

\$1.50

dr. dobb's journal of

COMPUTER

Calisthenics & Orthodontia

Running Light Without Overbyte

September, 1976

Box 310, Menlo Park CA 94025

Volume 1, Number 8

A REFERENCE JOURNAL FOR USERS OF HOME COMPUTERS



| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Realizable Fantasy: Computer Control of Music Tapes for Your Home Stereo – Jim C. Warren, Jr. <i>A systems project uniquely appropriate for your home computer</i> | 3 |
| A Realized Fantasy: Unlimited Speech Synthesis for Home Computers – Jim C. Warren, Jr. <i>A survey of computer speech synthesis systems including a \$395 completely assembled unit</i> | 7 |
| Plessy Does Right by Hobbyists – Significant Discounts on Memory & a 1000CPS Tape Reader | 11 |
| Proposal: A Board to Expand 8080 Instructions – Walt Ferris | 12 |
| Five Mass Memory Notes – Includes Info on a 256K Byte Core Memory for \$6500 | 13 |
| University of California Requests Suggestions About Stand-Alone Computers for its 120,000 Students | 14 |
| Personal Computers: A Bit of Wheat Amongst the Chaff – Jef Raskin <i>A critique of "little hidden gotchas" found in a multitude of kits</i> | 15 |
| 6800 SOFTWARE How to Make the 6800 Assembler Work in Your System – Tom Pittman | 19 |
| 6502 SOFTWARE A 6502 Disassembler From Apple – Steve Wozniak & Allen Baum | 22 |
| Mastermind: A Number Game for the 6502 – Steve Wozniak | 26 |
| 8080 SOFTWARE The Bouncing Beastie: A Random Walker for the VDM – Marvin R. Winzenread | 28 |
| TBX Additions: String Handling & a New Print Delimiter – The Penny Tiger | 30 |
| Dompier Music Mods, Memory Erase, Memory Test – Richard A. Kaapke | 32 |
| BUGS & FIXES A "Fix" for Denver Tiny BASIC; a Bug-Note on Processor Tech's 5K BASIC – F. J. Greeb | 34 |
| How Secure is Computer Data? Not Very, Say Stanford Experts – <i>The Stanford Observer</i> | 35 |
| Do You Know About People's Computer Company Newspaper? | 38 |
| -- AND -- <i>57 other exciting tidbits and short articles</i> | |

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, maybe 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 8; September, 1976
Box 310, Menlo Park CA 94025

Copyright © 1976 by People's Computer Company

Publisher
People's Computer Company
1010 Doyle, Menlo Park, California

Editor
Jim C. Warren, Jr. (415) 323-3111

Contributing Editors
F.J. Greeb
Steve Wozniak
Jim Day

Watchdogs
Bob Albrecht
Dennis Allison

Underdog
Rosehips Malloy

Circulation & Subscriptions
Laura Reininger

Bulk Sales
Dan Rosset

POSTMASTER: Please send Form 3579 to: Box 310, Menlo Park CA 94025. Return postage guaranteed. Application to mail at second-class postage rates is pending at Menlo Park CA.

Published 10 times per year; monthly, excluding July & December

U.S. subscriptions:

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

Discounts available for bulk orders.

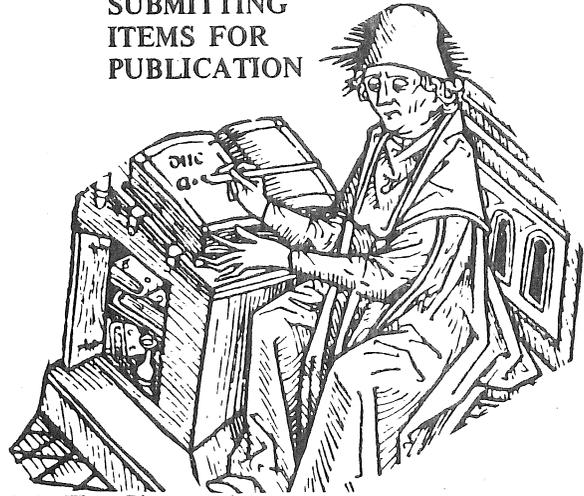
Disclaimer

We serve as a communication medium for the exchange of information. We do not guarantee the validity of that information.

Reprint privileges

Articles herein that are copyrighted by individual authors or otherwise explicitly marked as having restricted reproduction rights may not be reprinted or copied without permission from People's Computer Company, or the authors. All other articles may be reprinted for any non-commercial purpose, provided a credit-line is included. The credit-line should indicate that the material was reprinted from Dr. Dobb's Journal of Computer Calisthenics & Orthodontia, Box 310, Menlo Park CA 94025.

SUBMITTING ITEMS FOR PUBLICATION



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

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

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

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

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

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

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

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

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

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

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

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

realizable fantasy

Computer Control of Music Tapes for Your Home Stereo

by Jim Warren, Jr., Editor, DDJ

Do you use tape recorded music in your home music system? Do you have a number of bands or "cuts" recorded on a single tape? Wouldn't it be dandy to be able to play only desired cuts without having to manually hunt for each one, and then have the tape automatically rewound when a side is finished? You can do it with your personal computer; a project uniquely appropriate for home computing.

How you can use the system: Mount the desired stereo tape on the tape transport. Tell your computer which bands are to be played. You may instruct that they be played in a different sequence than they appear on the tape, and you may have any cut(s) played more than once. Tell your computer whether or not to rewind the tape when finished. Then, do nothing but enjoy the concert of selected pieces until the computer is finished with the tape. The computer will use the fast-forward / fast-reverse facilities of the tape transport to move between selected cuts that are nonsequential.

The following notes outline how such a system can be implemented at nominal cost with currently available components.

Pertinent points about how a good tape system can work: The tape transport can move the tape forward, fast forward, reverse and/or fast reverse, and can stop the tape. Playback and recording can only be done during the "forward" mode. The read/write head is retracted from the tape during "fast" modes. A position counter is provided, but is based on rotation of a tape spindle rather than linear tape motion (that is, a counter increment represents much less tape length at one end of a tape than it does at the other end). There are beginning-of-tape/end-of-tape (BOT/EOT) sensors that operate either by sensing a metalized strip attached to each end of the tape, or by photo-electrically sensing a reflective or transparent leader/trailer strip on each end of the tape. Recording is performed by first erasing with an erase head, and then recording with a read/write head. The erase head and the read/write head are adjacent, spaced well under an inch apart.

The usual format for a tape (Figure 1) is for there to be a BOT leader, followed by several seconds of unrecorded or "blank" tape, followed by some number of bands or cuts of music — each separated by several seconds of blank tape, and finally terminated by some length of blank tape followed by the EOT trailer.

***A realizable fantasy is a project that has not yet been done, but it's time to do it. That is: (1) It's interesting, entertaining, and/or useful. (2) It's feasible for hobbyists in that it can be done in a reasonable amount of time, at reasonable expense, and does not require great expertise (except, perhaps, that which can be furnished in the design article). (3) It is well within the limits of current technology and current techniques. Send us your realizable fantasies, so that we may share them with others.**

The system design: Most of the tape transport functions are obviously binary. The tape is/is not stopped. The tape is/is not in "forward" mode. The tape is/is not in fast reverse mode. The system is/is not in record mode. The tape is/is not at BOT or EOT. The counter can easily be made binary by having tape spindle rotation cause a strobe every turn or half turn or quarter turn, etc. Your computer can then accept these strobes and maintain its own position counter. Note that the position counter and BOT/EOT sensing are independent of whether the read/write head is retracted; i.e., they are independent of whether the tape is being moved in "fast" mode.

Given the number of audio cassette units being used for mass storage of digital data on home computers, it should be obvious that we can record digital information, as well as analog -- music (or voice) -- information on tapes. In fact, *both* digital *and* analog information can be intermixed on the same tape. All that needs to be done is to encode the digital information in such a manner that it can easily be discriminated from the analog information. (Quick! One of you analog types. Send the implementation details in a fool-proof encoding scheme for this context -- digital signals on the same tape as music signals.)

Thus, digitally-encoded identifiers can be recorded in the blanks preceding the music bands. (Figure 2). By recognizing and then reading that identifier, the computer can determine which band is about to pass over the playback head and can switch it to your hi fi system or skip it, depending on your instructions to the computer.

Since the read head is retracted during fast tape motion, it is not possible to read band identifiers while skipping bands in fast-forward or fast-reverse mode. Therefore, the position counter must be used to approximately position the tape to a desired cut. In order to do this, the computer must know the position counter value for each cut. When a tape is initially recorded, such information is accumulated and then placed in a "Tape Directory" at the beginning of the tape (Figure 3). Your computer can then read in the Directory whenever it begins to play a tape.

Hardware details: Your computer must have control over the system. It must be able to select tape motion: forward, fast-forward, fast-reverse, and stop. It should also be able to control whether the signal from the read head is allowed to go to the hi fi system, and should control whether a music signal can reach the record head during initial tape recording. And of course, it must also be able to control the recording of digitally encoded information on the tape.

Your computer must also be able to monitor all outputs from the tape system. It must be able to see BOT/EOT conditions. It must be able to monitor tape spindle motion via a rotation-periodic pulse or strobe. It must be able to monitor the analog signal coming from the read head so that it can maintain its constant vigil for digitally encoded information. It should also be able to monitor recording signals being sent from the hi fi system to the tape

recording head. Finally, it would be desirable for it to be able to sense whether the "write lock" tab has been punched out of the rear of a mounted tape cassette.

Software details: Two basic software modules are involved: A tape initialization module, and a tape playback module.

The *tape initialization module* functions as follows, interacting closely with the human user. The user will mount a blank tape on the transport, at least partially re-wound, and tell the computer the maximum number of bands he expects to record on the current side of that tape. The computer will look for BOT. If BOT is not seen, it will do a fast-reverse until BOT is sensed; then stop. It will then switch to "record" mode with no signal going to the record head, and go forward until BOT is no longer sensed, indicating the end of the leader and the beginning of the recording area of the tape. At this point, it will set its tape position counter to 0. It will continue to go forward, erasing the tape as it proceeds, for the amount of tape space it will later need to record a Directory for the maximum number of bands (already specified). It will then record a digitally encoded "1" to indicate that Band 1 is about to appear on the tape, stop the tape leaving it in "record" mode, and inform the user that it is ready for the first cut of music.

The user may then begin recording. He must signal the computer when to start recording and when a cut has been completed. When he signals that recording is to begin, the computer will place the tape transport in "forward" mode, and switch the hi fi output into the record head. Whenever the user signals the end of a cut, the computer will disable hi fi input to the record head (to avoid any possible "noise"), digitally record the number of the next cut on the tape (which also serves as an "end" flag for

the preceeding cut), and make note of the value of the position counter for later use in the Directory.

There should be several end-of-cut signals available for the user. One signal will indicate that the current cut is completed and that the next cut is immediately forthcoming, as is the case when continuously recording a multiple-band phonograph record. In this case, the computer will record the cut number and quickly switch back to hi fi input for the record head, leaving the tape in forward mode and record mode.

A second signal will mark the end of the current cut, but will indicate that there will be some delay before the next cut is ready for recording. The computer will respond by halting the transport after recording its next cut number. The start signal used for the first cut on the tape can then be used when the user is finally ready to continue recording.

A third signal specifies that recording for that tape is completed. In this case, after recording the "next cut" number (which serves only as the "end" flag for the preceding cut), the computer will disable recording, fast-rewind the tape until BOT, enable digital recording, move forward until BOT disappears, and record the Directory. The Directory will contain a sequence of entries indicating the values of the position counter the beginnings of each of the cuts.

Nitty gritties: While recording the digital cut numbers, the computer should monitor the hi fi output to, if possible, detect any music that begins prior to the computer completing the recording of the cut number. In this case, it should inform the user of a possible loss of the first part of the music.

Whenever the tape is moving forward, the computer should monitor the EOT flag and should output the appropriate error message if EOT is reached.

Software control over recording should take into

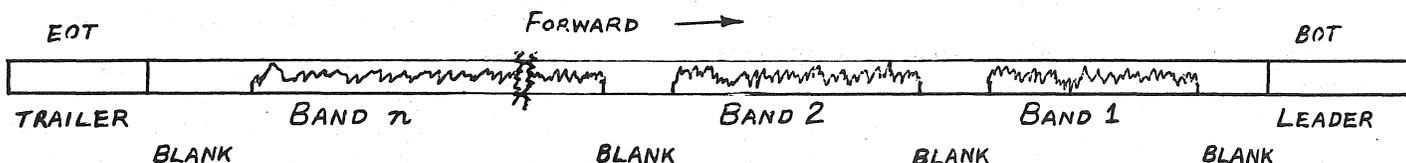


FIGURE 1

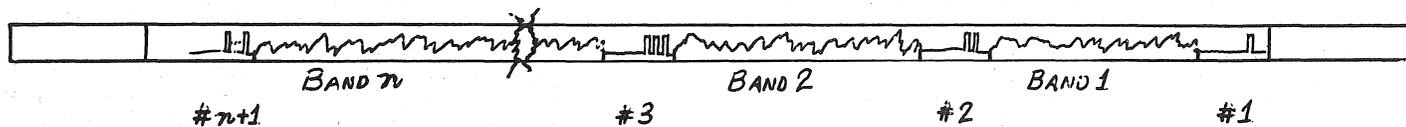


FIGURE 2

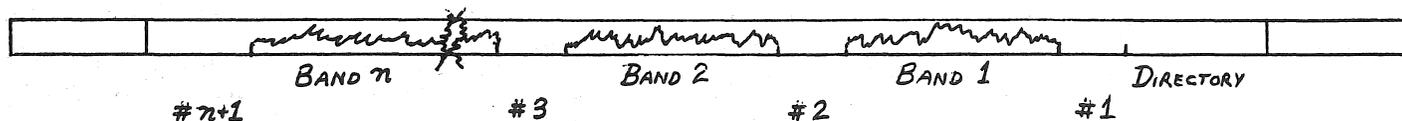


FIGURE 3

account the distance between the record/playback head and the erase head, whenever it is appropriate to do so.

The *tape playback software module* functions in a fairly obvious manner. First, it must accept a command sequence from the user, specifying which bands are to be played. When the computer notes the termination of this command sequence, it will then disable tape output to the hi fi system, hunt for the beginning of the tape, move forward and input the directory, setting the tape position counter to 0 at the beginning of the recording area of the tape.

To play the desired cuts, it looks at the next cut (or first cut, initially) to be played, looks in the directory that it now has in memory for the position counter value, and compares the desired value to the current value of the position counter. If the counters are significantly different, it uses the fast mode to position the tape. It positions the tape to a few counts preceding the desired count, then switches to forward motion and monitors the playback head output -- which, thus far, remains disconnected from the hi fi system. When the computer sees the digitally encoded cut number, it compares it to the desired cut number, and (presuming a match) switches the playback head output to the hi fi system as soon as the cut number has passed.

The computer then begins continuously monitoring the signal from the playback head, looking for the next digitally encoded cut number (that will indicate the end of the cut being played). As soon as it sees the first part of a digital encoding, it disables the output to the hi fi to avoid having the listener hear glibble. It continues to process the user's command sequence in the obvious manner.

Available hardware: Though we assume that there are several tape systems that provide the necessary capabilities for the described interface, we will detail the only one about which we have explicit information.

Triple I, a division of the Economy Company located at Box 25308 (1901 N. Walnut), Oklahoma City, OK 73125, (405) 528-8444, manufactures three models of the Phi-Deck cassette tape transport. The least expensive version has a single, fixed speed for recording and playback and is quite adequate for this project. The mamma transport is a variable-speed unit (which really isn't useful in this context), and the daddy transport is also a variable-speed unit with a King Kong motor on it. Other than for the variable-speed facility and the motor size, all three transport subsystems offer the same capabilities.

The standard speed for the fixed-speed transport is 1-7/8" per second. Other speeds are available via a drive-pully change. All transports have fast-forward/fast reverse modes that can run the length of a 60-minute audio tape in less than 30 seconds. Engaging and disengaging the record/player head and the pinch roller is accomplished with a DC motor, rather than a solenoid. The transports have three other motors: take-up, rewind, and capstan drive.

Our understanding from the local Triple I representative is that there are two stock options for recording/playback heads that come with the system: a 4-channel analog head, or a 2-channel digital head. The purchaser may choose either of these; their cost is included in the price of the unit. Alternatively, the buyer may choose to have the factory mount and align any other standard head that he may wish to furnish. There is no charge for factory installation of any standard head. Thus, the buyer can have as high a quality head as he can afford.

Available system features include: tape position

strobe, BOT and EOT sensing, "Side A" sensing, write-lock tab sensing, "cassette in place" sensing, and, of course, a PC board that facilitates digital control over all of the tape transport facilities.

Pricing is *great* for the experimenter: Triple I has a special introductory (one per customer) package for \$189.00. It includes *two* fixed-speed Phi-Decks (normally \$94.50 *each*), either one of the stock heads, an erase head (normally \$2.50), *all* of the sensors (normally \$1.25 to \$7.50 each), the tape position strobe (normally \$7.95), and the control board (normally \$99.50). That's not all, though. Since this introductory package is for prototyping and experimentation, it *also* includes a box that provides manual remote control over the system (normally \$89.00), and an AC power adaptor (normally \$7.50). Similar introductory packages are available using the variable-speed unit (\$199.00 for the package), and the super deck (\$299). Shipping charges are prepaid if the package price is sent along with the order.

Notes: Although these packages provide all of the electronics for control and sensing of the system, they contain no read/write electronics. That is, the wires come out from the record/playback head, are terminated at some wire-wrap posts on the edge of the PC board, and it's up to the user to take them from there. Incidentally, these are fully-assembled systems; they are *not* kits.

Although these packages contain two transports and one control board, the board is *not* designed to control two transports. The packages were designed to allow engineering prototyping in which one team builds a transport into a larger system which another team simultaneously develops the controls, software, additional electronics, etc.

Bells and whistles: There are some options available in this computer controlled system that have not yet been mentioned. We will mention them only briefly to whet your appetite.

The design that has been outlined implicitly assumes that cuts are identified only by number, and that the user will specify cuts to be played by entering their numbers in the desired sequence. There is no reason to be so primitive, however. With appropriate additions to the tape initialization software module, and the playback software module, one could easily input a name for each cut or piece of music. This information could be stored in the directory along with the tape position number. Appropriate software could then list an index to the recordings on a given tape, and such recordings could be chosen to be played by entering their names *or* cut numbers.

Primitive tape editing is another feature that could be added simply by expanding the software. More comprehensive editing facilities would be possible if two transports were under computer control.

Thus far, this discussion has concerned the control of a music system. The cuts, however, could as easily contain spoken information instead of music "information." For instance, it could contain instructions, questions, and answers. Coupled with devices for human-response input (keyboard, joy stick, push buttons, etc.), it would be ideal for many types of CAI (Computer-Aided Instruction) applications.

The design that has been specified assumes that the record/playback head is retracted from the tape whenever the tape is being transported in "fast" mode. If this is not the case -- if the head remains in "read" contact with the tape, even when the tape is being moved at high

speed — then it may be possible to avoid the need for a directory of cut position numbers, for the computer should be able to locate a desired cut simply by watching for the cut number to appear as the tape is racing across the read head. Such an alternative would provide some interesting timing or encoding problems, since the cut number would be recorded at one speed, but read at a quite different (and probably varying) speed.

Closing notes: This has been a *design* article; not an implementation article. We encourage you to send in follow-up articles, detailing your specific implementations, once you have them operational. The slogan is, "Let us stand on one another's shoulders, not on one another's toes."

We are publishing this now, rather than holding it in order to fill out the details and nitty gritty, for several reasons: Most importantly, we wish to get this idea out into the public domain — before someone else stumbles on it, and rushes off to the patent office. We are also publishing it in this somewhat incomplete form because these realizable fantasies are as much *your* projects as they are *ours*. We assume that you gain as much enjoyment from invention as you do from construction. We have left room for your own invention — for you to contribute your own good ideas and additions to this project. We hope that you will share them with us, so that we may share them with everyone else.

And, a final thought: Has your spouse been complaining about the money and time you have spent on your weird little computing machine? Have your friends been asking, "Yeah, but what can you *do* with it?" If so, perhaps this project will quell their unkind remarks. Be the first one on your block with a computer-controlled stereo system!

DENTAL PERSON REQUESTS DATA ON OUR OVERBYTE CONTROL

Dear Jim, 7-25-76
Saw John Craig's report of your new publication in *73 Magazine*.

Would you please send me a copy of *Dr. Dobb's Journal of Computer Calisthenics & Orthodontia*.

Orthodontia?? That part of the title caught my eye.
Thanks.

Dr. Jim Smith, Orthodontist 111 N. Regency Dr.
Drs. Lauder, Smith & Gordon Bloomington IL 61701
Orthodontics, Ltd. (309) 662-5331

HAVE AN OFFICE IN YOUR HOME? IT'S ABOUT TO LOSE ITS TAX DEDUCTIBILITY

We understand that the Senate Finance Committee has reported out a tax "reform" amendment which will, effectively, prohibit your deducting expenses incurred in maintaining an office at home . . . even if you are a completely independent consultant with no office elsewhere. If you have an opinion on this matter, you should write your Senator and Representative *immediately*. (In fact, you could use your computer and Selectric to type "individual" copies for *all* the key Congressmen.)

APPLICATION NOTES GIVE EXCELLENT INTRODUCTION AND COMPREHENSIVE DISCUSSION OF FLOPPY DISC DRIVES

We understand that PERTEC, 9600 Irondale Avenue, Chatsworth, CA 91311, has two Ap Notes that provide a complete and detailed discussion of floppys in a form that is understandable to the floppy novice (novice floppyite?). Their titles are "Soft Sector (Hard Sector) Formatting for FD400 Flexible Disk Drives," however our information is that they provide a really complete discussion of floppy drives that is essentially applicable to any model from any manufacturer.

