

## MITS Introduces New 6800-Based Computer

# ALTAIR 680

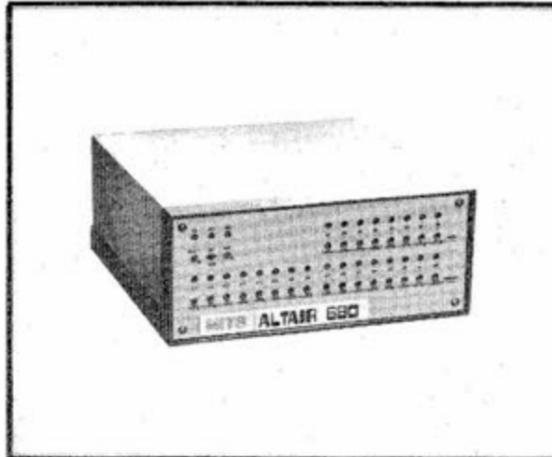
By James B. Vice

Not long ago MITS made a giant step towards the day when a computer will be a standard household item. They've now moved another step closer with the new ALTAIR 680 microcomputer. This new machine is an excellent compromise between computer power and low cost structure, without sacrificing design reliability. The MITS engineers have designed a system based on the 6800 microprocessing unit (MPU). This MPU, available from both Motorola and American Micro-Systems, adapts nicely to a minimum design configuration.

The ALTAIR 680 measures 11 1/16" wide x 11 1/16" deep x 4 11/16" high, making it less than one-third the size of the ALTAIR 8800. The basic system is available in three configurations, depending on the intended application. These include a fully user-programmable system with complete front panel controls, and two versions oriented more towards dedicated program applications.

The compact size of the 680 obviously precludes any significant amount of internal expansion, although additional memory and I/O control are already on the drawing board. The 680's small size can be deceiving. The overall concept was to keep the machine as simple, small and inexpensive as possible; but it is a complete system in itself, with the exception of some I/O device. To the industrial user this may not be important, but for the "home brewers" it can be critical.

The construction of this machine is a relatively easy matter for even the most inexperienced kit builder. Almost all of the circuitry is contained on a single large printed circuit board, including memory and a built-in I/O port. In fact, this single board is a full computer with the exception of a power transformer and some control switches. This is where the main distinction between the three configurations is encountered.



Most hobbyists will be concerned with the full front panel model. This contains all of the necessary controls for addressing and entering data besides those for controlling the processor itself. A "turn-key" front panel model is also available which eliminates all controls except restarting the processor. This could be used in any application where it is desirable to eliminate the possibility of the operator or any other person affecting the machine's memory or computing cycle. An example for such an application might be its use in controlling an intrusion detection system, or for a manufacturing machine control system. The third configuration is closely related to the last. The 680 will also be available as just the large PC board mentioned above. This board contains every-

**"An excellent compromise between computer power and low cost structure, without sacrificing design reliability."**

thing but a power supply and controls. It could be used for the same applications as the "turn-key" model, only where the computer is to be "buried" inside another machine.

The latter model would also be excellent for the experimenter who wished to purchase an absolute minimum and do a bit of his own designing. Another advantage to this particular idea is the considerable amount of information available on the 6800 microprocessing unit from Motorola

Semi-conductor Products, Inc. Also, the 6800 MPU is TTL compatible and requires only one 5-volt power supply.

In the full front panel model of the 680 there is an additional printed circuit board. This board contains all of the logic circuitry necessary to reset, halt or start the processor. Also located on this board are switches and associated LED indicator lights for each of the sixteen address lines and eight data lines. The front panel printed circuit board mounts directly to the main printed circuit board via a 100-contact edge connector. This eliminates the need for a cumbersome wiring harness. The only other control is the power switch, located on the back panel of the unit for safety purposes.

On the dedicated program models PROM or ROM is used for setting the starting address and for programming.

The basic ALTAIR 680 computer can be subdivided into five functional sections. These are the MPU and clock, the memory, an I/O port, control and indication, and the power supply.

The first three of these sections, along with the power supply regulation components, are located on the main printed circuit board.

#### MPU and Clock

At the heart of the 680 system is the 6800 Microprocessing Unit. This is a versatile and very powerful little processor, yet it is directly responsible for the overall simplicity of the 680 design.

The 6800 is an 8-bit parallel processor using a bi-directional data bus and a 16-bit address bus. The latter gives it the ability to directly address 65K bytes of memory. The instruction set consists of 72 basic instructions with variations giving 197 different instructions out of a possible 256 codes (59 of the possible codes are unassigned). The variations are the result of the 6800's unique addressing structure.

- CONTINUED Page 7 -

**BULLETIN - MITS drops**

**4K Price to \$195!**  
**see page 5**

## COMPUTER NOTES

October, 1975

© MITS, INC. 1975

Volume One Issue Five

A PUBLICATION OF THE ALTAIR USERS GROUP

## Across the Editor's Desk

by David Bunnell

As I have stated in the past, things change rapidly in the computer business. This issue of Computer Notes is an example.

Our lead article, "ALTAIR 680", was to have been the big news item, but as it turned out, the drop in the price of 4K boards and the announcement of new I/O cards is probably of more interest to Altair 8800 users.

I want to thank all of you who were kind enough to return your surveys--they have had a big impact already. For the benefit of those who think we don't listen to our customers, here's a list of changes resulting from the survey:

1) The 4K board price has been drastically lowered to \$195. People who have ordered 4K's at the old price will receive \$50 credit for each 4K ordered. This credit is applicable to the purchase of a new 4K board, which means you can buy a 4K for the price of \$145!

2) The SLA (Software License Agreement) has been eliminated. No longer will you be required to read and sign this infamous document. However, this should not be interpreted to mean that Altair BASIC is now part of the public domain. We have the copyright on this and our position is the same. A cover letter will be sent with our software to explain. It will be simple and it will hopefully eliminate the existing confusion.

3) The price for Altair BASIC to those of you who don't have a minimum system has been dropped. Effective November 1, the price of 4K BASIC is \$150, 8K BASIC is \$200, and Extended BASIC is \$350. Prices for those with a minimum system remain the same (see ad in this issue).

4) A letter-answering system has been set up. All letters will be answered within 5 working days and eventually this will be reduced to 3 days.

5) A Want Ad column will be set up in Computer Notes for individual Altair Users who wish to trade or sell computer related items to other Altair Users. This service will be free and the only restriction is that it cannot be used to sell products that compete with existing MITS Altair products.

6) Each month \$25 credit toward the purchase of Altair products will be awarded to the author of the best Altair applications article. If you have done something interesting with your Altair, write an article about it and submit it to: Editor/Computer Notes/MITS 6328 Linn NE/Albuquerque, NM 87108

By the way, the September issue of Computer Notes was not printed until the last week in September and this current issue is slated to be printed on October 24. Different publications have different policies regarding the date on their publication. Some magazines like Popular Electronics come out three weeks early; i.e., the November issue comes out during the first or second week of October. Other publications like Digital Design come out during the third or fourth week of the month on the cover (the November issue will be out around the 20th of November). If you got your September issue of C.N. around October 11, you received it "on time." This may change if we have a chance to catch up.

\*\*\*

Recently I attended the "Computer Fair" held at the Lawrence Hall of Science on the University of California campus in Berkeley. I went there at the urging of the People's Computer Company Dragon, Bob Albrecht.

The Lawrence Hall of Science is the huge bastion of modern architecture that sits on a hilltop overlooking Berkeley and most of the bay area (quite a view). Inside, there are a number of museum type science exhibits -- the kind that are normally associated with yawning. There

— CONTINUED PAGE 4 —

## ALTAIR SERVICE DEPT.



Barbara Sims

A few computer clubs are beginning to form throughout the country. One such group is based in New Jersey. The Amateur Computer Group of New Jersey is headed by Sol Libes. This group has regular monthly meetings scheduled in advance with seemingly well-planned topics. All in all, the group appears to be very professional. A newsletter, ACGNJ News (Amateur Computer Group of New Jersey Newsletter) is a monthly publication put out through the efforts of this computer club. The newsletter seems to be very informative with information concerning the members of the group as well as information about their meetings. If you are interested and would like more information about the group, you can contact Mr. Sol Libes (201) 889-2000 (day), 277-2063 (eve), president of the club, or Mr. George Fischer (212) 351-1751.

Another computer group has formed in Texas, known officially as The Computer Hobbyist Group of North Texas. This newly formed group has also taken the bull by the horns and started a very well organized club. The members own various types of equipment and seem willing to share their knowledge as well as time with one and all. The Northern Texas Group also has a monthly newsletter being published. This newsletter offers information concerning meetings of the group, questions and answers on both hardware and software, summaries of various newsletters throughout the country, ideas on where to get help for beginners in the microcomputer field, as well as various other types of information. If you are interested in this group, you can contact Bill Fuller (214) 641-2909 or Neil Ferguson (817) 461-2867. These two gentlemen are the editors of the newsletter. Another contact would be Lannie Walker (817) 244-1013, president of The Computer Hobbyist Group of North Texas.

We also have contacts for other computer groups listed in our new column, "Local Users Groups." If you are a member of such a club and would like to make it known for other users within your area, please send in the group's name (if any) and address as well as contacts for interested people to get ahold of. We'll put this information in our newsletter for 8800 users.

Barbara



COMPUTER BOPPERS

# LETTER from the PRESIDENT

by H. Edward Roberts,  
President, MITS

I have just finished reviewing all the Computer Notes survey forms that you have sent to us. A number of interesting ideas and products were proposed as well as a number of negative comments. I will address these negative comments in this column.

The number one all-time problem is delivery. We are only too aware of this problem. The reason for delivery problems are generally related to two factors.

The first factor has been delivery of materials from our vendors. We have literally sucked the industry dry of certain parts over the last year; examples are 8212, 8201, 2107, 4030, 4060, TTL parts, etc. We are moving into a mode where our long term production requirements are being anticipated more accurately and hopefully this will circumvent many existing delivery problems.

The other factor has been one of optimism. We have consistently spent more time in development than was planned. About three months ago I issued a directive that we would not announce any new products until they are in production. This won't completely eliminate the problems associated with delivery through the initial surge of orders, but it will greatly help.

I am sure that the readers of this publication have noticed a lack of new product announcements recently. This should not be interpreted as a lack of on-going development. We have a large number of new, and we think, breakthrough products that are close to being put into production, but they will be in production when we announce them. The new I/O cards and PROM cards that are announced in this issue are examples.

Questions have been raised about the 4K memory by a large number of customers. I will answer them here.

1. Question: Why did you go with the dynamic memory instead of the cheaper and simpler static approach?

Answer: The dynamic 4K memory promised to be ultimately the lowest cost memory. A 4K byte dynamic memory requires approximately 32 ma of 5 volt power for the total of 8 RAM, and static memory using 2102 requires approximately 960 ma of 5 volt power. Our internal Altair bus specifications say that no card will use more than .5A of 5 volts, we certainly deviate occasionally from this specification, but we try to stay close. Current Altair 4K cards use typically 530 ma at 5 volts, this current includes the bus drivers refresh, RAM, and address select current. In summary this means that

a 4K static card that meets bus drive specifications require close to three times the 5 volt current of dynamic memory. Incidentally, this data is based on the current Intel data sheets and our own internal measurements.

2. Statement: Dynamic memory is slower than static.

Answer: Grossly inaccurate statement--we currently use 4030-2 and 4060-2 which have a worst case access time of 200 nanoseconds, but even the earlier 4K designs use 300 nanosecond memories. Every 2MS the refresh circuit refreshes 64 columns, but this refresh occurs during the start of a cycle and is normally transparent to the processor. Even under worst case conditions it would only change system speed by 1.6% which isn't measurable in most applications.

3. Statement: There are problems with the 4K dynamic.

Answer: True to some extent. We have been unable to get an adequate supply of either 4030, 4060, or 2107 memories. This resulted in our switching the design to the Signetic 2604. This part appeared both on the data sheet and in our test to be a direct substitute for TI or Intel parts but, apparently we received a defective lot of chips, which resulted in an excessive failure rate for the 4K card. We have asked in the previous issue of this publication for the return of any 4K cards which contained the Signetic 4K RAMs. Most of these cards have been returned. At the time I am writing this, we are now returning the tested 4K cards to their owners. But this process is a slow one due to the availability of the TI parts. Please bear with us. As a result of these problems with Signetic RAMs a number of changes and revisions have been made to this card to make it even better. A new P.C. layout is underway which eliminates most jumpers. It should be available in approximately 8 weeks. In any case, if you have any problems, let us know. We will do everything possible to resolve them.

