Volume 1 Number 8

As promised, this issue of TCH starts two new series of articles, one on an IMP-16 system and the other on interfacing to the Altair 8800. Both of these series will include detailed discussions and examples of known good circuits. Upcoming issues of TCH will also see a new series, this one on a floppy disk interface that is buffered and can be hooked to any processor. MOS Technology has created quite a stir with their 6502 (see article inside) and we at TCH would like to know if readers are interested in a series about it.

There is a recording standards conference sponsered by BYTE Magazine coming up this month in Kansas City. This will be the first effort in the hobbyist computer industry to get the different manufacturers together along with users. TCH will be represented at the conference by Hal Chamberlin and Richard Smith. With luck this will be the beginning of a new era for the hobbyist. This conference could also lead to some standardization of interfaces in general, thus allowing one manufacturer's peripherals to plug into another's system.

This issue also contains two other items that should be mentioned. In the letters column there is a discussion of some errors in the listing of the 8080 code for the cassette interface. The second item is a listing of clubs. This is something many people have asked about since TCH ceased printing the listing of our subscribers.

One final matter, TCH special offerings. TCH is now sold out of both 2102's and relays for the cassette board. See note inside if you still need relays. The cassette boards and ROM programming offer are still good. As always, all back issues are available. Also, with this issue TCH starts offering a universal wire wrapping card and 5 volt regulator kits for the card, see the Altair interfacing article inside.

INTERFACING THE ALTAIR 8800 PART 1

by Hal Chamberlin

This is the first of what will probably be a long series of articles on direct interfacing of peripherals and memory to the ALTAIR 8800 parallel bus. Additional topics to be covered include direct memory access interfacing, a simple non-vectored interrupt system, noise control measures for large systems, and bus and power supply loading considerations. All circuits and techniques will be tested on a near minimal Altair 8800 with 1K of original memory. Most of the concepts are equally applicable to other just announced kits (see the new products column).

Although MITS is generally responsive to user's needs, particularly in software, we felt that there was a gap in the technical information on the bus system and interfacing. With nearly 10,000 machines in the field, we think that this will be particularly valuable information for our readers. Please keep in mind that we are not trying to compete with MITS or suggest that users not buy their boards. If a standard I/O board will do the job in a straightforward manner, by all means purchase one -- it will probably be less trouble and expense in the long run. On the other hand, many people have their own whizbang golly-gee I/O device to interface which is either impractical or too expensive to do with standard I/O interface cards. Music synthesizer control or graphics display interfaces are excellent examples. Additionally, an infinite variety of multi-function custom interfaces are possible with just enough of a particular resource to satisfy the application. The Altair is a great machine and full realization of its I/O system makes it even greater!

By far the best way to build a custom interface is on a so-called "wirewrap card" with wirewrap IC sockets and wrapped interconnections. Unfortunately, wirewrapping has a "high cost" stigma that has prevented many hobbyists from considering it. A very low cost wrapping tool ($1.70) made by Cambion and available through Solid State Systems exists however and is generally satisfactory for hobbyist use. If extensive custom interfacing is anticipated, Godbout has a relatively inexpensive ($47.50) motor driven wrapping tool. Unwrapping tools are outrageous for what they are ($5.58 from S.S.S.) but the time saved in making changes and correcting errors make them worth it. Wire for wrapping tends to be expensive also (1.5¢ per foot) but fortunately it goes a long way. Kynar insulation is easier to strip from the #30 solid conductor than is Teflon. The scissors type of stripper very carefully adjusted is OK for stripping Kynar insulation. It might be wise to buy such a stripper, adjust it once, and use it only for wirewrap wire. Wirewrap IC sockets are available from many sources with Solid State Systems being particularly good during one of their sales.

Wire wrap cards are now available from TCH for $20 and are made by the same PC shop that made our cassette borads. The wirewrap card is the same size as the other Altair cards and has a 100 contact edge connector for plugging into the bus. Also provided are power and ground connections for TTL IC's, numerous bypass capacitor pad sets, and two locations for 7805 regulators to supply regulated +5 from the unregulated bus +8. A photo of the artwork and an IC placement form are in the appendix. As a service, TCH also has a 5 volt regulator kit consisting of a 7805, heatsink, and two 47uF 10 volt bypass capacitors for $2.50 when purchased with a board. Connections between the custom I/O device and the interface card are made with DIP plugs into a row of 6 socket positions at the top of the board. Typically a short cable would go from the interface board to a rugged socket mounted in the holes provided on the Altair's back panel.

The only major problem with wirewrap boards is that they cannot be plugged into consecutive sockets since the spacing is only 3/4 inch. Alternate socket positions will have to be used making separate purchase of motherboards and 100 pin sockets advisable.

Now that the preliminaries are out of the way, lets get down to business with a discussion of system loading considerations. The Altair has three system resources that are subject to overloading. These are the I/O address space, the built-in power supply, and the parallel bus itself. We will take an objective, engineering approach in recommendations regarding allowable loading of these resources.

Since the 8080 CPU has the capability of addressing 256 input and output devices, there is generally no problem in having enough addresses to go around as there would be in an 8008 based system. The only real concern is that the user's custom interface addresses not interfere with any standard interface addresses. This is particularly true if one ever plans to use standard software or distribute his own software. We have not seen a comprehensive list of preferred device addresses for MITS interfaces (jumpers select the actual address) but the few that their software uses are all low addresses. One would probably be safe in assigning custom interface addresses in ascending order starting from 200 octal.

The internal power supplies have the following stated capacities: +8 unregulated at 8 amps, +16 unregulated at .8 amps, and -16 unregulated at .3 amps. All of these are connected to bus lines and are available for use by any plugin card. The +8 volt supply is regulated down to +5 volts on each card to power the TTL logic. This supply is the one most likely to be overloaded. The +16 supply is used primarily by the 8080 CPU and the 4K dynamic memory boards, as well as by op-amps on some standard interfaces. The -16 supply is very small and is used only for substrate bias on the 8080 and 4K chips (negliable drain) and to power op-amps. This supply may be overloaded if a large number of 1702 or 5203 PROMS are used on custom boards.

The primary guideline on power loading is don't overload the supplies. Since heat generated in the transformers is proportional to the square of the current drawn, even a minor overload will result in gross overheating. MITS's approach to preventing +8 power supply overload is to specify that no board is allowed to draw more than .5 amp. Thus a fully stuffed system with 16 cards will not draw more than 8 amps. Considerations on the + and -16 volt supplies are not as straightforward but the basic idea is to insure that any sensible combination of up to 16 cards be guarenteed not to overload any power supplies. This was undoubtedly a wise move considering the large number of non-technical people buying Altairs.