PC CARD HOLDS 60 MEGABYTE REMOV- ABLE MAGNETIC STORAGE AT LESS THAN HALF THE PRICE OF DIGITAL CASSETTES

Micro Communications Corp., 80 Bacon St., Waltham, MA 02154, (617) 899-8111, is manufacturing a tiny tape cartridge drive, small enough to be mounted on a PC board. It takes a wee, endless-loop tape cartridge that has a maximum capacity of about half a megabit. The drive will run at 3"/second. The tape is capable of 2400 flux-changes/inch. Thus, using ratio encoding involving three flux changes per bit, one can obtain 800 BPI, and 2400 baud transfer rates. That would load a 16K byte program in less than a minute.

Little Hidden Gotchas: MCC only manufactures the drive and cartridges. The buyer must supply power (12v), motor speed control and analog R/W electronics, as well as the digital interface. But, the price is right: In 5000-piece quantities (all right, which of you end-users is about to turn into a manufacturer?), the drives run from around \$22@ with a 100-hour toy motor to \$36@ with a 1000-hour Swiss motor. And, *certified* tape cartridges in that quantity run \$1.80 to \$2.20@ depending on length. Incidentally, the cartridges weigh less than a half-ounce apiece. Two can be mailed, First Class, for 13 cents (this month).

MOTOROLA COMPETITOR FOR THE Z-80

Motorola is reliably rumored to be working on their 6809, which supposedly will give Zilog's Z-80 some stiff competition. More details when we have 'em.

GAMES AND WAR

We hear "from a reliable source" that the Army has awarded \$25,000 to a computer games company for development of war games equipment.

Also, a year or so ago, the *Journal's* Editor (who is a long-time computer consultant) was contacted concerning a systems position in developing a project at the Hunter-Liggett Military Reservation south of Monterey, California. They were installing *multiple* PDP-10's *plus* about *ten* PDP-11's (as remote data concentrators for sensors in the field), for the single purpose of monitoring war games on the reservation. The system sounded most exciting. However, the application—including the incredible cost that was implied—made the project "uninteresting" to our Editor.

A Realized Fantasy: Unlimited Speech Synthesis for Home Computers

A Survey of Computer Speech Synthesis Systems, Including a \$395 Kit

by Jim C. Warren, Editor

Intelligible, useful, voice synthesis by home computers is now available to hobbyists at reasonable cost. This means that, not only can your computer talk to you and your spouse (who, admittedly, may not want to hear it), but it can also talk to your pre-reader and beginning-reader children. Thus, besides being fun and useful in a general context, your home computer can also become a significant educational tool for young children.

There are several manufacturers of computer speech synthesis subsystems, two of which are marketing specifically to hobbyists, and one of which is worthy of specific attention.

The best system we know of for hobbyists—i.e., financially feasible—is nearly available from *CompuTalker Consultants*, Post Office Box 1951, Santa Monica, CA 90406; (213) 392-5230. This brand new product has an introductory price of \$395 assembled. (Note: There is a significant possibility that this price will be available only for a short time and will then increase, as seems to be characteristic of vendors to the hobbyist community. Once the demand for a good—or bad—product grows to where personnel must be hired to handle the increased workload, prices often must be raised to pay for the added wages, plant facilities, etc.)

The first Computalker model is an assembled unit that mounts on a hobby-standard-bus board (that is, it plugs into an Altair/IMSAI/Polymorphic/etc. bus). The unit includes the PC board, two pre-assembled and tested CT-1 Synthesizer modules, and less than a dozen IC's for buffering, decoding, DAC, etc. The bus interface requires nine 8-bit, output ports. The block of nine ports is relocatable via switches on the board. The following seven ports should be reserved for future facilities. The unit requires +8V at a maximum of 250 mA (regulated on the board to +5V), and $\pm 16V$ at 85 mA (on-board regulated to $\pm 12V$). The audio output is via an RCA-type phone jack or directly wired to the board. It supplies a 1V output, peak-to-peak, into a 10K ohm load resistance. Units that will plug into other, non-hobby-standard buses are being designed, including interfaces to the GA-16 and DEC PDP-12.

During actual speech output, each output location or port must be loaded at least twenty times per second. If desired, the ports may be loaded as often as 1000 times per second. Good-quality speech synthesis requires that each of the nine ports be loaded about 100 times per second (presuming the locations are 8-bit bytes).

The user may develop the software entirely on his own, or he may purchase the driver software from CompuTalker for about \$35-\$40. A good write-up with extensive information on phonetics and speech mechanics will be included. The driver software will take less than 8K bytes for an 8080. Software for the 6800, 6502, etc. is planned for the near future.

At the time this article was written, there was some

question about the speed that would be required in an 8080 for real-time—on the fly—speech synthesis. At that time, it appeared that, for real-time synthesis, an 8-bit hardware multiplier might be required for calculation of the byte sequences that cause words to be synthesized. Check with Lloyd Rice for more definitive information. Under any circumstances, the system can still be used, however. One simply generates the desired byte sequences; places them in memory; then transfers them to the output locations as rapidly as desired. This way, no complex calculations would be required during the actual speech output. For such non-real-time operation, however, about a kilobyte of buffer space is required for each second of speech. Of course, another alternative is to use a faster processor such as the 4 megacycle Z-80 about to be available from CroMemCo (please see *DDJ*, Vol. 1, No. 7). A processor that is faster than your old, plain-Jane 8080 might allow real-time synthesis calculations without the need for hardware multiplication.

There is much more that could be said about this unit, including details of how it works, but that is the topic of future—and past—articles. We refer you to two articles by Lloyd Rice: The first one appeared in the April, 1976 issue of *Dr. Dobb's Journal*, "Hardware & Software for Speech Synthesis." The second one, though written earlier, appeared in the August, 1976, *Byte*, "Friends, Humans, Countryrobots: Lend Me Your Ears."

Incidentals: Lloyd Rice has been working in the area of speech synthesis for some time. He holds a B.A. in linguistics, and is currently working with the research computers in the Phonetics Lab at UCLA.

CompuTalker is another one of those very small companies that has developed an exciting product for the hobbyist community. The product has been developed, as much for the love of the project and the gadget, as for the purpose of making money. We are very impressed with Lloyd's expertise in the area of speech synthesis. We are also very impressed with his ethical standards and are quite sure that he will back his product and "do right by his customers." We tend to be much more impressed by people who are doing something because it is fun and exciting, than we are by those who are doing something to make money. We believe Lloyd fits the first category and recommend him and his product to you.

The second system being marketed to hobbyists is the Model 1000 from *AI Cybernetic*, Box 4691, University Park NM 88003; (505) 526-6842. First of all, we understand from people who have heard this system that its intelligibility is marginal. Some tend to feel that this system, rather than training the computer to speak, requires that the listener be trained to understand—an alternative which is opposite to our views concerning man-machine interfaces. Secondly, this unit is priced higher than the CompuTalker unit; the Model 1000 costs \$425. The unit does, however, plug into a hobbyist-standard bus, and its advertisements imply that it requires considerably less software support. We suggest listening to a unit, before you commit to a purchase. We also suggest careful queries concerning "on the fly" real-time speech synthesis and mem-

ory requirements for production of reasonable sentences.

Incidentally, we would be pleased to publish more positive statements concerning this equipment. We are interested in comments from end-users (as opposed to vendors) who own and have experience with the units.

The third system that is of *potential* interest to hobbyists is from Votrax, 4340 Campus Dr., No. 212, Newport Beach CA 92660; (714) 557-9181. (This is the address for their Regional Manager, John McDaniel; not for their home offices. However, John has expressed an active interest in the possible hobby market.) The last word we had was that Votrax was *considering* making their unit available in kit form for under \$1000 (see *DDJ*, Vol. 1, Nos. 2 & 5). Their system has been in the field for several years, being marketed as a turn-key system, ready to plug into big IBM systems for use in voice-answerback applications. In their commercial form, they run around \$3000-up, unit quantity. They are considering stripping out all the turn-key garbage—such as audio amplifier, volume control, power supply, fancy cabinet, etc.—and peddling the guts for a much more consumer-tolerable price. If they do, the kit-form would consist of four heavily potted, highly proprietary boards along with the necessary instructions for interfacing, adding power, providing input, and using the output. The four boards contain hardware phone generators. We have heard the units “speak”; they are quite intelligible and provide reasonably good-quality speech. They also require only very limited and simple-to-use software support. We will say more if/when Votrax tells us that they *are* going to market a reasonably priced kit. If they ever do, we expect to strongly recommend their well-proven and easy-to-use unit.

There are two other speech synthesis systems worthy of, at most, very brief mention here—primarily for the sake of completeness.

The first is a custom LSI component that has a limited and fixed vocabulary. This contrasts with the preceding systems which allow English speech synthesis limited only by the capacity of the computer's memory to store the byte-strings that specify the words and sentences. This LSI component was developed by Dr. Forrest Mozer, a physics professor at the University of California, Berkeley. The custom chip was implemented by Silicon Systems, Inc., in Santa Ana, CA. It will generate 24-64 complete words. It has been used in a “talking hand calculator” of particular value to the blind. This calculator is made by Telesensory Systems, Inc., 1889 Page Mill Rd., Palo Alto CA 94304; (415) 492-2626. It uses a TMS1000 micro-processor from TI, and the entire unit sells for \$395.

The second system is a voice response system for the PDP-8, PDP-11, and NOVA computers that uses a “solid-state speech synthesizer.” It is available from Interface Systems, Inc., 462 Jackson Plaze, Ann Arbor MI 48103, and sells for \$6,750 to \$47,500. Need we say more?

One final “maybe” system: We have heard that some crowd calling themselves “Master Specialties” is peddling voice synthesis equipment, but we have been unable to find out their location, the details of their product, or their pricing. Anyone know where they are—or know anything else about them, for that matter?

CROMEMCO NEW ADDRESS

CroMemco has moved from Los Altos. Their new address is 2432 Charleston, Mountain View, CA 94043, (415) 964-7400.

VIDEO TERMINAL TECHNOLOGY DOES HAVE A PHONE!

After publishing the article on the 64-character by 48-line TVT for \$499.95 [*DDJ*, Vol. 1, No. 6], we received a dozen or so requests for the vender's phone number, which we had neglected to include in the article. Here it is: (408) 255-3001 or (408) 734-2244, ext. 5179 (ask for Ralph Butler).

RELIABLE 110 CHAR PER SEC PRINTER (34 CHARS/LINE) FOR \$298, UNIT QUANTITY

Victor Comptometer Corp., Components Dept., 3900 N. Rockwell St., Chicago, IL 60618, (312) 539-8200 has a Model 130 matrix printer that prints about eleven times as fast as a TTY. The printer has been around for about two years; over 150,000 of them are now in use. It carries a *one year* parts and labor warranty to the original purchaser. In single-unit quantity (1-9), the price is \$298; 100-unit quantity is priced at \$183.25@. It takes roll paper from 2-9/32” to 3-3/4” wide by up to 220' long.

The user must supply power for motors and solenoids. The user must also supply solenoid print signals and paper feed control, but, then again. . . micros do that so well. Characters can be either 5x7 or 9x7. The 130 will only print 34 characters per line, but, at that price and speed, maybe it's worth it to tape two pieces of paper together to get 68 characters on a line.

A DOT-MATRIX PRINTER WITH 80-COLUMNS FOR AROUND \$305???

We are currently tracking down the details on a printer rumored to become available from Practical Automation, Inc. (address info when we get it) in about the middle of September, designated the DMTP-6, which will handle 8 1/2” paper, print 80 characters per line, and be priced slightly above \$300. Our guess is that the price is for OEM quantities, however, who knows. . .

PHILADELPHIA HOME COMPUTER SOCIETY

At last Philadelphia has its own computer society. Started as the result of a letter in *Byte*, the initial meeting drew over 80 interested individuals. So far we have had demonstrations of Sphere and the Digital Group computers. Plans are for field trips to the Moore School of Electrical Engineering (birthplace of the ENIAC) and to nearby MOS Technology. Our newsletter, *The Data Bus*, provides meeting information as well as articles on hardware and software, book reviews, letters, cartoons, etc. Interest is mushrooming and all of us here have visions of computer grandeur.

Dick Moberg
404 S. Quince St.
Philadelphia, PA 19147

MARIN COUNTY COMPUTER GROUP

Hobbyists in the Marin County area of California gathered for the first meeting of a new hobby group on July 21st. For information about this new group, contact the Byte Shop Computer Store, 509B Francisco Blvd., San Rafael, CA, (415) 457-9311.

ENGLISH-TO-PHONETICS TRANSLATOR FOR PHONEME SPEECH SYNTHESIZERS

Joe Mockus

Automatic Translation of English Text to Phonetics by Means of Letter-to-Sound Rules.

Honey Sue Elovitz, Rodney W. Johnson, Astrid McHugh, and John E. Shore.

Naval Research Lab Washington DC 21 Jan 76. 101p. NRL-7948
AD-A021 929/5WC Hardcopy \$5.50/Microfiche \$2.25

[available from National Technical Information Service, 5285 Port Royal Rd. Springfield, VA 22161]

Speech synthesizers for computer voice output are most useful when not restricted to a prestored vocabulary. The simplest approach to unrestricted text-to-speech translation uses a small set of letter-to-sound rules, each specifying a pronunciation for one or more letters in some context. Unless this approach yields sufficient intelligibility, routine addition of text-to-speech translation to computer systems is unlikely, since more elaborate approaches embodying large pronunciations dictionaries or linguistic analysis require too much of the available computing resources. The work described here demonstrates the practicality of routine text-to-speech translation. A set of 329 letter-to-sound rules has been developed. These translate English text into the International Phonetic Alphabet (IPA), producing correct pronunciations for approximately 90% of the words in an average text sample. Most of the remaining 10% have single errors easily correctable by the listener. Another set of rules translates IPA into the phonetic coding for a particular commercial speech synthesizer. This report describes the technical approach used and the support hardware and software developed. It gives overall performance figures, detailed statistics showing the importance of each rule, and listings of a translation program and a program used in rule development.

A NERVOUS SYSTEM FOR THE HUMAN RACE

Good morning,

June 10, 1976

We here are working on a plan for a series of "information stores" that would require computerization for large-scale information flow. Our plan involves store-front access points, an open catalog of information available, and economic self-sufficiency through both mass marketing of information and "per-bit" charges for both input and output information.

Ultimately, the plan would involve utilization of the information for specific problem-solving assignments, on-going studies of attitudes and values, selective dissemination of information and analysis to political figures and others likely to be influenced toward implementing the public good, and much more.

The plan has been compared to giving the human race a nervous system complete with a brain.

Please send us any information you can that relates to this task, and let us know if you would like to get more involved in what we are doing.

Sincerely,

Robert A. Moskowitz

403 W. School House Lane
Philadelphia PA 19144

BUSINESS WEEK NOTES HOBBYISTS

The July 12 issue of *Business Week* carried a full-page article about the computer hobbyist movement. They also devoted 7 pages, essentially, to microprocessors, in a major article in their July 5th issue: "The Smart Machine Revolution."

AN EXCELLENT BUY & SELL NEWSLETTER

There is a very useful little typewritten newsletter available, that is explicitly concerned with want ads for computer hobbyists, both for buyers and sellers. It is called *ON-LINE*, and is published by D.H. Bettle, 24695 Santa Cruz highway, Los Gatos CA 95030. It is published every third Wednesday, is sent by *First Class* mail, contains 6 to 12 pages in each issue, and has a very fast turn-around—all ads received as late as the preceeding Saturday are included in the issue for that month. The publisher states that they put out 17,3925 issues per year, but we assume that is merely an approximate figure. Subscriptions are \$1/four issues, or \$3.75/18 issues (= one year), or \$7/36 issues.

For those wishing to advertise, there are both line rates and word rates, and there are reduced rates for ads running in multiple issues. There are also much lower rates for non-commercial ads than for commercial ads. As far as we can tell, there are no display ads; it's all text, which means much more meat in much less space.

We highly recommend this publication to those interested in buying or selling goods for the computer hobbyist.

USED EQUIPMENT INFORMATION BROKER

The Computer Equipment Information Bureau, Box 163, Boston MA 02117; (617) 247-2290, has listings of used computer equipment. The listings are free to buyers, and are available to sellers for a fee. R. Ferrera is the President of the Bureau.

BLIND RUMOR: A FLOPPY DISC SUBSYSTEM FOR ABOUT \$1K

We heard the rumor that one of the companies that is marketing to the hobbyist community is putting the final touches on a floppy disc interface for Diablo's Model 12 disc drive. We also heard that the disc will store about three megabits, formatted, and that Xerox (Diablo's after-birth parent company) will be selling the unit for around \$1000.

We checked the story with the rumored company, and were told that there would be an announcement "soon." Since we think this company is one of the "good guy" crowds, we are going to abide by their plea that we not disclose the about-to-be manufacturers.

We do have a question, however: Why should anyone bother to wait for this system, when they can get one of Digital Systems' floppy units—right now, off the shelf, assembled and burned in—for \$1095 (or \$995 in quantities of 10)? Digital Systems' unit is already out in the field, has an excellent reputation from its users, uses a top-notch Shugart drive, and—perhaps the best thing of all—runs Kildall's CP/M, the DEC System-10-like floppy disc operating system that has been in production and instructional use for over two years. [See *DDJ*, Vol. 1, No. 7, for additional details.]

LOTTA FLOPPIES

Shugart, the manufacturers of one of the best-reputed floppy disc drives available in a highly competitive market, has shipped over 30,000 of their drives.

CASSETTE REPOSITORY, TRAC, OTHER LANGUAGES, CP/M, AND PROCESSOR TECHNOLOGY PRAISE

Dear Jim,

July 20, 1976

1) Many hobbyists may now be starting to do some heavy stuff in assembly language. With things like the Tarbell cassette, the capacity is now there to be able to store the 100K or so needed for a fair-sized program. However, a lot of hobbyists are stuck for a means of hard copy. Even if they have a teletype, not many people care to wait the 3 hours necessary to list a program of the aforementioned size. Such a listing can be a very useful thing to have. What am I getting at? We need a place where we can send our cassettes (it almost has to be cassette; I shudder to think of someone trying to handle 2 miles of tape. In any case, getting such a tape would entail the same problems as getting a listing) or maybe floppy discs to get a listing of them. Whoever does it could charge a nominal fee and you could get back a lineprinter listing. I don't know if anyone could feasibly provide this sort of service, but if it is possible, it'd be nice. Of course, you'd have to standardize the format of your cassette--but then, standards are necessary.

Speaking on the subject of standards, I think *DDJ* would make a good forum for discussion of such things. If we're ever going to really share stuff, we're going to have to tackle this problem.

2) Regarding TRAC: I'm sorry to hear that Calvin Moores is such a nut on the subject of copyrights. TRAC looks like such a fascinating language (from what I read in T. Nelson's *Computer Lib*). One thing I did notice in *Computer Lib* is that "Moores stands ready to accomodate . . . experimenters who wish to try their hands at programming a TRAC processor. An experimenters license may be obtained for use of the copyright material for a few dollars . . ."

I'm not sure whether this means you can try your hand at writing a TRAC interpreter for your own use, or whether this only means if you find a system that runs TRAC, you can write programs for it. Did you notice anything about this in your deluge of paper from him?

3) I'm very interested in your idea for small Pascal, even though I've never heard of it. But then, almost anything would be better than BASIC. One thing I'd like to see is some sort of list-processing language. BASIC is primarily oriented towards numbers, when a good deal of computer applications deal with text. That's where TRAC would be nice.

4) I recently visited a person who had CP/M (the much heralded floppy disk operating system) up and running. I was quite impressed. One thing that wasn't mentioned was that the system also includes an assembler. I also hear there is a compiler BASIC up to running with the system. I didn't get too much of a look at it, but I've been given the opportunity to learn to use it, and with this guy's permission, I'll send in my comments.

5) A word of praise here for Processor Technology's VDM. I've had a chance to compare Polymorphic's video board with the VDM (from a user point of view). When I first saw the Polymorphic's board it had the problem of white garbage being displayed on the screen (I understand that a fix has since been sent for this). I've seen no problems with the VDM. It is a *very* high speed device.

The thing that really makes the VDM so nice is the software. Processor Tech is really supporting this device. Their 5K basic, for instance, has the VDM driver built right in.

That driver is in itself interesting. It allows you to clear the screen and turn the cursor on and off, as it should, but it also allows you to change the rate of output. While the VDM is being outputted to, pressing one of the digits 1 through 9 on your keyboard changes the speed of output. 1 is equal to about 2 characters per second, and 9 is a maximum of 2000 lines per minute. In addition, the output can be stopped at any time by depressing the spacebar. Any other character pressed then resumes output.

This driver also works very nicely with MITS BASIC.

All in all, the VDM is a fine product. It makes program development.

6) Comments on Processor Tech's 5K BASIC: To start with, it's not really 5K BASIC: it needs 6480 bytes (decimal). It does have the feature of allowing more than one program in memory at a time; it does by the MEM command. When this command is issued, BASIC asks for a starting and ending address. Your program is put in this zone. It also asks you if the program is loaded. Note that BASIC does not concern itself with addresses. It blindly trusts you to give it address which are within your available memory. You also have to

NEW HOME COMPUTING RAGS

In the event that you have blindly bypassed the full-page ads in a number of ad-carrying magazines, two new hobby periodicals are in the offing:

Personal Computing, bi-monthly, \$8/yr (\$6/yr before Sept.1976), first issue October-November, Benwill Publ. Corp., 167 Corey Rd., Brookline, MA 02146

Microtrek, monthly, \$10/yr, first issue: September, 1976 Schneider Publ., Inc., Dows Bldg., Cedar Rapids, Iowa 52401 [Iowa???

Both will be your usual, run-of-the-mill, exciting hobby rags, filled with display ads, articles, letters, etc. *Personal Computing* is touting an interesting twist: each issue will carry a full-color, poster-sized computer graphic...a center-fold for computer phraeques. Betcha neither one of 'em will ever devote 15 or 20 pages to complete documentation & annotated source code listings of systems software, though [toot of own horn].

