$1.50

dr. dobb's journal of

COMPUTER Calisthenics & Orthodontia

Running Light Without Overbyte

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 9, October, 1976
Box 310, Menlo Park, CA 94025

Publisher
People's Computer Company
1010 Doyle, Menlo Park, California
(415) 323-3111

Editor
Jim C. Warren, Jr.

Contributing Editors
Marvin Winzenread
Steve Wozniak
Jim Day

Watchdogs
Bob Albrecht
Dennis Allison

Underdogs
Rosehips Malloy
M. Fingers

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:

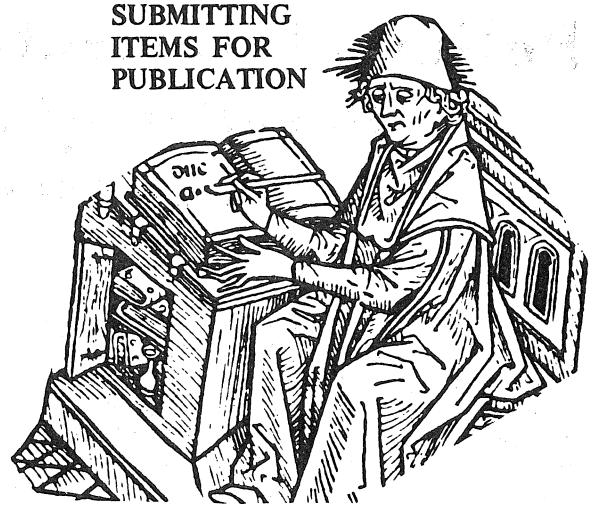
$10 per year.
$1.50 for a single issue.

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 purpose 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.

Policy Statement:

Handling of consumers' complaints regarding vendors' products & services

by the Editor

As our regular readers are aware, we have pursued a course of active consumer advocacy ever since we began publication. In fact, one of the reasons we refuse to carry paid advertising is to maintain the viability of our statements and comments regarding products and services being offered to the computer consumer. After some months of publishing compliments and complaints about vendors' offerings, we have slowly evolved a policy regarding the handling of consumer complaints which we believe will be fair to both the consumer and the vendor, to wit:

We solicit both compliments and complaints concerning vendors, from the hobbyist community. In either case, these comments should be explicit, and contain as much detail as may be appropriate. In the case of complaints, particularly those involving *service,* we will contact the vendor, discuss the complaint with him, and request that an explicit response from him be forthcoming within a reasonably short period of time. If the response is written, we will request that a copy of it be forwarded to the originator of the complaint.

If we receive a response within the time period we specify, and the response appears to us to adequately deal with the complaint, then we will do nothing more about the complaint.

If we fail to receive a response within a reasonable period of time, or the response is vague, ambiguous, or otherwise fails to deal with the specifics of the complaint, then we will probably publish the complaint (and probably the response, as well) in the next available issue of *Dr. Dobb's Journal.*

If we receive complaints about poor design of some item of hardware or software, and it is our engineering and systems judgment that they are valid, then we may publish the complaint without first contacting the vendor. If there is any question about the validity of the complaint, we will contact the vendor or manufacturer as outlined above.

We will take this path of action *only* in the case of *written* complaints that explicitly specify the problems.

In the case of compliments and praise, we will publish it whenever it seems appropriate to do so. It is almost certainly appropriate when we receive multiple complimentary remarks about the same product or service, coming from different parts of the country.

A Note and Warning for Hobbyists & Vendors

HURRAY & BEWARE: EVERYBODY'S JUMPING ON THE CONVENTION BANDWAGON

by Jim C. Warren, Jr., Editor, *Dr. Dobb's Journal*

So far, the computer hobbyist conventions have been entirely run by hobbyists. As such, they have been well-oriented to hobbyist needs and interests and have been really exciting to most of those who attended them. Also as a result, however, they have generally been less organized and more haphazard than is the case with a "professionally run" convention. They haven't furnished as much as they could (and have often been a hair-pulling frustration to exhibitors), but, what they lacked in smooth operation, they more than made up for in exuberence and enthusiasm.

Now, however, some professional convention organizers are starting to get into the act. With a proper attitude, and with close cooperation and leadership from the hobbyists, these "pros" can help make a good convention even better. They can smooth out the rough spots and tie up the loose ends. However, at least a few of them don't appear to give a damn about the hobbyists; they seem interested only in the maximum amount of profit for the minimum effort. In this case, I would say that "pro" stands for promoter with its worse possible connotation.

I became aware of this whole thing at the Atlantic City convention, because I passed out preliminary announcements of the First West Coast Computer Faire to be held in the San Francisco Bay Area, next Spring. As a result, I had half a dozen people contact me to say that they, also, were planning computer hobbyist conventions. I found this both exciting and disturbing. It was exciting to see that the personal computing movement has grown sufficiently that a number of people considered it worth their time and effort to organize a convention in their region. It was disturbing in that (1) I found that almost no one was paying any attention to what anyone else was planning, and (2) I encountered at least a few people who appeared to be interested in it strictly for profit without the slightest interest in benefiting the hobbyists. I think that the former can cause less-than-excellent conventions and gross confusion and irritation on the part of the vendors, and the latter can produce, at best, "cheap shows."

Sooner or later, the vendors will probably put together an association that will maintain a calendar of such events. Until that happens, there are some things that can be done by people considering holding a conference or convention to avoid (or at least know about) calendar conflicts.

1. They can phone me at People's Computer Company, (415) 323-3111. I will be happy to tell them of all the conferences I know of, and will maintain a Calendar of all hobbyist conventions of which I am informed.
2. Periodically, *Dr. Dobb's Journal* will carry a complete listing of that Calendar information. The list that follows is the first such listing.
3. They can contact Smith Dorsey, the Chairman of the ACM's Conferences & Symposia Committee. Smith maintains one of the most complete calendars of computer-related events available, including both ACM and non-ACM meetings. He can be reached at Rockwell International, Dept. 503/034, 3370 Mira Loma Ave., Anaheim, CA 92803, (714) 632-1391.
4. They can check the Events Calendar published each month in the Association for Computing Machinery's *Communications of the ACM,* and the Institute for Electrical and Electronics Engineers (IEEE) Computer Society magazine, *Computer.* This will at least point out possible conflicts with meet-

ings organized by the professional and engineering societies.

There are also some things that can be done to avoid being disappointed by the "cheap shows" (whether you are a hobbyist or a vendor).

1. See whether the hobbyist groups located in the area of an advertised convention are sponsoring or actively participating in the meeting. If they are, you can expect them to be listed in the advertisements. If they are *not* listed, you should assume that they are not involved, and that the "convention" is really little more than a trade show.
2. See whether a significant conference activity is part of the convention, along with the trade show/exhibition. If there is advertisements should note that there will be a number of talks, panel sessions, seminars, etc. If they don't, you should assume that it is primarily a trade show—and expect nothing more than that, if you attend.

COMPUTER HOBBYIST CONVENTIONS ALREADY HELD:

May 2, 1976	Trenton Festival Trenton, NJ Amateur Computer Group of New Jersey	1500 people 45 exhibitors
Jun 11-13, 1976	Midwest Regional Computer Conference Cleveland, OH Midwest Affiliation of Computer Clubs	1500-2500 people
Aug 28-29, 1976	Personal Computing '76 Atlantic City, NJ Southern Counties Amateur Radio Assn. of New Jersey	4500-5000 people; 103 exhibitors

COMPUTER HOBBYIST CONVENTIONS BELIEVED TO BE IN THE WORKS:

Nov 5-7, 1976	Technihobby—USA Commonwealth Pier Boston, MA [Computer & Radio Hobbyists]	Marketing Ventures, Inc. 5012 Herzel Pl. Beltsville, MD 20705 (301) 937-7177
Nov 10-14, 1976	Technihobby—USA Los Angeles Convention Center Los Angeles, CA [Joint Meeting with the Personal Electronics Show]	[same as above]
Nov 19-21, 1976	Technihobby—USA Atlanta Civic Center Atlanta, GA	[same as above]
Dec 3-5, 1976	Technihobby—USA Sheraton Park Hotel Washington, DC	[same as above]
Dec 16, 1976	Western Microcomputer Show Sheraton Anaheim, CA	
Apr 15-17, 1977	The First West Coast Computer Faire Civic Auditorium San Francisco, CA [Expecting 7,000-10,000 people, 50 sessions, 200 exhibitors]	[co-sponsored by a number of Bay Area hobbyist, professional, and educational organizations. See article, elsewhere in this issue]

IMS ANNOUNCES 44-COLUMN PRINTER FOR $399

IMS Associates is offering a 44-column dot-matrix printer for $399 (kit) or $549 (assembled). It prints 75 lines/minute, has automatic wrap-around for overlength lines, can print double-size characters under software control, can print on multicopy paper, and interfaces to a parallel output port using hand-shaking. IMS Assoc., 14860 Wicks Blvd., San Leandro, CA 94577.

June, 1977	Midwest Regional Computer Conference Cleveland, OH	Midwest Affiliation of Computer Clubs P.O. Box 83 Brecksville, OH 44141 (216) 371-9304
Jun 13-16, 1977	Personal Computing Section '77 National Computer Conference Dallas, TX [Expecting 30,000 people, 100 sessions, and over 250 exhibitors]	American Federation of Information Processing Societies 210 Summit Ave Montvale, NJ 07645 (201) 391-9810
Jun 18-19, 1977	Atlanta Computerfest Atlanta, GA [in conjunction with Hamfest]	*Kilobyte* & *'73 Magazine* 73 Pine St. Peterborough, NH 03458 (603) 924-3873
Jul 29-30, 1977	Seattle Computerfest Seattle, WA [in conjunction with Hamfest]	[same as above]
Aug 27-28, 1977	Personal Computing '77 Consumer Trade Show Atlanta City, NJ	PC '77 503 W. N.J. Av. Somers Pt, NJ 08244 (609) 927-6950
Oct 25-28, 1977	(Name unknown at press time) Anaheim Convention Anaheim, CA	*Interface Age* Box 1234 Cerritos, CA 90701 (213) 469-7789
Spring, 1977	Personal Computing (four shows being planned for: Boston, Chicago Philadelphia, and Southern California)	Benwill Publishing Corp. 167 Corey Rd. Brookline, MA (617) 232-5470
Fall, 1977	(Name unknown at press time) Los Angeles Area [Proposal to hold such a convention has been placed before SCCS Board of Directors]	Southern California Computer Society P.O. Box 3123 Los Angeles, CA 90051

Note: This list excludes a number of conventions directed towards computer professionals that are expected to have at least nominal activity in the area of personal and hobby computing. Although the '77 NCC is primarily for computer professionals, its Personal Computing Section will be a major activity with a number of significant sessions and events planned for personal computer enthusiasts.

A CONSUMER'S REPORT ON iCOM'S FLOPPY DISC SUBSYSTEM

Dear Editor, September 1, 1976

I read with interest, "The Time for Floppy's is Just About Now," in *DDJ*, 8/76. I too think a floppy peripheral provides a highly desirable mass storage device, not only for homebrews but for industrial micros and minis as well.

In particular, I would like to provide some answers from my own experiences, to the questions you had regarding the iCOM floppy system. I bought a single drive system in August 1975. At that time they had not yet introduced their Frugal Floppy but they allowed me to purchase just the controller-formatter, model CF360, Mux card (-20) and cable set (-22). They then told me exactly how to buy the mating Pertec drive and mating Adtech power supply, giving me part numbers and names of people to contact. At the same time, they mailed me their Interface Manual and their Schematics and Logic Diagrams to me. By the time all the equipment arrived, I had already purchased the necessary 2 PIA's (Peripheral Interface Adapter) for $15.00 each, connected them to my Motorola M6800 Evaluation Module board, and provided a connector to mate with the formatter-controller. Within about an hour, primarily required to add a fuse and power cord to the power supply and figure out which cables connected where, the system was all connected up and I was trying out the subroutines in my own version of a software driver. All the necessary interconnecting cables and connectors but one had been provided as part of the formatter-controller purchase. The manuals had provided suitable information about this cable so I had it ready and waiting too. (It is now included with the Frugal Floppy.)

Their documentation was thorough and accurate. Of the fourteen floppy commands that can be issued to the formatter-controller, I managed to correctly implement 12 of them in my driver without problems. It took me another week to debug my READ routine and the remaining two commands so that I was reading data and not status. I still had one problem left, but when I called iCOM, Terry Zimmerman and Bob Brusseau informed me how to interpret IBM format, and my problems were over. In addition, Bob went over the entire Logic Diagram and added all the signal names that had been omitted from various portions of the circuit and included two gates that had been added to the PC boards but not the Logic Diagram. During the year since, the Microcomputer/Floppy has been interfaced to a minicomputer as an intelligent peripheral, and it was found desirable to add another disk drive. I ordered another Pertec drive this time from iCOM, with an additional Mux board, and when it arrived and was unpacked, I simply attached it to the connectors that had come with the original system, and was able to use either drive just as easily as using the single drive. Unfortunately, the fuse on the power supply blew out several times. A call to iCOM revealed that we had been too conservative in sizing the fuse initially so we increased its capacity and added the recommended fan to provide cooling and now have an effective, dependable system.

Two additional programmers have "cut their teeth" on this system. One added routines to the core-resident driver including an automatic initial-program-load (IPL) feature. The other implemented a partially disk-resident operating system that makes the 1K of microcomputer RAM memory appear to be about 4K, by using overlay techniques and program chaining.

Throughout it all, the floppy system has always functioned properly. The compatability of the IBM format used was tested as follows. The minicomputer collected digital data, it was passed to the M6800 in ASCII, translated to EBCDIC and recorded on the diskette. The diskette was then loaded into an IBM 3540 peripheral on our IBM 370. The IBM 370 was then able to read and print the data as if the diskette had been recorded in the 3540. Similarly, IBM 3540-recorded diskettes were loaded into iCOM floppy system, read by the M6800, translated to ASCII, and used as data by the driver routines.

Since then, we have ordered a Frugal Floppy and a mating power supply ($250 for the power supply) and are contemplating purchase of a third system. These negotiations were handled by Alan Garfein who provided information and advice not only about iCOM equipment but was most helpful in providing leads to other related equipment. In summary, the equipment has been highly reliable and performed according to specifications, the documentation was well-written and complete, and the personnel were knowledgeable and helpful. We expect to be getting FDOS I, editor, and assembler with the Frugal Floppy and then we will be able to gain experience with iCOM software as well.

Sincerely,

Chuck Felber
Technical Staff
Box 51308
Tulsa, OK 74151

Rockwell International
Tulsa Division
(918) 835-3111

PERTEC (FLOPPY MAKER): BUYS iCOM (CONTROLLER MAKER), MAKES WATS LINE AVAILABLE TO CUSTOMERS

Pertec, a manufacturer of floppy disc drives for some years, has just completed purchase of iCom, the Southern California systems house. iCom has been marketing a floppy disc controller and operating system for Pertec drives since last spring, and has been actively cooperating with computer hobbyists.

Unrelated to this purchase, Pertec has announced installation of WATS-line (toll-free) telephone service for use by its customers and field service personnel. This 800-area code number allows you to call the factory trouble shooters, without cost.

Note: We just phoned WATS Information and found that the Pertec WATS number is (800) 423-5156 . . . but is accessable only *outside* of California.

USED TELETYPES & TERMINALS

National Teletypewriter Corp. (207 Newtown Rd., Plainview, NY 11803, (516) 293-0444 is an ongoing dealer in used computer terminals and a variety of models of teletype.

Some Personal Observations on *Personal Computing '76*

Jim C. Warren, Jr., Editor

Through the prodigious efforts of John Dilks, Davy Jones, and Jim Main (and with considerable help, guidance and support from Carl Helmers and *BYTE* Magazine), the August 27th-28th Personal Computing '76 Consumer Trade Fair was a *great* turn-on! So much was going on that I had only three to four hours of sleep per night, and even then, I missed some gatherings I would like to have attended. Details of particular note:

Statistics, first: About 4,500 computer enthusiasts and curiosity seekers attended the two-day convention. There were slightly over 100 exhibitors, and about 50 talks, panel sessions, and seminars. The banquet on Saturday night turned out over 360 people (in spite of its hefty $15 price tag).

We blew it. Contrary to the "reliable rumor" we published in our August issue, **Heathkit** was *not* one of the exhibitors. We understand, however, that Heath purchased a block of 16 tickets to the convention. Lou Frenzel, their Director of Computer Products was there. And, he reportedly was overheard saying that Heath would have a kit out "before Spring." So much for rumors mongered.

Idle gossip had it that **MOS Technology** was being bought by Commodore Calculators. (So what? They will undoubtedly continue to turn out hot products.)

I am delighted to publish another correction to information in our August issue: We noted (incorrectly) that the fine print of the literature concerning **Matrox** VRAM's (Video RAM) indicated that a minimum order was 100 units. Their rep, Lorne Trottier, dropped by to say "hi," and told us that the *stock* items we had listed in that August article *are* available in unit quantity for the prices we quoted. The 100-unit minimum pricing referred only to custom-built units. Therefore, take another close look at 'em. They have a 256 x 256 black-and-white graphics display for $630, as well as some interesting 64 and 80-character alphanumeric VRAM's (ASCII in, video out). That's Matrox Electronic Systems, Box 56, Ahuntsic Stn., Montreal, Quebec H31 3N5, (514) 481-6838.

I am also delighted to report a correction to comments contained in a letter that we published in our June/July issue. In that letter, one of our readers stated that **MITS Computer Notes** would only be available by subscription after the first of the year. Dave Bunnell the publisher of *Computer Notes*, announced at the banquet that this was incorrect (explicitly referencing the letter) and indicated that *Computer Notes* subscriptions would be available without cost, permanently. We applaud this fair-minded and sensible policy and apologize for publishing inaccurate information [note the Policy Statement on Complaints, published elsewhere in this issue.]

National Semiconductor was present and told me that they now have **Tiny BASIC** for their SC/MP microprocessor. It is available in paper tape form or in 3K of PROM. Incidentally, their SC/MP kit with 256 bytes of RAM, and 512 bytes of ROM including their KITBUG monitor is available for $99. In less than four months, Hash Patil said they had sold over 8,000 of these kits.

Someone mentioned this interesting tidbit: **Alexander Graham Bell** was not a technical type. Rather, he was an artist. In fact, he was a member of the National Academy of Design which, at that time, was a very prestigious national society.

I should mention **Leda Alpert**, the woman who put together the super low-cost travel package that helped a number of west coasters (including myself) afford to attend the convention. She almost worked her Southern California tan to the bone putting that mess together. At least in my case, her agency (Travel Co-Ordinators, Inc., Los Angeles) probably lost more than it made. We must have spent half a dozen long-distance phone calls, haggling over my eccentric travel desires. But ultimately, she put it together. Furthermore, she was at the show to troubleshoot travel problems for her clients . . . which was helpful to a coupla people.

I had the good fortune to meet **Neil Colvin**, the software type with **Technical Design Labs** of Trenton, NJ. Keep an eye on that crowd; they are doing some great things with their Z-80 system. Among other things, Neil said they already have a macro assembler modeled after DEC's MACRO-10 and using MACRO-11 syntax that produces relocatable code. They have a loader that accepts relocatable load modules. They have a BASIC that runs in 6.5K bytes and accepts Altair BASIC source code. They have a FORTRAN compiler very near completion (aww shucks; I thought that dinosaur was dead). And, at the time I spoke with him, he had just finished transporting a portable PASCAL compiler over to the Z-80. At that time, it took 32K bytes of grossly inefficient code (as portable software is prone to do), and Neil was starting to go through it and optimize it. The PASCAL compiler generates PCODE as output, which requites an interpreter (portable PASCAL had much the flavor of "portable" Tiny BASIC). Neil had managed to crowd the PCODE interpreter in one kilobyte on the Z-80. Also, he said they are working on a "word processor."

At the head table at the banquet, I was delighted to find myself sitting next to **John Whitney**, the computer graphics artist of international reputation, who was the last speaker at the banquet. As usual, he closed it with an absolutely fascinating exhibition of some of his computer-generated films. He is a most fascinating man, for his ideas, his art, and his exuberant and vibrant outlook. If you ever get a chance to hear him speak, or see his films, don't miss it! Incidentally, he travels so much that he has two watches on his watchband; one gives the local time, and the other gives "home" time for the west coast.

I saw the first issue of the much-advertised *Microtrek* magazine. It had 64 interior pages, of which less than nine were advertising. Almost five pages were devoted to "Just Getting into Microprocessors? Odds Are, You'll End Up With a Kit," reprinted from *Electronic Design*. The rest included: a ten-page article on the "KIM-1 Microcomputer Module," five pages devoted to a "Super-Simple Cassette Interface," Part I of an "Introduction to Assembly Language Translation" (eight pages including a glossary), almost nine pages concerning "Better Programming Though Flowcharting and Documentation," four pages of details for an M6800 "Software Leader," and a nine-page "Buyer's Report: The IMSAI 8080 Computer Kit." I got the impression that it was planning on being a "general content" magazine, just like *Byte*, but possibly with a

lower level of expertise expected of its readers. However, when I identified myself to the Editor and asked him what the focus of *Microtrek* would be, he said it was going to concentrate of software. Furthermore, some editorial notes in the first issue state that he is especially seeking articles on medical applications, display applications, floppy disc systems, computer control of electromechanical devices, pattern recognition systems, bit-sliced computers, and APL. So, I don't know where *Microtrek* is headed, but I am interested in seeing the future issues.

Some interesting tidbits gleaned from Chad Harris, Club and Training Manager for the **American Radio Relay League (ARRL)**: There are 281,000 ham operators, right now, and they expect 30,000 more by the end of the year. Their average investment in ham gear exceeds $1K. Only about 1% built *all* of their own gear; the rest built kits or bought turnkey systems. And . . . Chad estimated that approximately ¼th of the hams are seriously interested in amateur computing; that's more than 70,000 potential computer phreaques!