When custom interfaces are designed or alternate source interfaces are purchased, it is the user's resposibility to insure that no power supplies are overloaded by his intended configuration. On custom interfaces we would generally recommend waiving the "half amp" rule and try to cram as many different interfaces as possible onto each board. Besides saving space and cost, one can actually get more I/O function per amp! The reason for this is that the 8T97 bus drivers are real power hogs (65 MA typical) and every separate board will likely need a few.

CONTINUED ON PAGE 4

L E T T E R S

Gentlemen,

I have just completed a satisfactory one page write and read to/from tape using the TCH cassette interface and associated software. I would like to point out a couple of errors I have discovered in the program. Specifically the address specified for the jump line 177515 of the "Tape Read Record Routine" should read 107 instead of 042 and also the register specified for MOV on line 177512 should be C, not A. (change instruction to 161 from 167)

In addition you should indicate that the ROM software is based on the assumption of normal (uninverted) input to the data buss and replacement of IC 16 and 17 with non-inverting buffers or gates. As it is the program will not work with the interface if built per the schematic using the 7403's specified for IC 16 and IC 17. I finally did stumble across the suggestion of the DM8094 in the hardware description, but only on the third reading. No mention of this is made in the software description.

Also, I wonder if you have thought about using some sort of shift-register approach for the DID recognition on read. As it is, a lost bit on the leading zeroes can cause complete loss of sync with resulting loss of the record.

Nevertheless, I'm quite impressed with the overall product. Hope to be able to use it as an inexpensive replacement for a disk system. Will need to go to a deck such as the Phi-Deck with remote rewind, etc. for that. Wonder if you will be looking into any special prices on those. Also wonder about your thoughts on record-playback electronics for them. I've looked at the control electronics and that aspect seems quite good; the record-playback bugs me though.

Thanks again for a darn good job on the cassette system so far. Keep up the good work.

Fred LaPlante

Several readers caught the errors you mentioned in the 8080 version of the cassette software, the result of hand translation of known good 8008 code into untried 8080 code. The data ID recognition routine does indeed simulate a 16 bit shift register. It should tolerate all sorts of garbage at the beginning of a record as long as the DID itself is intact. TCH has two Phi-Decks and is working on a high-performance interface for them.

Gentlemen:

You ought to be ashamed of yourselves! Your 8080 cassette software had 4 bugs in it, 2 of them fatal. Address 177512 is 167, MOV M,A must be 161, MOV M,C. Address 177515 is 302:042:377 must be 302:107:377.

Also in the read data bit routine, the JZ and JNZ on the clock bit should be interchanged to be consistant with the flow chart and comments. 177661 is 302:255:377 should be 312:255:377 and 177670 is 312:264:377 should be 302:264:377. These are the only bugs I've found so far -I hope they are just typing errors, but that "MOV M,A" looks suspicious.

I might be starting some undergrad research here at UW, and I'm going to stick with microcomputers and peripherals. If you folks can suggest some interesting projects, either hardware or software, I'd appreciate it.

P.S. I hope you haven't programmed any 8080 ROM's yet!

Paul Gumerman

We always knew those programmers at the University of Waterloo were sharp (WATFOR, WATFIV, SPITBOL, PL/C). Fortunately, these errors were corrected in the ROM's that have been processed but a couple with incorrect load address and/or stack address may have escaped before a bug in our ROM customizer program was discovered. Interesting note, we have not come across any bad PROMS yet. As for projects, why not build a graphics display and experiment with software for automated drafting using cassettes for drawing storage.

Sirs:

An aquaintance was kind enough to loan me a few of the recent TCH issues, and I appreciated very much the informative articles on a Cassette Standard. Having personally traversed the problems involved in cassette interface design, I am painfully aware of some of the difficulties, and I appreciated finding out why my various designs failed. Two comments I should like to make with respect to your proposed standard:

1. One system requirement for the wide acceptance of a standard (aside from those propoqed with economic clout, a la IBM) is that it admit to a variety of implementations. In particular, not all users will be satisfied with the particular hardware/software mix defined by your circuit; some, like myself, prefer to minimize the external hardware at all costs when software can do the job; others wish to minimize the software at the expense of extra hardware (within reason). With the exception noted under item 2 below, this design does admit to a wide variety of implementations because of tl. basic simplicity of the format. In fact, I intend to alter my own cassette design to be compatible.

2. I was absolutely astounded that any serious attempt at proposing a standard should so blatently ignore one of the few widespread existing standards in the field of the serialization of parallel data, that is, that the least significant bit is transmitted first. If it were not for the facts outlined in item 1 above, this would be relatively unimportant; however since in this case it is not the hardware that does the serial-to-parallel and parallel-to-serial conversion, there is the non-trivial factor of user and programmer experience to consider also (in systems with other serial I/O, it is convenient to use common subroutines for some of the common functions such as serial conversion, CRC conversion, etc.). The real kicker is this: with a trivial modification to your basic circuit, a $20 USRT can be added and the whole system run byte-parallel on the processor's interrupt system (the minimum software version I mentioned in item 1 above), in a completely data-compatible mode, except that every USRT on the market follows the existing serialization standard and transmits and receives the least significant bit first, which would make all of the data upside-down in the processor (on input) or on tape (on output). Fortunately it is not necessary to alter either the physical standard or your already-designed and running hardware to correct this fault; the serialization sequence is entirely controlled by software, and to correct the problem in the software is not a major difficulty. The Data ID need not be altered physically if it is reinterpreted: as two bytes transmitted serially becomes (in hex) 91, F5 or (in octal) 221, 365.

As I mentioned earlier, I have gone through considerable gyrations in getting my own cassette interface running, and I have developed a paper which traces my experience in connection with a philosophy of peripheral design for microprocessors. I expect to present the paper at an upcoming meeting of the Bay Area Homebrew Computer Club, and would consider an opportunity to publish it to a wider audience. The title of the paper: The Quest for a 25¢ Cassette Interface, or A Microcomputer Design Philosophy". If published in TCH, I would include schematics and program listings to correspond to the TCH standard.