WE WILL TRY TO GIVE THE INFORMATION THAT YOU WANT

Ever get excited over a widget that's advertised as costing "only \$1195", only to find out that price is for 100 units to OEMers and excludes some essential component? We have.

We find it quite irritating to read an advertisement or article in a trade magazine or commercial [ad-carrying] mag that avoids giving crucial information and forces us to call or write for the desired details...and often an additional sales pitch. The other day, we suddenly realized that we could systematically correct that problem as far as *Journal* articles were concerned. Since we aren't paid for publishing product information, we are not constrained by advertisers' desires.

Therefore, to the extent possible, we will make it a policy to publish desirable product data such as: single-unit price, address, phone, availability, warranty, limitations, etc. We won't always succeed...particularly in the case of "hot stuff" that comes to us just before going to press...but we will try.

keep the programs separate.

BASIC also has the TSAU and TLOD (or TLOAD - I'm not sure) commands which allow user-written cassette (or disc) loading programs to be accessed.

It also allows assembly language subroutines to be accessed via the CALL and ARG functions. ARG(x) passes the argument x (as a 16-bit integer) to the subroutine. Y = CALL (7000) assigns Y the value returned by the subroutine at 7000 decimal.

This BASIC has all the features of "standard" BASIC, with the exception of strings. It includes READ-DATA, FOR-NEXT loops, SIN-COS-TAN functions, as well as the LOG and EXP functions.

I do have one objection to it: when you press CR (without putting anything on the line), it gives you a bad syntax error.

Well, I've rambled on fo' quite a while now, and so I'll quit here.

Computer power for the people!

Paul Holbrook
(alias Cap'n Quirk)

6104 Craterlake Ct.
Pleasanton, CA 94566

A FREE CATALOG OF MINICOMPUTER ACCESSORIES

A catalog of computer supplies, accessories, connectors, cables, etc. is available "for the professional minicomputer user" from Minicomputer Accessories, Catalog Sales Dept., [1015 Corporation Way] P.O. Box 10056, Palo Alto, CA 94303, (415) 969-5678.

FEW FAULTY LSI-II's FOUND

DEC (Digital Equipment Corp.) shipped more than 3000 LSI-II's in about six months. In that period, only 63, (2%) of them were returned as defective. DEC tested those that were returned and, in only 20 (0.6%) cases, agreed that the micro-computer was in fact, defective.

AN INTERVIEW WITH THE INVENTOR OF THE PDP-11

The May, 1976, issue of a trade magazine called *Research/Development* carried an interesting 5-page interview with Gordon Bell: "Computers—Past, Present and Future". It presented some quite interesting historical information about computer developments, particularly about the work done at Digital Equipment Corporation. It also illustrated some of the viewpoints and background of Dr. Bell, DEC's V.P. of Engineering, and a very well known computer engineer and researcher.

NEWSMANSTOWN, PA SYSTEMS GROUP: SORTIARI

Gentlepeople,

28Apr76

We are an embryonic, two-man company located in a relatively backward section of Pennsylvania. Our principal interest is in hardware maintenance and software support for personal and very small business applications. We're also interested in industrial controller systems. Our house machine will be a Sphere system with 36K core, and a line printer.

Sincerely,

David J. Beard

Rd. 1, Box 192C

Sortiarii

Newmanstown PA 17073

Minicomputer Systems & Applications (717) 949-6848

NYC STORE:

HOBOKEN COMPUTER WORKS

Dear Dragon People,

12 May 1976

The Hoboken Computer Works is another small, friendly computer store in the metropolitan New York area. A stroke of luck (and a memorable name) put us right on the front page of the *New York Times* last week.

... I can't end this note without expressing my personal appreciation for what PCC has been doing for the last few years. Obscurantism, ego-tripping, and elitism have dominated the world of computers for too long. Even though I note lingering evidence of these nasties in some of the letters published in *PCC*, I think you're doing a marvelous job of opening up these musty rooms to a little fresh air.

Yours,

Bob Radcliffe

56 - 2nd St.

Hoboken Computer Works

Hoboken NJ 07030

PLESSEY DOES RIGHT BY HOBBYISTS

A major manufacturer offers significant discounts to hobbyists on memory & a 1000 cps tape reader

Plessey Memories and Plessey Microsystems is the first large manufacturer, traditionally marketing its products to major industrial purchasers, that is making its very significant GSA (U.S. General Services Administration), and Educational discount schedules available to hobbyists.

They offer a 1000 char-per-second paper tape reader, including electronics and buffering, to hobbyists (and Educational, and GSA customers) for \$340. The 1-4 quantity price to their industrial customers is \$425. This 1000 cps reader has several quite interesting features. For one thing, it uses a capstan/pinch-roller to move the tape, rather than a much more wearing sprocket feed. For another, their phototransistors are infrared-sensitive to provide maximum coupling efficiency, and the light sources are LED's that never need adjustment.

They also offer a core memory module that is directly usable with National's IMP-8, IMP-16, PACE, and SC/MP micros, and can obviously be interfaced to most other microprocessors realtively easily. The fact that it is *core* means that it is *nonvolatile*; your programs stay there when the power goes away. These units are available to hobbyists, educators, and the feds for \$1085 for a 16K module, organized as 8K 16-bit words. To industrial customers, Plessey's price is \$1500 in single-unit quantities, and \$1365 in 10-14 quantity. They are really "doing right" by hobbyists.

The core memory modules require a controller. Plessey markets their controller card (already assembled, of course) to hobbyists, etc. for \$360 (\$500 to industry, unit quantity). This controller is rather flashy: it provides for memory expansion, data save, write protection, and address partitioning. Units are available 30 days ARO (After Receipt of Order).

Incidentally, Plessey policy is to not publish prices, but rather, to give quotes for specific orders to specific customers. The Plessey rep was less than overjoyed about our plan to publish these prices . . . but then again, the *Journal* is here to best serve its readers; not the vendors. Don't spread the good word about Plessey pricing too far, though. Otherwise, some of their greedier industrial customers might apply some unpleasant pressure.

We applaud Plessey for their fair-minded decision to market their products to the hobbyists on the same price schedule they use for educational customers. We believe it is a logical policy, and is certainly a policy that benefits the home computer user, who derives no financial gain from his or her system. If you also approve, why not *drop them a note of thanks*.

Plessey, 1674 McGaw Ave., Irvine CA 92714, has 22 national and international sales offices.

HOT IDEA

Why doesn't someone interface an LSI-11 bus to a hobby-standard (Altair/IMSAI) bus? Then people would be able to take advantage of the potency of an LSI-11 processor, *and* the economy of hobbyist sub-systems and peripherals?

PROPOSAL: A BOARD TO EXPAND 8080 INSTRUCTIONS

Dear Jim,

August 13, 1976

We plan to make a Hobbyist-Standard (Altair/IMSAI compatible) card which will add new commands and addressing modes, some of which will match or better the Z-80 commands.

We would like your opinion on what is desirable. Some of the possible extensions are listed below. If some variation seems more appropriate, as an addition or as a substitution, please mention this. Notes about any other function you wish added will be appreciated.

Of course, the more we put in, the more it costs. And there is the basic question is it worth it, since existing software would have to be modified to take advantage of it.

Please respond to:

Walt Ferris

2311 Middlefield Rd.
Palo Alto, CA 94301
(415) 321-1231

Editor's suggestion: Rank-order your preferences from most-desired to least-desired, and indicate the maximum you would pay for each facility (in kit form, and in assembled form). "J.W."

POSSIBLE NEW COMMANDS THRU "80F".

1. *Block Move.* 2 byte op-code, parameters such as starting addresses and ending address are in memory. Speed: 2 or 3 clock cycles per byte vs 21 for Z-80.
2. *Block Search for one character.* 2 byte op-code, 1 byte operand to search for, other parameters in memory. Speed: 1 clock cycle per byte vs 21 for Z-80.
3. *Block Search for a character string,* of programmable length, say up to 16 bytes. 2 bytes op-code, parameters in memory. Speed: 1 clock cycle per byte of area searched. Address found is left in memory.
4. *Block search for any of up to 16 different bytes.* You program a table in memory; when one is found it jumps to a programmed address. 2 byte op-code. Speed: 1 clock cycle per byte of area searched.
5. *Program-relative Addressing* for all 8080 jumps, calls and memory-accumulator instructions. 2 byte op-code. Jumps and calls have 2 byte displacement, $\pm 32K$ bytes range, and speed of 16 clock cycles vs 12 for Z-80. The mem-acc group could be either 1 byte or 2 byte displacement. 1 byte gives ± 127 byte range. Speed would be 17 clock cycles for 2 byte, 13 for 1 byte.
6. *Indexed Addressing* for all 8080 jumps, calls and memory-accumulator instructions. Index registers are in memory; I propose 3 and the maximum is 4 with a 2 byte op-code. With a 3 byte code it is unlimited. The index registers are 2 byte and the displacement (operand) is 2 byte. Speed: 16 or 17 clock cycles.

If we reduced displacement to $\pm 16K$ we could use one bit to specify yes or no to auto-increment. If we reduced displacement to $\pm 8K$, we could use the two bits to specify:

- Indexed, 2 byte operand
- Indexed, auto-increment, 2 byte operand
- Indexed, auto-decrement, 2 byte operand
- Indirect, use index register, 1 byte operand.

7. *Two-byte memory-accumulator commands* (base page or zero page type). 1 byte special op-code. 1 byte operand which has to specify both operation and address; thus we can have 8 operations and 32

Z-80 DETAILED IN *ELECTRONICS*

The August 19th issue of *Electronics* contains an excellent and comprehensive article detailing the features and capabilities of Zilog's Z-80 [see *DDJ*, Vol. 1, No. 7, for article on Z-80 based hobby kits]. The article was written by M. Shima, Federico Faggin, and R. Ungermann, three of the principals in Zilog. Incidentally, Faggin used to work at Intel, where he designed all Intel micros from the 4004 through the 8080.

GERMAN COMPANY ENTERS KIT MARKET

Siemens AG, a West German electronics manufacturer, has introduced a computer kit for around \$200. It uses a SAB 8080A micro, and includes a 256 word EPROM, some static RAM, a clock generator, and other minor tidbits.

[GIGGLE TIME] 4040 SYSTEM WITH 1K OF RAM IS PRICED AT \$1,195.00

International Microsystems, Inc. (address presumed unwanted by hobbyists) is trying to peddle an assembled microcomputer for \$1,195.00. That's a reasonable price until you find out that all it has is a 4040, 1K bytes of RAM, sockets for 1K of PROM, a simple front panel, a simpler system monitor, and a "Test PROM" program . . . and that's all.

COMPUTER CHESS TOURNAMENT

Entries are being solicited to the Seventh U.S. Computer Chess Championship to be held October 19-21, 1976 in conjunction with the ACM Annual Conference in the Hyatt Regency Hotel, Houston, Texas. A four round Swiss style tournament is planned with the first round on Tuesday, October 19 beginning at 8 p.m., and the final round on Thursday, October 21 at 8 p.m. The field will be limited to 12 teams. David Levy, International Master from England, will serve as tournament director.

For further information and for an application form, write to Dr. Monroe Newborn, School of Computer Science, McGill University, Montreal, Quebec, H3C 3G1, Canada.

addresses or 16 & 16. Speed: 12 cycles vs 13 for STA or 7 for ADD.

8. *Programmable interval timer* with associated programmable vector address when that interrupt comes. 2 byte op-code. The parameters could be in memory or be operands. There could be 3 or 6 channels. Management would be a problem.

We could have a memory address go high if each timer was in use; or we could have an automatic dispatcher if the parameters were in the operands, and have the memory address go high if all the timers were in use.

9. *Hardware Random 9-bit number;* could have also a random decimal number, or a random packed decimal number.
10. Hardware multiply, etc., etc.
11. What else?

SIGNETICS 2650 PROTO BOARD FOR \$190

A new hardware development tool for microprocessor-based systems that allows a user to configure an accurate prototype with a minimum of design effort is now available from Signetics. Called the Adaptable Board Computer (ABC), the prototyping tool matches the accuracy of a development system built from scratch with the convenience of a pre-assembled prototyping card, according to David Uimari, Microprocessor Product Marketing Manager.

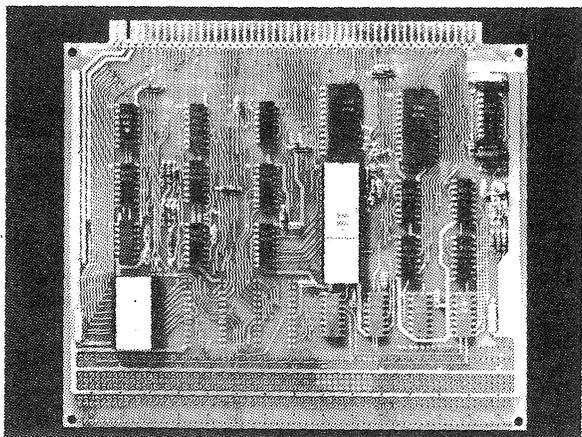
The ABC system consists of a printed circuit board with circuitry designed to meet a broad range of prototyping requirements; a 2650 microprocessor; 1000 bytes of ROM that includes PIPBUG, a Signetics-developed loader, editor and debug program, 512 bytes of RAM; both serial and parallel input/output ports; and a dual monostable on-board clock.

Even without adding components, other options can be selected by the user by simply moving jumper wires. These options include replacing either the ROM or the RAM with PROM, implementing asynchronous operation, adding external clock input, interrupt vector, and RS232 or TTY serial input/output.

Additional memory or control features, unique input/output circuits and the like, can be added by connecting selected components to the existing circuitry with jumper wires or wiretap connections. Room for new components is provided on the board, with plated through holes in place.

The Signetics ABC system is available in either card form or as a kit. The card, which is completely assembled and tested, is priced at \$275 in unit quantities. The kit, which comes complete with all ICs, resistors and capacitors, is priced at \$190. Both versions are available from stock through Signetics and its authorized distributors.

For further information on the Adaptable Board Computer contact Signetics, 811 East Arques Ave., Sunnyvale, CA 94086, (408) 739-7700.



There are 2^n possible combinations of data in an n-bit RAM and $2n \times 2^n$ cycles required to test them. It would take only two seconds to test all of the patterns in a 16-bit RAM at one microsecond per cycle, and about three days to test a 32-bit RAM. But, earliest man, working day and night since his birth around 450,000 years ago, could not have tested even 1% of a 64-bit RAM's possible patterns. Care to think about the 4K RAM?

-From a Fairchild propaganda leaflet

256K BYTE CORE MEMORY FOR \$6500

Dataram Corp., Princeton-Hightstown Rd., Cranbury, NJ 08512, (602) 799-0071, is offering a 14" x 16" board containing 128Kx18 magnetic core memory with 650ns access and 1.5 μ s cycle for \$6500 for a single unit and \$5070 for 25 units. They say their sales are only on an OEM basis, but I betcha they'd sell 'em to someone who'll buy 'em.

And, the nice thing about core is it remembers, even when the power dies in the middle of a six-hour program development session.

8K MEMORY INCLUDES BATTERY BACKUP

Seals Electronics, Box 11651, Knoxville, TN 37919, is offering an 500ns access time, 8K memory board that plugs into a hobbyist-standard* bus and includes battery backup to maintain memory contents when power is turned off (by you, or by accident). Its priced at \$295/kit or \$394 assembled, plus \$2 for shipping. You can use B of A, MasterCharge, or 75% C.O.D.

* Up to this point, most people have been referring to the default-standard bus that is very widely used in the hobbyist community as an "Altair/IMSAI bus". We think its more appropriate, now, to call it the "hobbyist-standard bus".

65K BIT CCD'S FROM T.I.

Texas Instruments expects to offer prototypes of a 65K bit CCD (charge-coupled device) memory chip by the beginning of 1977. It will have a serial/parallel/serial (SPS) loop configuration, using 32 2K-bit loops.

65KB CCD'S FROM FAIRCHILD...SOONER

Fairchild expects to deliver 65K CCD chips even sooner; beginning fabrication in September and offering samples to selected customers in October. These memory chips, numbered CCD465, also use an SPS organization, and have traded speed for low cost, low power and high density. Average bit-access time will be half a millisecond; worst case will be one ms... comparable to most fast discs and drums.

64K CCD BOARD FOR \$1500+ BY END OF YEAR

BISI, 410 N.E. 72nd, Seattle, WA 98115, (206) 524-4101 [and Vancouver, B.C.], has recently distributed preliminary information on a 64K byte CCD (Charge-Coupled Device) subsystem. It is to function as a BORAM (Block-Oriented Random Access Memory; one reads or writes an entire block of bytes for each data transfer); a disc replacement. The prototype has been used for about two months in a burroughs system, replacing a head-per-track disc and is said to have had essentially no failures.

BISI is targeting for October availability, but sez end-of year for sure. The subsystem will plug into a hobbyist-standard bus and will require three boards. The CCD board (about \$1500) the controller board, and the DMA board (no prices suggested on the latter two boards). They will probably be available only as assembled, burned-in, subsystems, and will have a 90-day warranty.

Right now, that's still too expensive for hobbyists palates, but you can be certain that the cost will drop dramatically in two years (or much less), as CCD's become more plentiful and cost-competitive.

UNIVERSITY OF CALIFORNIA REQUESTS SUGGESTIONS ABOUT STAND-ALONE COMPUTERS FOR ITS 120,000 STUDENTS

The University of California is soliciting advice from vendors of computer equipment and programs. This is an opportunity to consult with us in advance rather than tell us what we should have done after the fact. We are earnestly seeking your best thinking prior to considering the publication of a Request for Proposal which will establish our direction for the near future. This document is *not* a Request for Proposal (RFP) but rather an invitation to share with us your ideas and recommendations whether they be basic or "blue sky." Your response to the Request for Information (RFI) is voluntary and does not commit either you or the University.

The University wishes to dramatically improve the system of delivering computing capability to the 120,000 students on the nine campuses and associated sites. Instructional capability has been provided through both batch and timesharing services utilizing existing large computers housed in a computing center located on each campus. In the last two years, several of these centers have installed timesharing minicomputers which allow for the connection of several (typically, 16 to 32) remote terminals, through which the student may write programs in BASIC or access prewritten programs provided by an instructor ("Courseware"). These minicomputer-based timesharing systems have proven to be very effective and they have the capability of expanding the services to meet academic needs. Such systems are flexible enough to meet a very large percentage of the needs of a very large percentage of the students.

We feel that, now, with the changes being brought about by improvements in chip technology and the corresponding explosion in microcomputer capability, even lower cost solutions may be possible without giving up anything in capability. With any timesharing system, unless all terminals can be hard-wired, there will be additional costs (beyond the computer and the terminals) for modems and communications lines. Technology has been producing orders of magnitude improvements in price/performance of computers, but communications costs are rising. Therefore, it seems prudent to search for solutions regarding delivery of computing which do not depend vitally on communications. Such a model might be found in a single-user machine with the following design goals:

Desirable Features

Interactive programming capability in some language such as BASIC or APL. The implementation should include the advanced features of the language.

User space approximately equivalent to that provided in multi-user timesharing systems.

User-removable read/write storage capability with "reasonable" retrieval times. (diskette, cassette, videodisk, etc.)

Both character and graphics capability on some screen medium. Appropriate software (firmware) extensions to deal with graphics.

Full typewriter or teletypewriter keyboard facility. Additional controls for cursor positioning for alphanumeric as well as graphics use.

Some form of course-writing software/firmware such as PILOT or DIALOG.

Editor program for preparation of programs to be executed on another computer.

Provisions for attachment of communications interface. This should provide two functions:

1. Easy interface to a host computer.
2. Ability to service input data streams from other devices such as laboratory experiments.

Extension Capability

Additional or alternative language capability.

Software/firmware for word processing applications.

Color screens for graphics.

Provisions for attachment of high-quality impact printer of reproduction quality as output from word-processing applications.

Physical and Packaging

Two physical versions. The first might be built into a desk or tabletop and would be considered non-portable. The second would be able to be carried by one person for short distances at least. Thus, this version would be "portable." The first version could be located in groups of five or more in a monitored room, while the second could be used in individual offices and classrooms.

Since the majority of the machines would not have attached printers, an additional system type with the capability of producing hardcopy from the screen image would be needed. It would have the ability of printing a page at a time, on request of the user, from his removable storage medium. Both graphic and character printing would be required. This machine could be another version of the One-on-One with special software/firmware or an entirely different piece of equipment.

All equipment would operate from standard 110v 3-wire circuits. No special electrical circuits are to be needed.

Maintainability

The system would be modular in design, allowing for board replacement to correct any failure. Adequate stocks of spare boards for each campus should be available. Board repair would be done by the vendor.

Alternatives

Although this is primarily a request for information for a single user system, the University does not wish to prejudice the thoughts and development that many vendors may have already devoted to the delivery system problem. Since our desire is to have the most cost-effective solution possible within the state of the art, a discussion of the pro and con of the above suggested approach compared with other possibilities would be most helpful. For example, a little more software or firmware might produce a cost-effective solution that might use a small number of keyboard devices clustered around a single processor. But the questions of file handling, of portability, and communications costs would need to be addressed as a minimum.

Another possible alternative might allow sharing of disk storage and common, read-only firmware while providing a micro-processor with each local unit.

If you possess original ideas which might not satisfy the requirements anticipated in our possible RFP, the response to the RFI would be your opportunity to influence our determination of constraints to be incorporated into the RFP, so that your particular configuration would be eligible for consideration. This is important, since it is expected that only those responses to the RFP which satisfy State qualifications therein will be accepted, i.e., no alternative proposals. Any concept which might run the risk of having to be in the alternate proposal category should be presented as a response to the RFI, *not* the RFP.

While there is no monetary remuneration for responding to this document, efforts expended should provide plans and text-material needed for your response to the anticipated RFP. Hopefully, the uniqueness and enormity of our problem will be sufficiently tantalizing to serve as an inducement to stimulate your thinking.

Responses to this RFI should be returned by October 1, 1976.