Quoth one of the banquet speakers, alluding to computer foul-ups: "Failure to receive this bill is not an excuse for nonpayment."

A surprise award for Computer Hobbyist of the Year was given to **Saul Libes**, the leader of the Amateur Computer Group of New Jersey and one of the major organizers of personal computing activities on the East Coast.

Someone had a little booth in the Trade Fair portion of the convention and was selling **digital clock kits.** By Sunday afternoon, he had sold out, packed, and gone home.

IBM was/wasn't an exhibitor. (The following story came from John Dilks and is retold with his permission). In the early planning stages of PC '76, John contacted IBM and asked them if they were interested in being an exhibitor. Initially, he received no response. Then, IBM decided they were interested, sent in their exhibitor registration and fee, and indicated they would be there. So, as part of the promotion of the convention in some of the publicity releases, John (reasonably) mentioned IBM along with a number of others as being expected exhibitors. Shortly before the convention, IBM called up and said they wanted to cancel out. John said, "No way." IBM said he could keep the registration fee and rent the booth space to someone else. John refused. Then IBM wanted to get another vendor, on their own, and let him take the space. At that point, John told 'em that, no matter what IBM did, they were going to have the booth they had committed for, and it was going to have an IBM sign on it, whether they were there or not. Well, IBM showed up, refused to take the booth, but set up some of their gear in one of the hotel suites and placed a directional sign in the booth (strange people, those maxi makers). In one of the final verbal exchanges between John and an IBM rep, Dilks said, "You know, you people have your heads in the clouds watching for the money to blow by. You ought to look down on the ground and find the bucks laying at your feet." [Quote verified by John.]

Bob Jones, who has been the publisher of *SCCS Interface,* has chosen to go his own way. He was distributing the August issue of his new publication, called **Interface Age.** [Incidentally, the last we heard, SCCS is planning on continuing to put out their own publication which will be called *SCCS Interface.*

I spent a pleasant evening with Ron Davies and Roald Evensen, two of the principals in Marketing Ventures, Inc., which is putting together the **TechniHobby** shows [see "Hurray & Beware . . .", another article in this issue]. I was favorably impressed by them. They were low key, "up front," and seemed to have some good ideas. Basically, they were planning on their Technihobby shows being trade fairs covering not only computer hobbyists, but also ham radio and radio-controlled models. In this way, they expect to encourage considerable "cross-pollenation," particularly from the ham crowd into computery. They indicated that they were planning on cooperating with local hobby groups. They also had some innovative ideas for assisting small vendors who couldn't afford to travel all over the nation exhibiting their wares. They also publish an excellent trade newspaper, **Mid-Atlantic Electronics.** I got around to reading it on the way back to the West Coast and found that I tore out almost half the pages to keep in my technical reference files. They are at 5012 Herzel Pl., Beltsville, MD 20705.

I heard that the **National Security Agency** has an in-house club of computer amateurs that has over 100 members.

The **Chesapeake Microcomputer Club** appears to be an East Coast counterpart of the Southern California Computer Society. It already has Chapters in Baltimore, Columbia, Wheaton, Washington, McLean, Reston, and Richmond.

Wayne Green and John Craig were there with *73* Magazine and their new computer hobby mag, **Kilobyte.** [See John Craig's article elsewhere in this issue.]

Votrax was another exhibitor well worth noting. They have been making speech synthesis systems for several years. Last Spring, we contacted them and encouraged them to enter the hobbyist market with a low-cost speech-synthesis kit. I am delighted to report that they were handing out a press release detailing such a kit for $750, with first deliveries planned for early 1977. [See February, March, May and September issues of *DDJ* for more details.]

I heard some preliminary but very exciting comments about a general-purpose **string-processing language** that is about to become available, based on Strachey's GPM and McIlroy's M6.

Ted Nelson was there with his **Itty Bitty Machine** Company of Chicago, and 7'-high stacks of IMSAI's as a backdrop. We happened to share the same tables for several meals . . . a most interesting guy. He told me something of his progress on **Xanadu,** a very exotic electronic library and information facility (described in one part of his book, *Computer Lib*); a fascinating project that appears to be nearing reality. If you haven't head about it, better look it up. You are likely to be hearing about it before long. He is making heavy use of Calvin Mooers' TRAC (licensed, proprietary name) language, and expects to market the sortware for $100 (with, of course, an agreed upon royalty being paid to Mooers).

Carl Helmers gave me this hot news item for *DDJ* (and I took it, hook, line, and sinker): The Amalgamated Clothing Workers Union is considering unionizing programmers. They have decided they ought to do so since they heard that programmers make software. [*Byte* is too proud to publish it, but we're jus' plain folks, out here.]

And with that . . . I better quit.

San Francisco Bay Area finally gets it together

First West Coast Computer Faire

Jim Warren, Editor, *Dr. Dobb's Journal*

The first microprocessor was invented in the San Francisco Bay Area. One of the first computer clubs to be formed is located in the Bay Area. The Bay Area has the second largest such club in the U.S. The San Francisco Peninsula has a greater concentration of semiconductor and microprocessor manufacturers than any other area in the Western Hemisphere.

Yet, it has never had a conference or convention concerned with personal and home computers, in spite of the fact that three such conventions have met with great success and enthusiasm in the East and Midwest. Well, the San Francisco crowd has finally gotten it's act together. A convention of major proportions is scheduled for next April 15th through 17th, 1977.

The convention—called The First West Coast Computer Faire—will be held in the largest convention facility in Northern California, the Civic Auditorium in San Francisco. The two-and-a-half day event will include a Conference that will offer 50 to 100 different technical and tutorial sessions, several banquets and luncheons with major speakers, and an Exhibition that should have over 200 exhibits. Based on the location, the amount of planning and organization, and the experience with the other hobbyist conventions, 7,000 to 10,000 people are expected to attend.

The Computer Faire is being co-sponsored by a number of computer groups and educational organizations including:
The Homebrew Computer Club
The Association for Computing Machinery, Peninsula Chapter
Stanford University's Electrical Engineering Department
People's Computer Company
The Community Computer Center
The Amateur Research Center
and a number of other organizations are considering becoming co-sponsors as this article is being written.

The Conference portion of the Faire will include everything from formal papers and "heavy talks," through informal "shirtsleeve raps." It will also include a number of panel discussions, working sessions, and presentations of homebrewed computer projects. Though planning is still in its early stages, several exciting Conference Sections are already taking shape, some of which are "firsts" for the personal computing movement.

One such Section is that concerned with Personal Computers for Education. It will focus on educational uses of computers in the home and elementary and secondary school, as well as in one-computer/one-user situations in colleges and universities.

Another "first" is the Conference Section on Microprogrammable Microprocessors for Hobbyists. Presentations in this Section will include designs and kits for hobbyists who wish to have systems in which they can "customize" their processor's instruction set.

Sections are also being planned for Computer Music, Computer Graphics and Animation, Personal Computing for the Handicapped, and many other exciting subject areas.

There will be a number of panel sessions and working groups concerned with hardware and software standards. These will include Digital Cassette Tape Standards, Bus Design Standards, Peripherals Interface Standards, Software Modularization for Program Portability, and so on.

There will also be information exchange sessions for Club Officers, Newsletter Editors, and Conference Organizers. Additionally, several "social centers" will be set up around the auditorium for sittin', sippin', and gossipin'.

An exhibition of homebrewed projects and systems is also planned. Assuming that there are sufficient entires to make it interesting, these projects will be judged, and prizes and awards will be presented for the best systems. Nominal Grants-in-Aid will be made available on a refereed basis to some hobbyists living a significant distance from San Francisco who would like the opportunity to exhibit their homebrewed projects. These Grants will be to assist with the expense of shipping the systems to be exhibited.

Most of the banquet plans have yet to be solidified, however, one banquet speaker has already been chosen. He is Professor Henry Tropp who has spent almost three years as the Principal Investigator in the Computer History Project jointly sponsored by the Smithsonian Institution and the American Federation of Information Processing Societies. In this capacity, Professor Tropp has traveled quite broadly, conducting in-depth interviews with most of the "old timers" who created and invented the computer science and technology that is the basis for today's computers. Dr. Tropp is a quite fascinating speaker, and will be relating a number of stories concerning the personalities, foibles, mistakes, accidents, and eccentric technology of those early days of computers.

The Exhibition portion of the Faire will include exhibits by almost every manufacturer and vendor who is in any way interested in the personal computing community. These vendors have shown great enthusiasm for the Faire; many committed to being exhibitors as soon as they heard about it. In the first three afternoons of contacting possible exhibitors, the following have already indicated they expect to participate:

National Semiconductor
American Microsystems Inc. (AMI)
MOS Technology
Zilog
Southwest Texas Products
Apple Computers
Technical Design Labs
Polymorphic Systems
Cromemco
OSI
Microcomputer Associates
STM Systems
Quay
Project Support Engineering
Computer Converser
Shugart Associates
iCom
Action Audio Electronics

Byte
Interface Age
Dr. Dobb's Journal
People's Computer Co.
Kilobyte
73 Magazine
Microcomputer Digest
Osborne & Associates
Processor Technology
Solid State Music
National Multiplex
Monolithic Systems
Percom
Call Computer
Associated Electronics Co.
Byte, Inc.
CompuMart, Inc.

and that's just scratching the surface.

(Continued on next page.)

PAPERS AND PARTICIPANTS ARE BEING SOUGHT

If you are interested in participating in this Computer Faire, as:

- A Speaker
- A Panel Session Leader or Participant
- A Section Chairperson
- An Exhibitor of a Homebrewed System (hard or soft)
- An Exhibitor of Commerical Products
- An Assistant in the Administration and Coordination of the Faire
- Or Whatever

Or, if you wish to suggest something or someone for inclusion in the Faire:

- Topics for Panel Sessions or Talks
- Possible Speakers or Panel Leaders
- Interesting Exhibits (Homebrewed or Commercial)
- Special Activities (Tours, Meetings, etc.)

Or, if you have questions or desire additional information, then please contact:

Jim Warren, Faire Chairperson
[Editor, *Dr. Dobb's Journal* & Vice Chairman, Peninsula ACM Chapter]
c/o People's Computer Company
P.O. Box 310
Menlo Park, CA 94025
(415) 323-3111; 851-7664

[or]

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

FLOATING POINT PROCESSOR FOR 8080's

A PC card is available from North Star Computers, Inc., 2465—4th St., Berkeley, CA 94710, (415) 549-0858, that provides floating point functions to the 8080. It provides for addition, subtraction, multiplication, and division of BCD-formatted floating point operands with 14-digit precision. The subsystem is available for $499.

NEW CHIPS OFF THE OLD SUBSTRATE

by Jim Day

The September 2, 1976 issue of *Electronics* describes two new peripheral support chips that may prove useful to hobbyists.

On page 133 is a description of the MN3020 8-bit D/A converter chip from Micro Networks Corp., 324 Clark Street, Worcester, MA 01606. The 18-pin DIP includes an internal reference and an amplifier for the analog output. Priced at $30, the unit is TTL compatable and should be useful for music and speech synthesis applications.

Page 151 describes a new character generator chip from National Semiconductor Corp., 2900 Semiconductor Drive, Santa Clara, CA 95051. Designated the DM8678, the 16-pin DIP includes all circuits needed for parallel-to-series shifting, address latching, and character and line spacing. Priced at about $15 in lots of 100, the bipolar device could simplify the design of video display boards used by hobbyists.

HEATHKIT MARKETS VIDEO GAMES KIT

Heath Co. [Benton Harbor, MI] is starting to peddle a video game kit for $49.95 (mail order) to $54.95. It is designed to attach to the video and sound inputs of any of Heath's tee-vee kits, thus bypassing the FCC regulation quagmire. It also uses the television's power supply for its power source.

INSIDE DOPE FROM MOTOROLA

We announced the "reliable rumor" of a Z-80 competitor being planned by Motorola, the 6809, at the July 21st Homebrew Computer Club meeting. Afterwards, a Motorola engineer rushed over and, after an unsuccessful query concerning our source of information, volunteered to "tell all." Here is his information as of August 6th. *All of it is* tentative.

A **BASIC compiler** for the 6800 will be introduced in the 3rd Quarter of this year. It will be resident in the 6800 Exer Ciser, will be a legitimate *compiler* (*not* an interpreter), will take 4K to 6K for the compiler plus 2K—minimum—for the user. It will handle scalers, one and two-dimensional arrays (but no matrix operations), and will include such command facilities as IF-THEN-GOTO, GOSUB/RETURN, DATA, READ, RESTORE, INPUT, and PRINT. It will have the usual arithmetic functions and will allow linkage to user-generated object code. Features that will be added at a later date include matrix operations, string manipulation, and graphics. It is being developed at Ohio State.

Motorola's forthcoming *high performance microprocessor* will probably be labeled the 6809, although there was some in-house discussion about calling it the 6900. The architecture has been defined and has been just about finalized. The instruction set will be a superset of that of the 6800. It will have additional registers, including one or 2 more index registers. It will have additional addressing modes including indirect and "true indexing" with the base in the operand and the offset in a register. It will probably be an N-Channel device. It is expected to be available in the 2nd Quarter of 1977.

Lots of other tidbits are forthcoming from Motorola, however:

Note that the 6800 uses 197 of the 256 available op codes. The **6801** will have an on-chip clock. Sampling of the 6801 is expected in September or October.

The **6800A** is expected to be available in a coupla months. It will run at 1.5 MHz or better and will be a depletion-load device.

The **6802** will be available in the 1st or 2nd Quarter of 1977. It will be more of a "controller" than the 6800. Its instructions will be more oriented towards bit manipulation.

In October of this year, Motorola will introduce a **new 6800 kit** for about $200. It will be a two piece kit, will include a 7-segment hex display, hex keyboard, a 1K JBUG Monitor (the "J" stands for "Jim") that is located at the top of memory, and a Kansas City "standard" audio cassette interface.

RAW RUMOR: AN 8085 FROM INTEL?

The Silicon Gulch rumor mill grinds onward: someone announced at the September 15th Homebrew Computer Club meeting that Intel will shortly release the 8085. It is supposed to be a microprocessor with on-board clock and I/O, 12 new instructions, and requires only a single +5 volt power supply.

AN EXCELLENT, POINT-BY-POINT RESPONSE TO COMPLAINTS ABOUT THE DIGITAL GROUP

Dear Mr. Warren, August 24, 1976

I would like to take this opportunity to thank you for the kind words and your support in *Dr. Dobb's Journal*. As you know, Dr. Suding and the Digital Group have tried to be equally supportive of your efforts and of course will continue to be so.

In your August issue, you raised some very valid points about what we are trying to do and how well/not-well we are succeeding. Your assumptions were partially valid on the current situation and I would like to address them one-by-one. [The following three quotes were editorial comments published in the August article.]

1. "Customer interest in Digital Group products has exceeded their initial capacity to respond in a manner they would like."—obviously true.

2. "They are aware that they have a problem and are doing everything they can to expand their customer service department and procedures."—boy, are we aware of the problem!
3. "It is their wish to maintain an operation of unusually high integrity and responsiveness to their customers."—Exactly.

Agreeing with you is OK as far as it goes, but what are we DOING? Here are the actions we have taken to rectify the situation.

1. We have expanded our facility from 100 square feet to over 7000.
2. Our number of full time people has expanded from 6 to 16 plus part time as needed.
3. We have quintupled our investment in inventory (which was not trivial to begin with).
4. Of the new personnel, 2 went directly to customer service on a full time basis, effectively tripling our capability there.
5. We are in the process of automating our operation (a long overdue step) so that we can respond to order status more appropriately and do such things as acknowledge receipt of an order, keep better track of any shortages, and keep people informed if their order hits any unexpected snags.

All of these items have been implemented currently and you should begin to see their results soon. We are doing everything we possibly can to "dig out from under" and expect to be in very good shape soon.

Next item: You did print a couple of complaints about the servicing we were providing a couple of months ago. I find no objection whatsoever against complaints when they are on target, but I wish that just once a manufacturer would be given a chance to respond *before* you go to press—not censor, as that would impune your integrity—but at least a chance. **[We have. See editorial in this issue.]** The manufacturer is always automatically cast as the heavy against the little guy and does not dare respond. Please pick up the telephone and call me directly (collect, of course) so at least I can look into the problem and fix the situation. You would then be doing everyone a service by getting action and would still be able to print whatever you felt was appropriate. **[A reasonable request. Herewith adopted as *Journal* policy.—JCW]**

On the two items [Page 37, August issue]:

Christopher Hovey. Our turnaround time on servicing is now down to less than a week and rarely, if ever, has exceeded 2 weeks. Of course, Mr. Hovey is at an APO address and normal one-way shipment takes 3 weeks or longer. Amazingly enough, he's been waiting a full month for his equipment to go by boat to wherever overseas he is stationed. The latest information we have is that he is indeed up and running. **[My error. I should have spotted the APO address and noted that a one-month turn-around was reasonable.—JCW]**

Robert Gerald. This one's a pip. Yes, he is quite right that we got some mis-labeled parts from Motorola that did indeed generate upside-down characters. About 10 actually got out our door, and to my knowledge, all have been replaced in most cases before the kit builder even had his kit built. However, his complaint is misleading concerning the replacement of his chip. We shipped him a replacement the day after he placed his call, irrespective of Gus's comment (which was not true and very ill-advised—one of his bad days obviously). You will note that the date on Mr. Gerald's letter is the very next day. Now, the post office is doing pretty well, but not that well. Ah, but there's more. You will notice that after our repeated requests to send in his board, Mr. Gerald finally sent it in with a cover letter on July 31 **[Copies of correspondence were enclosed.—JCW]**. I've also included a copy of our service record for the unit and what we discovered—misplaced part, destroyed character generator, and a solder bridge. We returned his kit on August 12, up and running, with no charge. If that's "abandoning any kind of customer service to previous purchasers," I guess there's nothing more I can say. **[We agree.—JCW]**

Well, I hope I've made my point. As usual, there are 2 sides to every situation, and all I can hope for is that you might be able to do a little checking on what's really behind the letter before printing.

Sincerely,

Richard C. Bemis, President
The Digital Group, Box 6528, Denver, CO 80206
(303) 861-1886

Editor's Note: Mr. Bemis also included some notes about a delay in shipment of DG's Z-80 kits. The problem has been cleared up (as of September 6) and they expect to fill all Z-80 orders within three weeks of receipt.—JCW

ELECTRICALLY ERASABLE ROM'S

General Instruments is rumored to be planning announcement of ROM that can be erased simply by applying a short blast of voltage to appropriate pins. Reading is performed at usual voltage levels. Writing or erasing is done at 28 volts; 10-20 ms for writing and 100 ms for erase. There are to be two models of the ROM (EROM?, EPROM?, non-volatile RAM?): the ER2800 will have 8K bits and 2 mics. read cycle, while the ER3400 will have 4K bits and an access time of 650 ns.

A Consumer Reports

ASSEMBLING AN IMSAI MICROCOMPUTER

by Pierre duPont V

When I left for my first year at boarding school last September, my big interest was Spanish, and I had visions of a language major. However on my first day at school, I was introduced to a PDP8/e computer. Immediately I took interest and got to know the appropriate teachers and students. After a year of intensive training with a computer hobbyist who was never around when I wanted to learn, I received a key to the computer room, and a key to the computer itself. I now have visions of an electrical engineering major, and can't wait to complete my requirements in Spanish this year.

Just as school let out I became convinced that I should purchase a microcomputer. In about a week I had placed several orders, cancelled several, and ended up with an ALTAIR 680. It arrived on time (strange!), and I assembled it without difficulty. Just as I finished, I left for Denmark for some sleep. When I returned and had about one day's use of my 680, I went to the Personal Computing Convention in Atlantic City. When I arrived at the convention, my teaching friend began to think about trading his 680 in for an IMSAI. As it turned out, we both traded our VW 680's for Cadillac IMSAI's.

I didn't start assembling mine for about a week, but when I did, I had a couple of scares. The instructions were not as easy to understand as ALTAIR's. I had to remove the same IC socket twice, and nearly ruined the circuit board. It was a socket for a jumper to the main processor board, and there wasn't even supposed to be a socket there. It is assumed in the assembly manual that no sockets other than those given (3) were to be used. There was also a space that looked like a place for an IC, but nothing was to be installed there. I don't see why they don't mention the sockets in more detail, but they aren't too difficult to understand and install.

I nearly ruined the entire power supply when I installed four diodes in the wrong position. I put them in as required, but somehow I knew it wasn't right. I called the factory and found out that the updated page had been left out of my manual, and that they were sorry. Well, the power supply works fine now, although it supplies a little too much.

I finished building it in two days of slow work, and then proceeded to *wait.* I'm still waiting. It's been 1½ weeks, and it's killing me. I'm very disappointed because IMSAI supplies no memory with their basic kit. I have ordered 8K of RAM from a small discount store (8K for $180—worth the wait), and a video interface board from Polymorphic. Of course neither has arrived and I can't test my dream computer.

While waiting I've placed several more orders. I purchased a cassette interface, and expander board, several 100 pin connectors, and a cooling fan. I also hope to persuade the relatives to give me an advancement on my Christmas presents, and give me another 8K board.

So far I have no complaints with IMSAI, except that it comes with no memory. I realize that this is because their computer is meant for small businesses who may purchase 32K at one time, and have no use for 1K of on board memory that would clutter things up. The entire kit seems to be this way—the assembly instructions are not really meant for a small time hobbyist like me, and the computer itself is built like a tank. I do think that IMSAI makes a great kit, and offer praise to them for their relatively inexpensive tank-like kit.

In the next year or so I hope to expand to about 20K of RAM. I also plan to get 8K and 4K BASIC along with IMSAI's self-contained system on PROM. After that comes a hardcopy printer and several more cassettes for a Tarbell interface board.

The IMSAI kit is great and I recommend it to anyone who wants a microcomputer, but doesn't want to wait for an ALTAIR. Being relatively new like they are, IMSAI has quite a kit.