Tom Pittman

It is indeed true that the ASCII standard for data transmission in serial form calls for sending the least significant bit first. The reason for this is obvious, ASCII is basically a 7 bit code with the eighth used for parity. If you want both the high order bit to be the parity bit and the parity bit to be transmitted last, then the least significant bit must be transmitted first. IBM, true to form, transmits the most significant bit first in their data communications equipment. At any rate, the TCH cassette format is intended to be transparent, that is, not tied to any particular character code. There is no agreement on bit order when data is written onto magnetic media. Ignoring other factors, most-significant-bit-first is the logical way to do it because a printout of the bit stream can then be read simply by drawing lines separating groups of 8 bits. Likewise, a scope display of the signal can be easily read. The USRT problem can be solved simply by reversing the leads to the parallel data I/O leads on the IC package.

Your article proposal sounds great, so don't waste any time getting us a draft of it.

THE COMPUTER HOBBYIST
Founded October, 1974

Stephen C. Stallings - Managing Editor
Hal Chamberlin - Contributing editor
Jim Parker - Contributor
Steven Roberts - Contributor
Clyde Butler - Photographer
Richard Smith - Programming consultant

THE COMPUTER HOBBYIST is published at intervals approaching monthly near Raleigh, N.C. The subscription rate for 12 issues is $6.00 for persons in the U.S.A., U.S. possessions, Canada, and Mexico. Subscriptions in all other countries are $8.50 per year for surface mail or $13.50 per year for air mail. All back issues are available and cost fifty cents each for persons in the U.S.A., U.S. possessions, Canada, or Mexico. For all other countries back issues cost 75 cents each by surface mail or $1.25 each by air mail. Remittances from countries other than the U.S.A should be in the form of a bank or postal money order.

THE COMPUTER HOBBYIST is seeking paid contributors. Material to be submitted should be typed or neatly written and must not appear to be soliciting business for any firm. For details, please inquire.

Advertizing space is available to interested firms. Please inquire for rates and formats.

All correspondence should be addressed to:

THE COMPUTER HOBBYIST
Box 295
Cary, NC 27511

EDITORIAL

One would think that computer hobbyists, of all people, would be cool, level-headed, and logical in their thinking. But let the subject of conversation turn to octal versus hexadecimal notation for the Intel micros (8008 and 8080) and a shouting match may develop. One thing we can agree on though is that Intel created the problem by introducing and supporting the 8008 for more than two years with octal notation and then switching to hex notation for both machines when the 8080 was introduced.

Both notations are simply a more convenient way for human beings to communicate long strings of ONES and ZEROES. They are based on a fundamental law of human memory which states that the difficulty experienced in memorizing a string of symbols does not depend on the size of the alphabet from which they were drawn. An example would be the relative difficulty in memorizing a sequence of 30 random English words and a sequence of 30 bits.

Lets first look at the merits of the two number systems themselves without regard to a particular machine. Octal notation was invented sometime in the '50s for use on early binary computers. Its development was natural for two reasons. First, characters were 6 bits long in those days and binary computers frequently used 36 bit words. Second, octal notation confines itself to the already familiar decimal digits 0-7. This was important back then since line printers often had a 13 character set ($-.0123456789) to facilitate high speed operation. Hexadecimal notation was introduced by IBM along with the 360 line of computers in 1964. According to Knuth (1) the proper term for base 16 notation is "sexadecimal" but IBM probably didn't like the idea of "sex notation" or "sex dumps". The 360's were the first widely used computers that had an 8 bit byte thus making hex a reasonable choice. By itself, hexadecimal notation is probably the better of the two because it is more compact requiring less print space and time (except on Baudot TTY and Selectric typewriters where case shifting makes it slower) and memorization effort. Octal on the other hand is easier to learn and understand and can be easily displayed on 7-segment readouts using common decoder-drivers. Since hex is associated with IBM, some companies refuse to use it even though they should.

Now let's look at octal versus hex on the Intel micros. Since the machine word is 8 bits long, hex is a good first choice. However, the instruction set designers set up all of the instruction formats so that the bit field boundaries matched octal digit boundaries. This is evident on any instruction that referrs to registers because the register number (A=0, B=1, . . . M=7) becomes a digit in the 3 digit octal equivalent of the instruction. For example, LBM load B from memory, is coded 317 (in 8008) where 3 is a two bit "op code", the 1 is register B, and the 7 is memory pointed to by registers H and L. Things are shifted around in the 8080 but the octal oriented structure is unchanged. It should be obvious that with hex notation this correspondance between numeric instruction code and instruction function is lost requiring the some 200 instructions to be memorized separately or looked up in a table.

While octal is ideal for instructions, it is less than perfect for addresses. The problem arises because addresses are two bytes (16 bits) long and 3 digit octal notation implies a 9 bit number with the leftmost bit always zero. If addresses are considered to be 16 bit numbers, then the address 132764 is in page 265 at word 364. Note that the 6 digit octal address cannot be split into a 3 digit octal page and 3 digit word directly whereas the hex equivalent, B5F4 page B5 word F4, can. Actually the octal can be split with a little thought. To go from 6 digit address to page and word using the above address as an example, take the left 3 digits and double the octal number mentally (132 X 2 = 264). If the fourth digit from the left is 4, 5, 6, or 7, add one to your result (264 + 1 = 265) and you will have the page number. If the fourth digit is 0, 1, 2, or 3, the last 3 digits are already the word number. Otherwise subtract 4 from the fourth digit and you will have the word number (764 - 400 = 364). Going from page and word to 6 digits is also easy. First look at the page number. If it is odd, add 400 to the word number to form the last 3 digits of the result. If the page was even, then the last three digits are the same as the word number. The upper 3 digits are simply the page number divided by two in octal with the remainder discarded. Even these mental gymnastics are, I think, preferable to memorizing the hex op codes.

The ideal solution would, of course, be a notation that is good for both instructions and addresses. Some have suggested octal instructions and hex addresses. A better solution is called "split octal" notation. A split octal address is written as 265:364 which is page 265, word 364. The colon separates the two halves and reminds you that this is a split octal number, not a natural number. The only potential problem is in converting split octal to decimal. The digit weights are now 16384, 2048, 256 : 64, 8, 1. Possibly a better method of conversion with a pocket calculator is to convert each half separately and then multiply the page number by 256 and add in the decimal word number.

1. Knuth, Seminumerical Algorithms, Addison-Wesley Publishing Co., 1969.

TCH will be shifting to split octal notation in our 8008/8080 program listings when our new assembler is finished. We think that it offers the best balance between ease of learning and ease of use. For other machines, we will follow the manufacturer's recommendations unless they show the same indecision already demonstrated by Intel.

CLUBS ETC.

