Communications of the ACM
1133 Avenue of the Americas, NYC 10036

Computer Design
Box A, Winchester MA 01890

Computer Journal
British Computer Society
29 Portland Place
London, England W1N 4AP

Computers and Automation [now called *Computers and People*]
Berkeley Enterprises, Inc.
815 Washington St, Newtonville MA 02160

Datamation
Technical Publishing Co.
1301 South Grove Ave., Barrington IL 60010

ED *Electronic Design*
50 Essex St., Rochell Park NJ 07662

EDN *EDN Magazine*
270 St. Paul St., Denver CO 80206

EN *Electronotes*
203 Snyder Hill Rd., Ithaca NY 14850

IEEE *Institute of Electrical and Electronics Engineers*
345 E 47 St., NYC 10017

JAES *Journal of the Audio Engineering Society*
60 E 42 St
Lincoln Blg, Room 929
NYC 10017

JASA *Journal of the Acoustical Society of America*
American Institute of Physics
335 E 45 St., NYC 10017

JMT *Journal of Music Theory*
Yale School of Music
Yale University
New Haven CT 06520

NW *NUMUS West*
Box 135, Mercer Island WA 98040

Perspectives of New Music
Box 231, Princeton NJ 08540

Science
American Association for the Advancement of Science
1515 Massachusetts Ave. N.W.
Washington DC 20005

Scientific American
415 Madison Ave., NYC 10017

AMBIGUITY ABOUT ADVERTISING

The "Do You..." editorial in the preceeding issue mentioned the fact that we don't accept commercial advertising. This was in apparent contradiction to the "Advertising" note printed inside the front cover, as well as to the obvious fact that we carry articles about commercial products. To clarify this point:

The "Advertising" policy inside the front cover has been corrected to reflect our current position. The preceeding policy was a tentative one, chosen when we first began. We had been automatically reprinting that information in each issue and neglected to note that it was inaccurate (blush).

We do, and will continue to, carry articles about commercial products and services. We are not paid for this, however. It is provided as information for our readers. These articles often concern items that are unusually good deals for hobbyists (and thus worth special mention), or items being offered by very small companies that are too small to afford the hefty advertising rates charged by the other publications.

Incidentally, you might note that we try to include address, telephone, and single unit pricing in these articles (when we know them); information often missing from paid-for advertisements.

IT WAS *NOT* A "POLISH JOKE"

There was a "digital calculator" cartoon on page 18 of the April, 1976, issue of *DDJ* that included text: "Made in Hong Kong . . . Polish versions also available." Our production manager, who is an outstanding publications production person but not a computer phreaque, questioned our including a cartoon with an apparent racial slur. We explained the rather in-group pun that was involved, and put his mind at rest. It occurred to us that, although no one else has commented on the cartoon, it might be appropriate to explain the pun:

Handheld calculators use either infix or postfix processing for accepting a sequence of data and arithmetic operators. If infix processing is used, there must be some *explicit* means for indicating the sequence in which operations are to be performed. Commonly, parentheses are used (e.g., *3 x (4 + 5)* indicates that the + at the right is to be performed before the *x* at the left).

Postfix processing, however, is a very classy parentheses-free system in which the order of data and operator entry determines the order of operations (e.g., *4 5 + 3 x* produces the same result as the preceeding infix-coded example). The latter example says, *Store 4* then *Store 5* then *Add last two elements stored and store the result* then *Store 3* then *Multiply the last two elements stored*. Postfix processing is one of the major concepts utilized in information processing by digital computers. It possibly rivals the concept of an electrically stored program, or the application of Boolean algebra to digital circuitry in its importance to computers.

Why all this monologue on postfix processing? Because, more often than not, computer professionals refer to this as "Polish postfix" or simply "Polish" processing. It is so-named because its inventor was a Polish logician named Łukasiewicz. So, you see, if anything, the cartoon's pun was really a racial compliment.