Questions

Technical questions should be addressed to:

Charles W. Stevenson
Manager, Computer Planning
Systemwide Administration
University of California
Room 192, University Hall
Berkeley, CA 94720
(415) 642-0393

Procurement matters will be handled by:

Robert Brilliant
Purchasing Agent
University of California
2405 Bowditch Street
Berkeley, CA 94720
(415) 642-0881

Personal Computers: A Bit of Wheat Amongst the Chaff

A Critique on "Little Hidden Gotchas" Found in a Multitude of Kits

by Jef Raskin
Box 511, Brisbane CA 94005; (415) 467-4674

Let it be known: the computer for everyman is not here. Yet. Caveat Emptor! The tempting price tags you see in the ads are a mere drop in the wallet. These are not Heath kits - you need a good bit of understanding of electronics and some about software to use them. Either that or a lot of patience and good learning habits, or a friend who knows what you don't. Of course, you could wait until Heath does come out with one [which should happen by the time you read this.—Editor].

Having bought and built the two biggest names in the personal microcomputer field, the Altair 8800 and the IMSAI 8080, I know that there is much to praise. Both machines now run perfectly, reliably, and perform within their specifications as well as any computer I have worked with. They are dependable. I have purchased memories from Processor Technology, Dutronics, Solid State Music and Cromeco, a total of 52 Kilobytes, and *all* have operated, once debugged, without error, week after week. The same is true for my IMS parallel I/O board, the MITS SIO and 2-SIO serial boards, the PT Video board and so on. They all work. But, *very few of them would have worked if I had not had prior electronics and computer experience.*

Before I go on to details, I should mention that I write from the point of view of someone who wants to use the computers in various applications; not learn about computers per se. I chose the kits because they are cheaper than anything else currently available (which means until approximately next week) in the way of true general purpose computers. If you want to be forced to learn about hardware the hard way, the inadequate documentation (and just plain bad writing) that comes with almost all the personal computing products and the poor workmanship and parts of a few, will make you prematurely wise, and give you a bad disposition.

I will begin with an exception: the *Cromemco Bytesaver*. The quality of the printed circuit is up to the best commercial practice. The assembly manual did not lie once, and their software documentation was quite adequate, including a complete how-to-use-it for the novice and a source listing of the "bytemover" program. It is a study in How To Do It Right. Component locations are labeled on the board. A solder resist pattern keeps the wobbly hand from building bridges where none are called for. The board was straight and true, and well laid out. It worked the first time, and caused us no frustration whatever.

On the other hand we have some amazing areas of incompetence and oversights. Never in my years of computing have I seen an assembler which omitted any items from the computer's instruction set. Yet *Processor Tech-*

nology's [PT] Software Kit No. 1 omits, for example, the POP PSW instruction. It will just not assemble it. This explains the otherwise mysterious SP EQU 6 in the source listings of the Bytemover and in Tarbell's listings. They must have used the same assembler. You needn't understand assembler to understand the problem. An assembler is primarily a program that takes memorable abbreviations and translates them into the more easily forgotten numbers the machine feeds on. There is supposed to be a mnemonic for each instruction that the machine can use. A mere check of the instruction list published by (say) Intel would have discovered the fact that some were missing from the assembler. That such a check was not made is indicative of the malaise that the personal computing industry is suffering: Lousy quality control.

Quality control must be worked at. I always work on computers with my friend, partner and critic, Doug Wyatt, another professional computer nut. Whatever I do, from circuit design to documentation to soldering 100 pin connectors into mother boards (yech!), he checks. Whatever he does, I check. We use a magnifying glass, a dictionary, anything that will help uncover errors. And, surprise, our stuff works the first time and keeps on working! *RULE: You can't catch all your own errors.*

The *Dutronics 8KLST memory* shows good quality control. It can be assembled according to the instructions and work, right off. Though 9 of the memory chips had problems (well, pretty good quality control) the 180 or so others were fine. While perhaps a step down from Cromemco's sheer professionalism in board layout, documentation and QC (quality control), Dutronic's instructions were clear and current for the board, and the schematic was useful and easy to follow. I wonder this: If one or two companies can do it right and stay competitive, why can't the others? *Is it because the consumers are letting them get away with it?*

A Kind Word for the Palo Alto (California) Byte Shop is in order here. The nine bad chips were instantly replaced with new ones—of course, we had bought the memory boards there. And with service like that we'll be back. Not all the computer stores have been so helpful, unfortunately.

The *Oliver Audio Engineering OP-80A paper tape reader* is a very handsome unit. The instructions were not up to the same high aesthetic standards. For example, you are told to bend the wire guides that align the paper tape with the read head. But they come pre-bent. Nothing wrong with that (it does make the kit a lot easier), but it seems to me that if Oliver (or whoever) goes through the considerable to bend the wires for each kit (and very accurately, by the way), is it asking too

much that a new page of instructions be typed as well? Then, too, we find comments such as “. . . and align the sensor as shown below.” And we are never shown below or anywhere else. We are told “Install J1.” Let’s see. J1 is not on the parts list. On the schematic? Nope. There the little bugger is, on the parts layout!

I had problems of this sort with almost every kit. A typical kit’s parts list would tell you that there are 4 74LO4’s, the schematic would call them U34, U21, U8 and U5; the layout diagram would have them labeled IC5, IC19, IC32 and IC26; the printed circuit board itself would be unlabeled altogether. And, to add insult to insult, three of them will be placed with pin one facing left and the last one with pin one facing right. Better be awake when you assemble this one. A flagrant example is the *Processor Technology “3P+S” board*, to which I shall return later. On this board the chips actually occur in all four possible rectangular alignments. This is pretty poor for a board that is mostly jumpers anyway. But back to our typical kit. I should mention that there was a little slip of paper which fell out when I opened the plastic bag. It said that the 74LO4’s were actually numbered WT87-9-6221. And just for you hackers who solder in your chips, there is an erratum (as usual placed at the end of the manual) saying that U21 is shown backwards on the parts layout. While fuming, remember that you will not only have to unsolder the thing but that you will have to figure out by tracing the schematic and the lands on the board whether U21 corresponds to IC5, IC19, IC32 or IC26 on the parts layout *and*, since the schematic is for REV 02 and the board is labeled REV 03 and the parts layout has no REV number at all, you can’t tell if the change is still necessary. For newcomers, REV usually means “Revision.”

While I have not run into all these problems simultaneously on any one board or kit, they all have occurred in real life (if playing with computers is real life). *Incidentally, I always use sockets.*

And Caveat Emptor to you if you want three parallel ports and a serial port, and you buy the PT “3P+S” board [3 parallel + 1 serial I/O board]. It works fine, to be sure (remember that I began by saying that everything does, finally, work) but if you truly need an 8-bit port, then you will need an extra bit or so for “handshaking.” For example, say a keyboard puts out eight data bits. When these are all ready for the computer, the keyboard sends out a signal (usually called the *strobe*) which tells the computer that a new character is on the data lines. Do I hear you ask: “Why can’t the computer tell that a new character is there by just noting when the character changes?” Well, just consider the South American llama. A computer can’t tell the first “l” from the second “l”, and would think you had typed “lama” thus changing a beast into a priest. Surprising what computers can do these days. Anyway, you need the strobe so you take a bit from another port and use it for that handshaking signal. Since the second parallel port needs handshaking, you take another bit or so from the third port and use it for that function. And since the serial port also requires control bits and etc., you steal a few more bits . . . Anyway, the “3P+S” is a good solid and flexible 2P+S with a control port. And, indeed, the instructions (quite extensive, but poorly organized), and the schematic all make it clear that the third parallel port was intended from the beginning as a control port for the others. Once again, marketing wins

out over technical accuracy. Most other brands of serial I/O boards do include a parallel handshaking port without calling themselves, say, a 2SIO+P.

Now, a *kudo for Processor Technology* is in order. Their *Video Display board (VDM-1)* was easily built (even if it’s got lotsa parts) because the instructions are clear. The construction manual even had a good step-by-step checkout procedure. It seems to be a product of a different, and better, company. And out PT memories all worked perfectly in my Altair. Not so in my IMSAI. The memory-protect would come on and off capriciously. There is no hardware protect switch on the early IMSAIs. Since I am not a hardware hacker (ha!) and am entirely above entering long programs from the switches, I found the memory-protect feature less than useless, and unwired the feature from my Altair. The Processor Tech memories were rendered perfectly reliable in both machines by replacing the 7400 used as the memory protect latch with two jumpers. This also gave me five spare 7400’s.

The *Processor Tech Altair motherboard* is another example of lack of care. The fundamental intent was good, and the board works, but there is much wasted board space, and a pile of bothersome jumpers. This was due to too slavishly copying the original MITS board. The bus terminator regulator installation is a kludge. Included with the motherboard kit were two delrin rods with saw slots in them, meant to hold the various daughter boards upright. There was an interesting oversight in their design. A good way to bend a stiff rod is to cut a series of slots part-way through the rod perpendicular to its long axis. These slots, intended to hold the boards up, in fact cause the rods to arc neatly *away* from the boards they are supposed to hold. This gives a new computer meaning to the word “floppy.”

The *Tarbell cassette interface* is a prime example of good ideas followed by sloppy implementation and negligent documentation. A fault invisible to the naked eye—plated-through holes that weren’t—plagued both my boards. While some updated documentation I have seen gives a few more hints than I got with mine, nowhere are we told, for example, that to use the interface, data must be preceded by a start byte (3C) and then a sync byte (E6) and *then* comes the data. You can dope this out from the example program given and from following through the schematic and reading your IC manuals but *that should not be necessary*. You can’t be sure from the sample program alone, by the way, that these details are necessary to run the board. They might be needed, as far as you can tell from the given program, by the particular algorithm being used, or perhaps by some other software that Tarbell uses somewhere else. Here, too, the assembly drawing did not have part values marked, necessitating much flipping back and forth from page to page during assembly. *I get the impression from most kits that the instructions were never tested on persons unfamiliar with the device.*

A pet peeve: *Cassettes* have little tabs to break out if you don’t want the stuff on them written over. Not one tape pre-recorded by a manufacturer, including the BASIC from MITS that I paid for, had the write-protect tabs knocked out. Another pet peeve: Some manufacturers’ *printed circuit boards* have a specially shaped land where pin 1 of each IC fits. The peeve is this: those who don’t. A further peeve: The placing of modification and *errata*

sheets at the end of the documentation.

And here's a handful more of horror stories; little things that make computers seem much more frustrating than they need be. Like the un-numbered pages in so many instruction manuals. The *IMS PIO board* manual is completely unpaginated. Since I've mentioned this manual, let me show you how a bit of thoughtlessness on the part of the writers cost me a few hours. On page 12 (if you write in the page numbers yourself) you are told how to wire the address jumper sockets. Good. There are two, the one on top is called C8, and the one on the bottom, B9. I must be dumb, because I wired the top socket on the board as shown for C8, and the bottom socket as shown for B9. Wrong. The diagram is schematic in nature, and the sockets are not that way at all. If I had bothered to check my assembly drawing or memorized the labels for each IC position I would not have been so harried. B9, shown on the bottom in the illustration is, of course, *above* C8 on the board.

I think these problems stem from this: the people who know best what their equipment is intended to do seem to know it so well that they can't tell new users what is going on. Certainly Mr. Tarbell designed his board with the idea that a code of 3C was required to get it started. But the sentence that would have made this knowledge public: "The board must be sent a byte containing 3C (in hex) to start it and then a byte E6 (in hex) to synchronize it before sending data.", does not occur. Similarly, if the PT "3P+S" board advertisements and manual stated, "This board contains the hardware for one serial port and two parallel ports with a separate status and control port for them," I would have had fewer complaints. If the manufacturers would *test their assembly and user manuals on a few people not familiar with the design* of the module, they would quickly find out how to improve their style. And, if we users would stop buying badly kitted equipment, and demand well-written manuals, we'd get improved kits and better written manuals.

The current software situation is even worse. But that is a topic for another time.

A WORD ABOUT THE AUTHOR

Jef Raskin has been playing with computers since late in grammar school. Not that he was often late. He has been Director of a small computer center at U.C., San Diego, and taught such courses as computer programming, computer animation, computer music at San Diego, as well as a number of other colleges, universities, and in his basement where he has had both an Altair and an IMSAI running. He is well known for his heretical belief that people are more important than computers, and that computer systems should be designed to alleviate human frailties, rather than have the human succumb to the needs of the machine.

Editor's Note: Further lurid comments by Jef may be found on page 8 of August, 1976, *Datamation*.

A TIP ON SOLDERING FROM JIM DAY

A new type of solder, designed to prevent the dissolving of printed circuit traces during soldering, has been developed by Multicore Solders, Westbury, NY 11590. Called Ersin SAVBIT, this copper-loaded alloy also reduced the pitting of copper soldering iron tips. As every computer hobbyist knows, it can be a real nuisance to have to file a soldering iron tip every few minutes. (Iron-plated tips don't pit as easily, but cost more.)

IMSAI OWNERS - BEWARE OF THE MEMORY CLOBBERING PHANTOM!

You're perking along and suddenly the machine blows up. Careful autopsy of the wreckage reveals that a string of memory locations has been zapped to 00 or 02 or 06. And this is happening several times a week. The phantom strikes! I have spent nearly a month chasing this bug and have found that the culprit is the deposit one-shot (U-17) on the front panel. It gets triggered while the machine is running by noise on the power line which passes right by the one-shot timing capacitors - a major design boo-boo. There are several approaches to fixing this bug: You can disconnect the power switch somewhere else. This may still leave you vulnerable to really large noise spikes. You can install a pi filter on the power line (two 8 uH 5 amp inductors in series with each side with two .03 uF capacitors across) or you can disconnect the A input of the one-shot and connect it instead to the Q output of the Run/Stop flip-flop. That is, clip pin 9 of U-17 from ground and jumper it to pin 5 of U-22. This effectively inhibits the one-shot by holding the A input high while the machine is running. This last seems to work for me.

Let me know how yours is doing.

Pete Cornell Box 1290
Los Altos, CA 94022
(415) 948-8269

TARBELL COMPLAINTS

We are starting to hear a disturbing number of complaints about the Tarbell cassette interface. These are disturbing (1) because a number of hobbyists appear to be having problems with the units, (2) because we think the design principles of the Tarbell unit are sound and the speed is highly desirable, and (3) because we believe that Don Tarbell is sincerely trying to offer a good product at reasonable cost. Our distinct impression is that Don is one of the "good guys"; he is not greedy, and is not trying to rip off the hobbyists by intentionally selling imperfect equipment.

If you are having problems, we suggest that you write or phone him, directly. Tell him what the problem is. If you feel you know a solution, please pass it along to him (and to us; we'll publish it if we feel it appropriate to do so). Our impression is that Don will make every effort to "do right" by his customers.

To contact him: Don Tarbell, 144 Miraleste Dr. No. 106, Miraleste CA 90732; (213) 832-0182.

[Don, We would be delighted to publish any reasonable reply you may wish to forward.]

BIBLICAL MATHEMATICS

We hear that Noah told the animals in his ark to multiply, and two snakes refused to do so because they were adders. Noah overcame this objection, however. He quickly placed the snakes on a rough-hewn wooden table and told them to follow his command, for, as we all know, adders can multiply by using a log table. --Snickered around PCC

HIDDEN GOTCHA FOR 16K BASIC DOCUMENTATION

Gentlepersons: August 16, 1976
Anyone planning to use the 16K BASIC for the 8008 should be warned that the publication mentioned in *Dr. Dobb's Journal*, Vol. 1, No. 3, p. 4, does not include a listing. For more details, see the *Micro-8 Newsletter*, Vol. 1, No. 10, p. 15.

Sincerely yours,
Robert F. Miles

PALO ALTO TINY BASIC CONVERTED TO INTEL-STANDARD MNEMONICS

We published complete details of Li-Chen Wang's Palo Alto version of Tiny BASIC in our May, 1976 issue. This included a well-documented source listing for the program. The assembler mnemonics (abbreviations for machine instructions) that Wang used, however, were not those used by Intel and by many of the 8080 assemblers that are available. Those desiring a listing of Palo Alto Tiny BASIC, coded in Intel mnemonics, may contact Roger Rauskolb, 3766 Cass Way, Palo Alto CA 94306; (415) 493-1501x3608. If the demand isn't too great, he will furnish such listings. If the number of requests becomes more than he wishes to handle, we will consider publishing his translation in the *Journal*—however, we are not eager to do so, since there is so much other good stuff awaiting publication.

CONTRA COSTA COUNTY CLUB STARTING UP

The following person is organizing a computer club in the area of Contra Costa County, California. Contact him if you are interested in this club—the fourth computer hobby group in the San Francisco Bay area.

Eric Bergman, 3188 Kingsley Pl, Lafayette CA 94549; (415) 283-6256.

A BYTE LOSES SOME BITS IN SAN FRANCISCO

Al Cherin of the Computer Store of San Francisco (1093 Mission St., 94103; (415) 431-0640), phoned us on August 20th to tell us that his Store is now independent of Byte, Inc. He sez he is now dealing directly with his sources such as IMS Associates, Processor Technology, etc. (rather than routing his product purchases through the Byte, Inc. franchise).

HOMEBREW IBM OFFICE SELECTRIC INTERFACE DESIGN

Want to use a standard, IBM Selectric I office typewriter to obtain superb quality computer hard-copy? Page 218-220 of Don Lancaster's *TV Typewriter Cookbook* tells how to do so with a few solenoids and some simple, non-impairing mods to the typewriter.

SOURCES FOR 8080 FLOATING POINT ROUTINES & A STRUCTURED ASSEMBLER: ML 80

Dear Fleet-foot dentists: June 26, 1976
May I suggest the following source for floating point routines for 8080 systems.

Cope, S.N., Floating-point Arithmetic Routines and Macros for an Intel 8080 Microprocessor, Oxford Univ, 1976. This is available in the public domain through NTIS, 5285 Port Royal Rd., Springfield, VA 22151. Order number N76-11756 as microfiche for \$2.25 or paper copy for \$4. The routines take 335 bytes total, give 4½ digit precision and are relatively fast. Add in 0.25 ms, multiply or divide in .75 ms, and square root in 1.77 ms. These are typical times, maximums are slightly higher. The NTIS announcements are beginning to carry assorted microprocessor oriented reports now that funding and research/publication lead times have been passed by processor availability.

While it seems to have a bit of overbyte, their AD-AO20055 (\$2.25 in fiche, around \$8 in paper) reports on ML-80, a PL 360-like language for the 8080. It needs 16K to run however.

Sincerely,
Darrel J. Van Buer and 1516 Electric Ave., Apt.R
Hymie, the computer Venice, CA 90291
(213) 823-6016

P.S. Hymie is not an acronym. He is named in honor of the CONTROL robot.

INTERRUPT-DRIVEN PASCAL AND AN OFFER TO HELP

Dear Jim,

I am very interested in a project for a micro version of PASCAL. I think the I/O interface of an interactive programming system is even more important than the actual language structure. PASCAL's I/O system lends itself to the kind of things I think are needed. In particular it should be possible to have interrupt-driven input initialized without waiting for completion. This makes possible real-time programs such as Star Trek where entries can be made while the action continues. This is essential for the whole class of real-time games.

My interest is in the 6800 primarily. Let me know if I can provide any help in your PASCAL project.

Dr. Brian W. Johnson 1525 Westlake
Plano TX 75074

Can you help? Sure! Much is needed: tutorials describing how to program in a block-structured language, a brief historical article, notes on the advantages of PASCAL over other block-structured languages such as ALGOL and PL/I, well-documented sample programs and games written in PASCAL, a detailed comparison of PASCAL with BASIC including examples, and much, much more . . . LOTS to do!

Incidentally, PASCAL as Wirth originally designed it, did not have anything especially unusual or exciting for I/O. In particular he did not concern himself with interrupt-driven I/O. Per Brinch Hansen *did*, however. Brinch Hansen extended Wirth's original design into "Concurrent PASCAL," which is (1) explicitly designed to handle parallel processing problems, and (2) is implemented on PDP-11's. —JCW

I wish to express a certain disgruntlement with some of the gargoyles that have been constructed on the cathedral of LISP.

—John McCarthy, speaking at a Menlo Park SIGPLAN meeting.

How to Make the 6800 Resident Assembler Work in Your System

by Tom Pittman

(Reprinted with permission from the *Homebrew Computer Club Newsletter*)

These two programs were designed to co-reside in 8K of RAM with 762 bytes of workspace remaining for edit buffer or symbol table. If you have more, your workspace is thus larger. Also, both the editor and assembler are designed to be able to use the other's program space for additional workspace if desired. Figure 1 is a memory map of the system. The first two pages (locations 100-2B5) in these two programs are the same, and contain all of the I/O routines for both. If location 300 is non-zero, the Editor will use the assembler program space for additional buffer. The Editor is designed to use all available contiguous memory, but the Assembler will limit its symbol table to memory below 2000, or below the contents of 301-302.

The key to the I/O routines is a "transfer vector" starting in location 0100. There are 19 or so JMP instructions in this vector which permit access to all of the routines, so that the entry to the routines do not depend on their sizes. By careful analysis of these routines I have made the following determinations.

The I/O routines are designed to work with a Teletype ASR33 or Silent 700 with cassettes, where the same data is used for both printed and tape copy. The paper tape reader on the terminal may be controlled either by X-ON/X-OFF or by a relay connected to an I/O line on the computer. The punch is controlled by TAPE/TAPE- or the program is instructed to make separate punch passes. There are two versions of the I/O routines, one for MIKBUG which uses a PIA for the TTY interface, and one for EXBUG, which uses an ACIA. MIKBUG is a crummy operating system (it was designed as an evaluation program), and Motorola is pretty tight about giving out information on EXBUG except with the sale of an Exorciser, so unless you have one of these in your system you are probably better off rewriting the I/O.

The tape I have seen was done for EXBUG, and since it is most likely to be the same as your copy, the following remarks apply to that version.