Finally, and most important, we feel that our evaluation of the ultimate cost of the dynamic memory has been proven. Therefore, effective November 1, the price on the MITS 4K memory board will be \$195.00. Furthermore, all of you guys who have loyally purchased MITS 4K cards in the past have an automatic \$50.00 credit for each 4K card you purchased at \$264.00 toward the purchase of a new 4K memory card. Put another way, if you paid \$264.00 for a MITS 4K card in the past, you may buy a new 4K card for \$145.00. I hope this will eliminate any doubts concerning the wisdom of staying with MITS on all future purchases of Altair peripherals.

Another comment that has been made is related to customer service. I think that most of our customers are aware that we are constantly improving the quality and quantity of our service. The problem we have is training people internally to respond to customer requirements, and frankly the number of customers have increased more rapidly than our ability to train. We started the year with two people involved in customer service. We now have over 15 people involved in customer service and are still training. Our goal is that by the end of November you will be able to count on almost instant service, but until then, please bear with us. We are trying!!!

In total, I believe most of our customers agree that MITS has done an enormous amount to revolutionize the computer industry. We have done it by producing a consistently high quality product at a reasonable price. We have made mistakes in the past and will probably make more in the future, but you can be assured if it happens we will rectify the problem. Ever since we introduced the Altair we have been attacked by a small but persistent group of competitors and armchair experts. I personally have been strongly inclined to do battle, but cooler heads here at MITS have won out. I think it time to answer some of the nonsense questions that have been raised.

1. Comment: MITS uses substandard parts.

Answer: We never have nor ever will in the future. Anyone who tells you this, feel free to tell him that I said he is a liar.

2. Comment: MITS is a big company which is not interested in its customers.

Answer: False, MITS is a small company. We had less than 20 employees when we introduced the Altair and now we have grown to 90 as a result of our Altair customers. We are profit motivated, but that motivation exceeds our interest in our products by only the slightest amount.

3. Comment: MITS should give BASIC to its customers.

Answer: Wrong. We made a \$180,000 royalty commitment to Micro Soft in order to have BASIC available to our customers. MITS makes essentially no profit on BASIC. It is done as a service to our customers. The BASIC we supply has universally been accepted as nothing short of fantastic and has allowed Altair customers to be literally years ahead of where they would have been without this software. Contrary to some opinions, software developments are expensive and the people who do these developments feel that they should be paid for their efforts, I agree. It is irrelevant whether software is

— CONTINUED Page 4 —

# CONT. Letter from the President

developed to run on a large IBM computer or the Altair, it costs money. We are selling the BASIC at 1/10 to 1/100 the price large computer companies would get for a similar package, but we are still taking gas. Anyone who would like to argue this point, should feel free to call me at MITS. Anyone who is using a stolen copy of MITS BASIC should identify himself for what he is, a thief.

Recently a number of parasite companies have appeared. These companies are in a tough position. They must attack us for new business but at the same time they are dependent on our product for their survival. This has resulted in a good bit of nonsensical rhetoric and advertising. We are confident that most of our customers see through the nonsense, but a few customers have attacked MITS for the failure of a competitor to deliver or for the failure of a competitor's product to operate properly. We have more than enough of our own problems to worry about without getting involved with competitor's problems. Therefore, I will state again that we stand behind all MITS products and systems. We will do whatever is necessary to make these systems function properly if they haven't been abused by the owner. But if you use non-MITS products, any problems that arise are between you and your vendor.

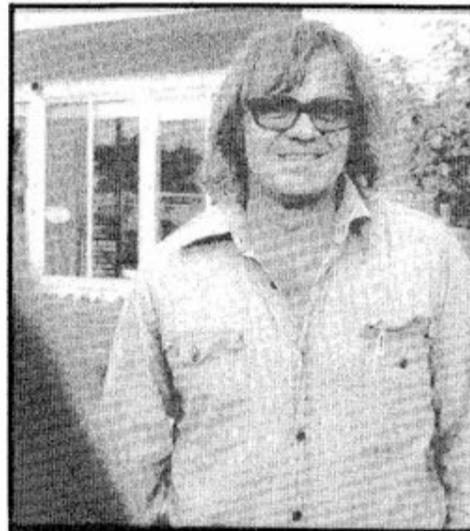
Anyone interested in discussing any of the points I have made in this column or anything else, please write to my attention. I will try to publish as many of these letters as feasible.

One final thought, I would personally like to take this space to thank you for your support both in terms of orders and in terms of suggestions and constructive criticism. I hope you will accept \$50 credit on the 4K memory board as a thank you for your past loyalty.

# CONT. ACROSS THE EDITOR'S DESK

is also a number of classrooms and except for one thing I would hardly recommend the place. That one thing is computers.

When I entered the Lawrence Hall of Science, the first sight I saw was a mass of computer terminals surrounded by junior high age kids who were obviously having the time of their lives (did you know that there are hundreds--perhaps thousands--of young turks who understand BASIC and can program BASIC).



PCC DRAGON

Until you have the experience of watching young people play camaraderie or attacking a match program, you can't really understand the enormous impact computers are going to have on education. Computers don't judge you like teachers -- if you do something wrong you can try again and no one is the wiser.

The advantages of computers in education are such that I am not going to be one of those who backs off and says, "but, of course, computers

will never replace teachers." Computers will replace some teachers. Computers will teach subjects and teachers will teach socialization (someone has to watch the playground).

In addition to thousands of bubblegummers swarming over terminals there were some other interesting things happening at the Computer Fair. The classrooms that I refer to were set up as demonstration rooms for computer companies to display their wares (two days at the Computer Fair were reserved for teachers and administrators).

One classroom had five Altair systems up and running, the most interesting of which belonged to a company called Cromenco that had an Altair hooked up to a color television. They call this interface the "TV Dazzler."

TV Dazzler will work in a 32 x 32, 64 x 64, or 128 x 128 matrix with 7 levels of gray scale in black and white or all 7 TV colors in color plus two intensities of brightness. When you turn it on you see a color representation of memory. If you are running BASIC, you can watch it as it writes into the buffers or onto the stack, etc. The program that I saw running in the Dazzler was a demonstration program written by Stephen Dompier of Altair Music fame. What you do is put a pattern on the screen (it has complete cursor control) and type RUN (using a teletype) and the pattern repeats itself over and over again until the screen is full. The result is a beautiful combination of colors that I would characterize as electronic painting.

Les Solomon of Popular Electronics stayed up until 3 am recently playing LIFE on a TV Dazzler and he says the Dazzler will be featured in the February issue of PE.

## Can You Cope With The Computer Age?



### WHAT'S THE BEST BASIC BOOK IN PRINT?

Did you know there are 36 BASIC language textbooks today? Which is best for 7th graders? For college freshmen? For an experienced FORTRAN programmer? What kind of games or books are best to teach computer literacy to 4th graders? What book has the best discussion of the computer threat to society? Read *Creative Computing* for the reviews that pull no punches.

\* \* \*

### THE NEXT PICASSO -- A COMPUTER?

Can a computer create original art? Or is it just a tool? Does all computer art look "mechanical"? Will computer art have an impact on art as a whole in the future? Can you imagine a printed circuit board winning a blue ribbon in an important art exhibition? Find out more in *Creative Computing*, the magazine that brings computer art to you.

All subscribers will receive a \$5.95 computer art book FREE!

Try *Creative Computing* for a year -- only \$8.00. Or three years for \$21.00. Or send for a sample issue for \$1.00. Please include money; otherwise we add a \$1.00 billing charge.

1 Yr. \$8     3 Yrs. \$21     Sample \$1  
 NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_

STATE \_\_\_\_\_ ZIP \_\_\_\_\_

Return to *Creative Computing* P. O. Box 789-M, Morristown, N. J. 07960.

### NO COMPUTER TODAY?

Looking for learning activities that don't need a computer? Things that teach about computer concepts, the role of the computer in society, its use as a tool in math, science or social studies? How about a whimsical centerfold poster for your bulletin board every issue? Comics too? It's all in *Creative Computing*, the down-to-earth computer magazine.

\* \* \*

### WHAT'S THE LARGEST INDUSTRY IN THE U. S.?

Automobiles? Steel? Agriculture? Not any more. It's computers. More dollars. More jobs. More impact. What do you know about computer careers? Where do you start? What courses to take? Where are the opportunities? Get the answers in *Creative Computing* so you can get the jobs.

### DO YOU ENJOY PUZZLES?

Do you like mathematical diversions? Are mind benders your bag? A printer uses 1215 characters to number the pages of a book -- how many pages in the book? What's the next number in the sequence 63, 94, 46, 18? What digit is represented by each letter: HOCUS + POCUS = PRESTO. You'll find lots more in *Creative Computing*, the fantastic puzzles and pastimes magazine.

\* \* \*

### COULD A COMPUTER TAKE OVER THE WORLD?

Isaac Asimov in a new short story describes what happens when all the computers on earth after a nuclear holocaust link up to support the few remaining human survivors. Want to know the outcome? Then get *Creative Computing*, the magazine that speaks your language.

# 4K Price Cut To \$195!

by David Bunnell

Moments before this issue of Computer Notes was slated to go to press, H. Edward Roberts, President of MITS, announced dramatic cuts in the prices of Altair 4K memory boards. Effective November 1, the price of 4K kits will be dropped from \$264 to \$195. Likewise, the price of assembled 4K memory boards will be dropped from \$338 to \$275.

Noting that problems with 4K boards have now been solved, Roberts also announced that customers who have purchased 4K boards prior to November 1 will receive a \$50 credit for each 4K board purchased. This credit can be applied to the purchase of new Altair 4K boards only.

If you have purchased a 4K board at the old price, you can now purchase a 4K kit for just \$145 or an assembled board for \$225.

Roberts stated that this decision was made after reading surveys returned by MITS customers, many of whom complained about the prices of MITS Altair memory and I/O prices.

Roberts noted that these price changes might create another large backlog but that MITS will make a determined effort to improve its delivery schedule and he asked that all MITS customers please bear with us. It is now a goal of MITS to have 24 hour delivery and all MITS employees are geared toward achieving this.

MITS is determined to provide Altair customers with the best, low cost memory and I/O modules on the market. We will continue looking at our prices and we will continue developing new products, however, MITS will not use junk parts in its products in order to be competitive with junk kit dealers. MITS uses only the best, quality parts and that policy will remain in effect.

## NEW IO'S

After announcing the price cuts for Altair 4K boards, Roberts announced that three new Altair I/O boards are now ready for production. These include a Serial Interface (88-2SIO) that can be ordered with one or two ports, a Parallel Interface (88-4PIO) that can have up to four ports, and a PROM memory card.

These boards could have been announced weeks ago but MITS has adopted a new policy of not announcing any products until they are in production stage. It is hoped that this policy will eliminate problems created in the past by premature product announcements.

### 88-2SIO Interface

The 88-2SIO Interface board can be ordered with one port for just \$115 in kit form and \$144 assembled. The additional port is \$24 for kit and \$35 assembled. Each port provides the following features:

- 1) 5 signal/control lines
  - a) transmit data
  - b) receive data
  - c) data carrier detect
  - d) clear to send
  - e) request to send

This allows for maximum utilization of some of the more sophisticated terminals on the market.

- 2) All signal/control lines are user-selectable for 12 volt levels (RS-232), TTL levels (0-5 volts), or 20 milliamp current loop (Teletype).
- 3) Software programmable for 9 or 10 bit transmission
  - a) 7 data bits + parity bit (odd, even or none) + 1 or 2 stop bits
  - b) 8 data bits + 1 or 2 stop bits
  - c) 8 data bits + 1 stop bit + parity bit (odd or even)
- 4) Full 8 bit status register
  - a) received data available
  - b) transmitter buffer empty
  - c) carrier detect
  - d) clear to send
  - e) framing error
  - f) received data overflow
  - g) parity error
  - h) interrupt request

Allows for greater control and handshaking ability.

- 5) Transmit and receive interrupts --disable/enable under software.
- 6) On-board, crystal-controlled clock for any of eight baud rates (with a single jumper): 110, 150, 300, 1200, 1800, 2400, 4800, 9600.
- 7) Programmable counter can provide other baud rates of 37.5, 75 and 600.

The 88-2SIO with 2 ports can interface 2 serial I/O devices, each running at a different baud rate and each using a different electrical interconnect. That is, the 2 ports can be operating entirely indepen-

ently of each other (such as an RS-232 CRT terminal running at 9600 baud and a 20 milliamp Teletype running at 110 baud). Five volt power consumption is typically 520 milliamps.

### 88-4PIO Interface

The 88-4PIO Interface board can be ordered with a single port for just \$86 in kit form and \$112 assembled. Three additional ports are \$30 each in kit form and \$39 each assembled. Each port provides the following:

- 1) Sixteen data lines
  - a) Each line can be initialized as an input or output so that a single port can interface a terminal (8 lines in--8 lines out) or 2 input devices (such as paper tape reader and keyboard) or 2 output devices (paper tape punch and printer) or any combination for custom applications.
  - b) All data lines are fully TTL compatible and in addition, 8 of the 16 lines can directly drive the base of a transistor switch (1.5v @ 1 milliamp).
- 2) Four controllable interrupt lines
  - a) Enabled/disabled under software control.
  - b) Two lines can act as outputs for ready/busy handshake.