CLASSROOM USE OF HAND-HELD CALCULATORS

The November 1976 issue of the ARITHMETIC TEACHER, published by the National Council of Teachers of Mathematics, will focus on instructional uses of hand-held calculators. Copies of this issue will be available for distribution at a special price of 50 cents each under the following conditions:

1). The minimum order is 100 copies.
2). All 100 copies must be sent to a single address.
3). Orders must be in the NCTM Headquarters Office by 31 August 1976.
4). NCTM will pay the shipping charges if full payment is received with the order.

To place an order, or for further information, contact Charles R. Hucka, Director of Publications Services, NCTM, 1906 Association Drive, Reston, Virginia 22091, or call (703) 620-9840.

REPORT ON ELECTRONIC HAND CALCULATORS IN EDUCATION

The body of the Final Report on the National Science Foundation supported project, "Electronic Hand Calculators: The Implications for Pre-College Education," is now available from the ERIC Information Analysis Center for Science, Mathematics, and Environmental Education, 1200 Chambers Road, The Ohio State University, Columbus, OH 43212. The 350-page complete Final Report will be available later in 1976 from the ERIC Document Reproduction Service, Box 190, Arlington, VA 22210.

DON'T PRINT ITEMS IN *DDJ* THAT APPEAR ELSEWHERE

Dear Dr. Dobb,

Please concentrate on infromation not available elsewhere—if *PCC* moves toward "blue sky," *DDJ* should move toward practicality. Draw a sharp line and find out what your readers *really* want. MOST OF ALL: don:t waste space on something that has already been printed nationally—we *have* that.

James R. Zimmerman
7835 Querida Lane
Dallas TX 75248

SMRT Will Hurt

by Leroy Finkel

(reprinted from *PCC* Vol. 4, No. 6)

EDITOR'S NOTE: In general, we are trying to avoid duplicating articles in *PCC* and *DDJ*, since our subscribers overlap. This information is sufficiently important, however, that we believe it warrants as widespread publicity as possible.

We believe that in and for the foreseeable future, home computer users will need use of the phone system. They will need it, not for traditional time-sharing, but for accessing large databases with their personal computers.

Since the phone system is the *only* public carrier that is reasonably available to home computer users, a government-licensed monopoly, it behooves us to assure that it meets our needs with reasonable economy.

On July 1, 1976 Pacific Telephone will implement a new phone tariff in major cities in California, on all business telephone service, called Single Message Rate Timing. SMRT, as it is known, will assess a supercharge of approximately 1 cent per minute on all calls over five minutes. That means that the dial-up, time-sharing setup that you presently enjoy for a cost of $7.50 a month, or thereabouts, to make local computer calls could increase to as much as $80 a month, based on a 6-hour day usage. HELP!

SMRT has come to California. How long will it be before it ZAP's you? How does SMRT work? For any call that exceeds 5 minutes a one message unit supercharge will be added to the bill for each 5 minutes after the first five minutes. Since one message unit currently costs Californians 5.75 cents, that means that a previously "free" 30 minute call will cost 28.75 cents under SMRT; a sixty minute call, 63.25 cents. Projecting hours into days, a school that uses dial-up service for 3 hours a day, 20 days a month can expect additional phone charges of $36.00 each month; a six hour day, $72.00 for EACH TERMINAL IN OPERATION. Can your budget afford that?

> The affects of SMRT could make you reconsider your hardware acquisition, location and use decisions. SMRT makes local microprocessors in each school look like a real wise decision. Think about it!

In our area, the phone company has computer printouts projecting the implications of SMRT on your bill. Ask to see this information, it is scary. In all honesty, I must admit that this is one new phone charge that does make sense. Those of us that use dial-up service for computer data, all day long, are not using the equipment "as intended." Therefore, we 'tie-up' valuable equipment for long periods without paying our fair share of the costs. SMRT will even all this out.... and then some!

THINGS TO DO

There are many things you can do to avoid or minimize the affects of SMRT. Your greatest problem will be getting all the right answers to your questions (We have not provided all the answers, just some suggestions.)