Since last issue the staff of TCH has been collecting information about clubs and other organizations which have sprung up to get computer hobbyists together. These organizations along with what information we were able to find are listed below:

LOS ANGELES, CALIFORNIA
Southern California Computing Society
Club newsletter - Interface
Contact - Hal Lashlee
Box 987
South Pasadena, CA 91030
213/682-3108

NORFOLK, VIRGINIA
Peninsula Computer Hobbyist Club
Contact - Larry Pollis
2 Weber Lane
Hampton, VA 23663
803/723-3117
Officers - Larry Pollis, President
Frank Pards, Vice President

NEW JERSEY
Amateur Computer Group of New Jersey
Club newsletter - ACGNJ NEWS
Contact - Sol Libes
995 Chimney Ridge
Springfield, NJ 07081
201/889-2000 day
201/277-2063 evenings

ATLANTA, GEORGIA
Atlanta Area Microcomputer Hobbyist Club
Club newsletter - AAMHC Newsletter
Contact - Richard Stafford
3144 Parkridge Crescent
Chamblee, GA
404/455-0118

SAN FRANCISCO, CALIFORNIA
Homebrew Computer Club
Club newsletter - Homebrew Computer Club Newsletter
Contact - Homebrew Computer Club
Box 626
Mountainview, CA 94040

SAN DIEGO, CALIFORNIA
San Deigo Computing Society
Club newsletter - Personal Systems
Contact - Personal Systems
10137 Caminito Jovial
San Diego, CA 92126
714/223-7853

CANADA
Tenative formation
Contact - G. Pearson
861 - 11th. Street
Brandon, Manitoba
CANADA R7A 4L1

DALLAS, TEXAS
North Texas No Name Computer Club
Contact - Bill Fuller
2377 Dalworth 157
Grand Prarie, TX 75050

Thats it for computer clubs, however TCH feels that two other groups are worth mentioning. Here they are:

BIT Users Association
3010 4th. Avenue South
Minneapolis, MN 55411
612/824-8247

This group, organized by Richard Koplow, is continuing support for the products of the now defunct BIT company. This should be of interest to those of you who acquired BIT-480 computers.

HP-65 Users Club
2541 West Camden Place
Santa Ana, CA 92704

This group is concerned with programmable pocket calculators (they no longer confine themselves to HP-65's) and they put out a fine newsletter, HP-65 NOTES.

Since it is inevitable that some of the above information is incorrect or out dated we will be printing updates next issue. Also if new groups start or we missed a current one, please let us know.

CONTINUED FROM PAGE 1

With several interfaces on one card, only one set of bus drivers is needed with multiplexors directing data from each interface to the bus drivers. Bus loading is also reduced since only one set of receivers is needed. The real limit is 2 amps set by the 7805's used on TCH's wirewrap board.

What about additional booster supplies if the standard ones are used to capacity? Adding the booster supply is somewhat tricky. Simple paralleling across the supply to be boosted is not recommended unless the components used are identical to the original. There is also a lack of space if the I/O connections are used and a fan is installed (recommended when the system becomes half filled). The only reasonable way to connect the booster supply is to cut the power bus for that voltage somewhere and make sure that the cards are installed such that neither the original nor the booster is overloaded. We would not recommend any substantial increase in the amount of +8 or +16 volt power available unless better internal ventilation is provided for. It would be permissable to double the -16 supply capacity by substituting a larger transformer and filter cap if necessary to power up to 4K worth of eraseable PROMS.

To illustrate the above points and also put a current controversy to rest lets compare MITS's 4K dynamic memory card with the 4K static memory cards available from alternate sources. The only real technical difference is in power resource use. The static RAM boards are stuffed full of 2102 equivalents (32 of them) and draw all of their power (about 1.5 amps) from the +8 unregualted supply. This power drain is constant, thus 4 cards for 16K would use up 6 amps leaving very little for I/O interfaces after the CPU and original 1K card are added in. Dynamic memory on the other hand draws power only when it is addressed (operating power) and very little when it is not addressed (standby power). Furthermore, most of the power is taken from the +16 volt supply; the +8 volt requirements are less than .5 amp to power the TTL overhead circuitry. If MITS had chosen to put 8K or even 16K of memory on a card, then even the TTL overhead power would have been negligible on a per K basis. Since only one group of 8 RAM chips can be addressed at a time, the power drawn from the +16 supply by a bunch of dynamic cards is only slightly more than that drawn by one card. The Altair could easily power 60K of dynamic memory but that much static memory would need a whopping 22 amps. So while it is true that <u>one</u> static card takes only a little more power in watts than <u>one</u> dynamic card, there is no comparison when several are used. As before, the main consideration is that power is a limited resource and it is up to the user to decide how that resource is divided up.

Bus drive capability is another limited resource. The 8T97 drivers used by MITS can drive 48 MA at .5 volts. The drive capability at the more common spec point of .4 volts is somewhat less, probably around 32 MA. MITS specifies a maximum bus loading of one <u>low power</u> TTL load (.16 MA at .4 volt) per card. Assuming a 32 MA drive capability, this figures out to a maximum of 200 boards in a system. MITS however routinely violates their own loading rules by connecting two and sometimes three low power loads to some bus lines. For examples, look at the address decoding on nearly any board, especially the 1K static board. If you don't intend to ever expand beyond the 16 board maximum in the case, it would be perfectly permissable to have one regular TTL load per card on a bus line and still have 6.4 MA of reserve drive capacity in a full system.

Besides DC current loading of the bus lines there is AC capacitive loading to worry about. It is this capacitive loading that contributes to internally generated noise. The noise arises from the fast, powerful bus drivers charging up the load capacitance when logic states are switching. The 8T97's can supply a surge of almost a quarter amp for each output. If a large number of bus lines change state at once such as an address change from 077:377 to 100:000 the large current surge flowing through the ground lines can induce noise in the critical strobe lines and possibly cause false triggering. As capacitive loading increases, the charging current's amplitude and duration also increases. The capacitance of a gate input and associated wiring is about 7 pF regardless of whether it is low power, standard, or Schottky. Thus, in order to reduce noise generation, each bus line should connect to a minimum of gate inputs through short connecting wires. In no case should a bus line be allowed to connect to an I/O cable that leaves the card without buffering. We have found that low power Schottky (74LS series) devices make excellent bus receivers. They are as fast as regular TTL, can drive 5 TTL loads, and have an input current of only .36 MA. They are also becoming more available on the hobbyist market. Purchase of a couple dozen 74LS04's should satisfy the bus receiver requirements of several boards. 8097's (also 8095, 8096, and 8098) can drive the bus as well as 8T97's although they are not as fast and don't have as much surge current capacity, thus cutting down on noise generation somewhat.