There are 31 references to upper memory in the I/O routines (none in the main part of the assembler and editor; I checked). Of these, there are four references to FF02 (locations 018B, 01CD, 023F, and 025F); this is apparently a flag to indicate whether the terminal is a Silent 700 or not, with 00=not. All four are TST instructions, which may be altered to test any zero memory location. There are two references to FF62 and one to FF53 (0154, 023C, and 0271), which seem to be related to the EXBUG I/O flags; they may be NOPed out (three NOPs will replace each reference).

It is assumed that location FCFD contains some kind of constant for the paper tape reader control; this is fetched in locations 01DE and 0205. Bit 5 of this byte should be 0, and bit 6 should be 1 if RTS of the ACIA is used for reader control, and zero otherwise; bits 0, 2,

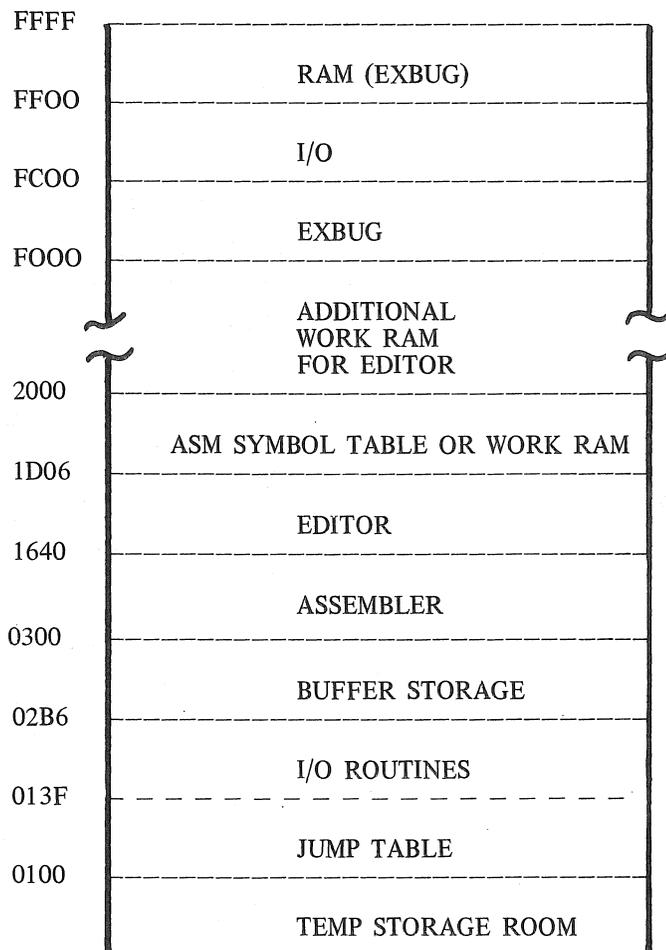


Figure 1. (EXBUG) Resident Assembler And Editor Memory Map

and 4 should be ones. It may be convenient to NOP out these two fetches, and replace the opcodes of the respective following instructions with LDA A immediately (instead of AND A). Note that bit 5 enables interrupts, which may not be desired, so No. 15 should be used instead of No. 35.

The ACIA is assumed to be at locations FCF4-FCF5. Two references to its control register are used to turn the paper tape reader on and off (locations 01E3 and 020A). Two references to the input data register (locations 0212 and 0215) occur to clear out any remaining input after turning off the reader. Location 0287 checks the status register for input during a timeout loop, and 028D reads the data character when it comes. If your ACIA is in a

different location, you may later the addresses in these instructions.

There are seven subroutines in EXBUG which are called from these I/O routines. Of these, four directly must have correspondence in your system. They input a single character from the ACIA, output a single character, and exit to the monitor or operating system; they are summarized in Table 2.

Table 2 also shows the calls for composite operations. These, however, may be encoded easily to rely only on the single character input and output functions. The discussion to follow describes how these functions may be encoded in an arbitrary system.

Assume that your operating system provides a single call each for one character input "INPUT", and one character output "OUTPUT". We also assume that if your terminal requires nulls between the CR and LF that your operating system will detect the CR and insert

| | | | |
|------------------------|-------|--------|-------------------|
| * Type CR & LF | | | |
| F021 | LDA A | #OD | Carriage return |
| | JSR | OUTPUT | |
| | LDA A | #OA | Line feed |
| | JMP | OUTPUT | |
| * Type String | | | |
| F024 | JSR | F021 | First type CRLF |
| F027 | LDA A | O,X | Get character |
| | INX | | Increment to next |
| | CMP A | #04 | Is it end? |
| | BEQ | EXIT | Yes. |
| | JSR | OUTPUT | No, type it. |
| | BRA | F027 | Go get another. |
| EXIT | RTS | | Quit when done. |
| * Input & Output Links | | | |
| F015 | JMP | INPUT | |
| F018 | JMP | OUTPUT | |
| F9CF | JMP | OUTPUT | |

Figure 2. I/O Linkages to substitute for EXBUG

| | | | |
|-------------------------------|-------|---------|--------------------------------|
| * File/Terminal I/O separator | | | |
| INPUT | TST | RDRON | Check if File Input |
| | BEQ | TERMIN | No, Terminal. |
| | BRA | FILEIN | Yes, file. |
| OUTPUT | CMP A | #11 | Reader-On Control? |
| | BEQ | XON | Yes. |
| | CMP A | #12 | Tape-On? |
| | BEQ | TAPE | Yes. |
| | CMP A | #13 | Reader Off? |
| | BEQ | XOFF | Yes. |
| | CMP A | #14 | Tape Off? |
| | BEQ | NTAPE | Yes. |
| | TST | PCHON | Is "Punch" on? |
| | BEQ | TERMOUT | No, Terminal. |
| | BRA | FILEOUT | Yes, file. |
| XON | INC | RDRON | Turn on "reader" = file input. |
| | RTS | | |
| XOFF | CLR | RDRON | Turn off file input. |
| | RTS | | |
| TAPE | INC | PCHON | Turn on file output. |
| | RTS | | |
| NTAPE | CLR | PCHON | Turn off file output. |
| | RTS | | |

Figure 3. Suggestion for Terminal/File separation

the necessary nulls in the output call. If you have separate devices for file input and output which do not go through the terminal I/O, the beginning of INPUT and OUTPUT will be complicated sufficiently to distinguish file I/O from terminal I/O as described a little later.

Figure 2 is a sample program to provide the functions supplied by EXBUG. The various entries are labeled in the listing by the address in EXBUG which provides the same function (see Table 2). Normally you would include these routines in your operating system monitor, and patch jumps to them from the I/O linkages in lower memory. If you do this, you probably also wish to modify the references to upper memory described earlier to point to your own ACIA.

For a completely generalized I/O system, you need

| LOCN | FUNCTION |
|------|---------------------------------------------------------|
| 0100 | Beginning of Assembler |
| 0103 | Beginning of Editor |
| 0106 | Initialize I/O flags in F2, F5, F6 |
| 0109 | Output character in A to Punch, adding LF to CR |
| 0100 | Punch character string pointed to by X, ended by 04 |
| 010F | Input (buffered) character from reader to A |
| 0112 | Read a line into input buffer; first char to A |
| 0115 | Type character in A, adding LF to CR, conv Tab to space |
| 0118 | Input character (from keyboard) to A |
| 011B | Input character to A (with no echo?) |
| 011E | Type text string on new line |
| 0121 | Type text string pointed to by X, ended by 04 |
| 0124 | Type CR & LF |
| 0127 | Punch 32 Nulls for leader |
| 012A | Turn on Punch (if not already on) |
| 012D | Turn off Punch |
| 0130 | Jump to monitor (exit from program) |
| 0133 | Jump to 011E |
| 0136 | Jump to 0121 |

Table 1. I/O Routines Jump Table

| CALL FROM | CALL TO | FUNCTION |
|-----------|---------|-----------------------------------------------------------------------|
| 0157 | F015 | Accept input character from terminal |
| 0177 | F018 | Type character to terminal (or punch) |
| 01A5 | | |
| 01E8 | | |
| 020F | | |
| 0130 | F564 | Monitor exit |
| 01D4* | F9CF | Output character to ACIA |
| 01D9* | | |
| 0246* | | |
| 024B* | | *these calls are skipped if location FF02 (or its substitute) is = 00 |
| 0250 | | |
| 0255 | | |
| 025C | | |
| 0266* | | |
| 026B* | | |
| 011E | F024 | Type string on new line |
| 0121 | F027 | Type string |
| 0124 | F021 | Type CR & LF |

Table 2. Assembler and Editor calls to EXBUG

to set up separate calls for I/O on the terminal-console and I/O on the data file. Operating system commands can then be used to designate different devices as file or terminal, so that for example, you can assemble from cassette or paper tape, edit from one device to another, etc. Figure 3 shows how two flag bytes in RAM somewhere (labeled PCHON and RDRON) can be used to direct a single stream of output to either the terminal or the output file, and conversely to accept a byte from either the terminal or the input file. Note that this routine traps the Teletype Tape controls to select the data stream, and does not pass these controls through to the output. If you are going to do this, you should delete references to the ACIA from the I/O routines in lower memory (see Table 3), and insert the following instruction in the input routine in lower memory:

```
O1EF      JSR      FILEIN
```

Note that the monitor subroutine FILEIN (as all good monitor routines) should not alter the Index register or the B accumulator. On return, FILEIN leaves the Carry flag set to one to indicate an end-of-file, otherwise the carry is cleared. In the case of an end-of-file, the A should still contain some appropriate character (I realize this is not the proper way of designing a system--so complain at Motorola) such as hex 1A (Control Z, which is not the ASCII EOF, but is used by these programs as an EOF character).

It should be noted that most of this information was inferred by analysis of the dis-assembled software, and lacks some of the insight afforded by true annotated assembly listings. Therefore, no warranty is made for the accuracy of the interpretations presented here. I have attempted to verify my findings, and it does seem to work. Please let me know if you find any differences or exceptions to this analysis.

| LOCN | INSTRUCTION | FUNCTION |
|------|-------------|------------------------------------|
| 0154 | INC FF53 | ?? |
| 023C | INC FF62 | Exbug Punch-on flag turned on |
| 0271 | CLR FF62 | Exbug Punch-on flag turned off |
| 018B | TST FF02 | Silent 700 flag (0=off) |
| 01CD | " " | |
| 023F | " " | |
| 025F | " " | |
| 01DE | LDA A FCFD | Fetch reader control type flag |
| 0205 | " " | |
| 01E3 | STA A FCF4 | Turn on RTS in ACIA |
| 020A | LDA A FCF4 | Turn off RTS in ACIA |
| 0212 | LDA A FCF5 | Discard pending input |
| 0215 | " " | |
| 0287 | LDA A FCF4 | Test for input during timeout loop |
| 028D | LDA A FCF5 | Input character in ACIA |

Table 3. Data References to Upper Memory

Dear Bob,

July 1, 1976

Some comments on D.D.J.C.C.&O!!!! First, Tiny BASIC itself: Basically OK (I'm sorry about that. . .really I am) but it lacks some flexibility because you don't have floating point math and a few simple functions like sin and cos. Look at it this way--with Tiny BASIC extended you are halfway there to writing 4K BASIC. . .25% of the way to 8K BASIC. In my opinion it is a waste of a \$1000 machine to be running just TB on it, when if you just added another \$100 or \$200 you could be running some superior software. Of course there is the problem of getting the software cheap too, so I may be wrong to say that.

Since I am sure everyone is patting everyone else on the back what with D.D.J.C.C.&O, I guess you can hear some criticism. You may have made a mistake by introducing a new newsletter and taking stuff out of P.C.C. The way I got interested in home computing was by reading all about assembler and machine code and hardware in P.C.C. If you take it out, you may be isolating high school people playing games on big systems from home computer people, since the high level language programmers won't bother with a newsletter for home system people. Furthermore there are now two newsletters, not one, and you know how bad it is to pay for all that stuff. Maybe you could offer a special rate when people subscribe to both at once, or send a free issue to people who subscribe to P.C.C. for the first time.

This may, on the other hand, work out OK since you will now have room in P.C.C. to explain stuff like assembler, machine code, and hardware in P.C.C. and not just list it or mention it. [Yes!]

Who is this Dr. Dobb character anyway? [A visitor from the planet Pern.]

One last item: I've seem BASIC take a lot of knocks lately, what with people saying, "BASIC is a very inefficient language and it doesn't suit the needs of the home computer user and besides it is hard to learn and who said we need it anyway and there must be a better language for games etc., etc., etc." but I have yet to see any concrete stuff in the way of ideas for a better language. Besides, you don't see people clamoring for FORTRAN or COBOL for their home systems! Or how about JCL! Think, if you were designing the ideal language what would you put in it? Well, first we need something to print stuff on the terminal so we'll use a PRINT statement and then we need to be able to jump from one spot to another so we'll use a GOTO. I guess maybe PL/M or something might be better but if all the software brains in all the computer software factories can't come up with anything better, knowing that their users will pay millions of dollars for it, why should some people who mess around with this stuff in their spare time think they can? Sorry to get crabby like this (again) but you know how it is.

Steve North

7 Deerhaven Lane
Newfoundland, NJ 07435

| LOCN | FUNCTION |
|--------|--------------------------------------|
| 00F0-1 | Temporary storage for Index register |
| 00F2 | Input Buffer non-void flag |
| 00F3-4 | Input Buffer pointer |
| 00F5 | Input flag (no timeout if set) |
| 00F6 | Punch Flag 0=print only |
| 00F7 | Input flag ?? |

Table 4. References to Page 0 by I/O Routines

A 6502 Disassembler from Apple

by Steve Wozniak & Allen Baum
 Apple Computer Co., 770 Welch Rd., No. 154
 Palo Alto CA 94304; (415) 326-4248

DESCRIPTION

This subroutine package is used to display single or sequential 6502 instructions in mnemonic form. The subroutines are tailored to disassemblers and debugging aids, but tables with more general usage (assemblers) are included. The subroutines occupy one page (256 bytes) and tables most of another. Seven page zero locations are used.

FEATURES

Four output fields are generated for each disassembled instruction: 1) Address of instruction, in hexadecimal (hex); 2) Hex code listing of instruction, 1 to 3 bytes; 3) 3-character mnemonic, or "???" for invalid ops (which assume a length of 1 byte); and 4) Address field, in one of the following formats.

| Format | Address Mode |
|----------|-------------------------------------------|
| (empty) | Invalid, Implied, Accumulator |
| \$12 | Page zero |
| \$1234 | Absolute, Branch (<i>target</i> printed) |
| #\$12 | Immediate |
| \$12,X | Zero page, indexed by X |
| \$12,Y | Zero page, indexed by Y |
| \$1234,X | Absolute, indexed by X |
| \$1234,Y | Absolute, indexed by Y |
| (\$1234) | Indirect |
| (\$12,X) | Indexed Indirect |
| (\$12),Y | Indirect Indexed |

Note that unlike MOS TECHNOLOGY assemblers, which use "A" for accumulator addressing, the APPLE disassembler outputs an empty field to avoid confusion and facilitate byte counting.

USAGE

The following subroutine entries are useful.

| | |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DSMBL | Disassembles and displays 20 sequential instructions beginning at the address specified by the page zero variables PCL and PCH. For example, if called with \$D2 in PCL and \$38 in PCH, 20 instructions beginning at address \$38D2 will be disassembled. PCL and PCH are updated to contain the address of the last disassembled instruction. Must be called with 6502 in hexadecimal mode ('D' status bit clear). All processor registers are altered (except S—stack pointer). Uses INSTDSP and PCADJ. |
| INSTDSP | Disassembles and displays a single instruction whose address is specified by PCL and PCH. Must be called in hexadecimal mode. All processor registers (except S) are altered. Uses PCADJ3, PRPC, PRBLNK, PRBL2, PRNTAX, PRBYTE, and CHAROUT. |
| PRPC | Outputs a carriage return, 4 hex digits corresponding to PCH and PCL, a dash, and 3 blanks. Alters A, clears X. Uses PRNTAX and CHAROUT. |
| PRNTX | Outputs the contents of X as two hex digits. Alters A. Uses CHAROUT. |
| PRNTAX | Outputs two hex digits for the contents of A, |

then two hex digits for the contents of X. A is altered. Uses CHAROUT.

| | |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PRNTYX | Same as PRNTAX except that Y and X are output. Alters A. Uses CHAROUT. |
| PRBLNK | Outputs 3 blanks. Alters A, clears X. Uses CHAROUT. |
| PRBL2 | Outputs the number of blanks specified by the contents of X (0 for 256 blanks). Alters A, clears X. Uses CHAROUT. |
| PRBL3 | Outputs a character from the A register followed by X-1 blanks. In other words, X specifies the total number of characters output. (0 for 256 blanks). Alters A, clears X. Uses CHAROUT. |
| PCADJ | (PCL,PCH) + 1 + (contents of page zero variable LENGTH) → Y & A (low order byte in Y). For example, if PCL = \$D2, PCH = \$38, and LENGTH = 1 (corresponding to a 2 byte instruction), PCADJ will leave Y = \$D4 and A = \$38. X is always loaded with PCH. |
| PCADJ2 | Same as PCADJ except that A is used in place of LENGTH. |
| PCADJ3 | Same as PCADJ2 except that the increment (+1) is specified by the carry (set = +1, clear = +0). |

RUNNING AS A PROGRAM

The following program will run a disassembly.

Supplied on APPLE-1 { 9F0 200 8 JSR DSMBL
 cassette tapes. { 9F3 4C1FFF JMP MONITOR

First, put the starting address of code you want disassembled in PCL (low order byte) and PCH (high order byte). Then type 9F0 R CR (on APPLE-1 system). 20 instructions will be disassembled. Hitting R CR again will give the next 20, etc.

Cassette tapes supplied for the ACI-1 (APPLE Cassette Interface) are intended to be loaded from \$800 to \$9FF.

NON-APPLE SYSTEMS

Source and object code supplied occupies pages 8 and 9. All code is on page 8, tables are on page 9. These tables may be relocated at will: MODE, MODE2, CHAR1, CHAR2, MNEM1, and MNEMR. The code may also be relocated. Be careful if you use pages 0 or 1. Page 1 is the subroutine return stack and page 0 must contain 7 variables (to use DSMBL). These may be relocated on page 0 but PCL must always immediately precede PCH for (Z-page), Y addressing.

| | | | |
|-----------|--------|--------|----------------------|
| | \$40 | FORMAT | Used |
| locations | \$41 | LENGTH | } by |
| used | \$42 | LMNEM | } INSTDSP, |
| by | { \$43 | RMNEM | DSMBL |
| supplied | \$44 | PCL | } Used by PCADJ, |
| code | \$45 | PCH | } INSTDSP, DSMBL |
| | \$46 | COUNT | } Used by DSMBL only |

MODIFICATIONS

- To change '#' to '=' for immediate mode change location \$955 (on code enclosed) from a \$A3 to a \$BD.
- To skip the '\$' (meaning hex) preceding disassembled values make the following changes:

946: 01 (was 81)
 947: 02 (was 82)
 94C: 11 (was 91)
 94D: 12 (was 92)
 94E: 06 (was 86)
 950: 05 (was 85)
 951: 1D (was 9D)
 95B: 00 (was A4)
 95C: 00 (was A4)

c) To have address field of accumulator-addressed instructions print as 'A'.

- 1) Must skip \$ preceding disassembled values by making modification b) above.
- 2) Change the following locations:
 949: 80 (was 00)
 957: C1 (was A4)

d) To add ROR and addressing modes, change the following locations:

991: 9C (was 00)
 9D1: 26 (was 00)
 919: 02 (was 00)
 91A: 45 (was 40)
 91B: B3 (was B0)
 91D: 08 (was 00)
 91F: 09 (was 00)

COUNT FOR 20 INSTR DSMBLY.
 DISASSEMBLE AND DISPLAY INSTR.
 UPDATE PCL:H TO NEXT INSTR.
 DONE FIRST 19 INSTRS.?
 * YES, LOOP. ELSE DSMBL 20TH.
 PRINT PCL:H.
 GET OP CODE.
 * EVEN/ODD TEST.
 * TEST B1.
 * XXXXXX11 INSTR INVALID.
 * 10001001 INSTR INVALID.
 MASK 3 BITS FOR ADDRESS MODE &
 * ADD INDEXING OFFSET.
 * LSB INTO CARRY FOR
 * LEFT/RIGHT TEST BELOW.
 INDEX INTO ADDRESS MODE TABL.
 IF CARRY SET USE LSD FOR
 * PRINT FORMAT INDEX.
 * IF CARRY CLEAR USE MSD.
 MASK FOR 4-BIT INDEX.
 \$0 FOR INVALID OPCODES.
 SUBSTITUTE \$00 FOR INVALID OP,
 SET PRINT FORMAT INDEX TO 0.
 INDEX INTO PRINT FORMAT TABLE.
 SAVE FOR ADDRESS FIELD FORMAT.
 MASK 2-BIT LENGTH, 0=1-BYTE,
 * 1=2-BYTE, 2=3-BYTE.
 * OP CODE.
 MASK IT FOR 1XXX1010 TEST.
 * SAVE IT.