Pierre duPont V
Salisbury School
Salisbury, CN 06068

CONSUMER NOTES ON DIGITAL GROUP & TINY BASIC

Dear Jim, Sept. 7, 1976

Since I wrote you last I have received an 8K board, Tiny BASIC and a C-104 superscope cassette player from Digital Group.

4/30/76 Order placed
5/10/76 Tiny BASIC and game tapes received
6/1/76 8K and Cassette deck received

UPS messed DG up by going on strike, so I reminded DG about my order a week after UPS went back to work.

The 8K board went together in 3½ hours including checkout and installation. Worked perfectly.

The "Tiny BASIC" (TB) tape was defective and they sent me another. Their soft ware is a little sloppy but it is very inexpensive and does work, $5.00 for TB.

The C-104 works great but I'm sure it's possible to buy it cheaper someplace else. I wanted it in a hurry. A good feature is its variable speed control. It allows a person to use tapes recorded on different players even if their speed was a little off. The tapes from DG have been running a little slow on my system.

My plans are to work on SCELBI's "SCELBAL" and 8080 Assembler programs. Tiny BASIC felt restrictive as soonaas I tried to apply it in physics problems. Integer arithmetic requires scaling.

Oh, by the way, the biggest disadvantage of DG's TB is they send no listing or include no software tools for modifying it. Fortunately they tell you how to use it. No listings are sent with their games. TB was set up for TTY and not for 32 column TV, a disadvantage.

Keep up the good work.

Yours,

Ed C. Epp
Educational Computer Laboratories
748 S. Main St.
Freeman, SD 57029

SOLAR ENERGY COSTS DROP 50% IN 18 MONTHS

Remember how the "nuclear people" were predicting, in last spring's California ballot initiative battle, that solar energy would remain uneconomical for the rest of the 20th Century? Well, the September 2nd issue of *Electronics* magazine carries a report of an announcement by ERDA (Energy Research and Development Administration) officials, stating that costs of photo cells designed for solar energy use had dropped from $30/watt to $15/watt within the last year and a half.

[And: Photocells aren't even the best way to obtain reliable electrical power from the sun. There are some excellent solar-thermal alternatives, requiring no significant technological breakthroughs that can provide 24-hour electrical power without requiring electrical sotrage for night-time service.]

PCC Research Engineer

Evaluates the 16-Bit PACER Kit

by John Snell

(Reprinted from PCC)

Editor's Note: John Snell is an unusually competant electronics engineer who joined the PCC staff some months ago. Prior to that time, he was working in the "real world" (or unreal world; it depends on your viewpoint) as an EE in the San Francisco Peninsula electronics industry. He found such work too time-consuming, keeping him from more important work, namely research into the design and implementation of very esoteric computer-music systems. He also needed more free time to work with some associates on their exotic video/lazer/lightshow systems.

So, PCC agreed to hire him for our usual paltry pittance, so he could do his research and keep our electronic widgits working. The PACER kit is one of those widgits, recently received. Incidentally, John is a graduate of Carnegie-Mellon University, one of the top computer engineering schools in the nation.

I recently put to gether a Pacer 3H microcomputer development system kit, made by Project Support Engineering. It has by far the easiest to use operational and debug functions of any computer kit that I have used. It has a NOVUS type (no tactile feedback) calculator keyboard built into the front panel. When you want to deposit a number such as 7532, you hit the 7, 5, 3, 2 keys on the keyboard instead of binary switches 1,1,1, 1,0,1, 0,1,1, 0,1,0. The Pacer uses the hexadecimal number system for keyboard input and display. If you do not know hexadecimal, the keyboard will convert from decimal to hexadecimal as well as perform hexadecimal arithmetic. Addresses are displayed to the left and data contents to the right in alphanumeric LED's. Using the keyboard you may examine and/or modify not only memory locations but also:

- 4 accumulators X 16 bits,
- a 16 bit program counter,
- a 16 bit status register,
- a 10 word X 16 bit stack,
- a 16 bit value register used for scanning memory to find the address at which a value is located,
- a 16 bit mask (for use if you are not certain about some of the bits or digits in the value),
- and 10 break point registers, 16 bits each.

The keyboard will also allow you to sequentially increment or decrement through memory or internal accumulators and registers for examination or modification of contents. Other front panel buttons include run, initialize (reset), restart (halt CPU but do not reset), and cancel last command.

The kit includes all parts one needs to have a working desk top microcomputer development system. The CPU board includes the PACE 16 bit MPU with necessary input and output buffers. On the control and I/O boards are two DM8531 (2038 x 8 each) ROMs for the system monitor. Also included on these boards are four MM2112 (256 x 4) static RAMs, one MM5740 keyboard encoder, two hex latches and LED driver circuits as well as all required support components to interface with the two 4 digit displays and 32 keypad. The control board has space for four more MM2112 RAMs. The memory board comes with four MM2112 RAMs. Space is provided for 12 more MM2112 RAMS and four MM5204 (512 x 8) PROMs for future memory expansion. The PAC II card has 2K x 16 of MOS RAM. The PAC I, PAC II, and PAC III cards are optional cards. PAC III is a prototyping card with voltage regulators. All other boards also have their own on board voltage regulators. PAC I is a TTY (or RS232C) interface and resident assembler card. With PAC I the user may perform all the front panel functions from a teletype (or similar device using current loop or RS232C) as well as the following useful functions:

- Load or punch a paper tape – no bootstrap need be loaded.
- Display a block of memory in one of several formats including assembly language (yes a dis-assembler! - very useful), ASC II, hexadecimal, unsigned decimal, or signed decimal.
- Set, list, or reset break or snap points (Break points are placed at strategic locations in a program. They halt execution and display the contents of specified registers and memory locations. Snap points do the same except program execution is not halted.)
- Enter programs in assembly language format (the assembler converts your programs line by line as you type them, to hexadecimal. No paper tape or cassette need be used for this. The assembler and other features listed here reside in two EA4900 type ROMs which hold 16K bits each.)
- Use symbols; the assembler does all address assignment and referencing. (One may also list the symbol table, delete a symbol or clear the table.)

In addition to the resident ROM memory in PAC I space is provided for twelve MM5204 (512 x 8) PROMs, for future user resident programs.

So what else comes standard with the basic Pacer kit? A case, keyboard, push buttons, alphanumeric LED's, all interface logic for keyboard and LED's, a power supply, a mother board (which reduces interconnect wiring and makes expansion easy), cables, connectors and a fan. Space is provided on the mother-board for 8 extra PC boards for future expansion. The power supply included in the kit will put out 8 amps total. If you used mostly CMOS on optional PC boards, the power supply would probably allow you to fill all the available space on the motherboard. P.S.E.[1] says the supply will power the CPU board, control board, front panel and logic cards, standard memory board, plus a TTY/line assembler (PAC I) and two RAM (2K x 16) memory boards (PAC II).

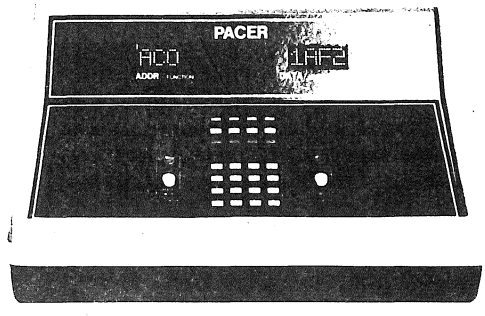
The assembly of the Pacer 3H was easy and quick. I had only a couple of minor problems. I forgot to connect the fan power lines at the right step of assembly. So I had to remove some heat shrink, solder wires, and then put new heat shrink on. Really no big deal, but I don't have a heat gun or hair dryer. Shrinking heat shrink with a match near the plastic case was somewhat of a hassel. I really should get a heat gun. They are also great for troubleshooting circuits with temperature sensitive problems.

There is a thin flexible cable that runs from the keyboard to the logic. This looked fragile, so I took my time installing it in its connector. Once installed however it will never be moved or flexed again, so this is really nothing to complain about.

The Pacer worked perfectly the first time I turned on the power. As I played with it, I began to appreciate the beauty of its high level front panel operational and debug capabilities.

Now I wanted to try our teletype with it. I quickly wired up our TTY to the connector and plugged in a PAC I pc board (TTY interface/resident assembler). The TTY would not work – oops, I neglected to ground the TTY select (low = select) pin on the connector. Once I did this everything worked perfectly, and I enjoyed exploring the fine operational capabilities of the unit.

Available soon from P.S.E. will be a PROM burning board, and an audio cassette interface, a CRT character generator and interface, a floppy disk interface, and BASIC (the debugged program burned into PROMs) as well as other programs in firmware. Since the Pace shares instructions with the IMP-16 (minor modification of programs might be needed), there is a lot of software already available. The *Bit Bucket* [2] newsletter is the best source of PACE and IMP-16 software. Program listings are free, source tapes $5, object tapes $3.



Overall I very much like the Pacer. I wish sockets had been provided for all

PACER continued

the IC's and a heftier power supply had been used, however, these additions would of course increase the cost. The front panel operation and debug capabilities are the best I have seen on any commercial computer kit. I have not used any PAC II operational memory cards yet; so I can't evaluate them. However, I would highly recommend the PAC I TTY interface/resident assembler optional card. Having an assembler and dis-assembler as well as a system monitor in firmware result in relatively quick and easy assembly language. programming and debugging. The 16 bit instructions and data provide for efficient assembly language programming as well as increased accuracy. One may use words as a whole or in 8 bit bytes. Common memory and peripheral addressing result in simple quick I/O instructions.

With the Pacer's 16 bit accuracy and easy I/O and a couple of floppy disks, one could program something like Music V[3] and Score[4] for composition and playing of high fidelity music. Of course you would also need a 16 bit DAC[5]. If a very fast hardware multiply card were added as well as a fast Pace IC (rumored to be coming out from National Semi) to replace the pMOS IC, a real time FM systhesis[6] of timbre might be possible. I'm not sure if the rest of the Pacer circuits would be fast enough. Oh well – back to the 4 bit bipolar slices for real time Fourier synthesis

(1) Project Support Engineering/750 N. Mary/Sunnyvale, Ca. 94086

(2) *Bit Bucket*/Compute-115/National Semiconductor/2900 Semiconductor Dr. Santa Clara, Ca. 95051.

(3) Described in *The Technology of Computer Music* by M. Mathews, MIT Press, Cambridge, MA 1969.

(4) "Score – A musician's Approach to Computer Music" by L. Smith in the *Journal of the Audio Engineering Society* (JAES) Vol. 20, No. 1, Jan/Feb, '72.

(5) "Digital-to-Analog Converters: Some Problems in Producing High Fidelity Systems" by R. Talambiras, *Computer Design*, Vol 15, No. 1, page 63, Jan '76.

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

PACER PRODUCT LINE RETAIL PRICE LIST (JUNE, 1976)

Quantity	Pacer Units 1H	2H	3H	PAC I	PAC II	PAC III	PAC IV	Fan Kit
1 – 3	$895	$1075	$1025	$180	$225	$50	$7	$25
4 – 9	$855	$1035	$985	$175	$245	$47	$6	$23
10 – up	$820	$995	$950	$170	$235	$45	$5	$22

Club group buys would help reduce costs.

Pacer 1H – totally unassembled (not recommended by P.S.E. for beginners.

Pacer 2H – completely assembled, tested and burned in.

Pacer 3H – unassembled except for logic cards which are tested and burned in.

PAC I – TTY interface/resident assembler card.

PAC II – 2K x 16 MOS RAM card.

PAC III – prototyping card with voltage regulators.

PAC IV – dual 43 pin mother board connector (this comes with PAC I, PAC II or PAC III).

Fan kit – designed for general purpose use

PHILA STORE OPENING

The Computer Room is opening in Philadelphia, located at Juniper and Sansom Streets. Though we are unclear of the exact relationship, it is evidently closely associated with Carol Groves of Castle Systems Co., 1028 Spruce St., Philadelphia, PA 19107, 928-1268.

DOC QUESTION ON 64-CHAR MOD FOR TVT-II

(Copy of a letter to Digital Designs)

To: Digital Designs
P. O. Box 4241
Victoria, Texas 77901

August 18, 1976

David O. Valliere, [see *DDJ*, Vol. 1, No. 6, page 29]

I received the circuit board to convert my SWTPC TVT-II to 16 x 64 characters from you last week. I have not installed it yet. In the meantime I have modified my TVT-II diagram to include the changes. At the same time I decided to check your modification to see if I could convince myself it would work.

I failed. But I think I know what's wrong. Three things bothered me.

1. There seems to be no source for the logic net comprising pin 9, IC28, pin 13, IC33 on the TVT board.
2. There seems to be no source for pin 15, IC42 comparator.
3. Connection I on your circuit board had no instruction referring to it.

I would like to know if the following correction is correct. Remove the following instruction from page 4 of the instructions: Connect pin 15, IC35 to H. Add in its place the following two instructions: Connect pin 13, IC33 to H; Connect pin 15, IC42 to I.

Walter Cole
5868 Pentz Way
San Jose, CA 95123

P.S. I am very interested in your upper/lower case and computer controlled cursor modifications for the SWTPC TVT-II.

ENDLESS LOOPS SUGGESTED TO AVOID TAPE-STRETCH PROBLEMS

Dear Dr. Dobb's,

Since we have been experiencing tape-stretch problems and other Natal problems when we have to run BASIC back and forth on the cassette machine to load our APPLE every morning, I was going to ask you for your best medical advice, but have solved the problem with a home remedy. Being new to this cassette business, I figured they should only run in one direction, but no, wham-bang, you have to rewind the buggers–this stretches and pulls and raises general hell with the tape! So why not just run in one direction? No reason why not if you use an endless loop tape! TDK puts out a high quality loop tape in 20 sec., 30 sec., 1 min., 3 min., 6 min., and 12 min. versions (something for everyone)– so you can save your favorite games or BASIC on a tape that will run round-and-round and just wait for the beginning to press load.

Sorry I didn't get the prices (they are expensive), but just finding the things after calling a dozen places threw me off. They are available from:

Qument Electronics
1000 S. Bascom Ave.
San Jose, CA
(408) 998-5900

If you have a loading indicator, file no. select, or can listen to the tape, endless-loop makes a fast single loading device.

What can you do about my wisdom teeth?

Bob Grater
Technical Services
Byte Shop No. 2
3400 W. El Camino Real
Santa Clara, CA 95051

The Computer at Puberty

The computer already seems to be everywhere, doing everything, but according to a group of prominent computer scientists, the industry has only just begun an explosive growth. Over the next two decades it will enter home, business and factory as never before.

This was the principal conclusion of preliminary reports on an extensive study of "The Future Impact of Computers" being performed by computer and social scientists at IBM, Stanford, M.I.T., Harvard, Yale, Princeton, and other prominent institutions.

The study, coordinated by Michael Dertouzos, Director of M.I.T.'s Laboratory for Computer Science, and Joel Moses, Associate Director, is sponsored by AT&T, IBM, the Office of Naval Research, and M.I.T. To be completed in 1977, it consists of an integrated series of carefully thought-out and hashed-over papers on key topics in the computer field. The participants in the study, key people in their fields, occupy excellent vantage points to ponder the computer's future, said Professor Dertouzos. But, perhaps just as important, their influence could make their predictions self-fulfilling prophecies.

At the symposium marking the 100th anniversary of the telephone, many of the participants in the study gathered to offer a fascinating glimpse into a future stuffed with computers.

The hardware of the computer should be about the least restraining influence on the computer's future, according to scientist-entrepreneur Robert N. Noyce, Chairman of Intel Corp. His analysis shows that the size of computer components could be reduced a hundred to a thousand times, from even the tiny circuits of today, before they begin bumping "down" against the fundamental thermal noise limits on size. And technological advance has allowed the complexity of integrated circuits to double every year since 1960, he pointed out, a trend that should continue, allowing ever lower costs for ever more powerful computers. The future almost certainly holds reductions in size and cost, and increased in complexity as remarkable as those that brought pocket calculators from the engineer's workbench to citizen's pockets.

The Computer at Home

The key to these technical advances is continuing profitability, said Dr. Noyce. His company is a leader in the high-technology semiconductor industry. For the computer to remain a profitable product, said Dr. Noyce, it must become a mass-market item. And according to Professor Moses, the computer will mount a massive invasion of the home over the next two decades.

The first wave, said Professor Moses, will consist of computers which perform tasks not requiring connection to a communication network - such networks will probably bring up the rear in computer developments.

Professor Moses foresees computer games and other recreations, "smart" household appliances, home education programs, and efficient household heating and cooling control as the first uses for home computers. Indeed, many such products have already appeared in the marketplace: computer-controlled machines now being marketed include microwave ovens, television sets, automobiles, and, of course, the popular electronic games, which can be played through one's TV.

Eventually, however, computer communications networks for the home will evolve, and subscribers will have enormous volumes of information at their fingertips - via home computer consoles that will cost about as much as today's color TV sets. Computerized encyclopedias, electronic mail, and enormous electronic "newspapers" which would allow the reader to delve into whatever subject he wished. Also possible would be a new form of electronic "junk" mail - when a subscriber wished to buy something, he could use his computer to summon all the current advertising for the product, and comparison shop from his easy chair. This divorce of news from advertising could also affect the content of news, noted Professor Moses. No longer would a reporter on auto safety worry about irate car dealers cancelling ads, should he criticize their product.

There will also be other changes in the kinds of information we receive, with the advent of the "information utility," according to John McCarthy of Stanford. Computerized publishing would increase competition, because the expensive printing process would be eliminated. As soon as a manuscript was ready, it could be instantaneously available. The nature of public controversy in the computer age would also change, said Dr. McCarthy. When one candidate attacked another over the computer news network of the future, the average citizen could immediately dial up the opposing statements. Thus, public stands would be much more carefully considered before being made public. Also, government information would be truly public for the first time. Instead of being stored away in dusty archives, available only to those with the wherewithal to dig it out, the data would be instantly available on the average person's home console.

Giving Computers the Business

Although business and industry are already heavy users of computer power, they will become even more so in the future, according to the study participants. One major trend will be toward the "paperless office," said Victor A. Vyssotsky of Bell Telephone Laboratories. Records and communications will be stored and transmitted by computers rather than as pieces of paper. However, said Dr. Vyssotsky, the clerical worker will by no means be out of a job; rather the nature of his job will change drastically, from paper shuffler to electron pusher. Computer methods for assisting managers would also make gradual process as business tools.

"Overall, I anticipate a modest steady improvement as a result of further computerization of business functions," said Dr. Vyssotsky. "At a very rough guess, this improvement, over the next twenty years, may be something like two per cent per year for a force which may run somewhere aorund forty per cent of the U.S. work force."

Manufacturing will also see much greater use of computers than even today, according to Professor Dertouzos, who wrote the portion of the study on that subject. The key technical developments in this area will be programmable robots with senses of touch and sight; microcomputer-operated control systems integrated into aircraft, autos, home and factories; and networks of robots or control systems capable of working together.

Out of these technologies, Professor Dertouzos sees emerging automated factories that could produce products tailored specifically to each customer. For instance, a customer might order shoes fitted to his foot measure-

ments, and the shoes would be manufactured on the spot. Such automation would revive U.S. industries currently being strangled by cheap foreign labor. Also, said Dr. Dertouzos, robots would continue to take over dangerous jobs, preserving human life; and they would relieve humans of mindless tasks, enabling them to do truly humane tasks.

"While the popular view of an automated future involves a heartless and mechanistic society geared to productivity and efficiency, I advocate the opposite view that computers will improve and humanize our way of life," he said.

The major plea throughout all the scientists' discourse was – as Terry Winograd of Stanford University put it – that computers be "peoplized." The computer must be taught to adopt the sytle of communication natural to people, and not the foreign language it is now allowed to use. Computers that understand human dialogue, that can take human speech as an input, and that can explain their actions and reasoning to humans will represent major steps in this direction.

Perhaps them the computer will become as benign and helpful an invention as is the telephone, whose 100th anniversary the conferees were observing. – *D.M.*

A THEATER COMPUTER USERS GROUP

Support of a Theater Computer Users Group is being considered by Theater Sources, Inc., a non-profit, tax-exemp corporation. Persons who have developed programs for applications in the area of live drama are invited to contact TSI with suggestions, information, and requests to be placed on a preliminary mailing list. Write TSI-TCUG, 4712 Northway Dr., Dallas, TX 75206.

The form of support has not been set, but may include a newsletter, reproduction of listings or decks on file and/or meetings for exchange of information. Suggestions are welcomed. Costs of services will be covered by modest fees.

Uses of the computer in theater might include rather ordinary uses such as bookkeeping, accounting, and mailing lists, but can also be extended to ticket sales, seating charts, budgeting and artist records. Further extensions might include creation of buying and cutting lists for sets and costumes, visualization and modification of designs, and other interactions between artists and machine to ease the tedium of craft.

Theater Sources, Inc., is a non-profit corporation under Texas law, created to "gather and distribute information about live theater" and granted a federal tax exemption. Its current activities include publication of *PROLOG*, a newsletter for playwrights, and *THEATER ACROSS AMERICA*, for community theater, and the National Theater File, timeshare access to theater information.

Contact: Mike Firth, 4712 Northway Dr., Dallas, TX 75206, (214) 363-5003.

SOUTH CAROLINA HOBBYIST GROUP FORMING

South Carolinians interested in forming a persona computing club should contact Dewey K. Barefield, Box 452, Hodges, SC 24653.

ROBOT FEATURED AT ACGNJ MEETING

BUSTER-II, a robot, will be the featured demonstrator at the October meeting [of the Amateur Computer Group of New Jersey]. BUSTER-II is a home-built robot controlled by an MOS Technology KIM-1 microprocessor.

BUSTER-II was designed and built by Rod Loofburrow. Rod, who is 14 years old, is another of John Loofburrow's young sons . . . you may remember Wayne Loofburrow (age 13) who demonstrated a 6800 Assembler at the March meeting.