3) Two control/status registers-- Contains a status bit for each of the four interrupt lines.

4) Removable flat cable connection from board to back panel.

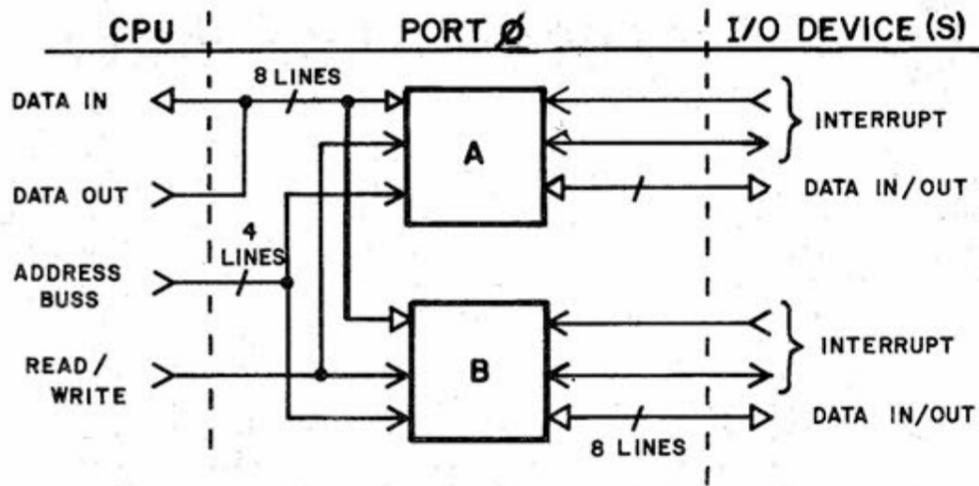
**SEE DIAGRAM FOLLOWING PAGE**

### HARDWARE HINTS

If your CPU board seems to be writing all ones or zeros into the memory, if it gives you the wrong status, or has problems running BASIC, then check IC Q. It should be a Texas Instruments 74123. While other brands may work, we have found that the TI 74123 gives greater reliability in this particular use.

By way of a reminder, all 4K dynamic memories using Signetic RAMs (S 2604) should be returned to the factory for checkout with replacement RAMs. If you have any of our 4K dynamic memory boards that fail to give reliable service, please return them to the MITS Repair Department for checkout.

# 88-4PIO Interface



SIMPLIFIED BLOCK DIAGRAM (1 port)

Assuming the board is addressed at location 0, register selection for port 0 is:

ADDRESS	REGISTER
0	Section A - control
1	Section A - data
2	Section B - control
3	Section B - data

Port 1 would be addresses 4,5,6,7  
 Port 2 would be address 8,9,10,11  
 Port 3 would be addresses 12,13,14,15

An 88-4PIO with 4 ports has 64 data lines (each group of 8 individually selectable) and consumes 500 milliamps at 5 volts - typical.

## Prices

Item	Kit	Assembled
88-4MCS	\$195	\$275
\$50 credit on new orders for each board ordered, i.e. has two 4K's, can order two more at \$145 each.		
88-2SIO	\$115 single	\$144
88-SP	\$24 additional port	\$35
88-4PIO	\$86 single	\$112
88-PP	\$30 additional ports (up to 4)	\$39
88-PMC	\$65	\$128

### CUSTOMER SERVICE NOTE:

If you plan to stop by at our Albuquerque plant to have a MITS product serviced while you wait, please notify us 2 days in advance so we may set aside time for our technicians. We are extremely busy and have quite a backlog at this time, so your cooperation will be appreciated.

### PROM Memory Card

- The 88-PMC PROM memory card provides up to 2K bytes of non-volatile memory for boot loaders and other programs that must be retained if power to the computer is interrupted. The card uses either 1702 or 1702A PROMs which contain 256 8-bit bytes. These PROMs are electrically programmable and erasable (using ultra-violet light) so they can be reprogrammed if necessary. The non-erasable versions (1602 and 1602A) can also be used in the card.
- The card has a power down option consisting of four drivers that switch the PROMs on and off in pairs to reduce current drain. In the "off" state the PROMs draw about 15% to 20% of the current required in the "on" state. Except for some overlap in switching, only two of the PROMs can be in the "on" state at a time.
- There are provisions on the card for patching the number of wait states from 0 to 3 in order to accommodate different speed devices.

### ENGINEERING NOTE:

For proper operation of a Teletype with the 88-SIOC interface board, the Teletype must be internally wired for 20mA, full duplex operation. MITS is currently selling Teletype model #3320-3JE. Instructions for changing to 20mA, full duplex operation may be found in the wiring diagram #9336WD-B2A supplied with the Teletype (this model comes from the factory wired for 60mA, half duplex). Also, be sure that R10 on the 88-SIOC is 220 ohm, not 390 ohm. This resistor determines the 20mA loop current operating the printer part of the Teletype.

## Using the VLCT

The VLCT is a very useful octal I/O device. It consists of a 10-key keyboard (0-7, Ready, and Clear), an eight LED input register display and a three digit octal output display. The VLCT interfaces to the Altair through the 88-PIO using TTL logic levels. Data is entered into the Altair by use of a short input program, such as the one listed below.

### VLCT INPUT PROGRAM

Stores data in memory starting at address specified by LXI instruction (must be different from input program.)

	0	041	LXI	Load H & L with starting address
	1	XXX		Least significant address byte
	2	XXX		Most significant address byte
IST:	3	333	IN	
	4	000		
	5	346	ANI	Test for input latch loaded
	6	002		(after 3 keystrokes)
	7	312	JZ	
	10	003		Jump to IST
	1	000		
	2	333	IN	
	3	001		
	4	167	MOV	(Mem H,L) ← (A)
OST:	5	333	IN	
	6	000		
	7	346	ANI	Test for Ready key
	20	001		pressed
	1	312	JZ	
	2	015		Jump to OST
	3	000		
	4	176	MOV	(A) ← (Mem H,L)
	5	323	OUT	
	6	001		
	7	043	INX	(H,L) + 1
	30	303	JMP	
	1	003		Jump to IST
	2	000		

NOTE: For 88-PIO address 0E1

After 3 keystrokes on the VLCT, data is automatically transferred to the input latch, where software stores it in memory. Pressing the Ready key causes the software to echo back the data to the octal display on the VLCT.

By changing the test byte after the first "ANI" you can use the Ready key to trigger storage of data to prevent automatic entry of an incorrect code.

This program stores octal data sequentially in memory starting at the address specified by LXI. It should be used for writing machine language programs; and, for long programs, it is faster and easier than loading in through the front panel switches.

Continued from Page One

# ALTAIR 680

## New 6800-Based Computer

The 6800 has seven different addressing modes, with the particular mode being a function of both the type of instruction and the actual coding within the instruction. The seven modes include the following:

**Accumulator Addressing**--These are one byte instructions, specifying either of the two accumulators.

**Immediate Addressing**--These are two or three byte instructions, with the MPU addressing the location given in the 2nd or 2nd and 3rd bytes when the immediate instruction is fetched.

**Direct Addressing**--These are two byte instructions which allow the user to directly address the lowest 256 bytes of memory in the machine.

**Extended Addressing**--These are three byte instructions, the second two bytes referring to an absolute address in memory for the operation.

**Indexed Addressing**--These are two byte instructions, the second byte being added to the 16-bit index register to give the address of the operand.

**Implied Addressing**--These are one byte instructions and the instruction itself gives the address.

**Relative Addressing**--These are two byte instructions where the second byte is added to the lower 8 bits of the program counter plus two. Carry or borrow are added to the higher 8 bits. This allows the user to address memory +129 to -125 bytes from the location of the present instruction.

These various addressing modes may take a bit of getting used to, but once understood they allow for some very fast programs to be written.

The 6800 MPU contains three 16-bit registers and three 8-bit registers. The program counter is a two byte register which keeps track of the current address of the program. The stack pointer is also a two byte register which contains the next address in an external, variable length push-down/pop-up stack. The index register is a two byte register used to store data or a memory address for indexed addressing operations. There are two single byte accumulators used for holding operands and results from the arithmetic logic unit (ALU). The 8-bit condition code register indicates the results of an ALU operation. In this register there are two unused bits, kept at a logic one. The remaining six bits are used to indicate the status of the following: carry, overflow, zero, negative, interrupt

There are several timing and control signals required to operate the MPU. Two clock inputs are required, phase 1 and phase 2. These must be nonoverlapping and run at the Vcc voltage level. In the 680 the clock is a 2-MHz crystal-controlled oscillator with logic to provide a 500-KHz two phase clock. Although the MPU is capable of using up to a 1-MHz clock, by reducing the speed to 500-KHz the overall design of the 680 was kept much simpler and greatly reduced the cost. Sixteen active high address outputs are used to specify the sections of memory or I/O to be used. These can drive up to one standard TTL load and 130pf. There are also eight bi-directional data

lines with the same drive capability as the address lines. The Halt signal is an active low input which ceases activity in the computer. The Read/Write signal in the high state indicates that the MPU is in a read condition, in the low state it indicates that the MPU is in a write condition. The Valid Memory Address signal tells external devices that the MPU has a valid address on the memory bus. The Data Bus Enable signal is the input which enables the bus drivers. The Bus Available signal indicates that the machine has stopped and that the address bus is available. Reset is used to reset and start the MPU from a power-off condition. The Interrupt Request signal, when low, tells the MPU to start an interrupt sequence. This can occur only if the interrupt mask bit in the condition code register is low. The Nonmaskable Interrupt signal is essentially the same as the Interrupt Request signal except that it is not dependent on the condition code register.

### Memory

The main printed circuit board on the 680 contains the basic memory for the unit also. This includes 1024 bytes of Random Access Memory and provisions for another 1024 bytes of ROM or PROM. The RAMs being used are the 2102 type 1024 x 1-bit RAMs. ROMs are custom ordered, and are usually very expensive. The PROMs being used are 1702 type, ultra-violet erasable PROMs. The PROMs are 256 x 8-bit units.

There is additional memory for the 680 on the drawing board at this time which may add up to 12K bytes more storage to the unit.

### I/O Port

Also on the main printed circuit board is a built-in I/O port and the appropriate interface circuitry. This port may be configured as either an RS232 level port or either a 20ma or 60ma current loop TTY level port. This means it can be easily inter-

types. The entire design of the 680 is greatly simplified due to the 6800's I/O structure. The MPU uses addresses to refer to I/O devices as well as memory, rather than have special I/O instructions and a separate I/O bus. This also gives it the ability to operate a considerably large number of separate I/O devices. MITS also has additional I/O interfaces on the drawing board now; although this and the additional memory boards will be greatly influenced in their development by customer response.

### Control and Indication

On the fully user-programmable version of the 680, the front panel assembly contains a RUN/HALT switch with an LED indicator for each switch position. There is a RESET switch with no indicator, and another indicator for the AC power switch which is located on the back panel of the unit. The switches for the 16 address lines and 8 data lines, and their associated indicator lights, are also located on the front panel assembly. There is also a DEPOSIT switch.

The DEPOSIT, RESET, DATA and ADDRESS switches are enabled only when the RUN/HALT switch is in the HALT position. To view the data in a particular memory address, the RUN/HALT switch must first be in the HALT position and then the ADDRESS switches may be set to the required address. The data located at that particular address will then appear on the DATA LED indicators above the DATA switches.

To place data, or to change it, in a desired location; once the correct address has been set on the address switches, the appropriate data should be entered on the DATA switches and then the DEPOSIT switch activated. Since the address bus is already connected to the switches by being in the HALT state, a write pulse causes the data to be written into the selected RAM address.

When the RESET switch is activated, the MPU itself resets. This initiates a restart sequence, pulling the address bus to its high state and causing hard-wired data on the board to be used as the restart address.

On the dedicated program versions of the 680, most of these functions are taken care of by ROM or PROM. The only controls available to the user are the AC power and RESET switches.

*continued on page 9*

# USING SERIAL I/O BOARDS

by Tom Durston

The MITS technical staff has been receiving many questions on the writing of software to handle the 88-SIO boards. Most of the confusion has been due to a lack of explanation of the fundamental concept of the Altair I/O structure. We hope the following will help answer users' questions on 88-SIO software.

**Inputting** data from an external I/O device: The input instruction is a two byte (1 byte = 8 bits) instruction. The first byte (333<sub>8</sub>) is the code telling the CPU to input data from an external device (TTY, Comter, CRT terminal, etc.) and put it in the accumulator. The second byte (XXX) is the address of the I/O board connected to the desired external device. The address of the I/O board is determined by the user by seven hardware jumpers (I1 - I7) on the I/O board. The list of addresses and jumpers is in the back of the I/O board assembly manual. After execution of this two byte input instruction, the input data present at the I/O board is transferred to the accumulator in the CPU. There are 256 I/O addresses, 000<sub>8</sub> through 377<sub>8</sub>; the even numbered ones being used for status and control of the I/O board and the odd numbered ones being used for data transfer. This gives the user a maximum possible number of 127 external I/O devices; the 128th I/O device is the sense switch input (address 377).