ASSEMBLER REV 1.4 8/11/76
 XREF
 FORMAT EQU #40
 LENGTH EQU #41
 LNAME EQU #42
 RNAME EQU #43
 PCL EQU #44
 PCH EQU #45
 COUNT EQU #46
 PREBYTE EQU #FFDC
 CHAROU EQU #FFEF
 ORG #800
 #13
 LDA STA COUNT
 #000 #09 13
 #002 #05 46
 #004 #20 12 00 DSMBL2 JSR INSTDISP
 #007 #20 EF 08 PCADJ
 #00A #05 44 STA PCL
 #00C #04 45 STY PCH
 #00E #06 46 DEC COUNT
 #010 #00 F2 BRE DSMBL2
 #012 #20 D3 08 INSTDIS JSR
 #015 #01 44 LDA (PCL,X)
 #017 #03 TAY
 #018 #04 LSR A
 #019 #00 08 BCC
 #01B #04 LSR A
 #01C #00 17 BCS ERR
 #01E #09 22 CAP #22
 #020 #00 13 BEQ ERR
 #022 #29 07 AND #27
 #024 #03 00 ORA
 #026 #04 4A LSR A
 #028 #04 4A LSR A
 #02A #04 4A LSR A
 #02C #04 4A LSR A
 #02E #04 4A LSR A
 #030 #04 4A LSR A
 #032 #04 4A LSR A
 #034 #04 4A LSR A
 #036 #04 4A LSR A
 #038 #04 4A LSR A
 #03A #04 4A LSR A
 #03C #04 4A LSR A
 #03E #04 4A LSR A
 #040 #04 4A LSR A
 #042 #04 4A LSR A
 #044 #04 4A LSR A
 #046 #04 4A LSR A
 #048 #04 4A LSR A
 #04A #04 4A LSR A
 #04C #04 4A LSR A
 #04E #04 4A LSR A
 #050 #04 4A LSR A
 #052 #04 4A LSR A
 #054 #04 4A LSR A
 #056 #04 4A LSR A
 #058 #04 4A LSR A
 #05A #04 4A LSR A
 #05C #04 4A LSR A
 #05E #04 4A LSR A
 #060 #04 4A LSR A
 #062 #04 4A LSR A
 #064 #04 4A LSR A
 #066 #04 4A LSR A
 #068 #04 4A LSR A
 #06A #04 4A LSR A
 #06C #04 4A LSR A
 #06E #04 4A LSR A
 #070 #04 4A LSR A
 #072 #04 4A LSR A
 #074 #04 4A LSR A
 #076 #04 4A LSR A
 #078 #04 4A LSR A
 #07A #04 4A LSR A
 #07C #04 4A LSR A
 #07E #04 4A LSR A
 #080 #04 4A LSR A
 #082 #04 4A LSR A
 #084 #04 4A LSR A
 #086 #04 4A LSR A
 #088 #04 4A LSR A
 #08A #04 4A LSR A
 #08C #04 4A LSR A
 #08E #04 4A LSR A
 #090 #04 4A LSR A
 #092 #04 4A LSR A
 #094 #04 4A LSR A
 #096 #04 4A LSR A
 #098 #04 4A LSR A
 #09A #04 4A LSR A
 #09C #04 4A LSR A
 #09E #04 4A LSR A
 #0A0 #04 4A LSR A
 #0A2 #04 4A LSR A
 #0A4 #04 4A LSR A
 #0A6 #04 4A LSR A
 #0A8 #04 4A LSR A
 #0AA #04 4A LSR A
 #0AC #04 4A LSR A
 #0AE #04 4A LSR A
 #0B0 #04 4A LSR A
 #0B2 #04 4A LSR A
 #0B4 #04 4A LSR A
 #0B6 #04 4A LSR A
 #0B8 #04 4A LSR A
 #0BA #04 4A LSR A
 #0BC #04 4A LSR A
 #0BE #04 4A LSR A
 #0C0 #04 4A LSR A
 #0C2 #04 4A LSR A
 #0C4 #04 4A LSR A
 #0C6 #04 4A LSR A
 #0C8 #04 4A LSR A
 #0CA #04 4A LSR A
 #0CC #04 4A LSR A
 #0CE #04 4A LSR A
 #0D0 #04 4A LSR A
 #0D2 #04 4A LSR A
 #0D4 #04 4A LSR A
 #0D6 #04 4A LSR A
 #0D8 #04 4A LSR A
 #0DA #04 4A LSR A
 #0DC #04 4A LSR A
 #0DE #04 4A LSR A
 #0E0 #04 4A LSR A
 #0E2 #04 4A LSR A
 #0E4 #04 4A LSR A
 #0E6 #04 4A LSR A
 #0E8 #04 4A LSR A
 #0EA #04 4A LSR A
 #0EC #04 4A LSR A
 #0EE #04 4A LSR A
 #0F0 #04 4A LSR A
 #0F2 #04 4A LSR A
 #0F4 #04 4A LSR A
 #0F6 #04 4A LSR A
 #0F8 #04 4A LSR A
 #0FA #04 4A LSR A
 #0FC #04 4A LSR A
 #0FE #04 4A LSR A

* OP CODE TO A AGAIN.
 FORM INDEX INTO MNEMONIC TABL.
 * 1XXX1010 -> 00101XXX
 * XXXYYY01 -> 00111XXX
 * XXXYYY10 -> 00110XXX
 * XXXYYY100 -> 00100XXX
 * XXXXXX000 -> 000XXXXXX
 * SAVE MNEMONIC TABLE INDEX.
 PRINT INSTR (1 TO 3 BYTES)
 * IN A 12-CHARACTER FIELD.
 CHAR COUNT FOR MNEMONIC PRINT.
 * RECOVER MNEMONIC INDEX.
 FETCH 3-CHAR MNEMONIC.
 * (PACKED IN 2 BYTES)
 SHIFT 5 BITS OF CHAR INTO A.
 * (CLEARS CARRY)
 ADD '?', OFFSET.
 OUTPUT A CHARACTER OF MNEMONIC
 OUTPUT 3 BLANKS.
 COUNT FOR 6 PRINT FORMAT BITS.
 IF X=3 THEN PRINT ADDRESS VAL.
 NO PRINT IF LENGTH=0.
 HANDLE REL ADDRESSING MODE
 SPECIAL (PRINT TARGET ADDR)
 * (NOT DISPLACEMENT)
 OUTPUT 1- OR 2-BYTE ADDRESS.
 * MORE SIGNIFICANT BYTE FIRST
 TEST NEXT PRINT FORMAT BIT.
 IF 0, DON'T PRINT
 * CORRESPONDING CHARS.
 OUTPUT 1 OR 2 CHARS.
 * (IF CHAR FROM CHARG2 IS 0,
 * DON'T OUTPUT IT)
 * RETURN IF DONE 6 FORMAT BITS.
 PCL:H + DISPL + 1 TO A,Y.
 * +1 TO X,Y.
 PRINT TARGET ADDR OF BRANCH
 * AND RETURN

| | | | | | | | | | |
|------|----|----|----|--------|---------|------|----|-----|------|
| 08D0 | 4C | DC | FF | JMP | PRBYTE | 052F | 22 | DFB | \$22 |
| 08D3 | A9 | 2D | FF | LDA | #80 | 0530 | 44 | DFB | \$44 |
| 08D5 | 20 | EF | FF | JSR | CHAROUT | 0531 | 33 | DFB | \$33 |
| 08D8 | A5 | 45 | | LDA | PCH | 0532 | 00 | DFB | \$00 |
| 08DA | A6 | 44 | | LDA | PCL | 0533 | 08 | DFB | \$8 |
| 08DC | 20 | CC | 08 | JSR | PRINTX | 0534 | 40 | DFB | \$40 |
| 08DF | A9 | AD | | LDA | #AD | 0535 | 09 | DFB | \$9 |
| 08E1 | 20 | EF | FF | JSR | CHAROUT | 0536 | 10 | DFB | \$10 |
| 08E4 | A2 | 03 | | PRBLNK | LDA | 0537 | 22 | DFB | \$22 |
| 08E6 | A9 | A0 | | PRBL2 | LDA | 0538 | 44 | DFB | \$44 |
| 08E8 | 20 | EF | FF | JSR | CHAROUT | 0539 | 33 | DFB | \$33 |
| 08EB | CA | | | DEX | #A0 | 053A | 00 | DFB | \$00 |
| 08EC | D0 | F8 | | BNE | PRBL2 | 053B | 08 | DFB | \$8 |
| 08EE | 60 | | | RTS | LENGTH | 053C | 40 | DFB | \$40 |
| 08EF | A5 | 41 | | PCADJ | LDA | 053D | 09 | DFB | \$9 |
| 08F1 | 38 | | | PCADJ2 | SEC | 053E | 62 | DFB | \$62 |
| 08F2 | A4 | 45 | | PCADJ3 | LDA | 053F | 13 | DFB | \$13 |
| 08F4 | AA | | | BPL | PCADJ4 | 0540 | 78 | DFB | \$78 |
| 08F5 | 10 | 01 | | DEY | #A9 | 0541 | A9 | DFB | \$A9 |
| 08F7 | 88 | | | DEY | #A0 | 0542 | 00 | DFB | \$00 |
| 08F8 | 65 | 44 | | PCADJ4 | ADC | 0543 | 21 | DFB | \$21 |
| 08FA | 90 | 01 | | INY | BCC | 0544 | 81 | DFB | \$81 |
| 08FC | C8 | | | RTS1 | RTS1 | 0545 | 82 | DFB | \$82 |
| 08FE | 40 | | | RTS1 | RTS1 | 0546 | 00 | DFB | \$00 |
| 08FF | 02 | | | MODE | \$40 | 0547 | 00 | DFB | \$00 |
| 0900 | 45 | | | MODE | \$45 | 0548 | 59 | DFB | \$59 |
| 0901 | 03 | | | MODE | \$3 | 0549 | 40 | DFB | \$40 |
| 0902 | 00 | | | MODE | \$00 | 054A | 91 | DFB | \$91 |
| 0903 | 08 | | | MODE | \$8 | 054B | 92 | DFB | \$92 |
| 0904 | 40 | | | MODE | \$40 | 054C | 86 | DFB | \$86 |
| 0905 | 09 | | | MODE | \$9 | 054D | 4A | DFB | \$4A |
| 0906 | 30 | | | MODE | \$30 | 054E | 85 | DFB | \$85 |
| 0907 | 22 | | | MODE | \$22 | 054F | 9D | DFB | \$9D |
| 0908 | 45 | | | MODE | \$45 | 0550 | AC | DFB | \$AC |
| 0909 | 33 | | | MODE | \$33 | 0551 | A9 | DFB | \$A9 |
| 090A | 00 | | | MODE | \$00 | 0552 | AC | DFB | \$AC |
| 090B | 08 | | | MODE | \$8 | 0553 | 03 | DFB | \$03 |
| 090C | 40 | | | MODE | \$40 | 0554 | A8 | DFB | \$A8 |
| 090D | 09 | | | MODE | \$9 | 0555 | A4 | DFB | \$A4 |
| 090E | 40 | | | MODE | \$40 | 0556 | D9 | DFB | \$D9 |
| 090F | 02 | | | MODE | \$2 | 0557 | 09 | DFB | \$09 |
| 0910 | 45 | | | MODE | \$45 | 0558 | D8 | DFB | \$D8 |
| 0911 | 33 | | | MODE | \$33 | 0559 | A4 | DFB | \$A4 |
| 0912 | 00 | | | MODE | \$00 | 055A | A4 | DFB | \$A4 |
| 0913 | 08 | | | MODE | \$8 | 055B | 00 | DFB | \$00 |
| 0914 | 40 | | | MODE | \$40 | 055C | 1C | DFB | \$1C |
| 0915 | 09 | | | MODE | \$9 | 055D | 8A | DFB | \$8A |
| 0916 | 40 | | | MODE | \$40 | 055E | 1C | DFB | \$1C |
| 0917 | 00 | | | MODE | \$00 | 055F | 23 | DFB | \$23 |
| 0918 | 40 | | | MODE | \$40 | 0560 | 5D | DFB | \$5D |
| 0919 | 60 | | | MODE | \$60 | 0561 | 06 | DFB | \$06 |
| 091A | 00 | | | MODE | \$00 | 0562 | 1B | DFB | \$1B |
| 091B | 00 | | | MODE | \$00 | 0563 | A1 | DFB | \$A1 |
| 091C | 40 | | | MODE | \$40 | 0564 | 9D | DFB | \$9D |
| 091D | 00 | | | MODE | \$00 | 0565 | 8A | DFB | \$8A |
| 091E | 00 | | | MODE | \$00 | 0566 | 1D | DFB | \$1D |
| 091F | 22 | | | MODE | \$22 | 0567 | 23 | DFB | \$23 |
| 0920 | 44 | | | MODE | \$44 | 0568 | 9D | DFB | \$9D |
| 0921 | 33 | | | MODE | \$33 | 0569 | 8B | DFB | \$8B |
| 0922 | 00 | | | MODE | \$00 | 056A | 1D | DFB | \$1D |
| 0923 | 8C | | | MODE | \$8C | 056B | A1 | DFB | \$A1 |
| 0924 | 44 | | | MODE | \$44 | 056C | 00 | DFB | \$00 |
| 0925 | 00 | | | MODE | \$00 | 056D | 29 | DFB | \$29 |
| 0926 | 11 | | | MODE | \$11 | 056E | 19 | DFB | \$19 |
| 0927 | 22 | | | MODE | \$22 | 056F | AE | DFB | \$AE |
| 0928 | 44 | | | MODE | \$44 | 0570 | 69 | DFB | \$69 |
| 0929 | 33 | | | MODE | \$33 | 0571 | A8 | DFB | \$A8 |
| 092A | 00 | | | MODE | \$00 | 0572 | 19 | DFB | \$19 |
| 092B | 8C | | | MODE | \$8C | 0573 | 26 | DFB | \$26 |
| 092C | 44 | | | MODE | \$44 | 0574 | 24 | DFB | \$24 |
| 092D | 9A | | | MODE | \$9A | 0575 | 53 | DFB | \$53 |
| 092E | 10 | | | MODE | \$10 | 0576 | 1B | DFB | \$1B |
| | | | | MODE | \$00 | 0577 | 23 | DFB | \$23 |

| Address | Content | | |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|-----|------|
| 0978 24 | DFB | \$24 | 09C2 74 | DFB | \$74 | 0978 24 | DFB | \$24 | 09C2 74 | DFB | \$74 |
| 0979 53 | DFB | \$53 | 09C3 72 | DFB | \$72 | 0979 53 | DFB | \$53 | 09C3 72 | DFB | \$72 |
| 097A 19 | DFB | \$19 | 09C4 44 | DFB | \$44 | 097A 19 | DFB | \$19 | 09C4 44 | DFB | \$44 |
| 097B A1 | DFB | \$A1 | 09C5 68 | DFB | \$68 | 097B A1 | DFB | \$A1 | 09C5 68 | DFB | \$68 |
| 097C 00 | DFB | \$00 | 09C6 B2 | DFB | \$B2 | 097C 00 | DFB | \$00 | 09C6 B2 | DFB | \$B2 |
| 097D 1A | DFB | \$1A | 09C7 32 | DFB | \$32 | 097D 1A | DFB | \$1A | 09C7 32 | DFB | \$32 |
| 097E 5B | DFB | \$5B | 09C8 02 | DFB | \$02 | 097E 5B | DFB | \$5B | 09C8 02 | DFB | \$02 |
| 097F 5B | DFB | \$5B | 09C9 00 | DFB | \$00 | 097F 5B | DFB | \$5B | 09C9 00 | DFB | \$00 |
| 0980 A5 | DFB | \$A5 | 09CA 22 | DFB | \$22 | 0980 A5 | DFB | \$A5 | 09CA 22 | DFB | \$22 |
| 0981 69 | DFB | \$69 | 09CB 00 | DFB | \$00 | 0981 69 | DFB | \$69 | 09CB 00 | DFB | \$00 |
| 0982 24 | DFB | \$24 | 09CC 1A | DFB | \$1A | 0982 24 | DFB | \$24 | 09CC 1A | DFB | \$1A |
| 0983 24 | DFB | \$24 | 09CD 1A | DFB | \$1A | 0983 24 | DFB | \$24 | 09CD 1A | DFB | \$1A |
| 0984 HE | DFB | \$HE | 09CE 26 | DFB | \$26 | 0984 HE | DFB | \$HE | 09CE 26 | DFB | \$26 |
| 0985 A8 | DFB | \$A8 | 09CF 00 | DFB | \$00 | 0985 A8 | DFB | \$A8 | 09CF 00 | DFB | \$00 |
| 0987 AD | DFB | \$AD | 09D0 72 | DFB | \$72 | 0987 AD | DFB | \$AD | 09D0 72 | DFB | \$72 |
| 0988 29 | DFB | \$29 | 09D1 72 | DFB | \$72 | 0988 29 | DFB | \$29 | 09D1 72 | DFB | \$72 |
| 0989 00 | DFB | \$00 | 09D2 88 | DFB | \$88 | 0989 00 | DFB | \$00 | 09D2 88 | DFB | \$88 |
| 098A 7C | DFB | \$7C | 09D3 08 | DFB | \$08 | 098A 7C | DFB | \$7C | 09D3 08 | DFB | \$08 |
| 098B 00 | DFB | \$00 | 09D4 04 | DFB | \$04 | 098B 00 | DFB | \$00 | 09D4 04 | DFB | \$04 |
| 098C 15 | DFB | \$15 | 09D5 0A | DFB | \$0A | 098C 15 | DFB | \$15 | 09D5 0A | DFB | \$0A |
| 098D 9C | DFB | \$9C | 09D6 26 | DFB | \$26 | 098D 9C | DFB | \$9C | 09D6 26 | DFB | \$26 |
| 098E 6D | DFB | \$6D | 09D7 48 | DFB | \$48 | 098E 6D | DFB | \$6D | 09D7 48 | DFB | \$48 |
| 098F 00 | DFB | \$00 | 09D8 44 | DFB | \$44 | 098F 00 | DFB | \$00 | 09D8 44 | DFB | \$44 |
| 0990 A5 | DFB | \$A5 | 09D9 44 | DFB | \$44 | 0990 A5 | DFB | \$A5 | 09D9 44 | DFB | \$44 |
| 0991 69 | DFB | \$69 | 09DA H2 | DFB | \$H2 | 0991 69 | DFB | \$69 | 09DA H2 | DFB | \$H2 |
| 0992 29 | DFB | \$29 | 09DB 08 | DFB | \$08 | 0992 29 | DFB | \$29 | 09DB 08 | DFB | \$08 |
| 0993 53 | DFB | \$53 | END | | 0993 53 | DFB | \$53 | END | | | |
| 0994 84 | DFB | \$84 | | | 0994 84 | DFB | \$84 | | | | |
| 0995 13 | DFB | \$13 | | | 0995 13 | DFB | \$13 | | | | |
| 0996 34 | DFB | \$34 | | | 0996 34 | DFB | \$34 | | | | |
| 0997 11 | DFB | \$11 | | | 0997 11 | DFB | \$11 | | | | |
| 0998 A5 | DFB | \$A5 | | | 0998 A5 | DFB | \$A5 | | | | |
| 0999 69 | DFB | \$69 | | | 0999 69 | DFB | \$69 | | | | |
| 099A 23 | DFB | \$23 | | | 099A 23 | DFB | \$23 | | | | |
| 099B 80 | DFB | \$80 | | | 099B 80 | DFB | \$80 | | | | |
| 099C D8 | DFB | \$D8 | | | 099C D8 | DFB | \$D8 | | | | |
| 099D 62 | DFB | \$62 | | | 099D 62 | DFB | \$62 | | | | |
| 099E 5A | DFB | \$5A | | | 099E 5A | DFB | \$5A | | | | |
| 099F 48 | DFB | \$48 | | | 099F 48 | DFB | \$48 | | | | |
| 09A0 26 | DFB | \$26 | | | 09A0 26 | DFB | \$26 | | | | |
| 09A1 62 | DFB | \$62 | | | 09A1 62 | DFB | \$62 | | | | |
| 09A2 94 | DFB | \$94 | | | 09A2 94 | DFB | \$94 | | | | |
| 09A3 88 | DFB | \$88 | | | 09A3 88 | DFB | \$88 | | | | |
| 09A4 54 | DFB | \$54 | | | 09A4 54 | DFB | \$54 | | | | |
| 09A5 44 | DFB | \$44 | | | 09A5 44 | DFB | \$44 | | | | |
| 09A6 C8 | DFB | \$C8 | | | 09A6 C8 | DFB | \$C8 | | | | |
| 09A7 54 | DFB | \$54 | | | 09A7 54 | DFB | \$54 | | | | |
| 09A8 68 | DFB | \$68 | | | 09A8 68 | DFB | \$68 | | | | |
| 09A9 44 | DFB | \$44 | | | 09A9 44 | DFB | \$44 | | | | |
| 09AA E8 | DFB | \$E8 | | | 09AA E8 | DFB | \$E8 | | | | |
| 09AB 94 | DFB | \$94 | | | 09AB 94 | DFB | \$94 | | | | |
| 09AC 00 | DFB | \$00 | | | 09AC 00 | DFB | \$00 | | | | |
| 09AD B4 | DFB | \$B4 | | | 09AD B4 | DFB | \$B4 | | | | |
| 09AE 08 | DFB | \$08 | | | 09AE 08 | DFB | \$08 | | | | |
| 09AF 84 | DFB | \$84 | | | 09AF 84 | DFB | \$84 | | | | |
| 09B0 74 | DFB | \$74 | | | 09B0 74 | DFB | \$74 | | | | |
| 09B1 B4 | DFB | \$B4 | | | 09B1 B4 | DFB | \$B4 | | | | |
| 09B2 28 | DFB | \$28 | | | 09B2 28 | DFB | \$28 | | | | |
| 09B3 5E | DFB | \$5E | | | 09B3 5E | DFB | \$5E | | | | |
| 09B4 74 | DFB | \$74 | | | 09B4 74 | DFB | \$74 | | | | |
| 09B5 F4 | DFB | \$F4 | | | 09B5 F4 | DFB | \$F4 | | | | |
| 09B6 CC | DFB | \$CC | | | 09B6 CC | DFB | \$CC | | | | |
| 09B7 4A | DFB | \$4A | | | 09B7 4A | DFB | \$4A | | | | |
| 09B8 72 | DFB | \$72 | | | 09B8 72 | DFB | \$72 | | | | |
| 09B9 F2 | DFB | \$F2 | | | 09B9 F2 | DFB | \$F2 | | | | |
| 09BA A4 | DFB | \$A4 | | | 09BA A4 | DFB | \$A4 | | | | |
| 09BB 8A | DFB | \$8A | | | 09BB 8A | DFB | \$8A | | | | |
| 09BC 00 | DFB | \$00 | | | 09BC 00 | DFB | \$00 | | | | |
| 09BD AA | DFB | \$AA | | | 09BD AA | DFB | \$AA | | | | |
| 09BE A2 | DFB | \$A2 | | | 09BE A2 | DFB | \$A2 | | | | |
| 09BF A2 | DFB | \$A2 | | | 09BF A2 | DFB | \$A2 | | | | |
| 09C0 74 | DFB | \$74 | | | 09C0 74 | DFB | \$74 | | | | |
| 09C1 74 | DFB | \$74 | | | 09C1 74 | DFB | \$74 | | | | |

A Number Game for the 6502

by Steve Wozniak

Apple Computers, 770 Welch Rd., Suite 154
Palo Alto CA 94304; (415) 326-4248

DESCRIPTION

MASTERMIND is a number guessing game. In this version the computer generates a 5-digit, random number where the digits are limited to 0 through 7. The user attempts to guess the number in the fewest possible tries. After each guess the computer informs the user as to how many digits were correct (contained in the random number) and in the correct position by printing '+' signs, and how many digits are correct and in the wrong position with '-' signs. The user is not informed as to exactly which digit positions the '+' and the '-' signs correspond. A skilled player can usually win in 6 or 7 tries.