Appendix 1 gives the timing diagrams for three types of data transfer cycles in the Altair, input, output, and read memory. Write memory will be discussed in a future article. The times given are measured values observed in our Altair. These are subject to some variation since most of them depend on characteristics of the 8080 chip itself. The relative timing relationships should remain the same however.

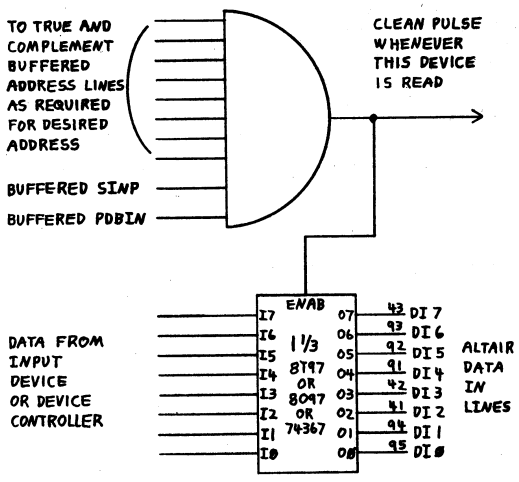


FIG. 1 SIMPLIFIED INPUT PORT SCHEMATIC

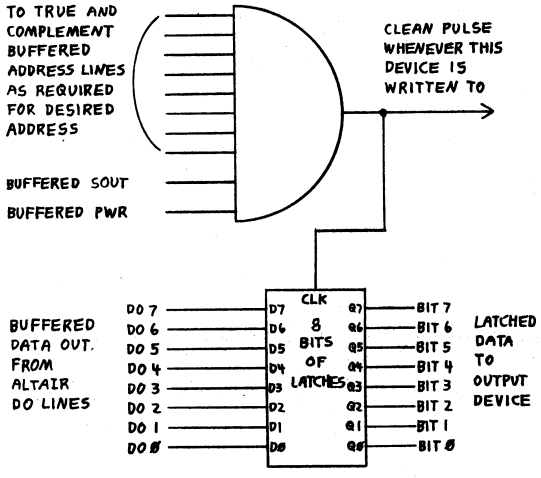


FIG. 2 SIMPLIFIED OUTPUT PORT SCHEMATIC

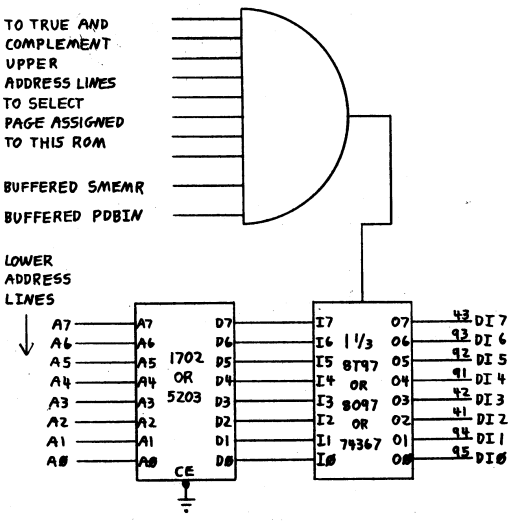
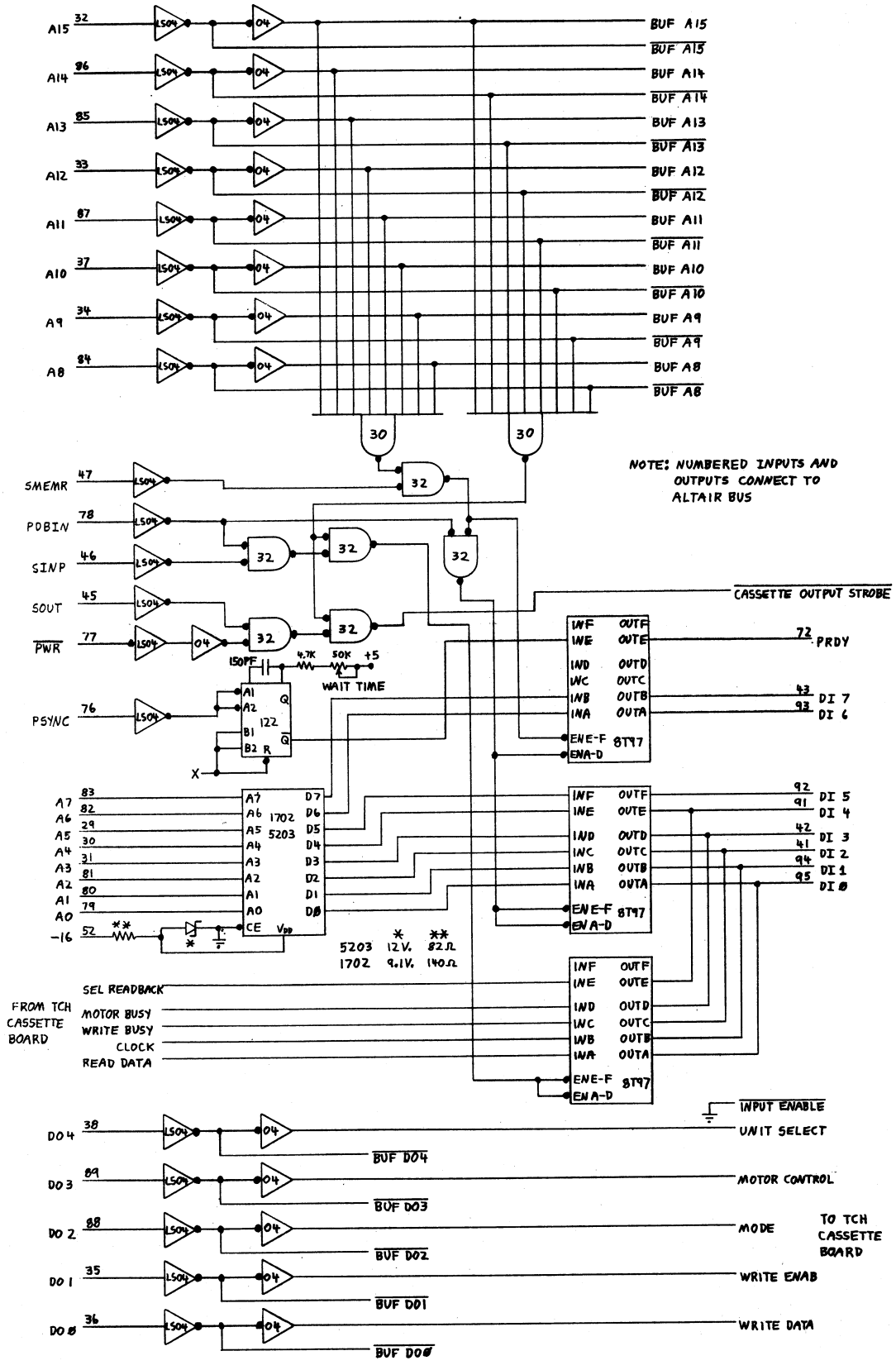