Rod will be explaining the technical side (or is it inside) of BUSTER-II, demonstrating the robot's abilities, and answering all questions (sorry, as yet, BUSTER-II does not have a voice).

The October meeting will be held at, and sponsored by, Middlesex County College, Edison NJ, on Friday, October 15th. The meeting will be held in Building No. 4, The College Center, Cafeterias B and C (same as last meeting at MCC).

There will be a Flea Market, starting at 6:30 p.m. Tables are available ($2 contribution is asked). Remember to bring your own extension cords.

BUSTER-II will be doing his thing at 8 p.m. We must vacate the building by 10:30 p.m.

Editor's Note: We are publishing this, not as a meeting announcement for the ACGNJ, but rather, because of the unusual and fascinating nature of the presentation being given at their meeting.

LOOT FOR USELESS HEADS

If you happen to have any crashed or worn-out disc heads, you might be able to peddle 'em for nominal cash. Trans-Data Corp [170 Glenn Way, Belmont, CA 94002, (415) 593-8545 (Don Collier)] is looking for heads for IBM 3330's and 2314's.

[ANOTHER] MICROCOMPUTER PRODUCT SURVEY

Dear Sir, Sept. 9, 1976

This letter is to announce to you the publishing of *The Microcomputer Product Survey*. As we at Small Business Systems are sure you realize, there are a number of individuals who have claimed to offer a microcomputer product survey which has turned out to be little more than a chip feature list. The *M.P.S.* is a two volume report on the hard and software being offered in the small and personal computer market.

The first volume will be released in November and will deal with Microcomputer Hardware, Peripherals and the companies that manufacture them. The second volume will deal with Software and the additional companies which supply only software and is scheduled to be released in January, 1977.

The organization of Volume 1 will be a series of 16 charts covering all the major mainframes and peripherals. The peripheral section will include plug compatible memory and I/O cards. All of the hardware will be cross-referenced to the company profile. The peripherals will also be cross-referenced by mainframe compatibility. The general form for Volume 2 will be three charts and a series of plain text discussions of applications program services. The charts will compare general types of instruction for the three general classes of software, operating systems, assemblers and higher level language.

The two volumes will be supplemented with periodic product supplements to allow the *M.P.S.* to be one of the most timely sources of performance parameters in the marketplace today.

SBS has planned to allow the individual three options to personally determine the worth of the *M.P.S.* A subscription to the *M.P.S.* is only $9.00 a year, a sample issue is $3.00 and complete details for only $1.00.

Yours truly,

Timothy D. Sisson
General Manager

Small Business Systems
1800 Wayne No. 1
Bellevue, NE 68005

How to Code Code
Jack Armstrong's Super Decoder Ring

by Jon O. Stedman, 1450 Alice St., Oakland, CA 94612

Nostalgia time, gang! Remember when Jack Armstrong was sending his secret messages over the radio into your home? For only two box tops of some breakfast cereal, you could order your own SUPER decoder ring. In no time, you can be passing secret messages like any international spy.

In reality, we have a super cryptographic machine in the home brew micro-computer! Complicated cyphering techniques can be programmed to baffle all of those government snoops who like to read other people's mail.

Simple letter substitution cyphers are easily broken with a little statistical analysis on the letter frequency of the code-text. Many other cyphering techniques are public knowledge (see your public library) so the secret part of a coded message is the code word which is the "key" to decoding the message.

For instance, John Kennedy used the key . . .HEAVY EQUIPMENT and the Playfaire cypher to send a rescue message from his wrecked PT 109. Had the Japanese intercepted and de-cyphered that rescue message, then history would have been changed!

Personally, computer games don't interest me much, so I hope that the following specifications will excite someone to write a Playfaire program and that cryptography will become a developing interest to hobbyist computer users.

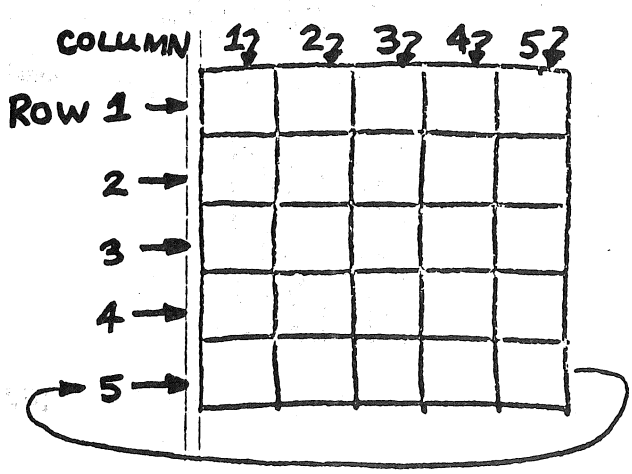
You can try your hand at Playfaire with the following bit of encyphered philosophy.

KPGHU WKPGA EWAQG EHBBZ PBVRN
EYUFZ ATSHY BMZQE VYVSN EVGUE
WZREU BMZQE VYUGG YEBZL

Use Kennedy's code word HEAVY EQUIPMENT.

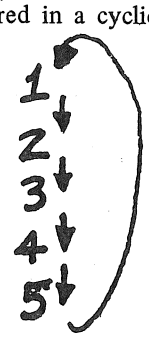
PLAYFAIRE–DIGRAPHIC CRYPTOGRAPHIC CYPHER

This code technique uses a 5 x 5 array matrix. The elements in this array can be considered to be ordered into rows and columns. Each row in turn also is ordered in a cyclic fashion.



The first cell of row number 5 is the one in the lower left-hand corner of the array. The second cell of row number 5 is directly to the right of the first cell. This ordering continues until the 5th cell is reached (the one in the lower right-hand corner), then the cycles begins again. Thus the elements in row number 5 are ordered in a cyclic way as follows:

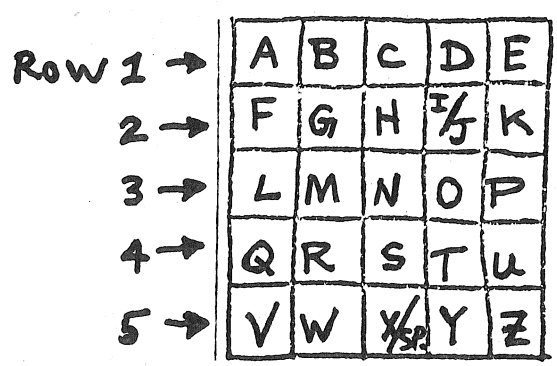
In a similar way, the elements of each column can be considered to be ordered in a cyclic pattern as follows:



Since the array has room for only 25 letters, the letters I and J will be considered to be interchangeable. I and J will occupy the same array element. In a similar way, X and space will occupy the same array element, but only X will be used as a code letter.

The key to filling in the 5 x 5 array with the letters of the alphabet turns on the use of a *"code word."*

In the absence of a code word, then the normal alphabetic order is followed in filling the array. The array is filled row by row, i.e., row number 1 through row number 5.



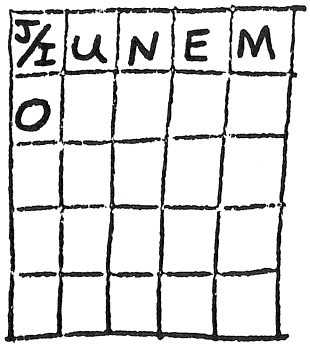
The use of a *code word* permutes the order shown above.

With a code word, fill up the array as follows:

Fill the array (row by row) with the letters of the code word in the order they occur in the code word. Remember not to use any letter twice from the code word. After the code word letters are exhausted, then complete the array

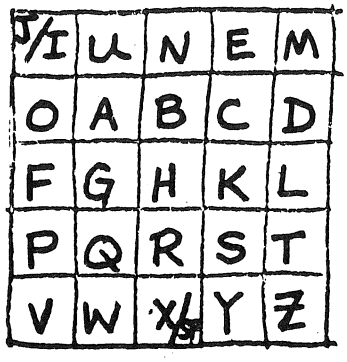
with the remainder of the alphabet (in alphabetic order, of course).

Example: code word is JUNEMOON



The code word has been entered into the array. Note: (J/I) J and I occupy same cell. O is used only once. N is used only once.

The array is then completed with the remainder of the alphabet.



Playfaire is a digraphic technique that destroys the letter frequency distribution found in a particular culture-language. The technique is to transform letter pairs into another letter pair, i.e., digraph is a letter pair. For instance;

Plain-text [F,A] ↔ [G,O] Code-text
[F,E] ↔ [K,J]

Of course, digraphs have a known frequency distribution, but the number of letters is small (approximately 26) and the number of digraphs is large (approximately 26 x 26); so a much larger volume of codetraffic is needed to build any statistics.

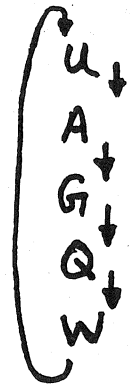
There are 4 cases to consider using the Playfaire array.

I. Both letters of a digraph are in the same row.
II. Both letters of a digraph are in the same column.
III. The letters of a digraph are in different rows and columns.
IV. Both letters of the digraph are the same.

Case I: Both letters of a digraph are in the same row. Method: each letter is replaced by the next letter in order in the given row. Example: Row number 2 from JUNEMOON array above.

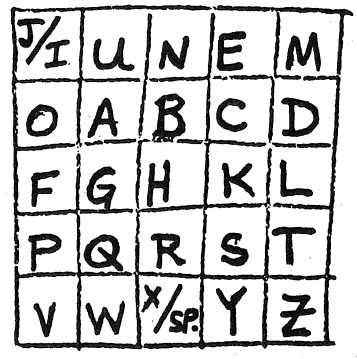
Plain-text [D,O] → [O,A] Code-text
[D,A] → [O,B]
[A,D] → [B,O]
[O,B] → [A,C]
[A,C] → [B,D]
[C,O] → [D,A]

Case II: Both letters of a diagraph are in the same column. Method: each letter is replaced by the next letter in order in the given column. Example: Column number 2 from JUNEMOON array.

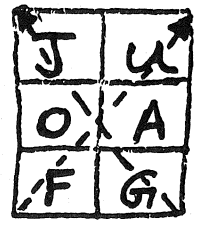


Plain-text [A,Q] → [G,W] Code-text
[A,U] → [G,A]
[W,Q] → [U,W]
[G,A] → [Q,G]
[G,W] → [Q,U]
[Q,W] → [W,U]

Case III: Letters of a digraph are in different rows and columns. Method: The vector formed by the line from the first to the second letter of a digraph is one of the diagonals of a rectangle. The two diagonals form the equivalence between the text-digraph and the code-digraph. Example: Using the JUNEMOON array:



For instance, the digraph [F,U] defines the diagonal and rectangle:



The mirror image of the [F,U] diagonal is the [G,J]

(Concludes, next page)

Music by computer

Reprinted with permission of Digital Equipment Corporation.

Yamaha has interfaced an electronic organ with a computer so that the organ can be played either by the computer or a musician.

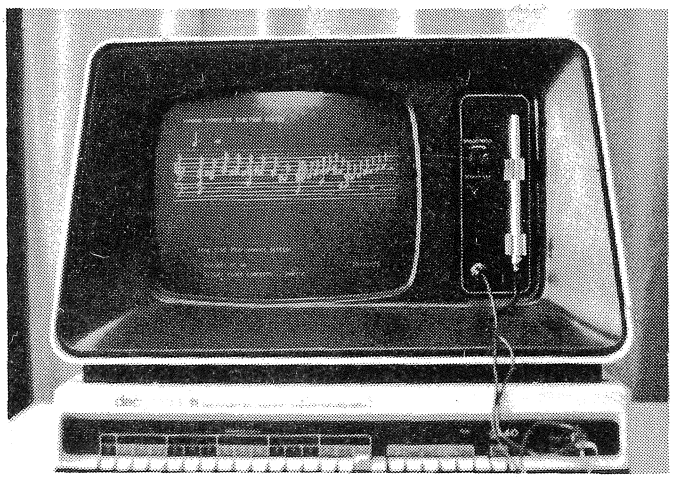
A PDP-8 minicomputer is able to control the instrument, reproduce previously recorded musical scores and display them on a cathode ray tube.

Using a core-resident real time operating system, the minicomputer can sense what is being played, interpret this data and compare it with storage.

Yamaha anticipates that the organ may one day be taught completely by this method. Students will be able to compare the written music stored in the computer memory with what they actually play as displayed on the cathode ray tube. Practice sessions can be recorded for critique by instructors at a later time.

Digital's GT40 graphics terminal was chosen for its interrupr handling capabilities, refresh-type display and cost effectiveness.

The refresh display is ideally suited to the dynamic pattern changes in a musical score. Since the GT40 converts the binary data obtained from the memory area (DMA) to analog form and displays it on the CRT independent of the central processing unit, the processor can be dedicated to the organ, making possible the real time applications of the recording and playing back.



(conclusion of Decoder Ring Article)

diagonal. Note that the first letter of both digraphs are in the same row.

Plain-text [F,U] → [G,J] Code-text
[U,F] → [J,G]

In general, using matrix notation–(row number, column number):

$$[(F),(U)] = [(3,1),(1,2)] \leftrightarrow [(3,2),(1,1)] = [G,J]$$

Therefore– $[(\alpha,\beta)\ (\delta,\zeta)] \leftrightarrow [(\alpha\ ,\ \zeta)\ (\delta\ ,\beta)]$

Case IV: Both letters of a digraph are the same. Method: the letter X is to be inserted into the plain-text to separate double letters. Example: BALLOONS would be broken into digraphs as [BA], [LX], [LO], [NS].

In order to break up the appearance of words, the plain-text is run together by leaving out the spaces between the words. Sometimes to improve on readability and avoid ambiguous situations, X's are inserted for the spaces between words.

The code-text output is in a standard fixes format of 5 letters per word.

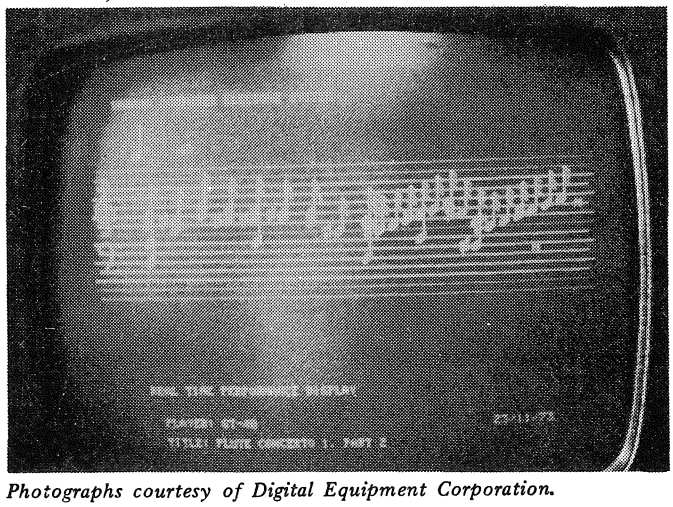
EUNFE RAHXJ UPZWO etc. . . .

PROCESSOR TECHNOLOGY ANNOUNCES WINNERS OF *Personal Computing '76* DRAWING

Processor Technology was one of a number of vendors which donated "door prizes" for the Personal Computing '76 convention in Atlantic City. The P.T. door prizes consisted of two $500 merchandise certicicates. The winners were: E. Eibling, Florham Park, NJ; Barbara Ataran, Montclare, PA.

AN 18-BIT PDP-11?

Rumor has it that Gordon Bell originally designed the Digital Equipment Corp. PDP-11 as an *18-bit* machine (note the bus width), but DEC Marketing decided to make it a "byte" machine and reduced the BRANCH range.



Photographs courtesy of Digital Equipment Corporation.

HURRY!

Renew Your Subscription Now

Don't miss a single issue!

This Volume Ends Next Month

DR. DOBB'S JOURNAL of COMPUTER CALISTHENICS & ORTHODONTIA

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

SURVEY OF VENDOR PRODUCTS & SERVICES

We have often published individuals' compliments and complaints about various products and services being sold to the hobbyist community. It is far more valid to publish evaluations derived from a considerable number of consumers' ratings, rather than those of a few joyful or irate individuals.

To this end, we request that you complete this rating sheet if you are a consumer of computer hobbyist equipment. In order for us to consider your ratings, you must complete the following identifying information about yourself (please *print* or *type*):

[This form may be copied and distributed as widely as desired.]

Name: _____

Home Address: _____

City, State, Zip: _____

Home Phone: _____

Occupation: _____

Signature: _____

VENDOR EVALUATIONS:

RATING:

Vendor & Product Information — Great — Acceptable — Awful

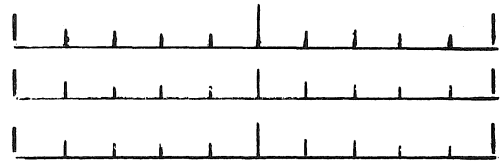
Name _____

City & State _____ Product

Product Model/Name _____ Documentation

Date Purchased _____ Customer Service

Purchased From: _____



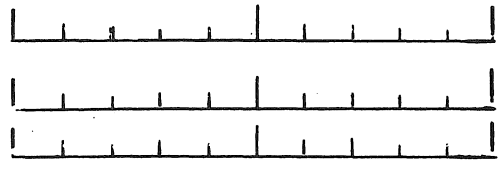
Name _____

City & State _____ Product

Product Model/Name _____ Documentation

Date Purchased _____ Customer Service

Purchased From: _____



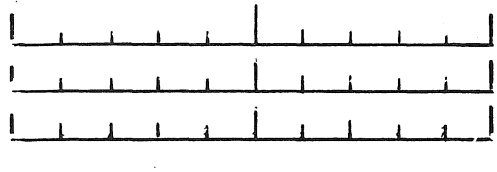
Name _____

City & State _____ Product

Product Model Name _____ Documentation

Date Purchased _____ Customer Service

Purchased From _____



Return this completed form to : *Dr. Dobb's Journal* P.O. Box 310
Menlo Park, CA 94025

TIMER ROUTINES FOR 6800's

Gentlemen: Sept. 3, 1976

I have found in writing programs, that often a time-delay can be used to spice up a program. The following subroutine will generate time delays between 1 and 255 seconds. To use the routine, load register-B with the number of seconds you wish to delay and then jump to the subroutine. The subroutine returns after the specified number of seconds.

To show how it works, I have enclosed a short program that uses the time delay routine to ring the bell on a teletype once every 30 seconds (make it 3 minutes and you've got an egg timer).

How it works:

LOOP1 is the basic time-delay loop. Traversing from line #11 to line #14 takes 20 clock cycles. This loop is executed 50,000 times (C350 Hex). Thus to complete this loop requires 20 x 50,000 or 1 million clock cycles. Since the clock frequency on the 6800 is 1 Megahertz, it takes one second to complete.

LOOP2 is the outer loop which by decrementing and testing register-B allows varying delays by repeating **LOOP1** the specified number of times.

Line #9 and #17 save and restore the X-register to the value it had when the routine was entered.

Note: If you put a radio next to the CPU and tune to the interference, there will be an audible click each second (a darkroom timer??). These routines were assembled by a cross-assembler I have written in IBM's BASIC.

Sincerely,

Gregory A. Worth

115 Campbell, Apt. 6
Rochester, MI 48063

```
LINE PC  OBJ CODE   LABEL    MNEMONIC  OPERANDS     COMMENTS
---- --  --------   -----    --------  ---------    -------
   0 0000                     ORIG     0000
   1     *
   2     *
   3     *TIME DELAY SUBROUTINE FOR THE MOTORLA 6800
   4     *NOTE:
   5     * 1. VALUE IN THE B-REGISTER WHEN ROUTINE IS ENTERED
   6     *     DETERMINES THE NUMBER OF SECONDS THE PROGRAM DELAYS
   7     * 2. REGISTERS A & X ARE LEFT UNCHANGED BY THE ROUTINE
   8     *
   9 0000 FF 00 16  &DELAY     STX      &SAVE       ;SAVE THE X-REGISTER
  10 0003 CE C3 50  &LOOP2     LDX      #$C350      ;LOAD X-REG WITH 50000

  11 0006 76 00 03  &LOOP1     ROR      &LOOP2      ;ROTATE->WASTE TIME
  12 0009 79 00 03             ROL      &LOOP2      ;ROTATE->WASTE TIME
  13 000C 09                   DEX                  ;DECREMENT X-REG
  14 000D 26 F7                BNE      &LOOP1      ;IF X<>0 GOTO LOOP1
  15 000F 5A                   DECB                 ;DECREMENT B-REG
  16 0010 26 F1                BNE      &LOOP2      ;IF B<>0 GOTO LOOP2
  17 0012 FE 00 16             LDX      &SAVE       ;RESTORE X-REG
  18 0015 39                   RTS                  ;RETURN TO MAIN PROGRA

  19 0016           &SAVE      RMB      2           ;STORAGE FOR X-REG
  20 0018                      END

*  WRITTEN AUGUST 25, 1976 BY GREGORY A. WORTH


LINE PC  OBJ CODE   LABEL    MNEMONIC  OPERANDS     COMMENTS
---- --  --------   -----    --------  ---------    -------
   0 0000                     ORIG     0030
   1     *
   2     * WRITTEN BY GREGORY WORTH
   3     *
   4     * PROGRAM TO RING THE BELL ON A TELETYPE EVERY 30 SECONDS
   5     * USES TIME DELAY SUBROUTINE FOR DELAYS  & THE CHARACTER OUT
   6     * ROUTINE IN  MIKBUG.
   7     *
   8     * NOTE E075 IS THE CHARACTER OUT ROUTINE IN MIKBUG
   9     *      AND 0000 IS WHERE THE DELAY ROUTINE IS
  10     *
  11 0030 86 07                LDAA     #$07        ;LOAD BELL CHAR.
  12 0032 BD E0 75  &BELL      JSR      $E075       ;RING THE BELL
  13 0035 C6 1E                LDAB     #30         ;LOAD B-REG = 30
  14 0037 BD 00 00             JSR      $0000       ;JUMP TO DELAY PROG
  15 003A 20 F6                BRA      &BELL       ;BRANCH TO &BELL
  16 003C                      END
```

MICRO BASIC PLUS FOR $15.95

INCLUDES COMPLETELY ANNOTATED SOURCE CODE

(NEWS RELEASE FROM TECHNICAL SYSTEMS CONSULTANTS, BOX 2574, W LAFAYETTE, IN 47906)

You have seen the "TINY BASIC" and "MICRO BASIC" for the 6800 which fit in a small system (2-3K) and offer 10-15 statements and commands for the user to write BASIC programs. These systems range from quite slow (due to 2-level interpretation techniques) to surprisingly fast for a small system. The limiting factor in these versions of BASIC is the incomplete selection of statements, commands, and functions. The lack of certain statements means having to do without, resulting in less efficient programs.