> CAVEAT EMPTOR. Do not expect to get all the "right" answers from the local phone company staff. They provide a limited array of services in the area of data communications. The right solution for you may not be in their array. You must ask Telco competitors about services they offer that fill the void of services provided by Telco.

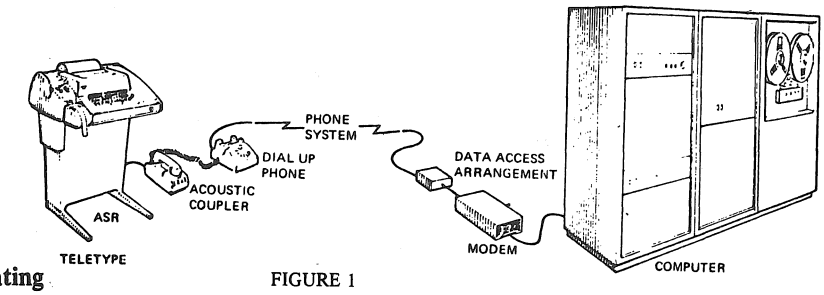


FIGURE 1

Your present phone-terminal-to-computer link probably like that shown in Fig. 1. Local costs will vary, but generally our costs break down like this:

At Terminal site - Dial-up business line	$7.50/month
At Computer site-Business phone line(not shown	7.50
Data Access Arrangement(DAA)	5.50
Phone costs per terminal	20.50/month
Add modem and acoustic coupler rental ($15@) (may be purchased for $300 each)	30.00
Total communications costs per terminal	$50.50/month

This is the type of configuration that SMRT will really hurt. But, such a dial-up system has some real advantages to the user: Terminals can be used from any location where there is a standard dial-up telephone(home included); you can 'oversubscribe' your system and users schedule themselves or compete for open phone lines.

Alternative No. 1 Probably the cheapest way to change your phone set-up to minimize the affects of SMRT is to convert all outside phone lines to run through your school or district PBX switchboard. (assuming you have one.) If you do not have a PBX, now may be the time for the school system to get one. Such a move should greatly reduce your normal monthly phone charges and because the system is "internal" via the PBX, no SMRT charges will be added to your bill. Schools can be located anywhere. SMRT charges are avoided as long as everything runs through the PBX. Flexibility continues as before. The only problem we have discovered is that there may not by space on your PBX to accomodate additional lines. Each computer line requires TWO PBX extensions (one at computer, one at the terminal). But do check this alternative out. We can't estimate how much, but can guarantee you substantial dollar savings.

Alternative No. 2. The phone company will likely suggest a Lease Line running from each terminal directly to the computer (picture a pair of wires running directly between the two sites). Lease lines are charged a fixed monthly fee for 24 hour usage that is based on distance from point to point. For the short distances we needed (no more than 12 miles), lease line costs ranged from $14 to $42 per month per terminal. Add to this a Telco modem at each end for $50/month (No DAA needed).

Lease lines avoid SMRT costs, and, depending on distances, should save you substantial money. However, you lose the flexibility of dial-up from any phone and it is unclear how you can "oversubscribe" a lease line communication system.

Alternative No. 3. Instead of using two Telco modems for $50/month, you can use lease lines as suggested in Alternative 2 but buy or lease modems from a company other than the phone company. Such modems may cost as little as $200 each and may lease for as little as $25 /month (for both). If you already own modems and acoustic couplers as part of a dial-up system, your present equipment MAY be usable with lease lines. Some will not be usable on lease lines. Contact your equipment supplier to see if your hardware is convertible for use on lease lines. We found our equipment was usable after a minor field modification made to our connecting cables.