THE PLAY

1. Computer prints 'READY?'
2. User hits any key when ready (not echoed). A random number is generated by counting from the 'READY?' message to key depression.
3. Computer prints '01' [↑] for move number
Cursor left here.
4. User enters his 5-digit guess following the move number. Remember that valid digits are 0 through 7 only. Entering any invalid character cancels the line and the computer repeats step 3 for the current move. This may be used to cancel errors.
5. Immediately after entering the 5th digit of his guess, the computer will print a number of '+' signs (for correct digits in correct positions) followed by a number of '-' signs (for correct digits in incorrect positions). This 'score' is indicated to the right of the guess and does not indicate the specific digit positions involved. Play resumes at 3 for the *next* move number except for a win. Examples follow:

Scoring Examples for Random Number 12154

| Move | Guess | Score |
|------|-------|---------------------------------------------------------------------|
| 01 | 33366 | (none correct) |
| 02 | 00018 | (line cancelled due to invalid digit, 8) |
| 02 | 00011 | -- (2 digits right, in wrong positions) |
| 03 | 11234 | ++-- (4 digits right, 2 in correct positions, 2 in wrong positions) |
| 04 | 11325 | +--- (4 digits right, 1 in correct position) |
| 05 | 13216 | +-- (3 digits right, 1 in correct position) |
| 06 | 44444 | + (1 digit right in correct position) |
| 07 | 55555 | + (1 digit right, in correct position) |
| 08 | 12154 | +++++ YOU WIN (win message) |

READY? (for next game)

RUNNING ON APPLE-1 SYSTEMS

The source and object listings provided will run on APPLE-1 systems. The program loads in locations \$300 through \$3AE and uses the following page 0 locations for variables:

| | | |
|-----------|-------|-----------------------------------|
| \$F2 | TRIES | (no. of tries - 0 to 99 BCD) |
| \$F3 | RNDL | Binary |
| \$F4 | RNDH | random number |
| \$F5 | RND2L | (Temp.) |
| \$F6-\$FA | N | (5 digits of unpacked random no.) |
| \$FB-\$FF | GUESS | (User guess) |

RUNNING ON OTHER SYSTEMS

1. The LDA STROBE instruction at loc. \$313 (on supplied listing) senses for a key down condition after the READY? message. This is used to generate a random number corresponding to the delay before a key is depressed. The code is written for a negative value to indicate 'key down' (b₇ = 1) and a positive value (b₇ = 0) to indicate

no key down. This instruction is followed by a JSR CHARIN instruction whose only purpose is to clear the strobe. The character returned is not used. Thus the user must provide an address for the LDA STROBE instruction corresponding to his hardware and must insure that the CHARIN subroutine clears the strobe.

2. A CHARIN subroutine must be provided to read one ASCII character with b₇ set. Do not use the CHARIN subroutine provided as it uses APPLE-1 I/O assignments. The character read must be returned in the A-Reg. The Y-Reg may be altered by CHARIN but not the X-Reg. If b₇ is returned clear (=0) then the EOR #\$B0 instruction at loc. \$34E (on supplied code) must be changed to an EOR #\$30.

3. A COUT subroutine must be provided which outputs one ASCII character (passed in A-Reg). If the user output device requires line feeds with carriage returns then the COUT routine must intercept the carriage return character (\$8D) and output the necessary CR-LF combination. All computer generated text has b₇=1. No registers (A,X,Y) may be altered.

4. PRBYTE subroutine must be provided which outputs one byte (passed in A-Reg) in hexadecimal (printing 2 digits). No registers need be preserved. The following routine will do:

| | | |
|--------|------------|-------------------------------------------------------------------------------------|
| PRBYTE | PHA | Save for LSD |
| | LSR | |
| | LSR | |
| | LSR | MSD to LSD position |
| | LSR | |
| | JSR PRHEX* | Output MSD first |
| | PLA | Restore A |
| PRHEX | AND #\$F | Mask LSD |
| PRHEX* | ORA #\$B0 | Add "0" |
| | CMP #\$BA | { May be skipped if used for MASTERMIND only, since only BCD digits supplied. |
| | BCC TOCOUT | |
| | ADC #\$6 | Output ASCII and return |
| TOCOUT | JMP COUT | |

THE BOUNCING BEASTIE

A RANDOM WALKER FOR PROCESSOR TECH'S VDM

by Marvin R. Winzenread

If you have a Processor Technology VDM or any display that reads directly from memory, and at least 256 bytes of other memory, here is a random walk program you might like.



The program was assembled using P.T.'s software No. 1 on a system of only 8K, so the comments were added later. I hope you enjoy my turtle.

| | | |
|---------------|-----------------------|---------------------------------------------------|
| 0000 31 A1 00 | 0900 LXI 6,AR+6 | Set Stack Pointer |
| 0003 3E 00 | 1000 MVI A,0 | |
| 0005 D3 8C | 1010 OUT 8CH * | |
| 0007 21 00 88 | 1020 AG LXI H,8800H * | Initialize VDM |
| 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 in Center of Screen |
| 0016 36 07 | 2010 N2 MVI M,7 | Looks like a turtle to me |
| 0018 78 | 2020 N4 MOV A,B | |
| 0019 E6 03 | 2023 ANI 3 | |
| 001B CA 24 00 | 2025 JZ N5 | |
| 001E 01 FF 1F | 2027 LXI B,1FFFH | |
| 0021 C3 27 00 | 2030 JMP N3 | A wait routine to slow the action to human speeds |
| 0024 01 FF 0F | 2033 N5 LXI B,0FFFH | |
| 0027 0B | 2040 N3 DCX B | |
| 0028 78 | 2050 MOV A,B | |
| 0029 FE 07 | 2055 CPI A,0 | |
| 002B C2 27 00 | 2060 JNZ N3 | |
| 002E 36 2A | 2900 MVI M,2AH ← | Experiment with different fonts |
| 0030 CD 72 00 | 3000 CALL RND ← | Could be IN0FFH If you want to control the action |
| 0033 47 | 3010 MOV B,A | |
| 0034 E6 01 | 3020 ANI 1 | |
| 0036 C4 5E 00 | 3025 CNZ UP | |
| 0039 78 | 3030 MOV A,B | |
| 003A E6 02 | 3035 ANI 2 | |
| 003C C4 69 00 | 3036 CNZ DN | |
| 003F 78 | 3040 MOV A,B | MOVE |
| 0040 E6 40 | 3050 ANI 40H | |
| 0042 C4 4E 00 | 3055 CNZ RT | |
| 0045 78 | 3060 MOV A,B | |
| 0046 E6 80 | 3065 ANI 80H | |
| 0048 C4 56 00 | 3070 CNZ LF | |
| 004B C3 16 00 | 5000 JMP N2 | |
| 004E 23 | 8100 RT INX H | |
| 004F 7C | 8110 MOV A,H | |
| 0050 FE 8C | 8120 CPI 8CH * | Move right |
| 0052 C0 | 8130 RNZ | |
| 0053 26 88 | 8140 MVI H,88H * | |
| 0055 C9 | 8190 RET | |

| | | | | | |
|------|----------|------|-----------------|---|---------------------------------------|
| 0056 | 2B | 8200 | LF DCX H | | |
| 0057 | 7C | 8210 | MOV A,H | | |
| 0058 | FE 87 | 8220 | CPI 87H * | } | |
| 005A | C0 | 8230 | RNZ | | Move left |
| 005B | 26 8B | 8240 | MVI H,8BH * | | |
| 005D | C9 | 8290 | RET | | |
| 005E | 11 C0 FF | 8300 | UP LXI D,OFFCOH | } | |
| 0061 | 19 | 8310 | DAD D | | |
| 0062 | 7C | 8320 | MOV A,H | | |
| 0063 | E6 FB | 8330 | ANI OFBH * | | Up |
| 0065 | F6 08 | 8340 | ORI 08H | | |
| 0067 | 67 | 8350 | MOV H,A | | |
| 0068 | C9 | 8390 | RET | | |
| 0069 | 11 40 00 | 8400 | DN LXI D,40H | } | |
| 006C | 19 | 8410 | DAD D | | |
| 006D | 7C | 8420 | MOV A,H | | |
| 006E | E6 FB | 8430 | ANI OFBH * | | Down |
| 0070 | 67 | 8440 | MOV H,A | | |
| 0071 | C9 | 8490 | RET | | |
| 0072 | E5 | 8600 | RND PUSH H | } | |
| 0073 | 21 9A 00 | 8610 | LXI H,SH+3 | | A random number generator from P.C.C. |
| 0076 | 06 08 | 8620 | MVI B,8 | | |
| 0078 | 7E | 8630 | MOV A,M | | |
| 0079 | 07 | 8640 | RTOP RLC | | |
| 007A | 07 | 8641 | RLC | | |
| 007B | 07 | 8642 | RLC | | |
| 007C | AE | 8650 | XRA M | | |
| 007D | 17 | 8660 | RAL | | |
| 007E | 17 | 8661 | RAL | | |
| 007F | 2D | 8670 | DCR L | | |
| 0080 | 2D | 8671 | DCR L | | |
| 0081 | 2D | 8672 | DCR L | | |
| 0082 | 7E | 8680 | MOV A,M | | |
| 0083 | 17 | 8690 | RAL | | |
| 0084 | 77 | 8695 | MOV M,A | | |
| 0085 | 2C | 8700 | INR L | | |
| 0086 | 7E | 8710 | MOV A,M | | |
| 0087 | 17 | 8720 | RAL | | |
| 0088 | 77 | 8730 | MOV M,A | | |
| 0089 | 2C | 8740 | INR L | | |
| 008A | 7E | 8750 | MOV A,M | | |
| 008B | 17 | 8760 | RAL | | |
| 008C | 77 | 8770 | MOV M,A | | |
| 008D | 2C | 8780 | INR L | | |
| 008E | 7E | 8790 | MOV A,M | | |
| 008F | 17 | 8800 | RAL | | |
| 0090 | 77 | 8810 | MOV M,A | | |
| 0091 | 05 | 8820 | DCR B | | |
| 0092 | C2 79 00 | 8830 | JNZ RTOP | | |
| 0095 | E1 | 8840 | POP H | | |
| 0096 | C9 | 8850 | RET | | |
| 0097 | | 9000 | SH DS 4 | | |
| 009B | | 9010 | AR DS 6 | | |

Move left

Up

Down

A random number generator from P.C.C.

*My VDM is addressed as: 8C = Port address
8800 - 8BFF = Memory
If yours is addressed differently you need to
make changes in these statements.


```

000207 170      MOV A, B
                  Determine if
                  all chars output
000210 261      ORA C
000211 310      RZ
000212 013      DCX B
000213 176      MOV A, M
000214 376 015  CPI CR
                  Is it C.R.
000216 310      RZ
000217 345      PUSH H
000220 315 026 022 CALL ASCOUT
                  Output the char
000223 341      POP H
000224 043      INX H
000225 303 206 000 JMP @OUT1
                  Changes to implement @ in the IN statement:
032232 027 273  JMP P4
027273 232 222      P4   TST P2A "@"
027275 300
027276 300 230      @IN routine
027300 032 240      JMP back
032222 133 310      P2A  CALL AVTEST
032224 032 234      JMP back
000230 325      @IN  PUSH D
000231 076 077      MVI A, "?"
                  Output prompt
000233 315 026 022 CALL ASCOUT
000236 076 040      MVI A, " "
000240 315 026 022 CALL ASCOUT
000243 315 000 020 CALL BUFIN
                  Read keyboard
000246 021 111 020 LXI D, BUFSTRT
000251 052 000 024 LHD H slot for
                  variable @
000254 116      MOV C, M
                  Set up count from
000255 043      INX H   DIM of @
000256 106
000257 043      @INI  MOV B, M
                  Put chars keyed in
                  into @
000260 170      MOV A, B
000261 261      ORA C
000262 312 304 000 JZ @INR
                  End of @
000265 013      DCX B
000266 032      LDAX D
000267 167      MOV M, A
000270 043      INX H
000271 066 000      MVI M, O
000273 376 015      CPI CR
000275 312 304 000 JZ @INR
                  C.R. terminates
000300 023      INX D
000301 303 257 000 JMP @INI
000304 321      @INR POP D
000305 311      RET

```

A NEW DELIMITER FOR THE PRINT STATEMENT

I have often wished that TINY BASIC could output messages catenated together without the intervening spaces provided by the semi-colon delimiters. For example, I would like to output a number as dollars and cents (e.g., \$4.36). Or I would like to generate

BRINCH HANSEN TERMINATES DISTRIBUTION OF CONCURRENT PASCAL & SOLO MATERIALS

Per Brinch Hansen has left Cal Tech and moved to USC. With this move, he has ended the availability (through him) of reports and PDP-11 system tapes for the Concurrent PASCAL/SOLO project. Papers describing the system may be found in *IEEE Transactions on Software Engineering* (June, 1975) and in *Software-Practice & Experience* (April-June, 1976).

PROPOSED STANDARD FOR MINIMAL BASIC PROGRAMMING LANGUAGE

A draft of a standard for a minimal BASIC was published July 1st and is being circulated for a four-month period of public comment and review. It was prepared by the X3J2 Technical Committee (T.C. for BASIC) of the American National Standards Institute (ANSI). Copies may be requested from, and comments should be directed to CBEMA/Secretary X3, 1828 L Street, N.W., Washington, DC 20036.

bar graphs. Or I would like to use the character string capability to compose text.

In order to make these possible, I implemented a new delimiter (colon) in the PR statement of the Whipple/Arnold version of TINY BASIC. It works like the semi-colon delimiter, except that no space is output.

It is implemented as follows:

```

032173 226 377      TST P5 ':'
026377 232 202      P5   TST S7 ':'
027001 272
027002 032 166      JMP back

```

NULL LINES AND THE REM STATEMENT

If you just hit RETURN when the Whipple/Arnold version of TINY BASIC expects a command, it puts a line of garbage at the beginning of your program. This change prevents this from happening:

```

020366 303 066 022      JMP P1
022066 041 111 020      P1   LXI H, BUFSTRT
022071 170      MOV A, B
022072 261      ORA C
022073 312 166 021      JZ END
022076 303 371 020      JMP back

```

I implemented the REM (remark) statement as follows:

```

032074 226 370      TST P2 'IF'
026370 232 112      P2   TST S3A 'REM'
026372 122 105 315
026375 325 246      LOOP

```

Those who have video terminals may wish to change the size of the print zone from 15 characters to 8 or 16. This can be done by changing the following locations to the desired number:

022036, 022355, 022365, 026107.

**MORE MODS TO DOMPIER'S MUSIC PROGRAM,
A FIVE-INSTRUCTION MEMORY-ERASE ROUTINE,
& ANOTHER MEMORY-TEST PROGRAM**

Dear Editors of *DDJ*:

Aug. 15, 1976

I was indeed interested in seeing some modifications to Steve Dompier's MUSIC PROGRAM [*DDJ*, Vol.1, No.3]. However, there is still an inconvenience in the original program. This is the necessity to change the tempo data in memory for each new piece that is loaded. I submit to you my re-written code that allows the user to input the tempo to the program from the sense switches of his machine, allowing him to change and "fine tune" the speed at which the music is played. The new code is as follows:

| ADDR | DATA | MNEMONICS | COMMENTS |
|---------|-------------|-------------|----------------------------|
| 000-000 | 041 XXX XXX | LXI H, XXX | ;ENTER START OF PIECE |
| 000-003 | 176 | MOV A,M | ;READ NOTE |
| 000-004 | 376 377 | CPI 377 | ;TEST FOR END-OF-PIECE |
| 000-006 | 312 000 000 | JZ 000-000 | ;JUMP TO BEG- GING IF END |
| 000-011 | 333 377 | IN 377 | ;INPUT TEMPO FROM SENSE SW |
| 000-013 | 127 | MOV D,A | ;MOVE TEMPO TO D REG |
| 000-014 | 005 | DCR B | ;MAIN PLAY PROGRAM FRAME |
| 000-015 | 302 021 000 | JNZ 000-021 | |
| 000-020 | 106 | MOV B,M | |
| 000-021 | 015 | DCR C | |
| 000-022 | 302 014 000 | JNZ 000-014 | |
| 000-025 | 025 | DCR D | |
| 000-026 | 302 014 000 | JNZ 000-014 | |
| 000-031 | 043 | INX H | |
| 000-032 | 303 003 000 | JMP 000-003 | |

Note that tempo zero on the sense switches will be the fastest, and tempo 337 will be the next fastest, tempo 001 being the slowest. (This is due to the nature of Dompier's program.)

The following are two very short programs for utility and recreation. The first is a memory clear program that leaves zeros in memory, effectively erasing itself, with the exception of two instructions.

| ADDR | DATA | MNEMONICS | COMMENTS |
|---------|-------------|-------------|--------------------------------|
| 000-000 | 041 010 000 | LXI H,8 | ;SET POINTER TO END-OF PRGM.-1 |
| 000-003 | 257 | XRA A | ;SET ACC. TO ZERO |
| 000-004 | 043 | INX H | ;INCREMENT MEM. POINTER |
| 000-005 | 167 | MOV M,A | ;MOVE ACC. TO MEMORY LOCATION |
| 000-006 | 303 004 000 | JMP 000-004 | ;LOOP TO INX H INSTR. |

This program will effectively try to write zeros into all available memory locations, and erases itself, up to the INX H instruction, which then will become a NOP and the program will infinitely loop, writing zero into the same memory location. The lights on your computer will indicate when the program has stopped; it usually will be through in a second or two, depending on the access time of your devices. You then may zero the remaining memory

memory locations (005, 006 & 007 in this lower-page assembly) with DEPOSIT.

The following program was written by my colleague, Steve Schosow, the first day that we got my Altair working. It effectively lets you use the sense switches on your computer like a keyboard on a synthesizer. This works by playing the note on the sense switches out an AM radio, just as the Dompier program does.

| ADDR | DATA | MNEMONICS | COMMENTS |
|---------|-------------|-------------|--------------------------------|
| 000-000 | 333 377 | IN 377 | ;INPUT NOTE FROM SENSE SW |
| 000-002 | 075 | DCR A | ;LOOP TO PLAY NOTE |
| 000-003 | 302 002 000 | JNZ 000-002 | ;CONTINUE DECREASING 'TIL ZERO |
| 000-006 | 303 000 000 | JMP 000-000 | ;GET NEW NOTE TO PLAY |

This program makes it obvious that the simple DCR or INR loop is essential to playing the "note." We tried putting more complex looping structures in this program, but simplicity seems to work the best.

MEMORY TEST PROGRAM

| | | | |
|--------|-------------|----------|-----------------|
| | LXI H,START | 000-000: | 041 XXX XXX |
| | LXI B,END | 000-003: | 001 XXX XXX |
| | LXI SP,\$+4 | 000-006: | 061 050 000 |
| AGAIN: | XRA A | 000-011: | 257 |
| NEXT: | MOV M,A | 000-012: | 167 |
| | MOV D, M | 000-013: | 126 |
| | CMP D | 000-014: | 272 |
| | JNZ ERROR | 000-015: | 302 040 000 |
| | DCR A | 000-020: | 075 |
| | JNZ NEXT | 000-021: | 302 021 000 |
| | INX H | 000-024: | 043 |
| | MOV A,H | 000-025: | 174 |
| | CMP B | 000-026: | 270 |
| | JNZ AGAIN | 000-027: | 302 011 000 |
| | MOV A,L | 000-032: | 175 |
| | CMP C | 000-033: | 271 |
| | JNZ AGAIN | 000-034: | 302 011 000 |
| | HLT | 000-037: | 166 |
| ERROR: | MOV E,A | 000-040: | 137 |
| | PUSH H | 000-041: | 345 |
| | PUSH D | 000-042: | 325 |
| | HLT | 000-043: | 166 |
| | | 000-044: | 000 000 000 000 |

The starting address for the memory to be tested should be loaded in the two locations reserved by the label START at the first instruction. If you wish the first available location outside the operating program, this will be location 000-050 (split octal). The ending address to be checked, however, must be increased. E.g., if you were testing a 4K memory, END would be set to 020-000, instead of the last memory location in RAM, 017-377. When the program is done, it will halt, regardless of error. To verify if there was an error in testing memory, examine the four locations beginning at location 000-044, if no error occurred in reading memory, these will remain unchanged. If there has been a change in these locations, the data they contain is interpreted, thus:

Location 044 is the WRITE data to memory

Location 045 is the READ data from the same memory location

Location 046 is the memory location detected (bad) lo-order

Location 047 is the memory location detected (bad) hi-order

Some hints on interpreting this information:

If the READ data was 377 and the WRITE data

000, the problem is probably that there was no memory at that address.

If the WRITE data is zero and the read data NOT 377, the problem probably is that the memory slice was protected.

Elsewise, you may "trap" a bad memory slice by zeroing the last four locations (after making a note of them!) and doing the following changes:

-Change the START data to be your LAST memory location (like 017-377 in a 4K memory)

-Change the END data to be the beginning memory address to be tested (as in 000-050)

-Change location 000-024 to a DCX H (053).