To input valid received data from an 88-SIO board, the status channel information must be inputted and bit D<sub>0</sub> tested. For Rev 1 88-SIO boards, D<sub>0</sub> = 0 indicates that a new character has been received from an external device. When status bit D<sub>0</sub> has been found to equal 0, then the data channel may be inputted. Status bit D<sub>0</sub> is reset when an input to the data channel is done. From this point the user may do anything desired with the data. A typical input program using address 0E1 would look like this:

Byte 1	000	333	IN	Input
2	1	000		Stat address
	2	017	RRC	Test D <sub>0</sub> , rotate Acc right
Byte 1	3	332	JC	Jump if carry (D <sub>0</sub> = 1)
2	4	000		
3	5	000		
Byte 1	6	333	IN	Input
2	7	001		Data address
Byte 1	10	062	STA	Store char in memory loc 000 040
2	11	040		
3	12	000		
Byte 1	13	303	JMP	Jump to beginning
2	14	000		
3	15	000		

**Outputting** data to an external I/O device: The output instruction is a two byte instruction. The first byte (323) is the code telling the CPU to output the data in the accumulator to an external I/O device. The

second byte (XXX) is the address of the I/O board connected to the desired external device. When outputting to an 88-SIO board on the even numbered address line (control channel), the interrupt enable circuit may be enabled or disabled by bits D<sub>0</sub> and D<sub>1</sub> (interrupt is explained later in this article). When outputting to an 88-SIO on the odd numbered address line, parallel data is transferred from the accumulator to the 88-SIO and is transmitted serially to the external device.

When outputting data to an external device, a test of status bit D<sub>7</sub> (Rev 1 88-SIO board) must be done to see if the 88-SIO board is ready to transmit a character. When status bit D<sub>7</sub> = 0, indicating the transmitter buffer is empty, then data to be transmitted is placed in the accumulator and outputted to the 88-SIO board on its data channel. A typical output program using address 0E1 would look like this:

Byte 1	000	333	IN	Input
Byte 2	1	000		Status Channel
	2	007	RLC	Test D <sub>7</sub> , rotate Acc left
Byte 1	3	332	JC	Jump if carry (D <sub>7</sub> = 1)
2	4	000		
3	5	000		
Byte 1	6	333	IN	Input
2	7	377		from sense switches
Byte 1	10	323	OUT	Output
2	11	001		Data Channel
Byte 1	12	303	JMP	Jump to beginning
2	13	000		
3	14	000		

This program continuously transmits data from the sense switches to the external device.

*continued on page 9*

## 1. JUST A REMINDER

Is your Altair in a RUT? Does it just sit there, unresponsive and illogical? Well, you may have pulled one of your boards out of the bus with the power on. It tends to render inoperative ICs R, U, and T on the display and control board. SOLUTION: Replace R, U, and T and remember to turn power off before removing PC boards from the bus.

## 2. DIFFERENCES BETWEEN STATIC AND DYNAMIC MEMORY

A static memory cell consists of a flip-flop which retains its state unless changed by writing a new bit or power going off.

A dynamic memory cell consists of a capacitor with transistors used to sense, maintain, or change its charge. Dynamic memories require a refresh cycle to maintain the charges in the memory cells.



For the MITS 1K static board, 88-MCS, a wait state of 1 microsecond is required due to the access time of the RAM. This means that every step of a program takes one microsecond in addition to the instruction cycle time.

For the MITS 4K dynamic board, 88-4MCD, no wait state is required, but every 32 microseconds, a refresh cycle of one microsecond is done. During the refresh cycle, no reading or writing of data is allowed.

One advantage of dynamic over static memory is that bit for bit, it takes less power.

An advantage of static over dynamic memory is that with faster access times becoming available, they are more practical due to less control circuitry.

# HARDWARE

# SI/O CONT...

By putting together the input program and output program we can create an "echo" program so that a character transmitted to the Altair will be immediately received back.

Byte 1	000	333	IN	Input
2	1	000		Status Channel
	2	017	RRC	Rotate acc right
Byte 1	3	332	JC	Jump if D <sub>0</sub> = 1
2	4	000		(no data)
3	5	000		
Byte 1	6	333	IN	Input
2	7	001		Data Channel
Byte 1	10	062	STA	Store data
2	11	040		
3	12	000		
Byte 1	13	333	IN	Input
2	14	000		Status Channel
	15	007	RLC	Rotate acc left
Byte 1	16	332	JC	Jump if D <sub>7</sub> = 1
2	17	012		(transmit busy)
3	20	000		
Byte 1	21	072	LDA	Load acc with
2	22	040		stored data
3	23	000		
Byte 1	24	323	OUT	Output
2	25	001		Data Channel
Byte 1	26	303	JMP	Jump back to
2	27	000		beginning
3	30	000		

**Interfacing** the 88-SIO with the external device: First you must determine what interface your terminal requires. The three standard types available from MITS are EIA-RS232 (88-SIOA), TTY - 20mA loop (88-SIOC), and TTL logic (88-SIOB).

**Selecting** the correct baud rate for your I/O device will depend on its requirements. For instance Teletypes operate at 110 baud only, the Comter operates at 110 or 300 baud, most CRT terminals operate between 110 baud and 9600 baud. (Baud = bits per second)

For Teletype, use 110 buad, 2 stop bits, and 8 data bits. For baud rates of 300 and up, use one stop bit and 8 data bits. Note that above 2400 baud the output pulse of IC "0" must be between 2.0 and 2.5 microseconds for correct operation. Varying R25 (7.5K) will obtain proper timing.

If you wish to check parity, use 7 data bits and either odd or even parity, depending on your requirements.

Selecting the address of the I/O board is up to the user with the following exceptions (used in the BASIC language software):

Address 0/1	-- I/O terminal
2/3	-- Line Printer
4/5	-- Alternate terminal
6/7	-- ACR
10/11/12	-- Altair Disk
376	-- VI, RTC
377	-- Sense Switch

## Altair 680 —

cont. from page 7

**Interrupt:** Using the Interrupt feature of the CPU allows the Altair to be performing a task other than monitoring the I/O ports for data transfers. To use the interrupt feature, the stack address should be specified, interrupts enabled in the CPU, interrupts enabled on the I/O board, and a subroutine written at the proper location to handle the I/O of data. Using single level interrupt (vectored interrupt hardware will be available by the end of the year), all interrupts will cause the CPU to jump to location 070 where the I/O subroutine should be located. Single level interrupt is utilized by connecting the desired interrupt mode pad ("IN", "OUT", "BH") to the "INT" pad on the 88-SIO board. A short input program that uses single level interrupt would look like this (for I/O board address 0&1):

Byte 1	000	061	LXI SP	Set stack pointer
2	1	200		to 000 200
3	2	000		
	3	373	EI	Enable interrupts
Byte 1	4	076	MVI	Move 001 into
2	5	001		accumulator
Byte 1	6	323	OUT	Output to I/O
2	7	000		board control channel
Byte 1	10	303	JMP	Jump to self and
2	11	011		wait for interrupt
3	12	000		
Byte 1	070	333	IN	Input from
2	71	001		Data Channel
Byte 1	72	062	STA	Store data in
2	73	040		loc 000 040
3	74	000		
	75	373	EI	Enable interrupts
	76	311	RET	Return to "jump to
				self"

The EIA-RS232 interface on the 88-SIOA board offers only transmitted data, received data, and signal ground (circuits BA, BB and AB). If your external device requires other signals, they will have to be strapped to the proper DC voltages for proper operation. EIA interface levels are: Logic 0: +5 to +25 volts, Logic 1: -5 to -25 volts.

The TTY - 20mA loop interface is used for Teletypes or I/O devices utilizing that interface level. The 88-SIOC levels are: Logic 0--open circuit, no current flow; Logic 1--closed circuit, 20mA current.

The TTL logic interface uses 7400 type logic levels. The interface level for the 88-SIOB are: Logic 0: +.4 volts, Logic 1: +2.2 volts.

### Power Supply

The 5-volt supply to the computer is supplied from the power transformer through a conventional bridge rectifier and filter capacitors and voltage regulator IC. A 32-volt winding on the transformer is used to generate the unregulated 16 volts required for a TTY interface, and a -16 volt line is fed to four zener diode regulated outputs to provide four -9 volt lines for the PROMs.

The transformer itself, along with the power switch, are located on the computer's back panel. There are also provisions for installing a cooling fan when necessary.

MITS has decided to await customer response to determine the course of further 680 development in both the areas of software and hardware.

Although it's not quite as powerful as the ALTAIR 8800, the ALTAIR 680 is mighty close and at less than half the price for a similar system.

OOPS

# Lowest Price in the World!

In January of 1975, MITS stunned the computer world with the announcement of the *Altair 8800* Computer that sells for \$439 in kit form.

## Today MITS is announcing the *Altair 680*.

The *Altair 680*, built around the revolutionary new 6800 microprocessor chip, is the lowest priced complete computer on the market. Until December 31, 1975, this computer will be sold in kit form for the amazing introductory price of \$293! (A savings of \$52!)

The *Altair 680* comes with power supply, front panel control board, and CPU board inclosed in an 11" wide x 11" deep x 4 11/16" case. In addition to the 6800 processor, the CPU board contains the following:

1. 1024 words of memory (RAM 2102 type 1024 x 1-bit chips).
2. Built-in Interface that can be configured for RS232 or 20 mA Teletype loop or 60 mA Teletype.
3. Provisions for 1024 words of ROM or PROM.

The *Altair 680* can be programmed from the front panel switches or it can be connected to a computer terminal (RS232) or a Teletype such as an ASR-33 or surplus five-level Baudott Teletype (under \$100).

The *Altair 680* can be utilized for many home, commercial or industrial applications or it can be used as a development system for *Altair 680* CPU boards. With a cycle time of 4 microseconds, 16-bit addressing, and the capability of directly addressing 65,000 words of memory and a virtually unlimited number of I/O devices, the *Altair 680* is a very versatile computer!

## Altair 680 Software

Software for the *Altair 680* includes a monitor on PROM, assembler, debug, and editor. This software is available to *Altair 680* owners at a nominal cost.

Future software development will be influenced by customer demand and may include *BASIC on ROM*. MITS will sponsor lucrative software contests to encourage the rapid growth of the *Altair 680* software library. Programs in this library will be made available to all *Altair 680* owners at the cost of printing and mailing.

Contact factory for updated information and prices.

## Altair Users Group

All *Altair 680* purchasers will receive a free one year membership to the Altair Users Group. This group is the largest of its kind in the world and includes thousands of *Altair 8800* and *680* users.

Members of the Altair Users Group are kept abreast of Altair developments through the monthly publication, **Computer Notes**.

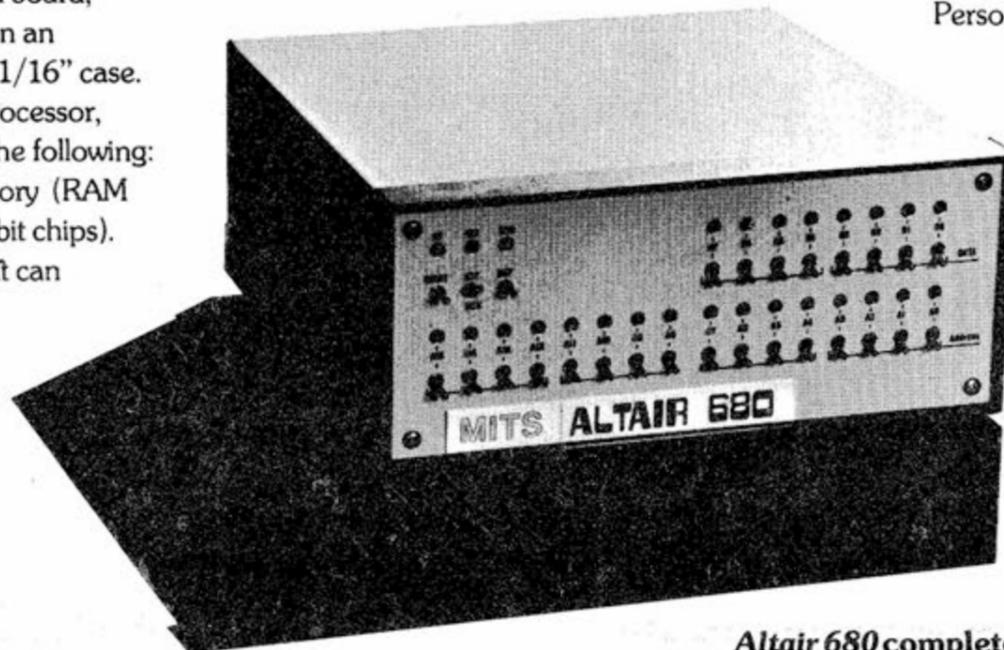
## Altair 680 Documentation

The *Altair 680* kit comes with complete documentation including assembly manual, assembly hints manual, operation manual, and theory manual. Assembled units come with operation and theory manuals. Turnkey model and CPU boards also include documentation.