Alternative No. 4 In the long run, the least-cost method to provide for multiple terminals at one site is to connect ONE phone lease line between computer and terminal site. (Fig.3) To this one line you connect all your terminals (19 teletypes, claims one firm- 12 is more real) - using a special modem called a Frequency Division Multiplexer (FDM). The FDM allows multiple terminals to share one phone line by varying the frequency at which the data is transmitted. Teletypes at 10 CPS and 30 CPS terminals can share the same line via the FDM's. At each terminal you need an FDM with another FDM at the computer end to "unscramble" the data. The FDM units cost about $450 each, more than the standard lease line modems. But you only need one phone line. Therefore, your initial cash outlay may be greater, but in the long run you will save money because phone costs are reduced dramatically.

Terminals can even be located in different classrooms using FDM's. There is even the possibility of using ONE lease line connecting a number of schools and a

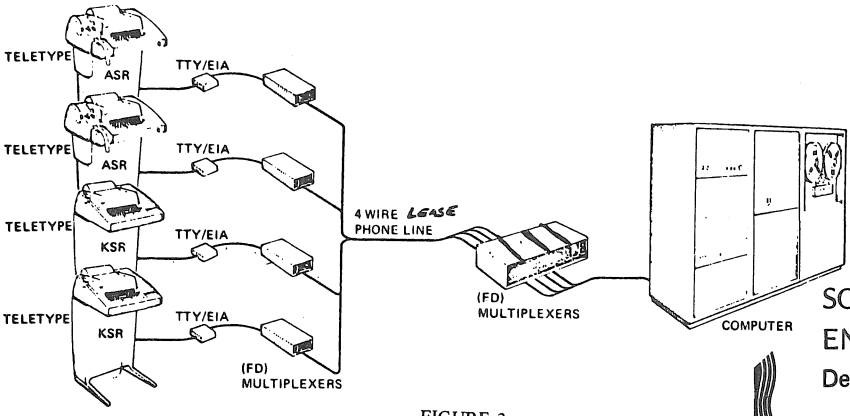


FIGURE 3

number of terminals. Technicians clam this Multi-Point lease line is not any more of a problem from a maintenance standpoint than a single point-to-point set-up. Thus, more phone money can be saved.

FDM capability is NOT available from the phone company at this time. Don't expect them to respond too warmly to your questions about this effective way to share lease lines and save money.

So now what? If this article serves no other purpose than to confuse you, we have accomplished something! Data communications has to be the most confusing topic we have investigated. Each call we made raised new confusion and new questions. Each "situation" at each new terminal site created a new set of problems, with a whole new set of answers.

What is the answer? It all depends on
Where you are in the greater US
Your computer
What equipment you already own
how much money you have
Should you spend big money NOW to save big money later
Should you spend less money now and have a
bigger phone bill forever...

and
so
on

SOLID STATE SALES SLIPS ORDER FOR 9 MONTHS

Dear Editor, April 3, 1976

I think it is worth telling about one supplier who is very remiss. I sent an order to Solid State Sales last July, for about $70 worth of electronic components. They sent roughly a third and backordered the rest. When October came and I had not received the balance I wrote them, with no reply received. My reminder in Decmeber brought a promise of 4-5 weeks delivery. When that passed I wrote again, in March. I have just received another promise of 4-6 weeks delivery. They have now had my money for nine months—I think maybe they ought to start paying interest. At any rate, if they can't supply the stuff they ought not to send out those enticing advertisements.

Kendall Stambaugh
5009 Guide Meridian
Bellingham WA 98225
206 734-9424

SAN FRANCISCO COMPUTER STORE OPENS

The Computer Store of San Francisco, affiliated with Byte, Inc., opened its doors for business in the second week of July. It is located at 1093 Mission Street; (415) 431-0640.

It used to be fun
To add one and one.
But it's no longer plain
What result to obtain
They say it could even be none.
--unknown

SOURCES OF ENERGY SOFTWARE SOUGHT

Dear Dr. Dobb,

I am trying to combine low cost computers, and alternative sources of energy. As a start, I'm collecting programs about energy use. Any programs or ideas, in any language, that readers could donate would be appreciated. If I get enough of a response, these will form the basis for a book.