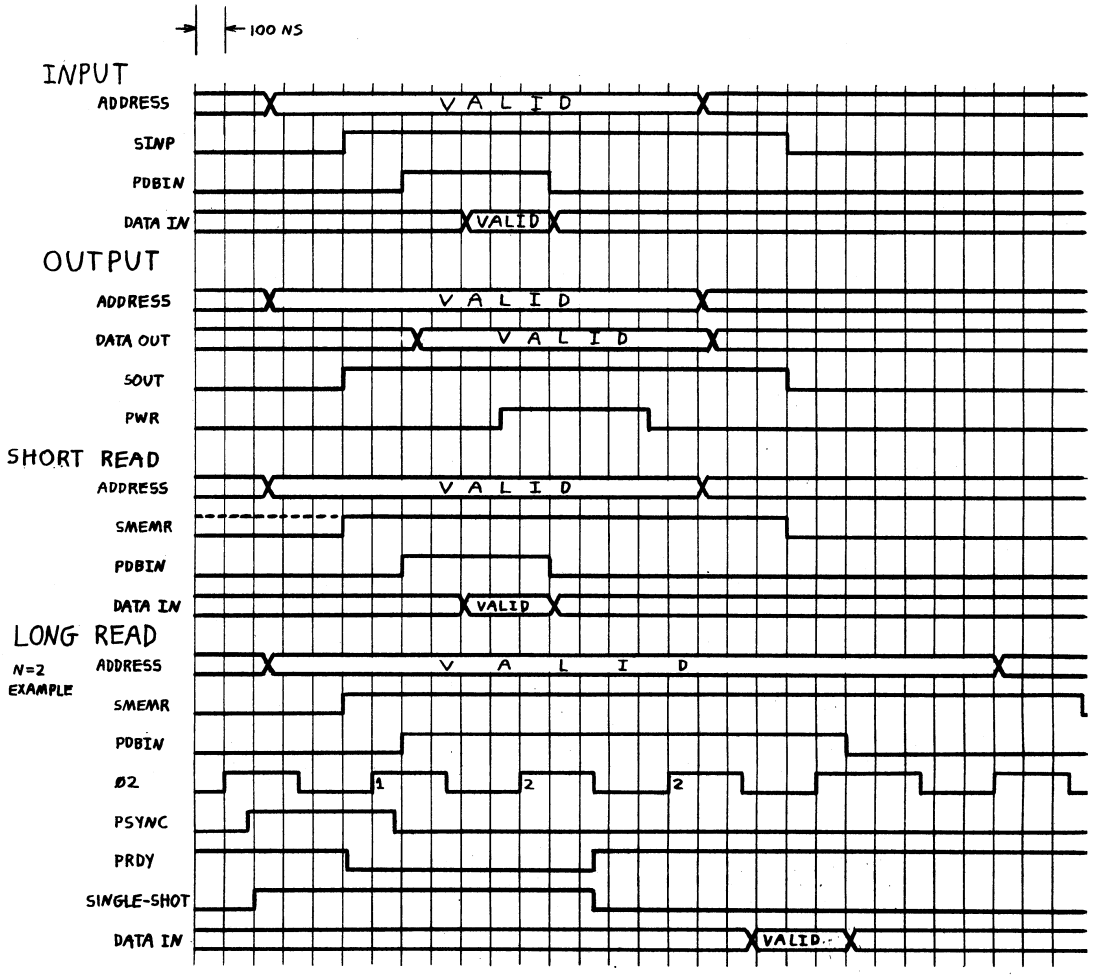
FIG. 3 SIMPLIFIED ONE PROM INTERFACE

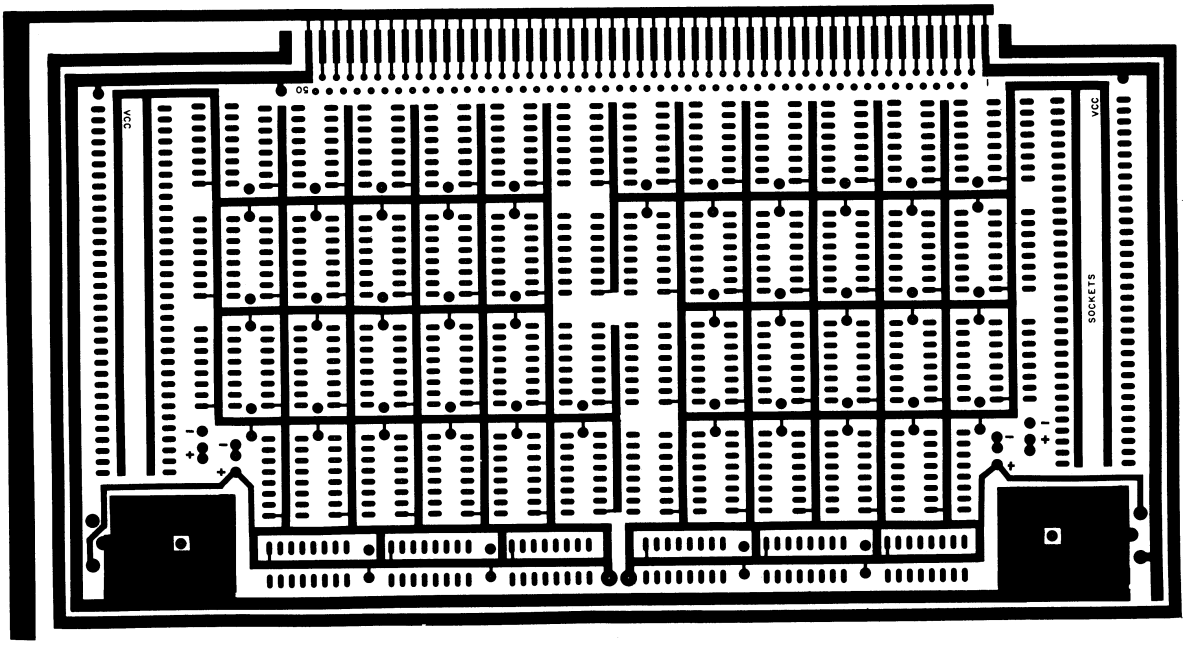
FIG. 4 EXAMPLE INTERFACE FOR CASSETTE AND PROM



NOTE: NUMBERED INPUTS AND OUTPUTS CONNECT TO ALTAIR BUS

APPENDIX 1. ALTAIR BUS TIMING





An input cycle is distinguished by the signal SINP (pin 46) going high on the bus. The input device address should have already settled on the address bus lines before SINP becomes active. An interesting property of the 8080 is that the 8 bit I/O device address is duplicated in the high and low order halves of the 16 address lines. This property can be used to distribute address buss loading evenly in a system with many I/O devices. After both the address and SINP have settled, PDBIN (pin 78) goes high. The coincidence (AND function) of the proper address, SINP, and PDBIN indicates that the input device data should be gated onto the data-in bus, lines DI0 through DI7. This is typically done with tri-state drivers such as 8T97's. A delay of up to 200NS is permissable from when PDBIN becomes active to when valid data is required to be on the input bus. If a clean pulse to inform the input device that it has been read is needed, it should be the coincidence of proper address, SINP, and PDBIN. At the end of the input cycle, PDBIN disappears first followed by the address, and then finally SINP disappears. That is all there really is to an input operation! Once the bus buffering overhead is taken care of, an 8 bit input port can be as simple as a 10 input AND gate equivalent to detect the coincidence of the address and control signals, and 8 bits of tri-state buffers (see figure 1).

An output cycle is distinguished by the signal SOUT (pin 45) going high on the bus. As with the input cycle the device address is duplicated on the high and low halves of the address bus and is settled before SOUT goes high. Shortly after SOUT becomes active, the output data from the CPU settles on the 8 data out (DO) bus lines. Finally, after everything is settles and valid, PWR (pin 77) becomes active for almost exactly 500 NS signalling that the DO bus data should be latched into the output device's register. Thus the trigger pulse used to clock the data from the DO bus into the device register is the AND combination of proper address, SOUT, and PWR. Since valid address, valid data, and SOUT brackets the PWR pulse, the register used to receive the data may be either leading or trailing edge triggered. Even level triggered devices such as 7475's or a 74100 may be used. This "load register" pulse is also clean and may be used to inform the output device that new data has been loaded into its register. Note that the PWR signal appears on the Altair bus inverted ($\overline{\text{PWR}}$) but is easily inverted again in the buffering process. As with input, once the bus buffering overhead has been taken care of, an 8 bit output port can be as simple as a 10 input AND equivalent and an 8 bit register (see figure 2).

We will discuss read memory now instead of later so that readers interfacing the TCH audio cassette board can also include an erasable PROM with cassette software on their interface. A look at the short memory read cycle timing diagram will show that it is the same as the input cycle except that read memory is distinguished by the presence of SMEMR (pin 47). The data out from the memory should be gated onto the DIN bus lines at the coincidence of proper address range, SMEMR, and PDBIN. Address range means the range of addresses assigned to the particular block of memory. For the case of a single 1702 or 5203 PROM, the circuitry can be very simple as shown in figure 3 once the required buffering has been done. For the case of reading multiple PROMS, the chip selects should be decoded directly from the buffered address lines in order to give the PROM maximum time to respond.