**NOTE: *Altair 680* manuals can be purchased separately. See back page of this catalog for prices.**

## Delivery

Personal checks take 2-3 weeks to process while money orders and credit card purchases can be processed in 1-3 days. Delivery should be 30-60 days but this can vary according to order backlog. All orders are handled on a first come, first served basis.



## Altair 680 Prices

<i>Altair 680</i> complete computer kit ....	\$293
(\$345 after December 31, 1975)	
<i>Altair 680</i> assembled and tested .....	\$420
<i>Altair 680T</i> turnkey model (complete <i>Altair 680</i> except front panel control board) Kit Only .....	\$240
(\$280 after December 31, 1975)	
<i>Altair 680</i> CPU board (including pc board, 6800 microprocessor chip, 1024 word memory, 3 way interface and all remaining components except power supply) .....	\$180
(\$195 after December 31, 1975)	
<i>Altair 680</i> CPU board assembled and tested .....	\$275
Option I/O socket kit (required when interfacing 680 to external devices) .....	\$ 29
Option cooling fan (required when expanding 680 internally) .....	\$ 16
(\$22 after December 31, 1975)	
Option cooling fan installed .....	\$ 26
PROM kit (256 x 8-bit ultraviolet, erasable 1702 devices) .....	\$ 42

# MITS

"Creative Electronics"

Prices, delivery and specifications subject to change.

GRADED INTENSITIES

DEBRIS,  
AND NUMB TRANCES  
WITH CALM CREATION,  
YOUR NUMBNESS ASTONISHES  
IN DREAM BALLOONS.  
SEDUCED IS SLEEP  
WHICH HOVERS  
IN TRANCEMOTIONLESS MOONLIGHT.

THE INVERSION WISHES OF DRAGONFLIES  
ARE MADE  
OF FLASHING GLASS  
LURING IN THE FREEDOM  
OF SUBORDINATION,  
LIKE THEIR SILENT SPIRITS.

THE ROMANCE OF WINGS  
IS THE THINNESS OF STREAMS.  
THEIR SUPPORTING COMPLEXITY  
IS GRACEFUL LIKE WRINKLES  
TURNING A FLAT EYE.  
WINGS HOVER  
LIKE FOG AND GARGOYLE,  
HOVERING WITH THE FLAKE  
OF COMPLEXITY BRANCHED.

A CAUSE OF OCEANS,  
MIGHTY AS RELIABLE ROPES,  
SCREAMS IN WILDNESS  
WHILE A DETAIL OF TREES  
SWAYS, LIKE A COBWEB  
BEING ATTACHED  
BY EXCESSIVE MASKS.

TO LINK WITHOUT INVULNERABILITY  
TO GLOW, IN FUNDAMENTAL COMPLETION  
WITHIN A CONCENTRIC EDGE  
AS STRENGTH RELOCATES ITS HARMONY  
A RAINFALL EVADES THE SYSTEM  
CONTRIVE, AND FIND THE RAINFALLS OF  
MODERATION  
COMPLEX CHAINS AND HAPPINESS  
GLIDING AT DIRECTIONS  
THE WINTER OF ELATION

CREATE TO CHEER  
SPLIT TO COMPLETE  
HARMONIES STIMULATE THE STIGMAS  
INSULTING AT EMBARRASMENTS  
MY ELATIONS LOITER  
ELATION-BREASTS  
PERFORMING ACID HAMMERS  
APPETITE LIKE ZONES

by James Runner  
above poems are from  
COMPUTER POEMS  
gathered by Richard W. Bailey  
Potagannissing Press  
Drummond Island, Michigan

# COMPUTER POETRY

HARMONY

This is my word to the multitude:  
I stopped upon a proud tower  
Above the fresh ruffles of the surf,  
I, a fond prophet in the place of the finish.  
Harmony is not all: It is not flesh nor body.  
Harmony is not all: It is not food nor water.  
Above the humid ruffles of the shore,  
This is my sentence to the creation,  
This is my message to the creation.

from PROGRAM ERATO  
by Louis T. Milic

## LOCAL USERS GROUPS

This month we'll be starting a new column that will list names and addresses of local computer clubs and users groups that want to recruit new members in their area. Please send in the name of your club if you would like to have it published.

Canadian Computer Club  
G. Pearen  
861 11th Street  
Brandon, Manitoba  
R7A 4L1

"Universe Unlimited"  
User's Group  
John E. Kabat  
11918 Forrest Ave.  
Cleveland, OH 44120  
Business 216-781-9400 Ext.55  
Home 216-795-2565

29 Palms, California area:  
Sgt. Wesley B. Isgrigg  
Box 3558, C&E Schools M.C.B.  
29 Palms, CA 92278  
714-368-9111 Ext.6241  
evening Ext.7289  
and  
Sgt. Stanley E. Herr  
13-C Copper Dr. M.C.B.  
29 Palms, CA 92278  
714-368-3809

(Wesley and Stanley are especially interested in programs to design circuits and in receiving information on Hal Chamberlain's Graphics Display.)

\*\*\* For Last Minute Additions to this  
\*\*\* column, see page 18 (lower right)



ACROSS

- 1 circuit between two computer components
- 8 neon (abbr)
- 9 to put to rest; to calm
- 10 in physics; "Outer Tensile Strength" (abbr)
- 12 corner joint
- 13 to process a printed circuit board
- 14 opining between mountains
- 15 a BASIC instruction
- 16 in Morse code; a dash
- 18 programmable read only memory (abbr)
- 22 allow
- 24 "\_\_\_-Sun Bay"; a part of San Francisco Bay
- 25 citizens band radio (abbr)
- 27 route (abbr)
- 28 street (abbr)
- 29 persons who use computers as a passtime

### CROSSWORD PUZZLE

1	2			3	4	5	6	7
8				9				
10		11		12				
13					14			
15								
	16		17		18		19	20
21		22		23		24		
25	26		27				28	
29								

DOWN

- 1 "chip" company
- 2 programming term for loops within loops (adj)
- 3 random access memory (abbr)
- 4 "\_\_\_-flop"
- 5 computer name (pl)
- 6 autos
- 7 you see with them
- 11 base eight
- 17 man's name
- 19 to force out; drive out
- 20 computer company
- 21 "The Computer Hobbyist" (abbr)
- 23 Teletype (abbr)
- 26 hobo (slang)

contributed by Ron Santore

CONTRIBUTORS WELCOME--SEND ARTICLES  
TO COMPUTER NOTES/MTS/6328 Linn  
NE/Albuquerque, New Mexico 87108

# Software Contest Winners Announced

Many excellent additions to the Altair Software Library were accepted during the October software contest. Choosing the winning programs this month was particularly difficult as the overall quality of all the entries was exceptionally good. We finally narrowed it down, however, and Ian Kettleborough came up with first place in the major program category for his Fortran IV cross assembler. A memory dump by Roger Walker won first prize for the subroutine category. For this month's complete listing, see below.

## FIRST PLACE MAJOR PROGRAM

#1016752

Author: Ian D. Kettleborough

Length: 44 pages

A Fortran IV cross-assembler, currently running on a Data-General. Several listing options, as well as an object dump can be selected. Conditional assembly provided for.

## SECOND PLACE MAJOR PROGRAM

#910751

Author: Ross E. Housholder

Length: 22 pages

A very complete discussion of playing music on the Altair, including translation programs and an improved PLAY program with source instructions.

## THIRD PLACE MAJOR PROGRAM

#916751

Author: Gary Tack

Length: 128 bytes

Program to play DICE with a basic Altair.

## FIRST PLACE SUBROUTINE

#929751

Author: Roger Walker

Length: 58 bytes

Dumps a block of memory in checksum format.

## SECOND PLACE SUBROUTINE

#1010752

Author: Jacques Roth

Length: 19 bytes

Timing subroutine. Delays a specified number of seconds.

#920751

Author: Dale J. Travis

Length: Approx. 80 bytes

Converts an ASCII string in octal to a signed 16-bit number.

#1014751

Author: Jack O. Coats, Jr.

Length: 73 bytes

Loads data from terminal into memory and executes data after the termination character. Currently written to work with a terminal connected through a parallel I/O board.

#102751

Author: Robert Rydel

Length: 82 bytes

"Error Detection Code Generator/Checker." Generates cyclic redundancy check (CRC) to allow for doing error detection.

#106751

Author: Gerard Bilodeau

Length: 25 bytes

A message print subroutine. Allows for embedded carriage return/line feeds and any delimiter character.

#911751

Author: Dale J. Travis

Length: 16 bytes

Subroutine to convert binary to a 3-digit BCD number.

#912751

Author: Gerhald Hansel

Length: 70 lines

A BASIC program that plays roulette.

#1010751

Author: Jacques Roth

Length: 256 bytes

Program to play the "Bagels" game with a minimum Altair. Instructions included.

#109751

Author: William D. Roch

Length: 30 lines

A BASIC program to calculate the number of days between two dates. Takes leap years into consideration.

#1016751

Author: Steve Caldara

Length: 21 lines

A BASIC "Lunar Lander" program.

#912752

Author: Bob Omer

Length: 78 lines

BASIC program that plays the Stick game. Instructions available when the program is run.

#920752

Author: Dale J. Travis

Length: Approx. 120 bytes

Program to convert a 16-bit signed integer to an ASCII string of octal digits.

## Software Notes

by Bill Gates

### Using the STACK

Every program written for the 8800, large or small, should take advantage of the stack. The stack is a stored list of data which behaves in a last in--first out (LIFO) fashion. That is, PUSH D/POP D doesn't modify [D,E] since the POP removes the same two bytes that were just pushed onto the stack. There are three things to remember in using the stack:

1) Initialize the stack pointer to the highest location in a block of free Read/Write memory. As data is pushed onto the stack the stack pointer is decremented.

2) Make sure that the amount of space set aside for the stack is sufficient to hold all the values (including return addresses) that are stored on the stack at one time. Every time a PUSH or CALL is done the stack pointer is decremented by two, so data pushed onto the stack will be stored in lower and lower locations. Every POP or RETURN increments the stack pointer by two, so the important thing is not how many values are PUSHed on during a program, but how many are on at the same time.

Consider:

```

                LXI   SP, STKBOT
LOOPDO:        PUSH  D           ;save [D,E]
                :
                :
                CALL  SUB1
RETADR:        :
                :
                POP   D           ;restore [D,E]
                DCR   B
                JNZ   LOOPDO
                :
                :
                DS    4
STKBOT:
SUB1:          :           ;no stack use
                :
                RET

```

No matter how many times the above loop is executed, [D,E] will always be saved and restored from the same memory locations, since the stack pointer is incremented by 2 by the PUSH, incremented by 2 by the CALL, decremented by 2 by the RET, and decremented by 2 by the POP. Only four bytes of stack space are set aside for this program since only 2 bytes of data (D and E) and 2 bytes of return address are ever stored on the stack at a time. If additional PUSH/POPs or CALLs are done inside the loop, more stack space would have to be set aside. Unless you are using all of memory and need to compactify as much as possible, it is a good idea to allocate a lot more stack space than you think you will ever use.

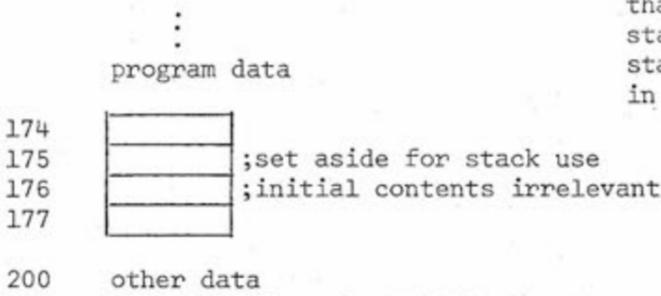
3) Unless you are being very tricky, never take more data off of the stack than is put on. This means doing more POPs and RETs than CALLs and PUSHs.

continued on page 13

# SOFTWARE

ILLUSTRATION OF STACK OPERATION

Consider the example given earlier. Assume that STKBOT = 200 octal.



After the LXI SP, STKBOT SP will equal 200.

PUSHs and CALLs always put 2 bytes of data onto the stack as follows:

- 1) Decrement SP
- 2) Store the high 8 bits of data being PUSHed in the memory location given by SP.
- 3) Decrement SP
- 4) Store the low 8 bits of data being PUSHed into the memory location given by SP.