TSC has solved this problem by creating MICRO BASIC PLUS. It is not a 2-level interpreter so it runs considerably faster than ones implemented using the 2-level technique. Statements available to the user include PRINT, INPUT, READ, DATA, RESTOR, IF . . . THEN, GOTO, GOSUB, RETURN, ON . . . GOTO, ON . . . GOSUB, FOR, with positive and negative STEP, NEXT, LET, DIM, for both single and double dimensioned arrays (up to 98 by 98), EXT, REM, and END. The commands available to the user are LIST, SCRATCH, RUN, and MONITOR. There are also several functions available which include TAB and SPC (for output formatting), RND, ABS, SGN, and ↑ (Used for exponentiation). This list of features does not stop here. MICRO BASIC PLUS also contains a complete line editor, multiple statements per line, direct execution of most statements, large arithmetic range than others (±99999), complete error identification including printout of the line number where the error occurred, and a very simple procedure for dumping the users BASIC program to paper tape or cassette and then reloading it at a later time. Also included is an EXTERNAL statement which allows the user to write 6800 machine language subroutines to be called during BASIC program execution.

You are probably thinking this all sounds great, but if the less extensive versions of "TINY" or "MICRO" BASIC require 2-3K of memory, this one must require 5 or 6K since the capability is doubled. Well, here is the "icing on the cake!" MICRO BASIC PLUS resides in just a fraction over 3K which means in a 4K system you still have room for a 30 to 60 statement BASIC program! Not only is the interpreter approximately the same size as others, TSC has made special provisions for keeping the users program source to a minimum (e.g. all commands and statements can be entered using the first 3 letters of its name).

Now you are probably saying it must be expensive. You are right if you think $15.95 is expensive, but we think this will also please you. Unlike others, we include the complete commented source listing, hex dump for fast entry using MIKBUG, sample programs, sample output, plus TSC's famous instructions and documentation for use. As all TSC 6800 software, this program uses MIKBUG I/O routines but instructions are provided for adapting to other routines. Write for details concerning availability of MICRO BASIC on "Kansas City Standard" cassettes.

Tiny BASIC for the 6800 & 6502 from Tom Pittman's "Company"

Editor's Note: Tom Pittman *is* Itty Bitty Computers. Tom is an avid computer hobbyist who is one of the more major figures in the San Francisco Bay area's Homebrew Computer Club. He is also an experienced software consultant who specializes in small-computer software and systems. We *strongly* recommend him to you because (1) we have received a number of communications praising Tom's Tiny BASIC for the 6800, (2) we have yet to receive a written or verbal complaint about the quality of Tom's products, (3) we have ample evidence that Tom consistently exhibits an unusually high level of responsiveness and good business ethics in dealing with his customers, and (4) we are delighted to see a competent systems software specialist offering excellent software products to home computer users at very nominal rates, and wish to do everything we can to encourage him and others to pursue this approach to development and distribution of good, low-cost software.

The following information is a duplication of the typewritten data sheet that Tom distributes to those wishing information about his products.

Itty Bitty Computers is dedicated to the principle of high quality, low-cost software for personal computers. Tiny BASIC is our first offering in this field; if it sells well, there will be more. We are not in the hardware business—software is all we sell. The emphasis of all our software is small size and flexibility rather than high speed. Tiny BASIC will run in as little as 3K of memory.

Tiny BASIC is a subset of Dartmouth BASIC, with only 16-bit integer arithmetic. There are only 26 variables (A-Z), no arrays, and no strings. The 12 commands (LET, PRINT, INPUT, IF. . .THEN, GOTO, GOSUB, RETURN, REM, RUN, LIST, CLEAR, END) are adequate for most programming needs. A machine language subroutine calling facility enables the user to extend Tiny BASIC to any degree necessary.

Single copies of Tiny BASIC cost $5. This buys an individual license to use it, a 26-page user manual, and a hexadecimal paper tape in the manufacturer's loader format (Motorola "S" or MOS Technology ";"). A hexadecimal listing will be substituted for the tape on request for users without paper tape facilities. Programs are mailed out first class within a few days of receiving the order, but at these prices we can accept prepaid orders only. For airmail service to foreign countries please add $2 per program. A 20% discount may be applied to quantity purchases of ten or more programs of the same version to the same address at the same time.

Tiny BASIC itself does not contain any I/O instructions; three JMPs link Tiny to the user's I/O routines. These are well documented in the manual. The primary difference between the following versions is the memory requirements (shown in parentheses), though all use memory page 00.

TB680R (0100-08FF) For MIKBUG, EXBUG, and most homebrew 6800 systems with RAM in first 3K of memory.

TB680S (0200-09FF) For Sphere; includes I/O routines.

TB680P (E000-E7FF) For AMI Prototyping Board; includes I/O routines (ACIA), and runs with on-board 1K RAM.

TB650K (0200-0AFF) For KIM and most homebrew 6502 systems with RAM in first 4K of memory.

TB650J (1000-18FF) For Jolt.

TB650W (0300-0BFF) For Apple. Includes I/O routines.

NOTE: The third digit of the version number refers to the revision level. We will ship the most recent version regardless of order specifications (e.g., we are now shipping TB682R for orders specifying TB680R).

MEMORY ALLOCATION STANDARDS FOR HOBBYISTS

by Jim Day

One of the benefits of owning your own computer is the fun of exchanging programs with other hobbyists. So far, this has been fairly easy to do. It is easy to load a BASIC program from an ASCII tape or to bootstrap a machine language tape into a "cold" machine. But this situation may not last very long. As more hobbyists begin to use ROM-resident operating systems and a wider variety of peripherals, compatibility problems may become much more difficult to resolve unless standards are established for memory allocation.

I suggest that the first 16K of memory be reserved for the operating system, including resident processors and I/O handlers. Standard port numbers and handler entry points should be established for all kinds of peripherals, including the exotic ones. Allocation standards need not work a hardship on those with smaller amounts of memory, since memory modules don't have to be contiguous. That is, a 2K machine could have 1K of memory starting at address 0 and the rest starting at 16K.

Needless to say, the standard handler "entry points" would simply be instructions branching to the real entry points somewhere within the first 16K of memory. Likewise for the entry points of language processors, text editors, etc. Attempts to use unimplemented programs should cause a branch to an error routine.

6800 CAN SHARE HOBBY-STANDARD BUS WITH 8080 WITHOUT CONFLICT

M.R.S. has developed a 6800-based board that plugs into a hobbyist-standard (Altair, IMSAI/SOL/Cromemco/Polymorphic/etc.) bus. Even while installed in the bus, it will allow any also-resident 8080 to run without interference. It gains control of the system via a single instruction, and returns control to the 8080, either via the halt switch or via software instructions. The 8080 unit handles all front-panel interface.

A fully assembled and tested unit is available for $180 with kits available at even lower cost. The mailing address for M.R.S. is Box 1220, Hawthorne, CA 90250.

SAVE AND LOAD MODS TO PITTMAN'S 6800 TINY BASIC

Dear Jim,

Here are modifications I've written to Tom Pittman's 6800 Tiny BASIC. The additions are confined within the 2K size for the interpreter. The Sphere 6800 user will now have the ability to SAVE and LOAD programs he has written in Pittman's Tiny BASIC. I do not think the modifications submitted infringe on the proprietary rights of Pittman's Tiny BASIC.

Sincerely,

Henry L. Kee
42-24 Colden St.
Flushing, NY 11355

```
            MODIFICATIONS TO PITTMAN'S TINY BASIC
            SAVE PROGRAM, LOAD PROGRAM, SYSTEMS BREAK

ENTER PITTMAN'S TINY BASIC WITH I/O SPHERE INTERFACES
IN LOCATIONS D8 TO FD AS PROVIDED

ENTER THE FOLLOWING MODIFICATIONS:

ORIGIN CØ
CE 0200        TINY BASIC START ADDRESS
DF 3C          SET PTR FOR START
CE 09FF        TINY BASIC END ADDRESS
DF 3E          SET PTR FOR END
86 42          LOAD A WITH VALUE "B"
97 34          STORE INTO PART OF IDENT
BD FB91        READ BLOCK
7E 0200        GO TO START ADDRESS FOR TINY BASIC

ORIGIN 9C1
BD FE4A        BREAKPT FOR FUTURE USE
C6 91          LOAD OPTION (DECIMAL ADDRESS 2500)
F7 09EB        MODIFY I/O FOR READ
F7 09F6        OR WRITE BY VALUE IN B
97 39          STORE 2ND CHARACTER OF ACIA
DF 33          STORE 2ND CHARACTER OF ID
86 FØ          SETUP ACIA ASSIGNMENT
97 38
86 FF          SET FLAG
97 3A
86 24          STORE "$" FOR INITIAL ID
97 33
CE 00C8        BEGIN ADDRESS OF TINY BASIC PARAMETERS
DF 3C          SET BUFFER START
CE 00CF        END ADDRESS OF TINY BASIC PARAMETERS
DF 3E          SET BUFFER END
BD FBØØ        INITIALIZE I/O
BD FBØØ        READ OR WRITE
DE C8          SET USER PROGRAM START PTR
DF 3C          INTO BUFFER START
DE CC          SET USER PROGRAM END PTR
DF 3E          INTO BUFFER END
BD FBØØ        READ OR WRITE USER BLOCK
39             EXIT
BD FE4A        SYSTEMS HALT (DECIMAL ADDRESS 2552)
C6 2D          SAVE OPTION (DECIMAL ADDRESS 2555)
               LOAD B WITH WRITE
7E 09C6        BRANCH TO ROUTINE

CODING FOR V3N SYSTEM

ENTER THE FOLLOWING:

ORIGIN 33
24 54          "$T" ID FOR TINY BASIC
ORIGIN 38
FØ 5Ø          CASSETTE 1
FF ØØ
00 C0          START OF BOOTSTRAP LOADER
00 FF          END OF BOOTSTRAP LOADER


ORIGIN AØØ
BD FB ØØ       WRITE OUT BOOTSTRAP
BD FB 2D
BD FE 4A

EXECUTE AT LOCATION AØØ


ORIGIN 33
24 42          "$B" ID FOR TINY BASIC
ORIGIN 38
FØ 5Ø
FF ØØ
02 00          START OF TINY BASIC
09 FF          END OF TINY BASIC


EXECUTE AT LOCATION AØØ

YOU NOW HAVE A MODIFIED VERSION OF TINY BASIC ON CASSETTE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


TO LOAD PROGRAM:

ENTER THE FOLLOWING:
ORIGIN 33
24 54
ORIGIN 38
FØ 5Ø
FF ØØ
00 C0
00 FF
ORIGIN AØØ
BD FB ØØ
BD FB 91
7E 00 C0


EXECUTE AT LOCATION AØØ

TO SAVE PROGRAM:

INTERRUPT SYSTEM
EXECUTE N=USR(2555,xx,yy)

N      VARIABLE THAT IS USED FOR ENTRY INTO SAVE
       COULD BE ANYTHING FROM A-Z
USR    USER CALL
2555   DECIMAL ADDRESS OF SAVE
xx     SECOND CHARACTER OF ID
       FIRST CHARACTER WILL ALWAYS BE "$"
       THE VALUE GIVEN SHOULD BE IN HEX
       e.g. 65=A,  71=G,  etc
yy     SECOND CHARACTER OF ACIA
       FIRST CHARACTER WILL ALWAYS BE "FØ"
       THE VALUE GIVEN SHOULD BE IN HEX
       e.g. 80=50, 96=60

TO LOAD PROGRAM

INTERRUPT SYSTEM
EXECUTE N=USR(2500,xx,yy)

2555  DECIMAL ADDRESS OF LOAD
all other parameters are the same as above


SYSTEMS BREAK

GOTO USR(2552)

2552  DECIMAL ADDRESS OF BREAK
THIS INSTRUCTION MAY ALSO BE PART OF CODING IN PROGRAM


TO INTERRUPT SYSTEM AT THE TINY BASIC LEVEL
HIT "REPEAT KEY"

TO IGNORE LINE JUST TYPPED
HIT "ESC" KEY
```

A BASIC TERMINAL EXCHANGE PROGRAM

Written by Robert Wilcox
920 N. Washington
Owossa, MI 48867

Sept. 13, 1976

The purpose of this program is to add the ability to talk to two terminals to those versions of MITS BASIC which do not have the console command.

The terminal change is accomplished by calling a machine language (USR) program which has been loaded into reserved high memory. The program may be loaded by use of the BASIC program given here, or in the case of 4K BASIC which does not have the POKE statement, by use of the front panel switches.

When the USR program is called, it interrogates each I/O handling location in BASIC. If Port Number 20 or 21 is found, it is changed to Port Number 22 or 23, respectively. If the BASIC is already addressing the higher port, it is changed to the lower one. Thus, if one terminal is addressed to Port 20-21, and another one to Port 22-23, control will be passed back and forth between them whenever the USR program is called.

The USR program may be called by a direct command, such as:

X = USR(1)

or, this may be an indirect (numbered) statement in a BASIC program:

110 X = USR(1)

The following statement:

10 X = USR(1):GOTO 10

will cause control to alternate between the two terminals. Typing a Control-C on either terminal will cause command to stop at that terminal.

The program is written for 8K BASIC, Version 3.2, supplied on paper tape. "*" indicates data to be substituted for the cassette version, and "**" indicates data for 4K BASIC.

BASIC PROGRAM WHICH LOADS THE USR PROGRAM:

```
5  REM     REMEMBER TO RESERVE MEMORY SPACE WHEN INITIALIZING BASIC
10 POKE 73,208:POKE 74,31
20 L=8144
30 READ D$:IF D$="END" THEN END
40 GOSUB  120
50 POKE L,D:L=L+1:GOTO 30
60 DATA 021,355,037,032,376,377,310,157,023,032,147,176,376,022
70 DATA 372,350,037,326,002,167,023,303,323,037,306,002,303,343
80 DATA 037
90 REM  USE LINE 100 FOR PAPER TAPE VERSION.
95 REM  USE LINE 110 FOR CASSETTE VERSION.
100 DATA 301,004,311,004,314,004,323,004,176,005,332,005,377,END
* 110 DATA 321,004,331,004,334,004,343,004,216,005,352,005,377,END *
120 D1= VAL(MID$(D$,1,1))
130 D2= VAL(MID$(D$,2,1))
140 D3= VAL(MID$(D$,3,1))
150 D=D1*64+D2*8+D3
160 RETURN
```

This is an assembly listing of the USR program. Note that the data is different for the different versions of BASIC. Before this program can be called by the USR command in BASIC, its starting address must be loaded in BASIC in locations called "USRLOC". This is taken care of in line 10 of the BASIC loading program. For 4K BASIC, you must load this information in by the front panel switches: load 320 at location 103, and 037 at location 104.

ADDRESS HIGH,LOW	LABEL	DATA	MNEMONICS	COMMENTS
037,320		021 355 037	LXI D DATA	D&E POINTS TO DATA
323	GET	032	LDAX D	GET DATA IN ACCUM.
324		376 377	CPI 377	IS IT END FLAG?
326		310	RZ	IF END, RETURN
327		157	MOV L,A	PUT DATA IN REG L
330		023	INX D	POINT TO NEXT DATA
331		032	LDAX D	PUT IT IN ACCUM.
332		147	MOV H,A	THEN INTO REG H
333		176	MOV A,M	GET DATA FROM BASIC
334		376 022	CPI 022	COMPARE W/HI PORT NO.

RASKIN REPENTS

Dear Dr. Dobb, Sept. 15, 1976

In my article in the last issue ["A Bit of Wheat Amongst the Chaff" I said that Processor Technology should have advertised their "3P+S" clearly as a *2* parallel channel device, with serial port and a control port. They should have. But I found a P.T. ad, I am happy to say, done a while ago, that mentions in the fine print the exact truth.

They should have, perhaps, called it the 2-1/3 P+S?

Jef Raskin

ERRATA FOR WRIGHT'S *ALPHA-NUMERIC MUSIC*

Dear P.C.C./Malcolm T. Wright:

Please be informed that your publication, *Alphanumeric Music with Amplitude Control,* contains a typographical error in the program listing:

Page 13, Address 001,103 is 000
should be 001

This JUMP references STORE in Page 0, but the object listing references Page 0. Source is correct; object is wrong.

A. Paejh

P.S.—Sounds fine. Forced me to write a hex-octal keyboard read routine.

[Why not submit the keyboard read routine for publication in *DDJ*? –JCW]

UNCLE SAM CONSIDERS ESTABLISHING A "SCIENCE COURT"

The National Science Foundation and the Department of Commerce are jointly studying a proposal to create a Science Court. Such a court would be especially organized to handle disputes involving highly technical matters, disputes that are currently adjudicated by courts having judges of no particular technical capability.

INTEL HAS RESIDENT PL/M

PL/M, the first and most widely used high-level language for microprocessors, was originally only compilable on large machines. A compiler for it is now available, however, that is resident on their 8080-based Intellec floppy disc system. It requires 65K bytes.

COMPUTER STORE WORKING ON BUSINESS SOFTWARE

A new computer store, Rainbow Enterprises, is reported to be developing programs for very small business systems. They are located at 10723 White Oak Ave., Granada Hills, CA 91344, (213) 360-2171.

PDP-11 CROSS ASSEMBLER FOR 6502's

For $900, you can get a PDP-11 based cross assembler for MOS Technology 6502 microprocessors. The cross assembler is written in MACRO-11 and runs under DEC's RT-11 operating system, using 5K words, minimum. The price includes one year of product support, as well as source code, distributed on a disc. Contact Mike Corder, Computer Applications Corp., 413 Kellog, Ames, IA 50010, (515) 232-8181.

Newsweek

NEWSWEEK NOTES HOME COMPUTING

The August 23rd issue of *Newsweek* carried almost a full-page article titled, "Computers: Home Cooking." It was the usual story of a couple of example installations—human interest stuff—with the usual inaccurate underestimates of number of stores and clubs, and the usual interview with Paul Terrell of Byte, Inc.

```
336          372 350 037      JM ADD          IF LO PORT, ADD 2
341          326 002          SUI 2           IF HI PORT, SUB 2
343    PUT   167              MOV M,A         PUT REVISED DATA BACK
344          023              INX D           NEXT DATA
345          303 323 037      JMP GET         DO IT AGAIN
350    ADD   306 002          ADI 2
352          303 343 037      JMP PUT

355    DATA  301      *321     **170          FIRST DATA IS FOR 8K
356          004              **003          BASIC ON PAPER TAPE.
357          311      *331    **200          * INDICATES DATA TO
360          004              **003          BE SUBSTITUTED FOR
361          314      *334    **203          CASSETTE VERSION.
362          004              **003          ** INDICATES DATA FOR
363          323      *343    **212          4K BASIC
364          004              **003
365          176      *216    **164
366          005              **004
367          332      *352    **377
370          005
371          377
```

OCTAL DEBUGGING PROGRAM (ODT-80) FOR THE MCS-80 COMPUTER

by E. R. Fisher

Abstract

ODT-80 is an octal debugging routine for use on the Intel 8080 microprocessor. This routine provides the capability to examine and modify all of the memory that is available to the microcomputer and transfer program control to the created program. ODT-80 makes use of simple keyboard commands from any terminal—such as a teletypewriter—that is attached to the system.

Introduction

ODT-80 (Octal Debugging Technique) is a program written for the MCS-80 that allows the user to modify a program via a teletypewriter keyboard. The program occupies 400 octal words and must be located in the lowest memory page of the MCS-80 system, since the program uses the RESTART instructions.

ODT has been proved to be an effective aid to debugging on microprocessors. The first version,[1] for the Intel 8008,* has been "front panel" for virtually hundreds of microprocessor applications. The author submits this ODT for the 8080 in hopes that the tradition of *soft* front panels may be perpetuated.

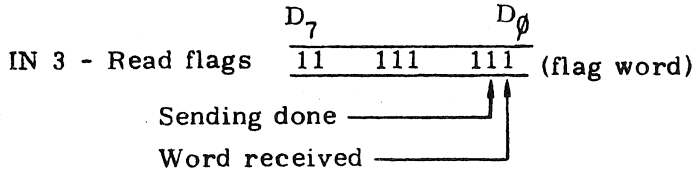
System Requirements

All addresses of memory locations and contents of memory locations are referred to in octal numbers. A question mark (?) will be typed for any illegal input.

The minimum system requirements for using ODT are as follows:

- MCS-80 computer set
- ODT programmable read only memory (PROM) at memory page 000_8
- 256 word (RAM) at page 010_8
- Teletype interface with the following codes:
  OUT 2—SEND ASCII character

IN 2 - Input word from TTY



Example

The following example illustrates a typical debugging operation. It is assumed that a program has been assembled and that the program is to be loaded by a loader in PROM 2.

A program listing and a detailed description of the command set follow the debugging description.