Unfortunately, almost any erasable PROM likely to be available for a reasonable price will be too slow to use the simple short read cycle. The standard speed for 1702A's is 1.0uS with speed selections from 650NS to 2.5uS available from the manufacturer. National's 5203 has similar speed capability. As can be seen in the short read timing, there is approximately 650 NS available from when the address settles to when valid data must be on the DIN lines. Allowing for buffering delays and variations in the 8080 chip, the short read should not be used if the PROM access time is greater than 500NS.

The PRDY line (pin 72) in the Altair can be used by the memory interface to instruct the CPU to provide a long read cycle rather than the short cycle. This technique allows additional time for memory access in increments of 500NS. The long read timing diagram shows the PSYNC (pin 76), PRDY, and Ø2 (pin 24) signals in addition to the others used for memory read. The CPU makes the decision whether to execute a short or long read based on the state of the PRDY line at the rising edge of Ø2 designated with a 1. If PRDY is high at this time, a short read is done; if it is low, the CPU enters the "wait" phase of a long read cycle. The PRDY line has a pullup resistor attached so that if PRDY is not connected, short reads automatically occur.

While the CPU is waiting, both SMEMR and PDBIN are active meaning that data from the memory is being gated onto the DIN bus before it is valid. Fortunately, the CPU only looks at the data in the 300NS period immediately before PDBIN drops. Also, once the wait phase is entered, the CPU examines the PRDY line at every rising edge of Ø2 designated with a 2. The first time a ONE is seen, the cycle is completed and PDBIN drops.

Since memory chips don't have a pin to tell when the data is valid, the user must provide a time delay equal to or longer than the data sheet specified access time to drive the PRDY line. Also, the circuit should drive PRDY only when the block of memory it serves is addressed. In the Altair, it is difficult to tell from looking at just the address and SMEMR lines when the memory access delay should start (note that PDBIN starts after the CPU has made its decision). Fortunately a signal, PSYNC, is available which marks the start of a data transfer cycle. Somewhat later, one of the "S" signals (SOUT, SINP, SMEMR, etc.) distinguishes what kind of cycle it will be. An easy way to provide the memory access time delay is to trigger a single-shot unconditionally with PSYNC. The coincidence of proper address range and SMEMR is then used to gate the single-shot output onto the PRDY line through a tri-state or open collector gate. If this block of memory was not addressed after all, the single-shot times out harmlessly without affecting PRDY. The optimum time setting is T=(500N)+150 NS where N is an integer such that T+350 is greater than the memory access time. In the case of unknown access time (likely with surplus PROMS) a trimpot can be connected to the single-shot to vary the time from about 500NS to 3uS. The pot can then be adjusted for the fastest reliable operation. Note that the single-shot must be of the retriggerable variety (74122, 74123) if the delay is longer than 1.2uS. Otherwise the single-shot may not have recovered from a preceeding short cycle.