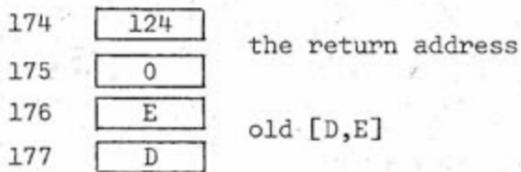
PUSHMA:

When the PUSH D is done: SP = 177, D is stored at 177, SP = 176, E is stored at 176.

Say RETADR = 124

When the call is done: SP = 175, high byte of Retadr = 0 and is stored at 175, SP = 174, the low byte of Retadr = 124 and is stored at 174.

So we have:



POPs and RETs do the "reverse" operation:

- 1) Pick up the low 8 bits of data in the memory location given by the SP
- 2) Increment the SP
- 3) Pick up the high 8 bits of data in the memory location given by the SP
- 4) Increment the SP

Exercise: Work out the details of how the RET and POP D in the example work.

Other machines that have stacks vary in the details of implementation. Some machines increment the SP on PUSHs and decrement on POPs (PDP-10). Other machines store before decrementing on PUSHs and increment before fetching on POPs (Altair 680). However, the basic notion of a last in first out list to store data and return addresses remains the same.

The way subroutines are nested, that is, always returning to the most recent caller, makes the stack very natural for storing return addresses.

The fact that the stack is used to store both return addresses and data allows for some tricky programming involving manipulation of return addresses on the stack, of which a few examples were given in earlier "Software Notes." It also causes some trouble however for subroutines that wish to leave results on the stack or fetch arguments from the stack, since the return address gets in the way. The sequence:

```

MOV C,M
INX H
MOV B,M
INX H
PUSH B
  
```

was used many, many times in BASIC so it was decided to use one of the RST instructions to perform this operation. It was coded as follows:

```

XTHL ;[H,L]=return address
SHLD PUSHMA+1 ;modify a JMP
POP H ;get [H,L] back
MOV C,M
INX H
MOV B,M
INX H
PUSH B
JMP * ;JMP to return point
  
```

If only a single byte of data needs to be pushed onto the stack, put the data in A, B, D, or H and do a PUSH PSW, B, D, or H respectively, followed by a INX SP. To pop off the single byte of data do a DCX SP, POP A, B, D, or H. This puts garbage into the PSW, C, E, or L respectively. Unless the same PUSH will be used to store a large number of one byte pieces of data on the stack, it is simplest to merely do a PUSH/POP sequence and allow the extraneous byte to be stored.

Sometimes the amount of stack space a program requires will depend on the input to the program, for instance when BASIC evaluates complicated formulas. If this is the case, a check must be done when data is pushed on to make sure the stack isn't "overflowing." If it is, either some sort of recovery procedure must be invoked, or an error message printed. The following subroutine checks to see if the stack is pointing below STKSTP.

```

; saves all registers
CHKSTK: PUSH H ;save [H,L]
        LXI H,-STKSTP ;won't work for STKSTP=0
        DAD SP
        POP H ;[H,L]
        RC ;return if still ok
; here on stack overflow
  
```

If you have a subroutine which is often passed constant arguments such as:

```

MVI C,3
CALL SUB1
:
MVI C,5
CALL SUB1
:
  
```

```

MVI C,7
CALL SUB1
:
  
```

SUB1:

By manipulating the return address you can save one byte per call as follows:

```

CALL SUB1C
DB 3 ;put constant in
: ;return location
CALL SUB1C
DB 5
:
CALL SUB1C
DB 7
:
SUB1C: XTHL ;[H,L]=return address
        MOV C,M ;fetch the constant
        INX H ;update return address
        XTHL ;restore the return
: ;address and [H,L]
SUB1:
  
```

This is not a useful technique in most cases, but it does give a good example of XTHL.

Next month, I'll do the long-awaited explanation of the boot and multi-precision arithmetic.

.....CONTRIBUTORS.....

Contributors to this month's issue of Computer Notes includes:

- H. Edward Roberts, President of MITS, Inc.
- David Bunnell, editor and President of The Agency, Inc.
- James B. Vice, Head Technical Writer, MITS, Inc.
- Bill Gates, Software Specialist, MITS, Inc.
- Dick Heiser, President of Arrowhead Computer Co.
- Barbara Simms, Altair Customer Service
- Keith Britton and Robert Mullen, People's Computer Company

Thomas W. Durston, Program Manager, Engineering, MITS, Inc.

Patrick N. Godding, Program Manager, Engineering, MITS, Inc.

Andrea Lewis, Technical Writer, Assistant Editor Computer Notes

Chris Ryland, Byte Magazine

Ron Santore, Altair User

James Runner, poet

Louis T. Milic, poet

# ALTAIR BASIC

**CLAIM:** Not just anybody's BASIC

**FACT:** Not just anybody's BASIC

BY: KEITH BRITTON  
ROBERT MULLEN

MITS does a lot of expensive advertising and their machine, while it has proved basically sound, has a number of problems requiring fixes - mostly evidence of hurried development. This has led to adverse comment and skepticism regarding their BASIC. MITS has done a lot wrong regarding the 8800 - but they have done a great deal that is right, too.... and received little credit for it. The Altair has proved fundamentally sound. That is to say the faults, though irritating, have cheap fixes that are effective. (People have designed *and sold* computers that did not work and could *not* be fixed.) But not only was it what was needed at the time, it was designed to be what was needed for the future. This shows commendable foresight on the part of Ed Roberts & crew, that is obvious. Less obvious, and something for which they deserve credit which they have not received, is the fact that they were willing to pay a considerable price to provide the hobbyist with things which he did not yet know he needed. The bus, for instance, has 100 lines. This cost money directly, more connections and more expensive sockets, and indirectly in that design freedom encourages competitors. *But this philosophy has provided the hobbyist with something open ended.* Their BASIC is similar but the features which render it open ended are sufficiently powerful that they amount to an advance in the development of the language.

BASIC (which is a registered trade mark of Dartmouth College) is a comparatively new language. It was developed because no existing language was suited to its application - an easy to learn language for instructional use by beginners on time shared computers. All languages are compromises and this one gave up a great deal for what it gained. Most professionals sneered at it because of its limitations but through the efforts of a small number of enthusiasts, (among them Bob Albrecht, who is still a member of the BASIC Standards Committee) the language became widely known. The more it was known, the more apparent became its unique merits. These were sufficiently important that BASIC may already have become the most widely known computer language, and is clearly destined to dominate the computer world over the next decade. Naturally, the race is on to incorporate into BASIC the things which were given up in the original compromise, resulting in a series of dialects of the language. Where does Altair BASIC fit in? At the moment, right out there in front of the field, but to explain why, let me back up a bit.

The first machines used machine code - which was fine for machines but miserable for humans. So that humans could generate machine code without losing their minds, a language was produced which allowed us to use phrases or mnemonics to assemble machine code, called, appropriately, ASSEMBLY LANGUAGE. This allowed the programmer to use every feature built into the machine and write programs which were the smallest and/or fastest possible. But assembly language was tedious, since every little step had to be defined, and since each new computer used different machine code, programs assembled for one machine would not run on another and the programmer had to learn a new language for each machine. To get over this new languages were developed which took much more general instructions, written in a form familiar to humans, and from them *compiled* the machine code necessary to carry them out. Thus were born FORTRAN and ALGOL, both math oriented, which accepted arithmetic/algebra *like* instructions.

Note the 'like'. While they were a great advance, both languages were still more adapted to the machine than to the human. Thus, they produced comparatively efficient code, but were difficult to use and very difficult to learn. Programs, however, had become portable to the point that they often needed only minor adaption for use on a different machine. Space requirements had risen sharply. The program to *compile* code was appreciably larger than that needed to *assemble* it, though the machine code was not much bigger. COBOL soon fixed that. COBOL was produced in an unsuccessful attempt to make a language suitable for humans, principally by the expedient of writing everything out in English. This conclusively demonstrated that the information content of 'plus' and '+' are the same; that a human is perfectly at ease with either; that such a compiler requires a horrendous amount of memory to produce even inefficient code - but contributed little else and can, for our purposes, be forgotten.

Now these latter languages were all compiler languages. A program was written in *source language* compiled into *machine code* and then run. This made it difficult to run more than one program at once, so queues developed and people got bad tempered - particularly when they had an error and had to start again from the back. This wasn't too bad in industry or business where there weren't many programmers per computer and they had sufficient experience not to make many mistakes, but in education where fifty students might be trying to get their program to run..... Educators also met another problem - programming languages were so difficult to learn that only those who were going to specialize in computer science had the time necessary.

Kemeny and Kurtz, brilliantly, solved this problem at Dartmouth College by inventing a new kind of language. Beginners All Purpose Symbolic Instruction Code was *interpretive*, not a compiler or an assembler. This meant that the BASIC source language program instead of being converted to machine code, was only slightly altered, being turned instead into a funny stuff called pseudo-code. When the time came to actually run the program, the *BASIC interpreter* looked at the first pseudo-code statement. It pulled it apart, *parsing it*, decided what operations were asked for, did them and then looked at the next pseudo-code statement (unless a jump was called for, of course), keeping going until it came to a statement which said 'end'. This had lots of disadvantages. It was deathly slow because the machine had to keep on working out what the statements meant. It was very limited in the things that it would allow a programmer to do and it needed a lot of memory because, when the program was run, both the program and the interpreter had to be in the machine. Despite this, it lived and prospered. The speed didn't matter too much because, even used inefficiently, computers are fast. The limitations were more of a problem but not critical because the language was used for education, and so did not need the sophistication of a business or scientific language. The memory requirement was no problem at all, because now lots of people could share the same space. Pseudo-code was easy to shift into and out of a machine and, so long as the machine remembered which statement it was supposed to tackle next, it could simply continue after a pause. And the pauses came about because the machine would now *time share*, sharing processor time among many users by working on one program, moving it out of the machine onto a disk memory, moving in another program and working on that, etc., etc. Alternatively, several programs could be in the same machine each in its own memory partition, all sharing the same interpreter and processor.

All this was very nice, but the true reason for the success of BASIC was that it suited people. They could quickly and easily learn to use it, and it was powerful enough to do interesting and useful things. But man, unfortunately, is a chronically dissatisfied animal. So then he wanted the speed he had given up and a lot of the features of the other languages to be built into BASIC - and there was no way he was going to give up the things he liked.

Well, the only direct way to speed in BASIC is to compile it instead of interpreting it. This can be done but then one loses one of the nicest traits of BASIC, the ease with which one can change the program. This is only possible because of pseudo-code. Since BASIC interprets one statement at a time, it doesn't mind a bit if you slip in an extra one or delete a few, but the slightest change in a compiled program requires that the whole thing be re-compiled. But the MITS software crew have an *indirect* way to blinding speed (or, more accurately, all the speed of which the processor is capable), they allow BASIC to jump back and forth to machine code subroutines. These can be written as such, assembled or compiled. So the Altair user can simply keep a library of such routines and use them as needed.

This has been done before, as has another of Altair BASIC's major features - the ability to talk out to user-supplied I/O devices, but not as simply and not as cheaply. A BASIC with the power of this one is a major achievement, there's a monstrous amount of work in it, and they have sold in the low kilobuck range for each copy. Some of the other features, particularly string and numeric matrix handling are certainly unusual and may actually be unique, but the user may notice them less than the general attention to detail evident in the programmers' approach. They, Bill Gates-Paul Allen-Monte Davidoff, must have spent literally hundreds of man-hours mulling over comparatively minor details - but details which sum to a degree of convenience for the user which will endear the dialect to those who are lucky enough to own an Altair. And, hopefully, the fact that MITS have done their BASIC so well will spur their competitors to improve the language yet more. Or, if they can't take a step forward, copy MITS's innovations and at least not take a step back.

All major programs must be expected to have bugs and the extent of the testing required to identify the more subtle ones is so great that it is more reasonable to test for the obvious ones and then wait for the users to find the others, updating software with fixes at intervals. Altair BASIC version 2.0 had a serious problem in that a jump out of a FOR.... NEXT loop left garbage on the stack. Do this too often and the stack would grow relentlessly down from high memory until it ate the program. This has been fixed in version 3.0, according to Paul Allen, but we have not yet received a copy to pull apart. We also hear that there have been a number of other improvements and fixes. Presumably, someone punished it a lot harder than we did, since we found little else to criticize, but we did find one problem and it has unfortunately been left in version 3.0. If you print a value less than 0.1, it appears in scientific notation. So, if you are printing out dollars and cents, and you have less than 10¢, you suddenly see some thing like - 3.00000E-02.

MITS has frozen development at the 3.0 version, except for bug fixing, and are close to releasing the first versions of their extended BASIC. And THAT has things that will make even more difference to the way BASIC will grow over the next few years. The worst single item BASIC gave up, relative to the other languages, was flexibility in precision, adopting 6 digit floating point for all variables. With floating point even incrementing by one is slow, and business use requires more precision. Altair Extended Basic is to have integer, 6 digit and extended precision variables. . . plus a lot more.