1. Set the memory field with the S command, and read the tape with the R command.

```
10S*      ; SET MEMORY FIELD TO 10
0R        ; START READER ON TTY
?         ; WHEN TAPE IS READ IN ODT
            RESTARTS (startup of ODT de-
            pends upon the reader program
            in PROM 2)
```

2. Start the program with the G command.

```
10S       ; START PROGRAM AT LOCAT-
            ION 0
0G
TEST PWOGRAM

          ; TYPED BY PROGRAM WITH AN
            ERROR
```

"*P.E.* ARTICLES TELL HOW TO BUILD A SYSTEM FOR UNDER $100"

Dear Bob, July 16, 1976

For some time now I've felt that the hardware oriented types (like me) who only want to spend $100 or so to get started have been completely neglected. I've also been approached by many hobby groups who want to know more about the COSMAC microprocessor. To satisfy both types of potential users, I've prepared a series of articles showing how to build a complete COSMAC microcomputer (called ELF) for under $100. This series of articles started in the August, 1976 issue of *Popular Electronics*, and should be of interest to many of your readers who are still looking for a lower cost way of getting started. The basic microcomputer doesn't need a terminal since input switches and an output display are provided. A novice user can actually do more with this system initially than he could with a $466 Altair 680 by itself (I have both systems). Of course, the COSMAC ELF has no cabinet or PC board and you must handwire 12 chips together. Other articles in the series will provide programs, applications, inexpensive interfaces, memory expansion, and a simple operating system. A very low cost graphic video display is also planned for the ELF microcomputer. I just want to make your less affluent readers aware that plans for a low-cost, do-it-yourself, micro-computer are now available via this series of articles.

Best wishes,

Joe Weisbecker 1220 Wayne Ave.
Erlton, Cherry Hill, NJ 08002

EDITOR'S NOTE: Joe is the architect/designer of RCA's COSMAC.

3. Change location (30) to correct output. Restart ODT.

```
?
10S
30 /   327 322)      ; CHANGE "W" TO AN "R"
0G
TEST PROGRAM         ; CORRECTED OUTPUT
```

EXAMPLE PROGRAM LISTING

```
1
 8080 MACRO ASSEMBLER, VER 1.1  ERRORS = 0 PAGE 1

                              ;ODT TEST PROGRAM
                              ;10-21-74
   000006                SEND EQU 6
   000370                CRLF EQU 370Q
                              ;
                              ;PROGRAM PROPER
                              ;
   004000                     ORG 4000Q
   004000   041 022 010       LXI H TABLE        ;SET UP ADDRESS
   004003   176          LOOP: MOV A,M            ;GET ENTRY
   004004   376 000            CPI 0
   004006   312 016 010       JZ ND              ;IF 0 END OF LIST
   004011   367                RST SEND           ;TYPE CHARACTER
   004012   043                INX H
   004013   303 003 010       JMP LOOP           ;LOOP

   004016   315 370 000 ND:    CALL CRLF          ;SEND CRLF

   004021   166                HLT
                              ;
   004022   324          TABLE: DB 324Q              ;T
   004023   305                 DB 305Q              ;E
   004024   323                 DB 323Q              ;S
   004025   324                 DB 324Q              ;T
   004026   340                 DB 240Q              ;SPACE
   004027   240                 DB 320Q              ;P
   004030   327                 DB 327Q              ;W FOR ERROR
   004031   317                 DB 317Q              ;O
   004032   307                 DB 307Q              ;G
   004033   322                 DB 322Q              ;R
   004034   301                 DB 301Q              ;A
   004035   315                 DB 315Q              ;M
   004036    Ø                  DB Ø
                               END

NO PROGRAM ERRORS

1
 8080 MACRO ASSEMBLER, VER 1.1  ERRORS = 0 PAGE 2

                        SYMBOL TABLE

   * 01

   A       000007    B       000000    C       000001    CRLF    000370
   D       000002    E       000003    H       000004    L       000005
   LOOP    004003    M       000006    ND      004015    PSW     000006
   SEND    000006    SP      000006    TABLE   004021
```

Commands

(n_8/) - SLASH — Open the n_8 address and type contents in octal.

(LF) - LINE FEED — Close the currently open address, and open the next sequential address. The contents of the open register may be changed by typing the octal number to be input and then typing a (CR).

(.) - PERIOD — This operator may be used before the (/) operator to open the address last used.

(n_8S) - The SET command is used to set the H register to the memory page to be accessed.
(EX) 10S ; H=010_8

(n_8R) - The READ operator is used to start a loader program in memory page No. 2, location 0. If this command is preceded by an "nS" command, the n value will be passed to the loader program in the H register.

(n_8G) - The GO routine is used to start a program in memory. The octal number typed before the "G" operator will set the starting address in memory. The "G" operator should be preceded by an "nS" command to select the desired page.

(CTRL-C) - CONTROL-C — Type out the top two locations in stack. (See trap.) Leaves the stack pointer at its initial position –2.

(CR) - CARRIAGE RETURN — Close the currently open address.

Utility Routines

The following subroutines are available to the user as utility routines for other programs:

Address	Call	
307	RST ODT	Restart ODT program. This is useful error branching in program debugging.
367	RST SEND	Send the ASCII character presently in the "A" register.
315 333 0	CAL READ	Wait for a character to be received from the teletype and return with the ASCII character in the "A" register. The "A" and "B" registers are used in this routine.
315 370 0	CALL CRLF	Send a carriage return and a line feed to the teletypewriter. The "A" register is used in this routine.
315 C 301 0	CAL OCTALP	Send a space and type in the three digit number in the "A" register. The A, B, and E registers are used in this routine.

Trap

The trap feature of ODT works as follows: When the central processor encounters a 377_8 as an instruction, the processor decodes this as a restart to location 70_8 in memory 0_8. An example of this is when a nonexistent memory is addressed, or when a 377_8 is placed in a program for a break point. At this time, ODT stores in the pushdown stack the address of the errant location and all of the registers. The trap indication is a "T" typed out on the teletypewriter. Control is now back in ODT and multiple Control-C's (CTRL-C) typed by the operator will yield, in sets of two,* the trapped information in the following format:

ERROR	ADDRESS +1	REGISTERS							
PAGE NO.	LOCATION	A	F	B	C	D	E	H	L
XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX

The "F" register is the status flags of the 8080, with the format shown in Fig. 1.

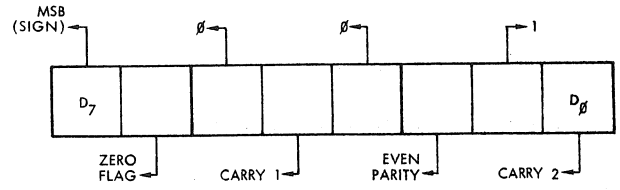


Fig. 1. "F" register format.

*One set for each CTRL-C typed.

A CTRL-C typed at any time will type out the push-down stack but *not* in the format shown above! The above above format is only available immediately after a "T" has been sent by ODT.

If a continuous string of "T" are sent to the TTY without stopping, this is an indication that the stack pointer is pointing at nonexistent memory. It will be necessary to restart ODT.

RESTART/INTERRUPT Locations

There are five segments of memory reserved in ODT for INTERRUPT or RESTART operation. These locations, shown in Fig. 2, contain jumps to the scratch RAM memory number 10, used by ODT. This allows interrupt service to be handled even though memory number 0 is preprogrammed to contain ODT.

Notes on the 8080

The stack pointer is reset to location 350_8 every time ODT is restarted via location 0 of ROM 0—in other words, whenever a "?" is sent by ODT. ODT and other programs use the stack pointer; therefore, memory locations plus and minus this location are apt to be overwritten. However, the stack pointer moves down in memory (high addresses to low addresses), so normal operation will keep the stack pointer in RAM 10_8.

Acknowledgements

I wish to acknowledge the work of Jim English, who coauthored with me the ODT for the 8080, from which a large part of this program was derived, and Walt Binge and Mike Maples, who helped on the preparation of this paper.

1. E. Fisher and J. English, *Octal Debugging Program (ODT) for MCS-8 Computer,* Lawrence Livermore Laboratory, Rept. UCID-16507 (November 13, 1972).

*Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Energy Research and Development Administration to the exclusion of others that may be suitable.

*Underlined typeout is typed by the user.

*One set of each CTRL-C typed.

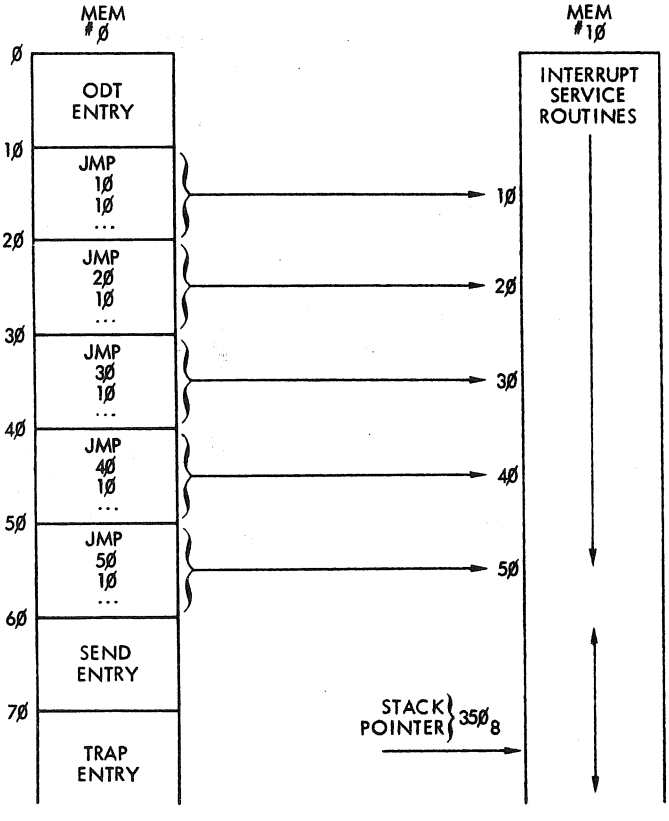


Fig. 2. RESTART/INTERRUPT locations.

ODT-80 Program Listing

```
8080 MACRO ASSEMBLER, VER 1.1

                        ;ODT PROGRAM FOR THE MCS-80
                        ;**********AUTHOR EUGENE R FISHER**********
                          ;4-20-74
                        ;
                              ORG 0
                        TTY EQU 2
                        SND EQU 6
                        FLAG EQU 3
                        ;
000000 061 350 010      ERROR:  LXI SP,4350Q    ;SET STACK POINTER,PAGE10,LOC 350
000003 076 277          MVI A,277Q              ;SEND A ?
000005 303 077 000      JMP ERR                 ;FIRST RESTART LOCATION
000010 303 010 010      JMP 4010Q
000013 000              NOP
000014 000              NOP
000015 000              NOP
000016 000              NOP
000017 000              NOP
000020 303 020 010      JMP 4020Q
000023 000              NOP
000024 000              NOP
000025 000              NOP
000026 000              NOP
000027 000              NOP
000030 303 030 010      JMP 4030Q
000033 000              NOP
000034 000              NOP
000035 000              NOP
000036 000              NOP
000037 000              NOP
000040 303 040 010      JMP 4040Q
```

```
000043  000                     NOP
000044  000                     NOP
000045  000                     NOP
000046  000                     NOP
000047  000                     NOP
000050  303 050 010             JMP 4050Q
000053  000                     NOP
000054  115             PER:    MOV C,L             ;ROUTINE TO MAKE PERIOD CURRENT LOCATION
000055  303 107 000             JMP NEXTC
000060  303 344 000     SEND:   JMP SEN             ;ENTRY POINT TO SEND ROUTINE
000063  315 370 000     GO:     CALL CRLF           ;SEND A CRLF
000066  151                     MOV L,C             ;SET THE L REG
000067  351                     PCHL        ;JMP TO STARTING ADDRESS
000070  343             TRAP:   XTHL                ;CHANGE ORDER OF STACK H AND L LAST OUT
000071  325                     PUSH D              ;FORTH OUT
000072  305                     PUSH B               ;THIRD OUT
000073  365                     PUSH    PSW             ;SECOND OUT
000074  345                     PUSH H              ;FIRST OUT IS THE ADR WHENCE WE CAME
000075  076 324                 MVI A,324Q  ;SEND A T
                                ;
                        ;THE STACK MAY BE DUMPED BY HITTING A CTRL-C TO GET THE FOLLOWING
                        ; SP+1  SP A F B C D E H L
                                ;
000077  367             ERR:    RST SND
000100  315 370 000             CALL CRLF
000103  257             BEGIN:  XRA A               ;CLEAR THE AC
000104  026 004         BEG:    MVI D,4
000106  117             SAV:    MOV C,A
000107  315 333 000     NEXTC:  CALL READ
000112  170                     MOV A,B
000113  326 270                 SUI 270Q            ;IS IT A NUMBER
000115  362 224 000             JP  TERM            ;270 MUST BE TERMINATOR OR ILLEGAL DIGIT
000120  170                     MOV A,B
000121  326 260                 SUI 260Q
000123  372 224 000             JM  TERM            ;YES
000126  025                     DCR D               ;>260 MUST BE DIGIT BUMP BUFFER CNT
000127  312 000 000             JZ  ERROR           ;BUFFER OVERFLOW
000132  107                     MOV B,A
000133  171                     MOV A,C             ;GET PREVIOUS INPUTS
000134  027                     RAL
000135  027                     RAL
000136  027                     RAL
000137  332 000 000             JC  ERROR           ;IF CARRY NUMBER WAS TOO BIG
000142  200                     ADD B
000143  303 106 000             JMP SAV
000146  151             SLASH:  MOV L,C             ;C CONTAINS ADDRESS
000147  176             GETCON: MOV A,M             ;GET CONTENTS
000150  315 301 000             CALL OCTALP
000153  076 240                 MVI A,240Q
000155  367                     RST SND
000156  303 103 000             JMP BEGIN
000161  172             LF:     MOV A,D
000162  326 004                 SUI 4
000164  312 170 000             JZ  NINP            ;IF CNT STILL 4 NO INPUT WAS RECEIVED
000167  161                     MOV M,C             ;IF CNT <4 DEPOSIT ONPUT IN MEMORY
000170  076 215         NINP:   MVI A,215Q
000172  367                     RST SND
000173  054                     INR L               ;BUMP ADDRESS
000174  174                     MOV A,H             ;GET HI PART
000175  315 301 000             CALL OCTALP         ;TYPE HI ADDRESS
000200  175                     MOV A,L             ;GET LOW PART
000201  315 301 000             CALL OCTALP         ;TYPE LOW ORDER ADDRESS
000204  303 147 000             JMP GETCON
000207  315 370 000     CR:     CALL CRLF           ;SEND A CRLF
000212  172                     MOV A,D
000213  326 004                 SUI 4               ;BUFCNT =4 ?
000215  312 103 000             JZ  BEGIN           ;YES NO INPUT SINCE LAST TERMINATOR
000220  161                     MOV M,C             ;LOAD MEMORY WITH INPUT
000221  303 103 000             JMP BEGIN           ;ERF
000224  170             TERM:   MOV A,B
000225  376 212                 CPI 212Q            ;IS IT A LF
000227  312 161 000             JZ  LF
000232  376 215                 CPI 215Q            ;IS IT RETURN
000234  312 207 000             JZ  CR
000237  376 322                 CPI 322Q            ;R
000241  312 000 002             JZ 1000Q            ;START READER PROGRAM
000244  376 257                 CPI 257Q
000246  312 146 000             JZ  SLASH
000251  376 307                 CPI 307Q
000253  312 063 000             JZ  GO              ;G
000256  376 256                 CPI 256Q            ;PERIOD
000260  312 054 000             JZ  PER
000263  376 203                 CPI 203Q
000265  312 357 000             JZ  CTRLC           ;CONTROL C
000270  376 323                 CPI 323Q            ;S
000272  302 000 000             JNZ ERROR           ;FOUND AN ERROR
000275  141             SETX:   MOV H,C             ;SAME HI ADDRESS
000276  303 100 000             JMP ERR+1
000301  006 004         OCTALP: MVI B,4              ;SET CNTR
000303  007                     RLC
000304  007                     RLC
000305  137                     MOV E,A             ;SAVE AC
000306  076 240                 MVI A,240Q          ;SEND A SPACE
000310  367                     RST SND
000311  173                     MOV A,E             ;GET THE AC
000312  346 003                 ANI 3               ;MASK
000314  005             DECR:   DCR B               ;DECR
000315  310                     RZ                  ;DONE ?
000316  306 260                 ADI 260Q            ;NO
000320  367                     RST SND         ;SEND A CHARAACTER
000321  173                     MOV A,E             ;GET AC
000322  007                     RLC
000323  007                     RLC
000324  007                     RLC
000325  137                     MOV E,A             ;SAVE AC
000326  346 007                 ANI 7               ;MASK
000330  303 314 000             JMP DECR
000333  333 003         READ:   IN FLAG             ;ROUTINE TO READ ONE CHAR FROM TTY
000335  037                     RAR
000336  322 333 000             JNC READ
000341  333 002                 IN TTY
000343  107                     MOV B,A
000344  323 002         SEN:    OUT TTY             ;ROUTINE TO OUTPUT AN ASCII CHAR
000346  333 003         SEN1:   IN FLAG
000350  037                     RAR
000351  037                     RAR
000352  170                     MOV A,B               ;RESTORE A REG FROM READ
000353  330                     RC                    ;DONE?
000354  303 346 000             JMP SEN1              ;NO
000357  341             CTRLC:  POP H
000360  174                     MOV A,H
000361  315 301 000             CALL OCTALP
000364  175                     MOV A,L
000365  303 150 000             JMP GETCON+1
000370  076 215         CRLF:   MVI A,215Q          ;SEND A CRLF
000372  367                     RST SND
000373  076 212                 MVI A,212Q
000375  303 344 000             JMP SEN             ;RETURN VIA SEND
                                END
```

NO PROGRAM ERRORS

SYMBOL TABLE

* 01

A	000007	B	000000	BEG	000104 *	BEGIN	000103
C	000001	CR	000207	CRLF	000370	CTRLC	000357
D	000002	DECR	000314	E	000003	ERR	000077
ERROR	000000	FLAG	000003	GETCO	000147	GO	000063
H	000004	L	000005	LF	000161	M	000006
NEXTC	000107	NINP	000170	OCTAL	000301	PER	000054
PSW	000006	READ	000333	SAV	000106	SEN	000344
SEN1	000346	SEND	000060 *	SETX	000275 *	SLASH	000146
SND	000006	SP	000006	TERM	000224	TRAP	000070 *
TTY	000002						

OOPS!: A Line-Drawing Game for Your Video Terminal

–Martin R. Winzenread

Go Only Where You Haven't Gone Before

You control a cursor with your right four sense switches in any combination. Try to draw a picture on the screen with a line that never goes through the same point twice.

E.g. *** can be drawn, but ***** can not.

An '⁓' indicates that you have been there before. You may then restart the program by putting the right four sense switches down.

To change the speed of the cursor, change memory locations 0026 and 002C.

*Change these statements if your VDM is not addressed as port 8C and memory 8800-8BFF

```
0000 31 90 00    0900     LXI 6,SP+6       Set stack pointer.
0003 3E 00       1000     MVI A,0          } Initialize screen
0005 D3 8C       1010     OUT 8CH *
0007 21 00 88    1020 AG  LXI H,8800H *
000A 36 20       1030 ST  MVI M,20H
000C 23          1040     INX H
000D 7C          1050     MOV A,H
000E FE 8C       1060     CPI 8CH *
0010 C2 0A 00    1070     JNZ ST
0013 21 20 8A    2000 N1  LXI H,8A20H *    Start at center of screen
0016 7E          2010 N2  MOV A,M          } If we have been there before halt
0017 FE 2A       2011     CPI 2AH
0019 CA 58 00    2012     JZ HLT
001C 36 07       2013     MVI M,7
001E 78          2020 N4  MOV A,B          Display cursor
001F E6 03       2023     ANI 3            } WAIT (It's longer for up and down
0021 CA 2A 00    2025     JZ N5              moves than for right and left
0024 01 FF 1F    2027     LXI B,1FFFH        moves)
0027 C3 2D 00    2030     JMP N3
002A 01 FF 0F    2033 N5  LXI B,0FFFH
002D 0B          2040 N3  DCX B
002E 78          2050     MOV A,B
002F FE 07       2055     CPI A,0
0031 C2 2D 00    2060     JNZ N3
0034 DB FF       3000 IN  IN 0FFH          } Read right four sense switches
0036 E6 0F       3001     ANI 0FH
0038 CA 34 00    3002     JZ IN
003B 36 2A       3005     MVI M,2AH        Insert '*'
                                           If you change this, change address
                                             0018 also
003D 47          3010     MOV B,A          } Move
003E E6 01       3020     ANI 1
0040 C4 76 00    3025     CNZ UP
0043 78          3030     MOV A,B
0044 E6 02       3035     ANI 2
0046 C4 81 00    3036     CNZ DN
0049 78          3040     MOV A,B
004A E6 04       3050     ANI 4
004C C4 66 00    3055     CNZ RT
004F 78          3060     MOV A,B
0050 E6 08       3065     ANI 8
0052 C4 6E 00    3070     CNZ LF
0055 C3 16 00    5000     JMP N2
                                   —SUBROUTINES—
0058 36 04       7100 HLT MVI M,4          } Halt and restart
005A DB FF       7110 A6  IN 0FFH
005C E6 7F       7115     ANI 7FH
005E FE 00       7120     CPI 0
0060 C2 5A 00    7130     JNZ A6
0063 C3 07 00    7140     JMP AG
0066 23          8100 RT  INX H            } Move right
0067 7C          8110     MOV A,H
0068 FE 8C       8120     CPI 8CH *
006A C0          8130     RNZ
006B 26 88       8140     MVI H,88H *
006D C9          8190     RET
006E 2B          8200 LF  DCX H            } Move left
006F 7C          8210     MOV A,H
0070 FE 87       8220     CPI 87H *
0072 C0          8230     RNZ
0073 26 8B       8240     MVI H,8BH *
0075 C9          8290     RET
0076 11 C0 FF    8300 UP  LXI D,0FFC0H     } Move up
0079 19          8310     DAD D
007A 7C          8320     MOV A,H
007B E6 FB       8330     ANI 0FBH *
007D F6 08       8340     ORI 08H *
007F 67          8350     MOV H,A
0080 C9          8390     RET
0081 11 40 00    8400 DN  LXI D,40H        } Move down
0084 19          8410     DAD D
0085 7C          8420     MOV A,H
0086 E6 FB       8430     ANI 0FBH *
0088 67          8440     MOV H,A
0089 C9          8490     RET
008A             9000 SP  DS 6
```

Trace font

SCRUNCH (Pass 1):

A Compactor for BASIC in 8080

—Martin R. Winzenread

Scrunch readable source code into a form that only a computer could love. Run bigger programs with less memory. Input original source code by paper tape, then dump your scrunched code to paper tape or cassette for later use.