Possible topics: Solar Parabolic Reflectors, Windmill Design, Thermal Loss, and Storage; Units Conversion (BTU, KW-HR), Battery Charge and Discharge, Flywheel Momentum, etc.

Keep up the good words,

Rich Roth
1052 Clark Ave
Mountain View
CA 94040

A COMMENT ON THE DIGITAL GROUP

Dear Dr. Dobb (alias Jim Warren?) July 15, 1976

I have (had?) a Digital Group 8080 system, which after being built didn't work quite right, and I had to send it back for de-bugging. It didn't care to read in the cassette tape, an initialization process needed to give it life. The still-born has been away for a month now, with no word from the doctor yet (it's under the care of Dr. Sudding and his staff - very competent I understand), but we're very hopeful.

Your journal and concept is the best. I have the D.G.'s version of TBX-TVCOS and would dearly love to see it run, but alas, (sob) I wait....

Christopher Hovey
ESD/OC
San Fran.APO
CA 96369

BAD SERVICE REPORT ABOUT THE DIGITAL GROUP

Dear PCC, June 30, 1976

Beginning to get bad vibes from The Digital Group. Purchased their television character generator cassette interface board. The Original character generator chip gave inverted characters on video monitor. The replacement was no good at all. On June 29th, I called about a replacement for the chip, and I requested that the chip be tested before being shipped. A Gus Calabrese stated that "They were swamped with orders and couldn't possibly get around to shipping the replacement for another two to three weeks."

I can understand their being quite busy, but I don't think that they should abandon any kind of customer service to previous purchasers of their equipment.

Thank you.

Robert L. Gerald
102 Coventry Dr.
Lakewood NJ 08701

LOU FIELD'S RESPONSE TO THE EDITOR'S NOTE TO SCCS MEMBERS IN THE MAY ISSUE

The May issue of the Journal carried a note that was essentially critical of Lou Fields, the current Vice President of the Southern California Computer Society, and their potential next President. As is PCC policy, equal space (in fact, much more space) is being provided for the response.

In response to Jim Warren's note to the members of the Southern California Computer Society:

In reply to your letter May *Dr. Dobb's* this is following my conversations by phone conversation with you and Bob Albrecht (after you wrote your letter but before I saw it — it arrived on the 22nd of July)

1. Your letter is unfair, inaccurate, and at best intemperate. If you really wanted to be constructive you could have called me first — or at least consulted Bob Albrecht who made the arrangement with me.
2. The facts are that I did not call to place an order. Bob called Ward Spaniol and Art Childs re: distribution of your products. He was told it was SCCS policy to handle no such material including its own magazine, Interface. They referred him to me since I had made arrangements for some other publications. I feel it in the best interests of the members to have magazines at meetings. Bob then called me and asked if he could send some (on consignment). I was not familiar with all the material but suggested he send what he thought was appropriate. Just before the meeting, 12 cases of papers arrived. I could not fit it all in my apartment or car.
3. On the day of the April meeting, I brought what I could fit into my car (along with the many supplies for the meeting).

Jim arrived at the meeting unannounced and proceeded to make a 45 minute presentation on *Dr. Dobb's*. The members were fascinated with his excellent talk, however it would have been better and more considerate if the Program Committee had been consulted about it so that the talk could have been scheduled, announced, and not have so disrupted two other excellent speakers.

Seeing the interest, I apologized to Jim because I thought I hadn't brought *Dr. Dobb's* (in fact I had and they were later uncovered and available). It was the seeming lack of the pre-shipped *Dobb's* which I believe may have really upset Jim.

4. Recognizing that the PCC was donated (but marked one dollar) I asked the staff to stamp them in some way so that they could not be inadvertently sold. A few (perhaps 12-20) were stamped as mentioned (in type approximately the size of a headline on the May letter). If I had been intending these for my own uses, I would have distributed them all this way. The newsletters were shipped to me, but no slight was intended and Jim made no mention of it at the time.
5. Though these items were on consignment as confirmed with Bob and Jim during my telecon, I have sent you a total of $259.75 as well as the remaining material. I trust you will refund the difference.
6. This unfortunate incident is perhaps the most dramatic example I've seen of the need for an ombudsman (consumer advocate) service.