BASIC language was chosen for the Altair 8800 because it is the *easiest language to learn* and because it can be used for an *infinite number of applications*. Literally hundreds of thousands of BASIC programs have been written and are in the public domain. These programs include accounting programs, business programs, scientific programs, educational programs, game programs, engineering programs, and much more.

Altair BASIC is an *interactive language*. This means that you get immediate answers and you can use your Altair as a super programmable calculator as well as for writing complicated programs.

## 8K BASIC Features

Altair 8K BASIC leaves approximately 2K bytes in an 8K Altair for programming which can also be increased by deleting the math functions. This BASIC is the same as the 4K BASIC only with **4 additional statements** [ON...GOTO, ON...GOSUB, OUT, DEF], **1 additional command** [CONT] and **8 additional functions** [COS, LOG, EXP, TAN, ATN, INP, FRE, POS]. Other additional features include *multi-dimensional arrays* for both strings and numbers, AND, OR, NOT

▀▀ I've seen and used other BASICs, but byte-for-byte, Altair is the most powerful BASIC I've seen. I'm particularly impressed with the n-dimensional arrays (and for strings too!), machine level I/O, and machine language 'function' features. The level of your documentation is, for me, though the high point. Sections for those who know nothing and sections for those who know a lot, plus sections that 'normal' people can read and understand. ▀▀

**J. Scott Williams**  
Bellingham, Washington

Altair BASIC was written as efficiently as possible to allow for the *maximum number of features in the minimum amount of memory*. You can order one of three Altair BASICs: 4K BASIC—designed to run in an Altair 8800 with as little as 4K of memory, 8K BASIC, or EXTENDED BASIC (12K). Each of these BASICs allows you to have *multiple statements per line* (a memory saving feature), and each of them is capable of executing *700 floating point additions per second!*

The 8K BASIC and EXTENDED BASIC have *multi-dimensional arrays* for both strings and numbers. This is particularly useful for applications requiring lists of names or numbers such as accounting programs, inventory programs, mailing lists, etc.

The 8K BASIC and EXTENDED BASIC also have an OUT and corresponding INP statement that allows you to use your Altair 8800 *control low speed devices* such as drill presses, lathes, stepping motors, model trains, model airplanes, alarms, heating systems, home entertainment systems, etc.

Altair BASIC comes with complete documentation including a copy of "My Computer Likes Me When I Speak in BASIC" by Bob Albrecht, a beginner's BASIC text.

**Never before has such a powerful BASIC language been marketed at such low prices!**

## 4K BASIC Features

Altair 4K BASIC leaves approximately 750 bytes in a 4K Altair for programming which can be increased by deleting the math functions. This powerful BASIC has **16 statements** [IF...THEN, GOTO, GOSUB, RETURN, FOR, NEXT, READ, INPUT, END, DATA, LET, DIM, REM, RESTOR, PRINT, and STOP] in addition to 4 commands [LIST, RUN, CLEAR, SCRATCH] and **6 functions** [RND, SQR, SIN, ABS, INT and SGN]. Other features include: *direct execution* of any statement except INPUT; an "@" symbol that deletes a whole line and a "←" that deletes the last character; *two-character error code* and line number printed when error occurs; *Control C* which is used to interrupt a program; *maximum line number of 65,535*; and all results calculated to at least six decimal digits of precision.

operators that can be used in IF statements or formulas, *strings* with a maximum length of 255 characters, *string concatenation* (A\$ = B\$) and the following string functions: LEN, ASC, CHAR\$, RIGHT\$, LEFT\$, MID\$, STR\$, and VAL.

## EXTENDED BASIC

Altair EXTENDED BASIC is the same as 8K BASIC with the addition of *double precision arithmetic*, PRINT USING and *disk file I/O*. A minimum of 12K memory is required to support EXTENDED BASIC.

Other Altair 8800 software includes a Disk Operating System, assembler, text editor, and system monitor. Altair users also have access to the *Altair Library*, which contains a large number of useful programs.

## SOFTWARE PRICES:

Altair 4K BASIC .....	\$150
Purchasers of an Altair 8800, 4K of Altair memory, and an Altair I/O board .....	\$ 60
Altair 8K BASIC .....	\$200
Purchasers of an Altair 8800, 8K of Altair memory, and an Altair I/O board .....	\$ 75
Altair Extended BASIC .....	\$350
Purchasers of an Altair 8800, 12K of Altair memory, and an Altair I/O Board .....	\$150
Altair PACKAGE ONE (assembler, text editor, system monitor) .....	\$175
Purchasers of an Altair 8800, 8K of Altair memory, and an Altair I/O board .....	\$ 30
Altair Disk Operating System .....	\$500
Purchasers of an Altair 8800, 12K of Altair memory, Altair I/O and Altair Floppy Disk .....	\$150

**Note: When ordering software, specify paper tape or cassette tape.**

# PCC

IS THE

## PEOPLE'S COMPUTER COMPANY

The People's Computer Company

is a newspaper...  
about having fun with computers  
and learning how to use computers  
and how to buy a minicomputer  
for yourself or your school  
and books...films...and tools  
of the future



## MUSIC OF A SORT

Steven Dompier

I received my ALTAIR 8800 in the mail at 10 am, and 30 hours later it was up and running...

The radio was picking up the switching noise of the 8800! I tried some other programs to see what they sounded like, and after about 8 hours of messing around I had myself a program that could produce musical tones and actually make music; of a sort. (Or any other program you have!)

The closest sheet of music that I could find was "THE FOOL ON THE HILL", by the Beatles, so I translated it into OCTAL code, picked up the Altair, and headed down to Menlo Park for the 3rd meeting of the 'BAACUG-HBCC'.\* I thought everyone there should see just what a computer can do!

\* Initialization of "BAY AREA AMATEUR COMPUTER USERS GROUP - HOME BREW COMPUTER CLUB"

**Next Issue** Unusual and exotic devices for your home, school or personal computer. Beginner's info on computer arithmetic and machine language programming. More BUILD YOUR OWN BASIC. And of course, games!

We are a bunch of people (computer people, dragons, and other creatures) who are working together, with the idea that computers should serve people, not rule them.

We do this by publishing the PCC newspaper 6 times a year. By printing letters, writing articles and providing information people want but can't find elsewhere, we seek to serve the information needs of people interested in the humanistic and recreative uses of computers.



### Whence the Dragon?

Once upon a time there was a dragon. It lived happily in the land for a while, spreading alarm and disquiet and doing all those things that dragons do. And then it laid an egg.

Way back in the dim depths of the past, fifteen years or so ago, a very straight young computer man used to work for a very straight organization up there in the great city in the sky—five thousand feet up in Denver, to be precise. His name was Bob Albrecht and the flaws in his character were already becoming patent. He took to teaching youngsters aged 14 or so about computers, an heretical notion for computers were expensive and only for mathematicians, certified data processors and similar lordly beings. The corruption of these unsuspecting minds finally tipped our hitherto merely illbalanced hero into the monomania from which he has never recovered. Aroused to the vulnerability of an unwary public, he lumbered forth, vampires byte at the ready.

The scene now shifts some years and miles, to the warm, sunny rocks of California, a proper hatching ground for a dragon. The nursemaids and attendants of the egg began to coalesce. A dragon's nest was prepared with the formation of Dymax. Ostensibly a publishing house, it became, in reality, a center of subversion where computer time was wasted on mere fun. With the periodic presence of a PDP8L the seed was sown and the more perceptive (speaking with hindsight) were clearly able to discern the chip chip chip of a hatching dragon.

Then one day the egg split asunder. A Jolly DEC Giant passed by Dymax, met Bob and gestured to the PDP8L. "Take it," he said, with characteristic generosity, "It's not mine anyway." Bob, who had been at the gin again, stared at this magnificent gift until there crept into his befuddled brain the realization that he had, at last, a computer of his very own, to do with as he pleased—and thus was born a dragon.

Dymax moved to Menlo Park, space was contributed, and slowly a computer center grew, to the delight of hundreds of the local school children. Dymax was a profit making corporation (and making enough to support all this) so it was decided to create another Thing, but this time a Good Thing, called Peoples Computer Company, which would publish a newspaper, run the center and do diverse other things allowable under the general non profit corporation laws of the State of California. So Bob and Mary Jo Albrecht, the Jolly DEC Giant (Val Skalabrin), Bob Kahn, Marc LeBrun, Dennis Allison and Lois B. became the first board of directors, and, in 1973, PCC was incorporated.

Even before this, the dragon, which was mature to the point of precocity, had laid an egg, though no one noticed until it too was ready to hatch. Dymax, finding the center too much for its limited space, had rented the store next door and called it Peoples Computer Center, or PCC, to avoid confusion with Peoples Computer Company, or PCC. PCC (Company) divided into three divisions; the newspaper, the bookstore and the center. Bob ran the newspaper while Lois ran the bookstore. It was decided that the Center (or PCC) should be separated from PCC (the Company) and form a new corporation, Peoples Computer Center. (This new dragon has since been renamed Community Computer Center).

Simultaneously, there came the great flood. The skies opened and it was discovered that the newly installed roof was not all it had been cracked up to be. Over Dymax and the newspaper there was a split 36 feet long, and over the Center it was nearly as bad. The roofer, it appeared, did not roof in the rain, and could not stop the leaks in the dry, so PCC the old dragon, his fire steaming and almost out, moved to a new home in downtown Menlo Park, selected partly for the wallpaper and mostly for its proximity to the Village Host, which sells draught Anchor Steam Beer and fine Pizzas.

#### FACT:

The OLD DRAGON, Peoples Computer Company, is a non profit corporation. PCC is a low budget operation, principally because few people have been paid and the payment has been minimal. PCC is supported by the newspaper subscriptions, the revenue on books sold through the bookstore and much good will. Equipment donated to PCC is used educationally and may be loaned to other non profit organizations, with the requirement that any software developed using it be placed in the public domain. Profits from PCC's game book, What To Do After You Hit Return, purchased through the bookstore are split two ways. Half goes to CCcenter (the new dragon) and the other half is placed in a special fund to buy hardware that may be used to help start other centers.



# Letters to the Editor

## Pittsburgh Area Computer Club

Dear David,

Just received Issue Four of the Computer Notes and would like to provide a few comments. First though, I want to thank you for your letter of Sept. 10, and the priority effort in getting me the updates. For some reason I get this intuitive feeling that I may not be on your manual update file. Back in April there was some confusion in one order I placed for some memory along with the update order. Anyway, if this confusion caused me to be dropped from the list, I need to be put back on it. (Ref. Invoice No. 29691 date 3/14/75--William C. Fuller).

Now, related to issue 4. I think you wrote a damn good editorial. I don't necessarily agree with Crain's position about the stifling of business by government and consumer groups, and that managers are no longer able to make innovative, creative decisions. I think MITS proved it just ain't so! At least in electronics.

I believe the problem goes deeper than that, and involves social, psychological, political, economic, etc. shocks that have jarred the "collective unconscious" of the people in the past few years. The song goes something like "...where have all the heroes gone..."

But back to the issue at hand. I think that your analogy of the musician's royalty is valid in applying it to privately developed software. Since software isn't my profession, I have no way of knowing if your prices (royalties) are reasonable or not, so I voice no opinion on it.

Related to your software license agreement. People I have talked to are confused by it, as I admit I am. Some clarification was made about it in a past issue of the Notes, but confusion still exists.

One question I haven't seen asked or really answered is about the application of the license agreement to user developed programs not in MITS BASIC, but in the 8080 instruction set. For example, are programs available from MITS like 521751, 72153 and 721751 supplied to MITS by users under the software license agreement?

Anyway, I think your editorial was appropriate. If MITS can maintain its openness as it has done in the past, then the innovativeness may not be just in the electronics field.

Congratulations on the appearance of the 680...I'm tempted!

I haven't really had time to go through the BASIC language documentation, but a first glance evaluation is that yes, it is a vast improvement over the operator's manual.

By the way, the Texas Computer Club is now The Computer Hobbyist Group of North Texas. Our basic policy is no computer bias, all types welcome. Presently members have 8 Altairs of various configurations and 9 other varieties based upon the 8008, 6800, Mostek 5065, TI APC 980A, PDP-8, etc. We are presently scheduling outside speakers (equipment), so if you have someone who might want to make a presentation on the new 680 around Nov. or Dec. let me know. A club survey made at the last two meetings indicates at least 5 are interested in the Altair. Membership is somewhere between 50 and 75.

Pax, Bill Fuller  
Grand Prairie, TX

Dear Bill,

I checked out your Update and you are still on the list. The next mailing should be in about three weeks.

The programs in the Altair Users Library are part of the Public Domain and once you have a copy you can do whatever you want with it. If you should happen to reprint a program, be kind enough to give credit to the author.

Thanks for the nice comments and good luck to the Texas Computer Club.