... And run the program again. It will now stop, hopefully returning a new value in the ERROR locations. From these two addresses (the old address pointer you wrote down from locations 046 & 047, plus the new address pointer currently there) indicating between what two addresses (inclusive) that bad memory was found, you may have an indication that one of your IC's was bad (for instance, one bit would never go off: WRITE 000, READ 001 - it will usually be detected on WRITE 000.) and the memory pointers' difference will most likely be 1024. The memory slice and bit that is bad will indicate, with the help of a schematic, a bad memory chip.

Richard A. Kaapke 4485 Vision Dr., Apt. 9
 KLUGES, Inc. San Diego, CA 92121
 (All rights released to DDJ from Kaapke's Little Used but Greatly Esoteric Software, Incorporated)

REPRINTED ALPHANUMERIC MUSIC WITH AMPLITUDE CONTROL INCLUDES CORRECTIONS FROM AUTHOR

Malcom Wright has sent in three letters containing corrections and additions to his booklet on Altair computer music generation [see DDJ, Vol. 1, No. 5, for notes on this publication]. These letters (dated April 13, May 17, and June 7) have been included in the most recent reprint of the booklet, available from the PCC Bookstore.

COMPUTING CAREERS FOR DEAF PEOPLE

Proceedings of the 1975 ACM Conference on Computing Careers for Deaf People have been published by the Association for Computing Machinery. The Conference, sponsored by the ACM Special Interest Group on Computers and the Physically Handicapped (SIGCAPH), was held last April in the Washington, D.C. area. It featured 30 presentations (including 11 by deaf professionals) covering such topics as educational opportunities, special training programs, placement problems and solutions, federal legislation, on-the-job problems and solutions, and success factors.

Industry and government employers will find that these proceedings provide sound input to their plans for compliance with the requirements of the Rehabilitation Act of 1973 (Public Law 93-112).

The proceedings contain 125 pages and cost \$6.75 for ACM Members and \$9.00 for non-members. They are available, prepaid, from:

ACM Order Department
 P. O. Box 12105
 Church Street Station
 New York, NY 10249

ERRATA TO ZEIGLER'S 6502 "BUG PROGRAM"

The March issue of *Dr. Dobb's Journal* [Vol. 1, No. 3] contained a "Breakpoint Routine for 6502's" submitted by John Zeigler. The final paragraph of the documentary text contains an error. It *should* read:

"... it is necessary to load the interrupt vector, FFFE and FFFF, with 54 amd 02, respectively ..."

PROPOSAL FOR HANDY SOFTWARE, WITH EXAMPLE

A STRING OUTPUT SUBROUTINE FOR THE 6502

Dear Jim:

August 10, 1976

I have been noticing that in the *Journal*, the main subject has been large programs (BASIC's, monitors, text editors). I agree there is a need for large programs such as these, but I believe you should also concentrate on HANDY (Helpful Algorithms for Novice Do-It-Yourselfers) programs to save bytes in space-limited systems. I enclose my example: a string immediate-output subroutine for 6502-based systems. This routine saves pointers, loops, etc. normally used for string output by sequentially outputting the ASCII characters represented in hex in the bytes immediately following the Jump to Subroutine. After reading a terminating character (null), it returns to the instruction following the end of the string. No string addresses or lengths are needed.

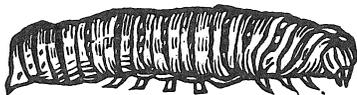
The subroutine uses 40 (hex) contiguous bytes for program and intermediate storage and 2 zero-page bytes for indirect addressing. Calling the routine affects none of the registers, nor the stack. It has been implimented on an Apple Computer and copies of the program are being delivered to the Homebrew library and the CCC repository.

Chris Espinosa

```

LOC.  OBJECT CODE  SOURCE STATEMENT
                                           ;;; STRINGOUT: HANDY STRING OUTPUT ROUTINE
                                           ;;; DEVELOPED FOR THE MOS 6502 BY C. ESPINOSA
                                           ;;; PUBLIC DOMAIN 9/11/1976
                                           ;
                                           ORG $400
AKEEP EQU $043D
YKEEP EQU $043E
KKEEP EQU $043F
OUT EQU $EFFF
LO EQU $FE
HI EQU $FF
LO DPZ
HI DPZ
0400 8D 3D 04  BEGIN STA AKEEP
0403 8C 3E 04          STY YKEEP      SAVE REGISTERS
0406 68              PLA           GET RETURN ADDRESS
0407 85 FF          STA LO
0409 68              PLA
040A 85 FF          STA HI
040C A0 01          LDY #1       SET UP INDEX
040E B1 FE          LDA (LO),Y   GET NEXT CHAR
0410 F0 07          BEQ EXIT    END IF 00
0412 C3              INY
0413 20 FF EF      JSR OUT     OUTPUT IT
0416 4C 0E 04      JMP NEXT
0419 3C 3F 04      EXIT STY KKEEP
041C A5 FE          LDA LO
041E 38              SEC
041F 6D 3F 04      ADC KKEEP   ADD STR. LEN.
0422 85 FE          STA LO     TO RETURN ADDRESS
0424 A5 FF          LDA HI
0426 69 00          ADC #00    CARRY
0428 85 FF          STA HI
042A AD 3D 04      LDA AKEEP  RESTORE REGS
042D AC 3E 04      LDY YKEEP
0430 5C FE 00      JMP (LO)   RETURN

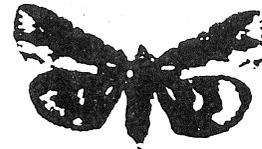
```



BUGS



FIXES



A "FIX" FOR DENVER TINY BASIC, PRAISE & A BUG-NOTE ON P.T.'S 5K BASIC

Dear Jim, July 30, 1976

I just finished looking through Vol.1, No.6, thought it was great, as usual, and decided to send along a few comments.

The following mod should be made to get dimensioned variables to work properly in my Denver version of Tiny BASIC [DDJ, Vol. 1, No. 3]:

| | | | | |
|------|-----------|-------------|------|-----------|
| TSV6 | DCX D | | TSV6 | DCX D |
| | XCHG | | | XCHG |
| | DAD H | Change to → | | DAD H |
| | DAD D | | | LDAX D |
| | MOV E, M | | | ADD L |
| | INX H | | | MOV L, A |
| | MOV D, M | | | INX D |
| | XCHG | | | LDAX D |
| | CALL ASPH | | | ADC H |
| | POP PSW | | | MOV H, A |
| | RET | | | CALL ASPH |
| | | | | POP PSW |
| | | | | RET |

A modification to the text editor which would be useful would be a "Test for valid data". I occasionally get control characters into a file, or lower case ASCII, which prints as upper case on my TTY, and don't detect it until I run a program through the assembler, which generates an assembly error. A test of this type, either as a separate command, or in the input routine, or both, would prevent these types of errors. Since I modified my TV-Typewriter (it will scroll now), I will probably change the editor to eliminate the clear screen routine, at which time I plan to add the above mod also. It shouldn't be difficult. The test is used in the INSM command, and all that is necessary is to pull that portion out as a separate command.

I finally received Processor Technology's listing of their 5K BASIC, and thought that I would pass along some comments on that, also. First, for \$5.00, I think it's a real bargain. (So is their assembler/editor/etc. package). I have it running now and it appears to work fine. I did find one apparent bug in the listing, in the "INT" function, subroutine AINT. This routine sets the floating point exponent to zero if the absolute value of the variable is less than 1. This is correct, but apparently at least some of the other routines require the entire 5 byte number to be zeroed for a zero value. I haven't traced this completely, but I was getting some strange results. I changed the rou-

tine to zero the entire value, and the problems disappeared.

The only other problem I encountered concerns a feature of the assembler used to generate the listing. There are numerous 5 and 6 letter symbols used, so rather than reassigning new symbols for my 4 character max limit, I modified my assembler (again) to accept longer symbols. I now can have symbols up to 11 characters long. I store only the actual number used in order to keep the memory requirements down, and plugged in 2K more of memory to be able to handle decent size source code blocks. Anyway - I used the symbols directly from the listing, and on the first pass through the assembler discovered that even though 6 character symbols are used, only the first 5 are unique and recognized. In several places the sixth letter had been dropped in the listing, which my newly modified assembler assumed were two different symbols. Numerous "Undefined symbol" errors occurred. I also had a duplicate symbol error, and sure enough, the symbol "PRIN2" is used two places in the listing. After straightening those problems out, along with numerous errors I made in entering the code, everything appears to work fine. A little more documentation would have been nice (I received only a listing, but there are enough comments to figure it out), and 5K BASIC requires over 6K, but I still would recommend it.

I still think I will implement my own high level language. I have the floating point routines running now, using binary arithmetic at 4 bytes per value, rather than decimal at 5 bytes per value, and have begun investigating different techniques for implementation of a new language. I'll send along a listing of the floating point routines as soon as I can get a good copy off. I have the listing on cassette, but haven't had time to run it through the printer yet.

Happy computing,
F. J. Greeb

1915 So. Cape Way
Denver, CO 80227

EDITOR'S NOTE: Processor Technology is located at 6200 Hollis St., Emeryville, CA 94608, (415) 652-8080.

How secure is computer data? Not very, say Stanford experts

(reprinted with permission from May, 1976, *The Stanford Observer*)

The proposed key for a security system designed to lock up the vast quantities of information stored in U.S. computers so that only authorized persons can get at them is "too small" and "within 15 years will be rendered totally insecure" in the opinion of two Stanford computer experts.

Electrical Engineering Prof. Martin E. Hellman and graduate student Whitfield Diffie criticize the computer key, proposed by the National Bureau of Standards (NBS) and the National Security Agency (NSA), in an article in *Communications of the ACM* (March 1976), professional journal of the Association for Computing Machinery.

The two computer experts have been working on related research in data security under National Science Foundation sponsorship.

If the Secretary of Commerce approves it, the key will become a required standard for most federal agencies and a *de facto* standard for all computer users.

Numerous computer thefts of company secrets and even of money in recent years have caused the industry to look hard for better methods of securing their trade secrets and other private information stored in computers. At the same time there has been wide public concern about the privacy of individual's computerized files held by the Internal Revenue Service and other government agencies.

So far the only way to tell whether computerized secrets are safely stored is to use trial-and-error methods to find the code words or signals that will enable one user to read another's files. Such eavesdropping already occurs now and then quite by accident.

The system proposed by NBS originated at IBM and consists of a kind of combination lock called an "algorithm," which is an encoding computer program. When the user

NOW LET'S SEE
IS IT 10101?
OR IS IT
11100?...
BETTER BRING
THE DYNAMITE
JUST IN CASE.



puts his key into the algorithm lock and feeds information into the machine, it is stored or sent to the recipient in coded form.

The same key must be used to get the information out again, or for the recipient to get the sender's message. Any other key will produce only garbled versions.

If approved the algorithm lock will become a standard item of integrated circuitry, mounted on a tiny silicon chip and manufactured in quantity for installation in any computer whose users want it.

The key to the lock, on the other hand, is a series of 56 digits, zeros and ones, arranged in any order. The large number of possible arrangements of these digits—approximately 100 million billion—gives the appearance of a high level of security.

But Hellman and Diffie hold that the so-called "56-bit" key is not all that secure, that such a key could be broken in a day by anyone with enough money to build the trail-and-error machinery to search the 100 million billion keys.

The machine would cost about \$20 million by their estimate, averaging out to \$10,000 per key

broken. Such a prodigious search has only become economically feasible within the last few years due to revolutionary advances in electronics.

They believe it significant that this would probably be too costly for a private firm to invest in prying into a rival firm's secrets, but not too much for a government agency—say the NSA or CIA. The implication is that by setting a 56-bit standard the government wants to be able to pry in the name of national security or other pretext, but prevent others from doing so.

"While it is well established that a larger number of keys does not guarantee security," says the Stanford pair, "too small a number of keys guarantees insecurity.

"It is widely held that computation costs will continue to decline by an order of magnitude every five years. In 10 years our \$20 million machine will then cost approximately \$200,000 and be within the reach of any large organization. If adopted, the proposed standard would have to be changed before that time.

"The cost of modifying termi-

nals, systems, etc. to use the new standard, and the cost of [reencoding] old files which are still deemed private would be high. These costs seem unjustified since the cost of enlarging the key of the proposed standard would be minimal."

Enlarging the 56-bit key to 64 bits would escalate the cost of key-breaking machinery to about \$5 billion, or \$2.5 million per key broken, they estimate. This cost might give pause to governments today, but in 10 years the expected advances in technology would reduce these figures to \$50 million and \$25,000 respectively.

The Stanford experts have urged a much higher standard with at least 128 key bits, which the computer users could employ wholly or in part. It would then be impossible to break the key by trial and error.

"We strongly encourage NBS to consider modifying its proposed standard," they conclude. "At the least it should define the level of security it believes the proposed standard offers and make public and reasoning behind its belief. Adoption of a standard with built-in obsolescence is not justified."

REVIEW OF THREE PERIODICALS CONCERNING ENERGY, TECHNOLOGY, AND PEOPLE

by "The Old Soldier"

Intermediate Technology
556 Santa Cruz Ave.
Menlo Park, CA 94025

Self Reliance
1717 - 18th St., N.W.
Washington, D.C. 20009

People & Energy
1757 S Street, N.W.
Washington, D.C. 20009

The three titles above are from newsletters that recently came into P.C.C. All are concerned with energy and technology, and each has something to recommend it, but since the combined subscription prices come to \$23.50 a year, some discrimination becomes necessary.

People & Energy is the most political of the three and a strong advocate for the corporate conspiracy theory of explaining our energy problems. For those who feel that way, this will help reinforce that attitude. No article on the suppression of the 100 MPG carburetor, but perhaps next issue . . .

Intermediate Technology is trying to put together working groups to examine production of energy, goods and foodstuffs to seek more rational production units in this time of rising transportation and energy costs. Looks like a good way to pick up multiple

RAIDS ON ALLEGED ELECTRONICS COMPONENTS THIEVES

Marshals raided Valcomp, Inc. in Westlake Village, CA, Pacific Semiconductor, Inc., in Inglewood, CA, and Idea Total Printing Services in Torrance, in the first week of July. At least part of the action concerned \$8.7 million worth of Xerox Corp. computer modules, stolen in El Segundo. The raids were based on information furnished by a private investigator hired by National, Intel, Motorola and Fairchild; all being companies quite concerned about recent major thefts of components.

ideas without a lot of rhetoric.

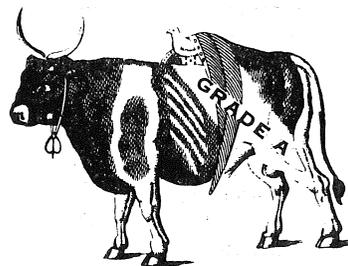
Self-Reliance, issue No. 1, got to me. The article "Do We Need Large Companies?" advances the thesis that, beyond a certain size, corporate growth does not enhance efficiency. Those advantages that do accrue are, greater leverage for equal percentage advertising expenditure, better and less expensive access to credit, and the possibly mistaken impression that large company employment is more stable. Another article, "Towards Sewerless Cities" brought an 'amen' from me. The majority of the water we use in the home is used just to carry away sewage. Los Angeles, from just one plant, The Hyperion Sewerage Works, discharges between 350 and 750 million gallons per day of sewage containing water into the ocean. Modern technology, and some technology not so novel, offers viable alternatives to this waste, and some of the alternatives are explained here. *Self-Reliance* has a fan in

—the Old Soldier—

A reference Journal for home computer users
from the People's Computer Company—

DR. DOBB'S JOURNAL of COMPUTER CALISTHENICS & ORTHODONTIA

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



Content regularly includes:

Complete documentation on systems software

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

Design notes for build-your-own software

Detailed 'blue skying' about practical systems projects for the immediate future

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

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

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

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

Active consumer advocacy for home computer users

- Supported by magazine sales—not by ads
- No vested interest in good will of manufacturers

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

- \$1.50 for a single copy: Vol. ___ No. ___
- \$10. per year (10 issues/year), to begin with Vol. ___ No. ___ This is a renewal.
- for foreign subscriptions
- add \$4. per year for surface mail
- add \$12. per year for air mail

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

Name _____

Mailing Address _____

City _____ State _____ Zip Code _____

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

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

COMPANY

THESE ARE NEW STORES WE'VE
LEARNED ABOUT SINCE OUR

NEWSLETTER

Byte Shop No.4
1225 Ocean St
Santa Cruz, CA 95060

The Computer Mart
625 W. Katella N
Orange CA 92667
(714) 633-1222

The Computer Store
3120 San Mateo
Albuquerque N.M.
The Computer Store
55 West 39th
New York N.Y.

The Computer Store
310 E. Wash
Ann Arbor

The Computer Store
63 So. Main
Ann Arbor

THESE ARE NEW
CLUBS WE'VE LEARNED ABOUT
Amateur Computer Society
c/o Walter Marvin
408 Thurber Dr. W
Columbus OH 43201

Ann Arbor Comp
C/O Roger Gregg
1485 Newport F
Ann Arbor MI 48106

Chicago User Group
195 Ivy Ln
Highland Park IL 60030

Computer Club of America
University Laboratory
1362 E. 57th
Chicago IL 60649

Computer Club of America
Contra Costa
Eric R. Berman
DACS Newsletter

ACGNJ News
Amateur Computer Group
UCT I
1776 Raritan Rd.
Scotch Plains NJ 07076

ACS Newsletter
Amateur Computer Society
260 Noroton Ave
Darien CT 06820

Cache Newsletter
Chicago Area Computer Hobby Enthusiasts
P.O. Box 36
Vernon Hills IL 60061

Cola Insight
Computer Organization of Los Angeles
P.O. Box 43677
Los Angeles CA 90043

Computer Hobbyist Group -
Computer Hobbyist Group
2377 Dalworth No. 157
Grand Prairie TX 75050

LETTERS

TIES - Total Information for Educational Systems is a cooperative effort among Independent School Districts in Minnesota. The enrollment of member districts is in excess of 260,000 students. TIES employs advanced systems concepts and computer technology to support the functions and needs of the district.

districts in Minnesota. TIES has been organized to offer special workshops on a regular basis. Contact TIES, 1925 West County Rd. B2, St. Paul, Mn. 55114.

A friend and I recently entered the micro-computer world with the purchase of an Altair. We also ordered a Riker-Maxson "terminal" from MiniMicro Mart in New York, along with their PIO board and TVT III board and ASCII encoder board.

The Altair hardware was good quality for the most part, but assembly instructions were fair to poor! Several parts were missing (but could easily be obtained from many local supply houses). The hardware failed to function on initial power up and required several hours to debug. Most parts were traced to errors in assembly.

Every so often we receive a letter at PCC which makes us blink. In that category is this one.

I read your piece on pg. 12 of Sept. '75 issue of PCC.

I didn't like it.

If you have to run a full page article to explain what it means, then it's not worth the effort.



BOOK REVIEW

MATH, WRITING & GAMES
Herbert R. Kohl, 1974, 252 p.
A careful analysis of the ways in which descriptions of countless games are written so they can comprehend probability theory. Herb Kohl loves kids. 1976. 294 pp; \$ 6.95.

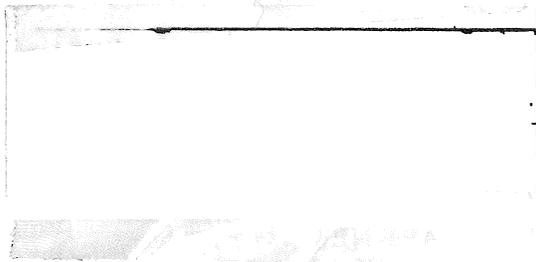
PROCESSOR/MICROPROGRAMMING
The blurb indicates this book is "an authoritative microprocessor construction, operation, programming, and maintenance manual for the Intel 8080 and 8085." For the most part, it is simply a reference manual found in specification manuals for the Intel 8080 and 8085. \$1.69.

COMPUTER LIKES ME WHEN I SPEAK IN BASIC
"Computer likes me when I speak in BASIC" is a humorous, yet informative, book by Ph.D. author, 1972, 64 p., \$2.00. It is a "do-it-yourself" workbook for the novice computer user. It includes numerous examples of BASIC programs and their results. \$2.00.

MOS Technology 6502

Our first reaction, upon opening the box, was to recoil from the sheer volume of paper. Aside from the things from EBKA, there were two massive manuals from MOS Technology. A closer inspection showed that these manuals were written with commendable simplicity and clarity. Documents of such quality are instructional as well as being references. This was despite the instructions regarding assembly and operation were somewhat scanty. The all sized layout sheet and correctly oriented components were correctly oriented. This was a kit maker's dream.

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



10% DISCOUNT
FOR DR. DOBB'S JOURNAL READERS
on Selected Items

KENTUCKY FRIED COMPUTERS
OPENS RETAIL STORE

The Kentucky Fried Computer store is now open for business. Hours are 11 a.m. to 6 p.m. daily. We are fully stocked, and we service what we sell—both kits and assembled units. We sell many products at a substantial discount from manufacturers' list prices. We also give special prices on complete system purchases. We can help you "put it all together."

Our product line includes:

IMSAI, full line*
Processor Technology, full line
CroMemCo, full line
Polymorphic, full line*
Lear Seigler: ADM terminal kit
Morrow's Micro-Stuff: Smart Cassette interface*
Oliver Engineering: Optical paper tape reader*
Keyboards, monitors, accessories, and publications

*STARRED ITEMS ARE AVAILABLE BY MAIL TO DR. DOBB'S JOURNAL READERS AT 10% OFF MANUFACTURER'S LIST PRICE. For mail orders, send money order or cashier's check (personal checks require 3 weeks' processing) plus 2% for shipping and handling. Californians add sales tax. This offer is for a limited time only.

Kentucky Fried Computers
2465 Fourth Street, Berkeley CA 94710
(415) 549-0868

A COMPUTER IN EVERY POT.