The SCCS will, I hope, soon offer this, so that some reasonable inquiry can be made of the facts before damaging, insulting and unfair material is distributed for which no timely or effective rebuttal is available.

7. I think your technical work is worthwhile, clever, imaginative, and creative. It is my heartfelt with that all of these good virtues will soon be matched with responsibility, accuracy, and a respect for the feelings, reputation, and good name of others.

I, too, am working very hard to bring the knowledge and experience of computing to our friends. I would like to hear directly what the readers think of this matter.

Louis G Fields, 11662 Sunset Blvd Suite 2, Los Angeles CA 90049
(213)478-0388

We stand by our original May note. 1. Bob read and agreed with that note, prior to its publication. 2. The for-sale materials were not *sent on consignment. This was a legitimate misunderstanding in the original phone conversation. It was very explicitly stated, however, in a later conversation (referred to in point 5). 3. I had been invited to fly down and attend the meeting by Ward Spaniol and Art Childs. By agreement with Art, who was running the Announcements Period between the two major speakers, I spoke briefly about the Journal. Due to my own longwindedness and a flood of questions, I took almost 15 minutes. Others with other announcements both preceeded and followed me, thus it is unlikely that I disrupted the morning or afternoon speakers. 4. Stamping the give-aways as being complimentary copies was a good idea. The copy I have, however, was stamped with two different stamps, one stating 'Compliments of Louis G. Fields' and the other being a 7-line stamp with Lou's name being the largest and in boldface print. 5. As of July 29th, we have received $259.75 and publication returns with a wholesale value of $70.60. We assume that the remaining $29.40 will be forthcoming. We also assume that the publications were sold for their retail value and the resulting profit was handled in a manner acceptable to the SCCS. If any excess payment is received, it will be immediately refunded. 6. We agree that there should be consumer advocates. We can think of many more dramatic examples of their need than this. 7. Rather than avoid controversy, we prefer to present statements from both sides (or all sides), and let our readers make up their own minds.*

This particular controversy, however, has taken up enough space. THE END.

WHAT THIS WORLD NEEDS IS A GOOD . . .

Dear Dr. Dobb , May 1, 1976

As I see it, the hobbyist could use three different publications.

1. a monthly news magazine similar in content to the present *PCC*. This would be written for the average hobbyist who just wants to know what's going on in general terms without being inundated by a mass of technical detail. It could include:

Reprints of articles of general interest from the club newsletters.

News items related to home computers.

Articles on social implications of computers (although *Creative Computing* seems to be doing this pretty well).

Letters of general interest.

Suggestions, gripes, comments, and blue-sky stuff.

General information on new hardware, software, etc.

Listings of useful or novel applications programs or computer games.

Indices of selected non-technical or semi-technical articles, etc. from other publications.

Classified ads.

2. A monthly or quarterly technical journal similar to *DDJ*. This would be written for the hard-core hobbyist who is grimly determined to cram all the features of OS-360 into his homebrew machine. It could include:

Schematics from the club newsletters.

Indices to selected technical articles, etc. from other publications.

Letters of technical content.

Listings, hopefully legible, of useful non-applications software.

3. An annual catalog or directory of reference-type information. This could include:

Computer clubs.

Hobbyist-oriented publications (e.g., *BYTE*).

Computer stores and distributors.

Manufacturers of computers, kits, and other hardware.

Brief descriptions and prices of all current stuff of use to hobbyists and commercially available.