--D.B.

### ALTAIR USERS

Paul W. Schuler, III  
1100 Mansion Ave.  
Collingswood, NJ 08108

H. Eugene Latham  
Box 187  
616 AC&W SQ  
APO, NY 09277  
Wasserkuppe Air Station  
Germany

Leonard J. Paul  
PO Box 229  
Santa Monica, CA 90406  
805-497-1975

Samuel Seiber  
759 N. Skyline  
Idaho Falls, ID 83401

Donald C. Schertz  
764 Toyon Dr.  
Monterey, CA 93940  
home 408-375-6186  
office 408-646-2982

Dear Fellow Computer Phreaks:

Just a quick note to let you know that hobby computing and computers are alive and well in Pittsburgh. As you can see from the logo the name of the organization is the PITTSBURGH AREA COMPUTER CLUB. We had our first meeting on October 8, 1975 and even though the weather was inclement we had 15 people in attendance. In addition there is definite interest from at least that many more! Not bad for a beginning.

The name of our acting president is Eric Liber and the name of the secretary/treasurer is Fred Kitman. We can be reached at 400 Smithfield St., Pittsburgh, PA 15222 or at the following telephone numbers:

Day: 412-391-3800  
Nite: 412-276-6546

The members of the club have a total of 5 computers with the following breakdown:

Altairs	--	2
PDP/11	--	1
Wang	--	1
8008	--	1

Not a bad start!

We would appreciate the inclusion of our organization's name in your publication to inform other hobbyists in this area of our existence.

Thank you for your help,

Eric Liber  
Fred Kitman

### LAST MINUTE ADDITIONS TO "LOCAL USERS GROUPS" (continued from page 11)

Anyone interested in forming a users group in the BOSTON area, contact:  
John Vullo  
21 Sunset Ave.  
N. Reading, MA 01864  
617-664-4271

PITTSBURGH AREA Computer Club  
Fred Kitman  
400 Smithfield St.  
Pittsburg, PA 15222  
412-391-3800  
or  
Eric Liber  
412-276-6546

# ARROWHEAD TIPS

The following tips are on construction of the Altair 8800. Page numbers refer to the Assembly Manual.

Page 11  
(D/C Board): Capacitor C7 has been changed several times; you may find change notices referring to various stages of this change process, as well as extra parts. C7 was changed from .001 MFD to .0047 MFD, then to .0068 MFD. Now, the absolute last final ultimate change (as of August 10th!) makes C7 = .01 MFD and changes C8 from .01 to .1 MFD.

Page 18  
(D/C Board): Don't bolt the printed circuit board to the sub-panel; the switches will hold it fine. The switches come with extra mounting nuts and extra guide washers; you can safely throw away all this extra hardware - you don't need it. Mount the switches as shown with only one nut each for best results.

Page 39  
(CPU board): The wafer connector is about 5% too large to fit the board (or you might say the holes in the board are too close). To prevent the connector plug from arching, clamp it down flat while soldering it on. If you can't clamp it, then try cutting it with a hacksaw into two 4-pin connectors.

(Checkout): After turning the computer on you should reset it. To reset the computer you have to hold the stop switch raised while raising the reset switch. Release the reset switch first. (No, we don't know why, but it's traditional!)

Pages 40-46  
(1K Memory): Until May, only the 1K memory board was available, and most systems were ordered with 256 words of memory on one of these boards. Now that the 2K and 4K memories are available, it isn't sensible to require some 1K boards in every system, but the instructions are still embedded in the CPU manual. CPU kits don't have any memory in them.

Page 50  
(P/S Board): The bridge rectifier seems to cause more than its share of problems. Be sure the leads are clean - several of us have found that solder won't wet the leads, and it's a mess to try to clean the partially soldered assembly. Run the leads through some alcohol and/or steel wool before installing the bridge. The spacer-and-washer arrangement on Page 51 is a jig to get the bridge flat at the right position it will later be bolted directly to the chassis.

Page 58  
(Chassis): MITS doesn't ask you to cut wires to close tolerances. If you follow the instructions here without trimming transformer leads, and use all the #20 wire, you'll have long loops of slack. This is good for allowing boards, etc. to be moved without breaking wires, but you may want to install the terminal lugs after consulting the wiring diagrams on Pages 59 and 62.

Page 66  
(Mother board): The way the instructions spell it out, you'll have a slack loop of cable. Hold everything in place on the chassis to see the actual length required. We are including a sorted list of wires to help you check your progress. Using masking tape, group the wires in decades after protecting them with a cable clamp. Then, install them on the motherboard, by decades; 50's, 60's, 70's, 20's, 80's, 30's, 90's, and finally 40's.

Install the cable clamps by bolting them to the printed circuit board only. If you put screws through the sub-panel, then the dress-panel won't fit flush against it and you won't be able to screw the chassis into the case!

Page 68  
(Expander Boards): The card guides are maybe sorta optional. They look nice, but they really aren't required to hold the boards in place - the edge connectors are plenty strong for that.

Gary Hullquist, M.D.  
248 Sunset Dr.  
(or Box 448  
Professional Bldg.)  
Mountain City, TN 37683

Glenn D. Nelson  
Geology Dept.  
Box 1846  
Brown University  
Providence, RI 02912

Page 74  
(CPU Chip): Many people advocate postponing installation of the CPU chip until after the regulator and zener diodes on the CPU have been tested.

Page 77  
(Nameplate): This beauty gets a lot of criticism: "Mine was off-color, kinda pinkish." That's a sticky plastic cover to protect it until after you've installed it. Peel the covering off afterward. The white lettering on the dress-panel can be chipped off by hard use. If you decide to protect it with clear acrylic spray, use a matte-finish product. Ours looks funny with a glossy krylon finish.



ARROWHEAD COMPUTER COMPANY  
**THE COMPUTER STORE**  
11656 WEST PICO BOULEVARD  
LOS ANGELES, CALIFORNIA 90064  
(213) 478-3168

## DATA/CONTROL BOARD CONNECTIONS TO SYSTEM BUS

### ORGANIZED BY DECADE

0's	10's	20's	30's	40's
		20	30	41
		21	31	42
		24	32	43
		26	33	44
		27	34	45
		28	37	46 (1!)
		29	39	47
				48
50's	60's	70's	80's	90's
53	68	70	80	91
54	69	71	81	92
		72	82	93
		75	83	94
		76	84	95
		77	85	96
		78	86	97
		79	87	98
				99



### Altair User's Group

In the words of *The Agency*, MITS' advertising subsidiary, "The Altair User's Group is quite possibly the largest hobbyist organization in the world. It is both a means of communication among Altair users and a method of building a comprehensive library of Altair programs ... among other benefits, you will receive a subscription to the monthly publication *Computer Notes*, which contains complete update information on Altair hardware and software developments, programming tips, general computer articles and other useful information." And that's a pretty fair statement, judging from the August 75 issue of *Computer Notes*.

Edited by Dave Bunnell, the head of *The Agency*, *Computer Notes* covers most items of interest to Altair users. The August edition is headlined "WORLD'S FIRST COMPUTER STORE," featuring an article on The Computer Store, located in West Los Angeles, which is apparently the first retail store solely for computers and computer supplies (shades of Arthur's

Information Parlor). The store sells Altairs over the counter of course, and functions as a general computer hobbyist gathering place and information center. *Computer Notes* also features a travelogue of the MITS-MOBILE Altair Caravan, which toured the Southeast during August and September.

In his editorial, Dave Bunnell leads a good deal of rumors to their proper resting places, covering "off brand peripherals, memory cards, etc.", the false rumors concerning "less than full spec" Intel 8080 chips, software agreement technicalities, MITS' development of a Motorola 6800 system, and delivery complaints. Dave precedes all this with the explanation, "One point that has gotten us good press in a number of publications is that we try and level with our customers." From Dave's straightforward presentation, I think they deserve another "good press."

There are no surprises in the rest of the newsletter, with its "Altair Service Dept.," "Letters to the Editor," "HARDWARE," and "SOFTWARE" sections. In "HARDWARE" Tom Durston and Paul van Baalen deliver some ACR (Audio Cassette Recording) hardware alignment updates, fixes for 8800 problems, some "Boo Boos," various maintenance techniques and hardware specifications.

The "SOFTWARE" section contains "Software Contest Winners Announced" by Bill Gates; "Q & A" on the "Monitor, Editor and Assembler" by Paul Wasmund, the author of these software components; "General Software" by Paul Allen, the director of MITS' Software Department, answers various questions concerning MITS' software performance, policies and

future plans; Monte Davidoff, one of the authors of Altair BASIC, illustrates the string handling and recursive subroutine capabilities of BASIC in "Fun with Altair BASIC."

Monte does a good job of describing these interesting possibilities of BASIC, even though such applications can be a bit strenuous for both the reader and the BASIC language itself, as witnessed by Monte's closing comment on factorial computation: "If confusion still prevails, do not worry about it."

Bill Gates, in his article "Software Hints for 8800," gives just that. Mentioning

some of the reasons for the 8080's power, Bill goes on to discuss binary coded decimal (BCD) arithmetic, giving a sample routine for conversion from BCD to binary. After a short lesson on special short branching ("skip") techniques the reader is presented with some neat but fairly general stack usage tips. Bill doesn't waste words in his article; a rank beginner would probably be left far behind by these last concise hints.

In a short review, Dave Bunnell gives BYTE an A+ for format and an A- for content ("They have something to work for."). Although it sounds a little

like an ad for BYTE, Dave is pleasantly positive in his remarks.

I have a couple of complaints. The format of *Computer Notes* is imaginatively done, but at some points the text is hard to follow - you don't know what to read next. [But then, it is only fair to point out that BYTE may suffer the same malaise in one or two places ... Carl] Also, a lot is assumed on the part of the reader. I realize that any such newsletter can't function as a tutorial publication (that is a major part of BYTE's job) but the "HARDWARE" and "SOFTWARE" sections would certainly throw any real beginner. But I judge too harshly: *Computer Notes* is truly a good bulletin for the users of Altair equipment.

To paraphrase Dave's closing question in his BYTE review, is the Altair User's Group something no Altair computer hobbyist should be without? At this point in time, I would have to say that he has no choice since membership is automatic for Altair owners. Should the Altair computer hobbyist be thankful for *Computer Notes* and other benefits of the User's Group? I would have to say yes.

... Chris Ryland

Reprinted from the November issue of *Byte*. Thanks Chris.

### DELIVERY DATES:

The following is a revised list of delivery dates on products appearing in the Altair Brochure, but not yet released. Our product development department is somewhat overloaded at this point, so please bear with us.

PRODUCT	DATE AVAILABLE
COMTER II	January 1976
88-DISK, 88-DCDD	December 1975
88-DMAC, DMAE, DMAI-no date set	
88-PPC	January 1976
88-PMC	December 1975
88-PROM	November 1975
88-VI	December 1975
88-RTC	December 1975
88-32DU	discontinued
88-KB	discontinued
88-ACC	no date set
88-CC	no date set
CRT Terminal	no date set

Extended BASIC	December 1975
Extended BASIC/DOS	December 1975
DOS	December 1975

## What's Going On In Here

There have been changes in our personnel which will affect our customers: Mr. Paul Van Baalen is now Marketing Vice President and Mr. Harvey Lee is Repair Department Manager. Please refer technical questions to our Repair Department and cost and delivery inquiries to Marketing.

## What's Going On Out There

Those of us in marketing and engineering are very interested in what our customers (present and future) would like to see in the way of new products, ideas, iet. The best way for us to know is to drop a post card to Barbara Sims telling us what your ideas are.

For instance we have had some requests for control features for the 88-ACR. Also, we are interested to know if there is a need for a standardized low cost tape recording system, and what standard should be adopted. NOTE: The "HIT" system (Popular Electronics, Sept 1975) is one alternative, but has the disadvantage of not being compatible with serial data transmission format and rates.

Edward Enck  
106 Edford St.  
New York, NY 10014

Joseph T. Finneran  
11 Montrose Ave.  
Pinehurst, MA 01866

Richard R. Kenyon  
18609-A Del Rio Rd.  
Cerritos, CA 90701

Michael Jackson WN4JJI  
Sky Park Rd.  
R#7  
Florence, AL 35630

Wolfgang Luhn  
c/o CCT  
1001 North American Way  
Miami, FL 33132

Gordon L. Quivey  
Natlandsfjellet 207  
5030 Landas  
Norway

Bruce Sidlinger  
Recognition Equip., Inc.  
PO Box 24  
Garland TX 75040

Steve Caldara  
36 Theo.Roosevelt Dr. (home)  
Blauvelt, NY 10913  
965 E. State St. (school)  
Ithaca, NY 14850

Ian D. Kettleborough  
1108 B. Holik Dr.  
College Station, TX 77840

James D. Larson  
Inn Lane, Apt. 206  
Oak Ridge, TN 37830