I am presently working on PASS 2 which eliminates unnecessary line numbers by using the multiple statement per line feature.

```
0000 31 B7 00  1000  LXI SP,AR+6      Highest available memory
0003 01 FF 8B  1010  LXI B,MAX        Lowest available memory
0006 21 00 88  1020  LXI H,START      Could use 00B8
0009 11 00 88  1030  LXI D,START
000C CD 8B 00  1050  CIRA CALL BYTE
000F FE 22     2030  QUO CPI 22H (")  Keeps text between quotes intact
0011 C2 22 00  2040  JNZ LET
0014 CD A6 00  2050  AG CALL STBY
0017 CD 8B 00  2060  CALL BYTE
001A FE 22     2070  CPI 22H (")
001C C2 14 00  2080  JNZ AG
001F C3 85 00  2090  JMP CIRB
0022 FE 4C     2100  LET CPI 'L'      Eliminates "LET" from LET state-
0024 C2 3C 00  2110  JNZ PR.           ments.
0027 CD 79 00  2120  CALL NEXT
002A FE 45     2140  CPI 'E'
002C C2 85 00  2150  JNZ CIRB
002F CD 79 00  2160  CALL NEXT
0032 FE 54     2180  CPI 'T'
0034 C2 85 00  2190  JNZ CIRB
0037 2B        2200  DCX H
0038 2B        2210  DCX H
0039 C3 0C 00  2220  JMP CIRA
003C FE 50     2500  PRI CPI 'P'      Replaces "PRINT" with "?" in
003E C2 58 00  2501  JNZ BLK           PRINT statements.
0041 CD 79 00  2530  CALL NEXT
0044 FE 52     2540  CPI 'R'
0046 C2 85 00  2550  JNZ CIRB
0049 CD 8B 00  2590  CALL BYTE
004C CD 8B 00  2600  CALL BYTE
004F CD 8B 00  2610  CALL BYTE
0052 2B        2620  DCX H
0053 3E 3F     2640  MVI A,'?'
0055 C3 85 00  2650  JMP CIRB
0058 FE 20     3000  BLK CPI 20H ( )   Eliminates spaces
005A CA 0C 00  3010  JZ CIRA
005D FE 52     3020  CPI 'R'
005F C2 85 00  3030  JNZ CIRB
0062 CD 79 00  3040  CALL NEXT
0065 FE 45     3060  CPI 'E'          Eliminates REM statements
0067 C2 85 00  3070  JNZ CIRB           entirely
006A CD 79 00  3080  CALL NEXT
006D FE 4D     4000  CPI 'M'
006F C2 85 00  4010  JNZ CIRB
0072 EB        4020  XCHG
0073 CD 8B 00  4030  MO CALL BYTE
0076 C3 73 00  4040  JMP MO
0079 CD A6 00  4800  NEXT CALL STBY   Useful subroutines
007C CD 8B 00  4810  N2 CALL BYTE
007F FE 20     4820  CPI ' '
0081 CA 7C 00  4830  JZ N2
0084 C9        4840  RET
0085 CD A6 00  4980  CIRB CALL STBY
0088 C3 0C 00  4990  JMP CIRA
008B DB 00     5000  BYTE IN 0        Input Routine
008D E6 20     5005  ANI 20H
008F CA 8B 00  5010  JZ BYTE
0092 DB 01     5020  IN 1
0094 D3 01     5030  OUT 1
0096 E6 7F     5040  ANI 7FH          Used as end of file
0098 FE 12     5050  CPI 12H (CNTR R) Might use JZ to a binary dump of
009A CA 9A 00  5060  HLT JZ HLT             memory from START to H,L
009D FE 0D     5080  CPI 0DH (CR)     Points to last (CR) for use in REM
009F C0        5090  RNZ                routine
00A0 D1        6000  POP D
00A1 5D        6010  MOV E,L
00A2 54        6020  MOV D,H
00A3 C3 85 00  6030  JMP CIRB
00A6 77        7000  STBY MOV M,A     Store a byte and check for end of
00A7 23        7010  INX H              usable RAM
00A8 78        7020  MOV A,B
00A9 BC        7030  CMP H
00AA C0        7040  RNZ
00AB 79        7050  MOV A,C
00AC BD        7060  CMP L
00AD C0        7070  RNZ
00AE C3 AE 00  7080  HE JMP HE
00B1           8000  START EQU 8800H
00B1           8010  AR  DS 6
00B7           8020  MAX EQU 8BFFH
```

TINY HI

by Martin Buchanan

Dear *DDJ*, Sept. 2, 1976

For the past eight months, I've worked on designing my own programming language. I've now conceived several clean and fairly powerful procedural languages. The simplest is described in the attached article. I also expect to specify TINY LISP, and TINY SNOBOL in coming months. Core is cheap compared to people, and the major economy I want is in syntax, number of data types, and number of data structures seen by the user. One key to such economy is the elimination of the *multilevel approach* and the *specialization* that characterizes much existing system software. Software that is "OS dependent" is even less portable than machine code on many systems. I don't want the user to have to learn assembler, editor, JCL, file handling packages, etc. I want him to see conceptually, and syntactically integrated systems.

During the latter part of my spinning of imaginary languages, I've become dissatisfied with my previous definition of HI, as a procedural language which is both cleaner and more useful than PL/1, and incorporating associative retrieval, pattern matching, and list processing facilities. Several ideas have changed my direction:

1. A tiny language (or any language) should be *extensible.*
2. Most existing extensible languages (LISP, TRAC, FORTH) are limited by primitive syntax.
3. The concept of "set" or "relation" is powerful enough to include all the diverse data structures found in computer applications, and to manipulate them with common tools.
4. A further equivalence can be established between relations and functions; one is defined extensionally; the other is defined operationally.
5. The relational calculus is a powerful form for manipulating such structures. In the past, it has been restricted to data base operations (Codd's DSL ALPHA).
6. Declarations make it feasible for the user to see only a single data structure, ignoring the different *internal* representations of structures.

I'm attempting to design an extensible relational language with a context-sensitive grammar. Work is presently stalled by a 50-hour work week, 17 semester-hours of school, my home computer, and a promised article.

FORTH is a fascinating new language ($1000 for 8080 systems). I've ordered manuals, and will report on what I find out.

DDJ is great; things like your floating point routines for the 6502 (which MOS Technology did not have available) really help.

Design for the future! In 5 years your home computer will be a 64K memory, 16 bit dynamically microprogrammable processor, four billion bytes of fast mass storage, and goodies I can't imagine.

Keep up the good work.

Martin Buchanan
2040 Lord Fairfax Rd.
Vienna, VA 22180
(703) 893-7978

HI is a family of general purpose programming languages designed in 1976 by Martin Buchanan. TINY HI (TINY) is the simplest version of HI designed, and will be implemented on the author's IMSAI 8080 system when I find the time.

TINY is a hybrid interpreter-compiler running on a dedicated virtual machine with virtual memory. TINY is completely structured. There are no GOTO statements or labels. There is no COMMON or RETURN; procedures have one entrance and one exit. TINY is a debugging translator, insuring that variables are defined, monitoring subscript range, etc.

TINY supports two data types, INTEGER, and STRING, and one data structure, the vector. Memory is allocated dynamically, and data type is determined at execution time, or by context, eliminating declarations.

The simplicity of TINY syntax, the small number of well defined operators, the ability to manipulate vectors, convenient input-output, and flexible commenting and mnemonic naming facilities, all combine to make TINY a very clean, easy to learn and use, language.

In the examples below, both interpretive sessions and programs are excerpted. TINY outputs are underscored. ↓ is used to indicate a carriage return.

Arithmetic:

Numbers in TINY are integers i: $-2^{15} \leq i \leq 2^{15}-1$

4 infix arithmetic operators are defined:

+ addition
- subtraction
*multiplication
/ division

Division by zero or overflow will produce an error message.

1 prefix arithmetic operator is defined:

- negation

Negation cannot produce an error.

The hierarchy of operators in TINY is:

highest	()	[]	nesting or subscripting
	♭		(Blank) concatenation
	#	(number)	- (negation)
	*	/	
	+	−	
lowest	=	< >	<= > = < >

Expressions are evaluated from left to right.

Vectors:

Numbers may be concatenated to form vectors:

```
A ← 1 2
A ← A 3 4
A
    1   2   3   4
```

The left pointing arrow, ←, is assignment. Any expression standing alone on a line is output. Vectors of

equal length may be combined in expressions:

```
1 2 3 + 4 5 6
    5   7   9
```

Vectors of length one may be added, subtracted, etc. to another vector and be "distributed":

```
1 + 4 5 6
    5   6   7
```

The prefix operator # (number) returns the number of elements in a vector or string:

```
#5 73 -1
   3
```

Both integer vectors and character vectors (strings) are limited to 255 elements.

Strings:

A string is a sequence of characters, other than ", enclosed in quotes:

```
"THIS IS A STRING"
```

A string may contain no characters:

```
" "  /* the null string */
```

Strings may be concatenated:

```
A ← "JOHN"
B ← "AND"
C ← "MARY"
A  B  C
JOHNANDMARY
```

Subscripting:

Vectors and strings can be subscripted by integer expressions:

```
A ← "ALGEBRA"
A[5 1 3 2]
BAGL
```

Note that: x[i j k] is equivalent to: x[i] x[j] x[k]

Global variables:

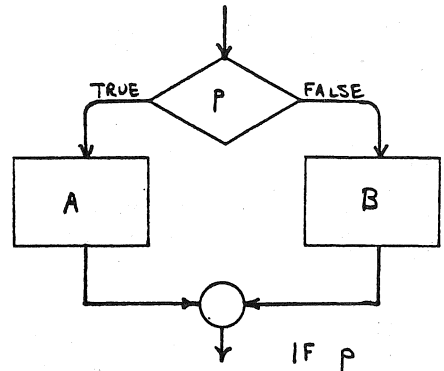
Globals are available to all TINY programs, and normally exist before and after program execution. I/O devices, data files, programs, and convenient system constants (.TRUE) are all good globals. All variables defined in the interpreter are globals. Globals are distinguished by use of the period prefix when referenced:

```
.SIN
.EMPLOYEE-FILE
.PRINTER
```

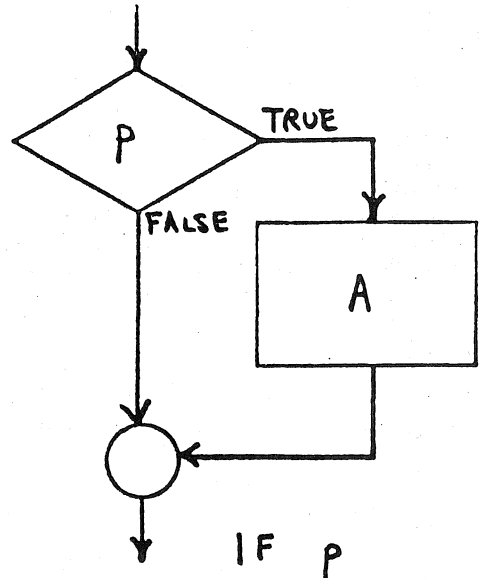
Programs:

TINY programs have only one entrance and one exit. Parameters are passed by location. Assignment to parameters is prohibited within a program. Programs can return vector or string values, and are invoked by use in

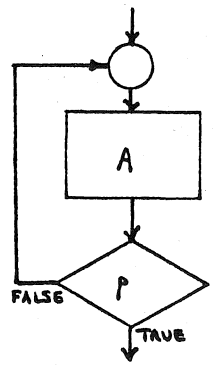
TINY CONTROL STRUCTURES



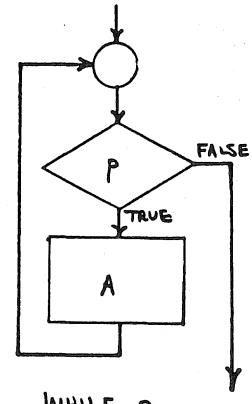

```
IF p
   code A
ELSE
   code B
END
```




```
IF p
   code A
END
```




```
UNTIL p
   code A
END
```




```
WHILE p
   code A
END
```

an expression. Sample program:

```
BEGIN MOD (A,B)
/*
  RETURN THE REMAINDER OF A/B
*/
  MOD ← A−B*(A/B)
  IF MOD<0
    MOD ← −MOD
  END
END

BEGIN GCD              /*
    ?X ?Y                USE EUCLID'S ALGOR-
  IF X <Y                ITHM TO FIND GREAT-
    T ← X                EST COMMON FACTOR
    X ← Y                OF X AND Y.
    Y ← T              */
    T ←
  END IF
  R ← Y
  WHILE R > 0
    R ← .MOD (X,Y)
    X ← Y
    Y ← R
  END WHILE
  "GCD=" Y
END GCD
```

The program GCD has several unusual features. ?X ?Y forces the evaluation of the expression ?X ?Y, which inputs X and Y, without making any assignment. T ← deletes T, allowing the compiler to reuse the allocated space. As shown by END IF, END WHILE, and END GCD, "noise" may be added to END statements; normally this is used to indicate the structure ended. /* begins the reservation of all columns to the right of, and including /* for comments, until */ is encountered. In TINY it is easy to implement Gerard Weinberg's concept of placing comments on the right hand side of the page, to be covered during debugging.

The program as shown is unusual; except in introductory programming classes, main programs do not calculate greatest common factors; also the mnemonic name is inconsistent with the use of "greatest common *factor*," rather than "divisor." To rewrite the program as a function, it should also be changed so that the values of the parameters are not changed.*:

*This change is now *required* for program correctness reasons.

```
BEGIN .GCF (XD,YD)     /*
  X ← XD                 USE EUCLID'S ALGOR-
  Y ← YD                 ITHM TO FIND GREAT-
  IF X <Y                EST COMMON FACTOR OF
    T ← X                XD AND YD.
    X ← T              */
    Y ← T
    T ←
   END IF
  R ← Y
  WHILE R > 0
    R ← .MOD(X,Y)
    X ← Y
    Y ← R
  END WHILE
END .GCF
```

Flow of control:

TINY control is always based on an expression which is true or false, called a *predicate.* TINY does not include logical variables or operators. All TINY predicates are of the form:

<exp > <relational > <exp>

The two expressions must agree in type. Vectors or strings must be of equal length to be equal. The first element of a string or vector is most significant when evaluating a predicate; the last element is least significant. TINY has two basic control structures with a total of four variants:

Do one thing or another depending on a predicate;
Repeat something until a predicate changes.

These structures are illustrated in the accompanying figures.

Input-output:

To output an expression, simply place it on a separate line. If a string expression, the string is output. Integers are converted to a 6 character string with leading zeroes replaced by blanks and any minus sign to the immediate left of the most significant digit. In output, concatenation between integer and string expressions is allowed:

```
"A+B=" (A+B)
```

To input a variable, *anywhere,* place a question mark before the variable. Input is free format. Input and output may be combined in one statement, like the INPUT statement of some BASICs:

```
"A=" ?A
A=37
```

Operating System (OS):

TINY is its own operating system. Users run on a dedicated machine (their own micro) or a virtual dedicated machine. Programs are automatically compiled when defined and recompiled whenever source code is changed. TINY will include access to a virtual machine level where relocatable and reentrant machine code may be written using an assembler. This level cannot be defined until subroutine linkages and data formats are defined in greater detail. TINY has a virtual memory; the user sees a single large homogeneous memory space. Files, programs, utilities, and system variables are all represented as global variables in this memory.

Conclusion:

I'd appreciate feedback about TINY design and implementation. My address is at the end of this article.

Program correctness:

It is straightforward to construct correctness proofs for TINY programs. Relevant features include:

Restricted control structures;
All programs have only a single entrance and exit;
Programs return only a single structure;
Parameters may not be altered by programs;
No floating point arithmetic (10.0*.1≠ 1.0);
External variables share a common name table, and must be explicitly indicated with the . prefix;
The deletion feature explicitly delimits the range of

temporary variables;
The simplicity of the language.

These features do not follow from the desire for a small language. GOTOs would be easy to add, and checking for assignment to parameters will actually cost core. I maintain that one should not add features to a language indiscriminantly, considering only available core or CPU time. Psychological factors can also make a language good or bad. PL/1, APL, "SUPER FORTRAN", "BASIC PLUS" are all baroque languages, whose descriptions occupy sizeable books and whose complete features are rarely mastered.

"I see a great future for very systematic and very modest programming languages." — *The Humble Programmer* by E. W. Dijkstra

TINY LANGUAGE SUMMARY

Vocabulary: BEGIN END IF ELSE WHILE UNTIL
Comments: /* */
Arithmetic infix operators: + − * /
Arithmetic prefix operators: -
Concatenation: ♭ (blank)
Length operator: # (number)
Relational operators:
Assignment:
Input: ?

Global: .
Data types: INTEGER, STRING
Data structures: the vector
Maximum number of elements in a vector: 255
Formats: Integers convert to a six character string

TINY ERROR MESSAGES

compile: assignment to parameter
mixed mode
value undefined
global undefined
unmatched parenthesis
missing end statement
syntax error

execution: overflow
division by zero
vector too long
string too long
subscript out of range

Direct letters about TINY HI to: Martin Buchanan
2040 Lord Fairfax Rd.
Vienna, VA 22180

TINY HI SYNTAX

Top Down

⟨PROGRAM⟩ ::= BEGIN ⟨NAME⟩ { (⟨NAME⟩ { , ⟨NAME⟩ }$_0^{\infty}$) }$_0^1$ ↵
{ ⟨PROGRAM⟩ }$_0^{\infty}$
⟨XCODE⟩
⟨END⟩

⟨XCODE⟩ ::= { ⟨XSTMT⟩ }$_0^{\infty}$

⟨XSTMT⟩ ::= ⟨IF⟩ | ⟨LOOP⟩ | ⟨ASNMT⟩ | ⟨IO⟩

⟨END⟩ ::= END ♭ ⟨ANYTHING⟩ ↵

⟨IF⟩ ::= IF ♭ ⟨PREDICATE⟩ ↵
⟨XCODE⟩
{ ELSE ↵
⟨XCODE⟩ }$_0^1$
⟨END⟩

⟨LOOP⟩ ::= ⟨LOOP TYPE⟩ ♭ ⟨PREDICATE⟩ ↵
⟨XCODE⟩
⟨END⟩

⟨LOOP TYPE⟩ ::= WHILE | UNTIL

⟨ASNMT⟩ ::= ⟨VARIABLE⟩ ← { ⟨EXPR⟩ }$_0^1$ ↵

⟨IO⟩ ::= ⟨EXPR⟩ { ♭ ⟨EXPR⟩ }$_0^{\infty}$ ↵

TINY HI SYNTAX

Bottom Up

⟨LETTER⟩ ::= A | B ⋯ | Z

⟨DIGIT⟩ ::= 0 | 1 ⋯ | 9

⟨ALPHA⟩ ::= ⟨LETTER⟩ | ⟨DIGIT⟩ | _

⟨INTEGER⟩ ::= { - }$_0^1$ { ⟨DIGIT⟩ }$_1^5$

⟨OPERATOR⟩ ::= + | - | * | /

⟨RELATIONAL⟩ ::= > | < | = | >= | <= | <>

⟨NAME⟩ ::= { ? }$_0^1$ ⟨LETTER⟩ { ⟨ALPHA⟩ }$_0^{254}$ |
{ . }$_0^1$ ⟨LETTER⟩ { ⟨ALPHA⟩ }$_0^{254}$

⟨VARIABLE⟩ ::= ⟨NAME⟩ { [⟨INTEGER EXP⟩] }$_0^1$

⟨INTEGER EXP⟩ ::= ⟨INTEGER⟩ | ⟨VARIABLE⟩ | ⟨FUNCTION⟩ |
⟨EXPR⟩ | - ⟨INTEGER EXP⟩ |
⟨INTEGER EXP⟩ ⟨OPERATOR⟩ ⟨INTEGER EXP⟩ |
⟨INTEGER EXP⟩ ♭ ⟨INTEGER EXP⟩

⟨FUNCTION⟩ ::= ⟨NAME⟩ (⟨EXPR⟩ { , ⟨EXPR⟩ }$_0^{\infty}$)

⟨EXPR⟩ ::= ⟨INTEGER EXP⟩ | ⟨STRING EXP⟩

⟨STRING EXP⟩ ::= ⟨STRING⟩ | ⟨VARIABLE⟩ | ⟨FUNCTION⟩ |
⟨STRING EXP⟩ ♭ ⟨STRING EXP⟩

⟨PREDICATE⟩ ::= ⟨INTEGER EXP⟩ ⟨RELATIONAL⟩ ⟨INTEGER EXP⟩ |
⟨STRING EXP⟩ ⟨RELATIONAL⟩ ⟨STRING EXP⟩

Book Review

COMPUTER POWER AND HUMAN REASON

Reviewed by Dave Caulkins, 437 Mundel Way, Los Altos, CA 94022, (415) 948-5753

Computer Power and Human Reason
by Joseph Weizenbaum
W. H. Freeman & Co., 1976
300 pp. $9.95

This is a book that should be read by anyone seriously interested in or affected by computers, a recommendation that includes most of the population of the U.S. The broad interest of the book is indicated by its selection as a Book-of-the-Month alternate. Professor Weizenbaum is a talented man who has been doing inpressive work in computer science for a long time. In this book he performs the difficult feat of humanistic perspective. Weizenbaum's central theme is that the range of human capabilities and concerns is both profoundly different from and immensely larger than anything a computer can do. Too many people in the Artificial Intelligence community of computer science research believe that there is no human activity that cannot eventually be done by a computer. Weizenbaum is strongly critical of this view; he rightly sees it as the worst kind of blind hubris.