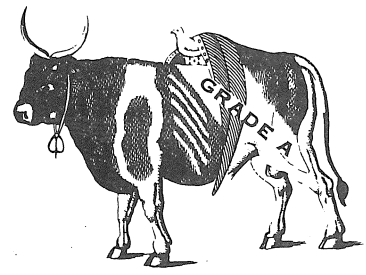
Jim Day
17042 Gunther St.
Granada Hills CA 91344

We're tryin'. Let us know, in a year, if you think we've succeeded. --JCW.

A reference Journal for home computer users

from the People's Computer Company—

DR. DOBB'S JOURNAL of COMPUTER CALISTHENICS & ORTHODONTIA



- 8½ x 11 inch magazine format
- "all meat" content; no display ads
- published monthly, except July & December

Content regularly includes:

Complete documentation on systems software
- Tiny BASIC, interpreters, debuggers, assemblers, compilers, cassette & floppy disc file systems, TV Dazzler software, graphics programs, music programs, etc.
- User documentation, implementation details, complete annotated source code listings

Design notes for build-your-own software

Detailed 'blue skying' about practical systems projects for the immediate future
- Tiny BASIC was the first such project (proposed, March, 1975; detailed, September, 1975; 5 systems up & running, March, '76)
- English language voice synthesis kits
- Computer music & graphics systems
- Shared mass storage
- & much, much more
- Electronic telephone book
- Community memory
- Biofeedback

Reprints of articles & schematics from computer club newsletters (*all* of 'em)

Directories: used equipment sources, users & their equipment, clubs & organizations, etc.

Indices: *All* articles in *all* major hobbyist publications, & selected articles from other publications

Active consumer advocacy for home computer users
- Supported by magazine sales—not by ads
- No vested interest in good will of manufacturers

Published ten times per year, monthly except in July and December.(Volume 1, Number 1 is January, 1976.)

☐ $1.50 for a single copy: Vol.____ No.____

☐ $10. per year (10 issues/year), to begin with Vol.____ No.____ ☐ This is a renewal.

for foreign subscriptions

☐ *add* $4. per year for surface mail

☐ *add* $12. per year for air mail

Payment must accompany order. We do not invoice for individual subscriptions or single copies. Please make your check or money order payable to People's Computer Company. Thank you.

Name _____

Mailing Address _____

_____

City _____ State _____ Zip Code _____

This information may be published in directories and lists of individuals interested in computers in non-commercial environments: ☐ *YES* ☐ *NO*

Please return this form to: PCC, Box 310, Menlo Park CA 94025; (415) 323-3111

DR DOBB'S JOURNAL OF
COMPUTER CALISTHENICS & ORTHODONTIA
PCC
Box 310
Menlo Park CA 94025

TV DAZZLER

SOFTWARE CONTEST

Sponsored by People's Computer Company
P.O. Box 310. Menlo Park, Ca. 94025

FIRST PRIZE: $500 certificate for hardware from CROMEMCO

SECOND PRIZE: $250 certificate for hardware from CROMEMCO

OBJECT: Develop a program resulting in a new and interesting display using the Cromemco TV Dazzler. (The Dazzler is an interface that permits a home color TV set to be a graphic terminal for certain microcomputers.)

RULES:

- All entries must use the Cromemco Dazzler display and must not require more than 20K of computer memory.
- All entries will be judged by People's Computer Company on
  1 – originality
  2 – general user appeal
  3 – clarity of documentation
- Entries should include source code and object code on punched paper tape. A listing of an appropriate bootstrap loader should also be provided.
- Software should be compatible with MITS REV 1 serial I/O port convention for I/O requirements (i.e., data transfer is on port 1, bit 7 [active low] of input port 0 is used to indicate receiver ready, and bit 0 [active low] of input port 0 is used to indicate transmitter empty).

Microcomputers can be incredibly versatile. The Dazzler adds the dimension of full-color graphic display to the microcomputer.

What can you develop? – games? – business? – education? – art? – others?

SEND ALL ENTRIES TO: PEOPLE'S COMPUTER CO
P.O. Box 310
Menlo Park, Ca. 94025

ENTRIES MUST BE RECEIVED BY SEPT. 30, 1976