Some quotes from the book give the flavor of Weizenbaum's arguments:

What could be more obvious than the fact that, whatever intelligence a computer can muster, however it may be acquired, it must always and necessarily be absolutely alien to any and all authentic human concerns? The very asking of the question, "What does a judge (or a psychiatrist) know that we cannot tell a computer?" is a monstrous obscenity.

The theories–or, perhaps better said, the root metaphors–that have hypnotized the artificial intelligentsia, and large segments of the general public as well, have long ago determined that life is what is computable and only that. As Professor John McCarthy, head of Stanford University's Artificial Intelligence Laboratory said, "The only reason we have not yet succeeded in simulating every aspect of the real world is that we have been lacking a sufficiently powerful logical calculus. I am currently working on that problem."

Sometimes when my children were still little, my wife and I would stand over them as they lay sleeping in their beds. We spoke to each other in silence, rehearsing a scene as old as mankind itself. It is as Ionesco told his journal: "Not everything is unsayable in words, only the living truth."

In support of this central theme Weizenbaum deals with some necessary background material. He presents an explanation of the theoretical basis for the power of machine based information processing, including a treatment of Turing machines. These sections are not what I would call light reading, but attention to them offers even the lay reader considerable insight into computer functioning. Weizenbaum also has a fascinating theory about the existence of hackers (compulsive programmers). As one who enjoys periodic fixes of code-writing and debugging, I found this theory especially interesting.

Professor Weizenbaum has written a book of substantial importance. As man's information processing machines become ever more powerful and pervasive, it is vital that we use computers for human purposes, and not the other way around.

TUTORIALS FROM THE IEEE COMPUTER SOCIETY

Minicomputers and Microprocessors: A Tutorial, Gaithersburg, Maryland, May 26, 1976–271 pp.

Review of recent developments in both minicomputer and microprocessor technology.

Each chapter begins with a short section which introduces the topics to be discussed and also highlights the significant points found in the chapter. Topic headings include technological advances, microprogramming, minicomputer architectures, microprocessors, mini and micro computer system development, and interfacing and peripherals.–$12.00

Microprogramming: A Tutorial, Long Beach, California, May 17, 1975–309 pp.

Text includes an introduction by the instructors, J. Michael Galey of IBM and Richard L. Kleir of Browne and Ramamoorthy, Inc., and 19 papers on microprogramming in six general categories: introductory papers, microprogrammed architecture and languages, production of microprograms, emulation and direct execution, microprogram applications other than instruction execution, conclusions, and perspectives.–$12.00; $15.00 Hardbound

Minicomputer Realtime Executives, from COMPCON 74 Fall, September 9, 1974–208 pp.

Of central importance to the economic realization of minicomputer applications is the realtime executive. Applications such as data acquisition, process control, and communication, are briefly discussed to identify their needs in a realtime executive. The main elements of these executives discussed include memory management, scheduling, data structuring, file handling, interprogram communication, and control structures.–$15.00

Tutorial on Computer Peripherals, from COMPCON 74 Spring, February 25, 1974–48 pp.

Digest presents broad perspective of peripherals industry. Excellent preparation for students, generalists, and those who have not been working directly in the field. Topics include trends in computer hardware and remote terminal systems, magnetic recording technology, use of peripheral devices in remote systems, and future projections in hardware and software. –Microfiche only, $9.00

Structured Programming, from COMPCON 75 Fall, September 8, 1975–241 pp.

This tutorial includes an introduction to the major ideas of structured programming, and their theoretical and practical foundations; a detailed presentation of a major application program system, and a presentation of program development stressing the top-down evolutionary character of the process.–$12.00; $15.50 Hardbound.

Computer Networks: A Tutorial, Gaithersburg, Maryland, June 18, 1975–296 pp.

The tutorial convers the areas of network topology and communications media, network technology and its effect on network performance, resource sharing requirements and techniques, and new approaches to network performance measurement, network access and network management. The general approach enables presentation of terminology and concepts followed by specific examples. The presentation stresses breadth rather than depth, and in turn provides network novices with a firm basis for becoming better educated on specific topics.–$12.00

Send Order To: IEEE Computer Society, 5855 Naples Plaza, Suite 301, Long Beach, CA 90803

PASCAL, SEXISM, VRAM, STRUCTURED PROGRAMS, VARIABLE SPACING & GAMES

Dear Sir: Aug. 23, 1976

I just received issue no. 7 and back issues 1-6 in the mail Saturday and spent all afternoon reading them.

Specific comments: Vol. 1, no. 4, pg. 4; and no. 6, pg. 6 refer to PASCAL as being developed by Niklaus Wirth—the language was actually developed by Niklaus Wirth *and* Kathleen Jensen. **[Hmph! He never mentioned that when I took his compiler course at Stanford!]** If you insist on dropping Kathleen's name someone might accuse you of being sexist and . . . (by the way, it is just coincidence that everything published in *DDJ* so far has come from males, isn't it?). **[No. It's the frustrating fact that we have yet to receive any manuscripts from women.]**

It was mentioned that you were planning on writing a Small PASCAL **[Nope. Just thinking of writing some design articles to encourage implementation of something better than BASIC.]**; two documents that would be *very* helpful are: *The PASCAL 'P' Compiler: Implementation Notes,* by Nori, Ammann, Jensen, and Nageli; and *PASCAL-S PASCAL-S: A subset and its Implementation,* by N. Wirth. These are numbers 10 and 12 from:

Institut fuer Informatik
Eidgenossische Technische Hochschule
Clausiusstrasse 55
CH-8006 Zuerich, Schweiz

Institute for Information
Federal Institute of Technology
Clausiusstrasse 55
CH-8006 Zurich, Switzerland

PASCAL-S is the "semi-official" subset of PASCAL. It consists of one program subdivided into a compiler which generates code for an abstract stack computer and an interpreter which interprets this code for an actual machine. The abstract computer is organized as a stack, 2 stack indexes, a program counter, an instruction register, a program status register (and a DISPLAY array used to speed up the addressing mechanism).

You might also be interested in the PASCAL Users Group and especially the PASCAL Newsletter, which contains a lot of information about PASCAL implementations, programming, features, etc. The PUG address is Univ. Comp. Ctr., 227 Exp. Engr., Univ. of Minnesota, Minneapolis, MN 55455. If you do finish a PASCAL implementation, please write to the Editors of the Newsletter because they are trying to keep a complete listing of all PASCAL implementations (as far as I know, the PDP series are the smallest machines on which PASCAL has been implemented to date).

From Vol. 1, no. 7, pg. 10—your comment on VRAM: there's another ringer besides the one you mentioned—unless this device meets the radiation limits of Part 15 of the FCC regulations, it's an illegal television transmitting station. **[Not applicable. Matrox VRAMs output video for a video monitor; not RF for television.]**

Vol. 1, no. 6, pg. 40—In F. J. Dickey's letter, he seems to equate structured programming with 'lots of control structures.' This seems to me to be precisely the opposite of the truth; one of the basics of structured programming is the use of only a very few control structures. Structured programming usually involves the use of only three basic control structures: sequence, if-then-else, and loops (usually repeat-until or while-do form). In fact, one of the important events in the history of structured programming was the publication of a proof that all programs could be written using only these three control structures. I hardly consider three structures "a lot of control structures." Perhaps Mr. Dickey was confused by PL/1, which he seems to consider as a simple structured programming language. PL/1 is most definitely not a structured programming language. Indeed, many advocates of structured programming have strongly criticized several aspects of PL/1.

Vol. 1, no. 6, pg. 28—Jim Day's article on variable character spacing is very interesting and such a scheme would be quite useful in producing good looking output. However, this must be restricted to use on final, human oritented output; it can not generally be used for program listings, etc. Variable character width destroys the visual character of program, and makes it nearly impossible to determine in which column a specific word begins, which is a significant factor in most languages. **[Not so. Most "good" languages are free field format.]** There are many examples in computer literature of much confusion caused by typesetters who reset program listings with variable width character spacing.

I've noticed several comments on games in *DDJ*. While I'm more interested in systems tape software being published, I'm somewhat upset at the large number of computer people who seem to enjoy running down games. I think games are very important as logical exercises. The importance of games in developing mental capabilities has been frequently pointed out by people ranging from Albert Einstein to R. Buckminster Fuller. As an example, the main field of study of John Von Neumann, one of the developers of the digital computer at the Institute for Advanced Studies, was in Game Theory. Some every interesting commentary to real world affairs is contained in the introduction to *Computers, Chess, and Long Range Planning,* by Mikhail Botvonnik, a book that I highly recommend. **[Agreed. But, then again, why not just play games for the fun of playing games?]**

Thanks very much for your kind attention in wading through this lengthy letter.

Yours truly,

Tim Bonham D605/1630 S. 6th St.
Minneapolis, MN 55454

"STM BABY & VIKING VISION"

Bob, July 29, 1976

. . . Speaking of neat things, the STM BABY! computer-in-an-attache-case [STM Systems, Mount Vernon, NH 03057] is the neatest thing I have seen in a long time. Not just a computer but also an ASCII-plus-Greek keyboard, cassette interface, video display generator, ROM monitor, etc., with a loudspeaker for audio output! Now, if they could only cut the price to $500 instead of $850 for their 2K system. [Photos and write-up in August *Byte*, pp. 122-124].

JPL reports that the on-board Viking computer is working OK. I hope it keeps on working--the nearest customer engineer is probably in Helium City. As astronomer Carl Sagan pointed out recently, animals on Mars may sleep during daylight hours because of the uncomfortably high temperatures (-20 F) and harmful ultraviolet rays. No one thought to mount a spotlight on Viking to allow pictures of Martian nightlife. The Viking camera did spot what looked like an inscription on a nearby rock, though; when translated (where is John Carter now that we need him?) it will probably be found to read, "Vikings go home!"

Sincerely,

Jim Day 17042 Gunther Street
Granada Hills, CA 91344

Another Hobbyist Magazine

WAYNE GREEN TO START PUBLISHING *KILOBYTE*

John Craig, Editor–KILOBYTE Magazine
RFD Box 100 D, Lompoc, CA 93436, (805) 735-1023

Editor's Note: John Craig, the new Editor of the about-to-be *Kilobyte* magazine, submitted this article to *Dr. Dobb's Journal* after a conversation with us. We are publishing it because (1) it gives some explicit information about what may be expected from the forthcoming magazine and (2) we wish to support active competition among the vendors of products to the computer hobbyist community. We believe that such competition in this free marketplace can do nothing but benefit the consumers.

Readers should *not* take our publication of this article as an implication of our endorsement or special support of *Kilobyte* over any of the other excellent periodicals serving our community. We enjoy *BYTE* and *Interface* and what we have seen of *Microtrek*. We expect to be equally pleased with *Kilobyte* and *Personal Computing*.

We would be happy to publish similar articles from the editors of *Microtrek* and *Personal Computing*. Our only requirement is that they contain explicit information and details about the magazine, along with the unbridled enthusiasm for his publication that should be expected from any magazine's Editor.

Well, Wayne Green [the original publisher of *BYTE*] has finally got around to doing it! The hobby community has been expecting him to start another magazine for quite some time . . . and here it is . . . *KILOBYTE*. If the name of this magazine brings about a reaction, don't worry about it, because you're not alone. The important thing is that it's a name you won't easily forget . . . and besides, what's in a name? Your reaction when you read the first issue is what will really count. (And you certainly ought to plan on getting that first issue, because it will very likely become a collector's item in the months to come. Look at what happened to the last magazine Wayne Green started.) Never mind getting any of the following issues . . . just get that first one!

Now, for the important question: "What will *KILOBYTE* have to offer that would make me want to shell out my hard-earned bucks for a subscription?" One of the most important areas we're going to cover will be basic articles concerning both hardware and software. This doesn't mean that *KILOBYTE* is going to be just a "beginner's magazine." Most of us are either hardware or software oriented and could use some good fundamental material in the "lacking" area which will bring us up to speed. And, when I say "good material", I mean just that. We're not going to be publishing articles of a dry, tutorial nature. If there's a hardware or software concept, or "lesson", to be passed on through the pages of *KILOBYTE*, it will be done using practical examples. The approach will not be one of, "This is how it is done . . ." but instead, "This is *why* it is done a certain way." Hardware articles will be related to various applications and the software articles will be tied in with programming problems and situations. Needless to say, the person who is a beginner will benefit from *KILOBYTE* as though it were directed at him exclusively.

Another area which *KILOBYTE* will be covering is applications. We will take a good look at small computer applications in both the home (entertainment, education, etc.) and especially in small businesses.

The *KILOBYTE* Software Library will be one of our big efforts, and probably one of our big headaches. There's a tremendous need for a central distribution point for small systems software. Much has already been developed . . . but it isn't getting around! We're going to be selling programs at a modest cost on cassettes with the author/programmer receiving a 15% royalty for his efforts. Hopefully, this will provide an incentive to some of those who have developed some good programs to share them (and make some good cash in the process).

Standardization is something else *KILOBYTE* will be tackling. There's no reason why a person shouldn't be able to buy a microcomputer which has standard plugs on the back for connecting his serial or parallel peripherals. The cassette situation is something which needs looking into, also. We've got the Kansas City standard for software exchange at a rather slow speed, but perhaps we need to come up with a standard for high-speed cassette operation, too. People sit around and talk about standardization all the time . . . others write about how something should be done . . . and then there are those who give seminars on the subject. The only way anything is ever going to be done about this (and other situations) is for *the industry* to get together and discuss it! Wayne Green and *KILOBYTE* will be working on this. The address is *KILOBYTE* Magazine. Petersborough, NH 03458. Be sure to include the "KILO . . ." Subscriptions will be $12/year or $25/3 years ($2 at the news stand). It will be in circulation around the end of December.

(AS USUAL) *ELECTRONICS* HAS A PASSLE OF GOOD HARDWARE ARTICLES

The September 2nd issue of *Electronics* magazine carried the following articles (among many others):

"Conferees Close in on 16K Standards" (for 16K-bit RAM)

"Shortage of Field Technicians Looms" (training programs for minicomputer techs)

"Fiber-Optic Data Transmission: A Practical, Low-Cost Technology" (Gigahertz bandwidth, noise-free)

"PROM Decoder Replaces Chip-Enabling Logic" (Design details)

"Check List for 4,096-bit RAMs Flags Potential Problems in Memory Design"

"Microprocessors Dominate Scene at Sold-Out Wescon"

"Character Generator Fits on One Chip" (Bipolar LSI gives 64 alphamerics for CRTs and matrix printers)

Electronics is published by McGraw-Hill, Inc., Box 514, Hightstown, NJ 08520. Subscriptions are $12/year or $30/3-years in the U.S.

JIM DAY NOTES . . .

Dear *DDJ*, September 8, 1976

I see that Martin Gardner has already reviewed Conway's book in the September issue of *Scientific American*. Note the ad on page 213 of the same issue, offering a FREE (for $1) GIANT CATALOG (64 pages) of computer stuff from Newman Computer Exchange, 3960 Varsity Drive, Dept. 23, Ann Arbor, MI 48104.

"Microprocessors Have Their Problems Too," by Jerry L. Ogdin, on p. 35 of the August issue of *Instruments & Control Systems*, is an interesting user's-eye-view of the state of the art. But one should keep in mind that the criteria of the industrial user are not always those of the hobbyist or educator.

Another interesting item is "Things That Go Byte in the Night," on p. 38 of the August issue of *Computer Decisions*. While amusing, this bit of whimsy would probably be of more interest to readers of *Creative Computing* rather than a dignified, scholarly journal like *PCC* or *DDJ*.

Jim Day
17042 Gunther St.
Granada Hills, CA 91344

BOOKSTORE

ACTIVE FILTER COOKBOOK
Don Lancaster. 1975. 240 pp. $14.95.

ADVANCED APPLICATIONS FOR POCKET CALCULATORS
Jack Gilbert. 1975. 304 pp. $5.95.

ADVANCED BASIC
James S. Coan. 1976. 184 pp. $6.95.

ALPHA-NUMERIC MUSIC WITH AMPLITUDE CONTROL
Malcolm Wright. 1975. 23 pp. $2. Xeroxed.

BIOFEEDBACK: Turning on the Power of your Mind
Marvin Karlins & Lewis Anderson. 1973. 190 pp. $1.25.

BIOFEEDBACK AND THE ARTS
Edited by David Rosenboom. 1976. 163 pp. $12.95. Hardbound.

BASIC
Albrecht, Finkel, & Brown. 1973. 325 pp. $3.95.

BASIC BASIC
James Coan. 1970. 256 pp. $3.95.

BASIC PROGRAMMING
Kemeny & Kurtz. 1961, 1971. 150 pp. $6.95.

THE BEST OF CREATIVE COMPUTING, Vol. 1
David Ahl, editor. 1976. 328 pp. $8.95.

BODY TIME
Gay Gaer Luce. 1973. 411 pp. $1.25.

THE BUGBOOK I & II with INSTRUCTOR'S WORKBOOK
Rony, Larsen, & Braden. 1974. $16.95. 2 volumes + workbook.

THE BUGBOOK III
Rony, Larsen, & Titus. 1975. $14.95.

BUILD YOUR OWN WORKING COMPUTER
David L. Heiserman. 1976. $5.95.

CALCULATOR CALCULUS
George McCarty. 1975. 254 pp. $8.75.

COMPUTER LIB/DREAM MACHINES
Theodore Nelson. 1974. 186 pp. $7.

COMPUTERS & COMPUTATION
Scientific American. 1950 - 1971. 280 pp. $6.

ELECTRONIC PROJECTS FOR MUSICIANS
Craig Anderton. 1975. 134 pp. $6.95.

FUNDAMENTALS & APPLICATIONS OF DIGITAL LOGIC CIRCUITS
Sol Libes. 1975. 192 pp. $5.98.

FUN & GAMES WITH THE COMPUTER
Edwin Sage. 1975. 360 pp. $5.95.

GAMES, TRICKS, & PUZZLES FOR A HAND CALCULATOR
Wallace Judd. 1974. 100 pp. $2.95.

GAMES WITH THE POCKET CALCULATOR
Thiagaragan & Stolovitch. 1976. 64 pp. $2.

GETTING THE MOST OUT OF YOUR ELECTRONIC CALCULATOR
William Hunter. 1974. 204 pp. $4.95.

INTRODUCTION TO MICROCOMPUTERS I & II
Adam Osborne & Associates, Inc. 1976. 384 pp. & 400 pp. $7.50 each.

MATH WRITING & GAMES IN THE OPEN CLASSROOM
Herbert Kohl. 1974. 252 pp. $2.45.

MY COMPUTER LIKES ME WHEN I SPEAK IN BASIC
Bob Albrecht. 1972. 64 pp. $2.

METRICALLY YOURS
Helen Hunter and Paul Wallach. 1976. 70 pp. $2.95.

MY COMPUTER LIKES ME WHEN I SPEAK IN BASIC
Bob Albrecht. 1972. 64 pp. $2.

MICROPROCESSOR/MICROPROGRAMMING HANDBOOK
Brice Ward. 1976. 294 pp. $6.95.

NEW MIND, NEW BODY; BIO-FEEDBACK: New Directions for the Mind
Barbara Brown, Ph.D. 1974. 523 pp. $2.50.

NUMBERS: SHORTCUTS & PASTIMES
Jack Gilbert. 1976. 336 pp. $6.95.

101 BASIC COMPUTER GAMES
David Ahl, editor. 1974. 250 pp. $7.50.

ON THE SENSATIONS OF TONE
Herman Helmholtz. 1885. $6.00.

PRINCIPLES & PRACTICE OF ELECTRONIC MUSIC
Gilbert Trythall. 1973. 214 pp. $6.95.

PROBLEMS FOR COMPUTER SOLUTION
Gruenberger & Jaffray. 1965. $7.95.

PROBABILITY
D.J. Koosis. 1973. 163 pp. $2.95.

PROFESSOR GOOGOL'S MATH PRIMER
Sam Valenza Jr. 1973. 144 pp. $3.25.

PROF E. McSQUARED'S (ORIGINAL FANTASTIC & SATISFYING CALCULUS PRIMER)
Swann & Johnson. 1975. 111 pp. $2.95.

PROF E. McSQUARED'S, PART II
Swann & Johnson. 1976. 103 pp. $2.75.

PROGRAMMING PROVERBS
Henry Ledgard. 1975. 134 pp. $5.95.

PCC GAMES' PROGRAM LISTINGS
PCC. 1974. 31 pp. $2.

STATISTICS
D.J. Koosis. 1972. 282 pp. $3.95.

TEACH YOURSELF BASIC, Volumes 1 & 2
Technica Education Corp. 1970. 64 pp each. $2.95 each.

TTL COOKBOOK
Don Lancaster. 1974. 328 pp. $7.95.

TV TYPEWRITER COOKBOOK
Don Lancaster. 1976. 256 pp. $9.95.

II CYBERNETIC FRONTIERS
Stewart Brand. 1974. 96 pp. $2.

THE UNIVERSAL TRAVELER
Don Koberg & Jim Bagnall. 1974. 128 pp. $4.95.

WHOLE EARTH EPILOG
Stewart Brand, editor. 1974. 318 pp. $4.

WHAT TO DO AFTER YOU HIT RETURN or PCC'S FIRST BOOK OF COMPUTER GAMES
PCC. 1975. 157 pp. $6.95.

List title and quantity for each item you wish to order. (Orders to be shipped within California require a sales tax remittance of 6%.) For orders less than $10, add $1 for postage and handling; for orders $10 and more, add $2. Send your order, along with your check or money order, to: PCC, Box 310, Menlo Park, CA 94025. Thank you.

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