As an example to illustrate some of the concepts discussed, figure 4 shows a possible schematic for an interface between the TCH cassette board and the Altair. Also included is provision for a 1702 or 5203 PROM containing the cassette software. This circuitry would use only about 1/3 of the available space on a wirewrap board leaving room for other interfaces.

The cassette board in TCH's demonstration system is mounted in a separate Ten-Tek box with two relay override switches and four LED indicators mounted on the box front. Since there is a two foot cable between the cassette board and the Altair, all of the signals sent to or received from the cable are buffered. To use this circuit, the cassette board should be jumpered for true data input and the 7403's replaced with 7408's or 74125's so that the data out will also be true. The 74125's would allow a number of cassette boards to be paralleled if desired. Altair bus lines DO0 - DO4 are buffered by 74LS04's and reinverted by 7404's. While the second set of inverters could have been eliminated if the cassette board was jumpered for inverted data, it is quite likely that the buffered DO lines might be needed elsewhere on the board. Complete separation of signals sent to or received from a cable from other circuitry is always desirable for maximum noise immunity. The 5 bits coming back from the cassette board connect to a single 8T97 or equivalent to drive the DIN lines. The various control signals are also buffered with 74LS04 inverters.

The address decoding is done in a completely general way in anticipation of added interfaces on the same board. Each of the high 8 address lines is double inverted to provide both the true and complemented form. By connection of 8 input NAND gates to various combinations of true and complemented address lines, any I/O address or page address can be selected. Using the address connections shown, the cassette interface is at 200 octal and the cassette ROM is at page 377.

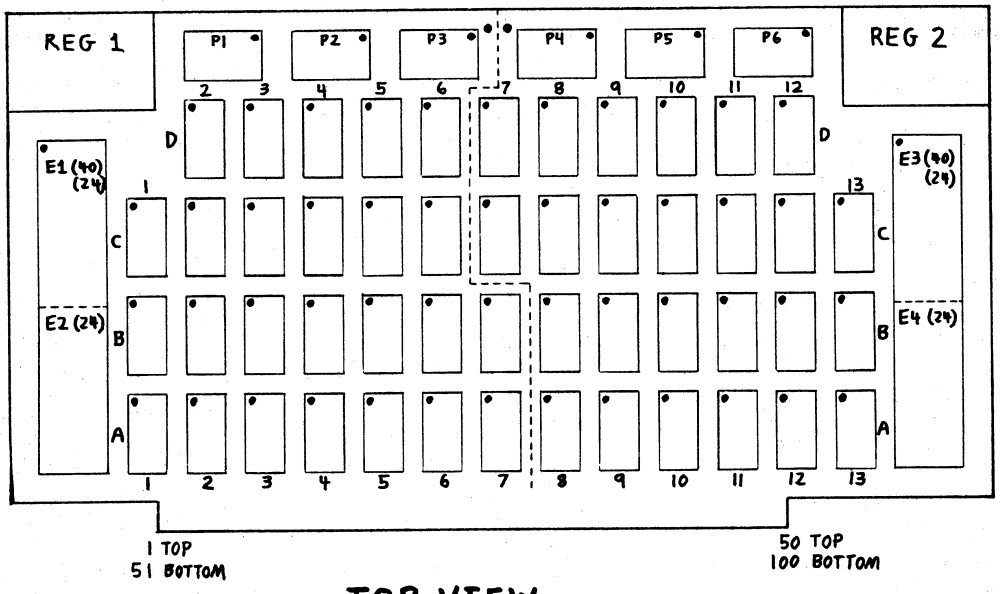
Type 7432 positive OR gates are used as NAND-NOT gates (see TCH # 2) to detect the coincidences of addresses and control signals. This eliminates the need for many inverters. The cassette output strobe is the coincidence of cassette address, SOUT, and PWR. The bus driver for the data back from the cassette is enabled by the coincidence of cassette address, SINP, and PDBIN. The drivers on the cassette board itself are always enabled by grounding the INPUT ENABLE signal to the cassette board.

If only one PROM is to be used, it is permissable to connect its address inputs directly to the low 8 address lines as shown. Although MOS circuits present no DC loading, the address input capacitance of several PROMS in parallel would increase the noise on the bus. The outputs of the PROM are only rated for one TTL load so the PROM chip enable is grounded and the DIN bus is driven with 8T97's or equivalent. The driver is enabled by the coincidence of ROM page address, SMEMR, and PDBIN. A 74122 retriggerable single-shot is triggered every time PSYNC goes up. An unused section of an 8T97 gates the inverted single-shot output onto the PRDY line when ROM address and SMEMR are present.

Power for the TTL and cassette board is taken from a 7805 regualtor on the wirewrap board. When only one PROM is used, the -9 or -12 volts required can be derived from the -16 bus supply with a simple zener diode regulator. If two or more PROMS are interfaced, a series regulator of some sort should be used to conserve power.

Additional interfaces can be added later to the board with very little additional circuitry since most of the bus signals are already buffered. If a number of input type interfaces are added (those that would drive the DIN lines), it might be wise to use only one set of 8T97's and a multiplexor to select data fed to the 8T97's. This would reduce capacitive loading on the DIN bus substantially. For example, we will probably put a keyboard interface, a graphics display interface (the XYZ signal generator would be in the display cabinet), and a couple more PROMS on our experimental board. As before, if the standard products meet your needs, use them. If not, the information presented here should make the job of custom interfacing easier and surer.

TCH 8080 WIRE-WRAP BOARD



TOP VIEW

NEW PRODUCTS

It seems that everybody and his brother is offering microcomputer kits of some sort these days. Three different ones have come to our attention in the last few weeks. In many cases the prices this month are half of what they were last month thanks to the bold lead taken by MOS Technology with their $20 microprocessor offering.

MITS has announced their long rumored MC6800 system. Logically it is called an ALTAIR 680 and not supprisingly it looks like a baby ALTAIR 8800. MITS has gone to larger, somewhat denser boards in the ALTAIR 680. For example, the CPU board has the MC6800 chip, 1K bytes of 2102 RAM memory, and a serial I/O port. In addition there are four sockets for 1702 erasable PROMS. This represents a significant departure from the ALTAIR 8800 philosophy which would have required 4 separate, smaller boards for the same function. The front panel is a separate board. In order to accomodate the slow speed of the 1702 PROMS, the clock frequency to the MC6800 is 500kHz, half of the maximum specified by Motorola.

The complete machine is enclosed in a smaller Optima cabinet (11"wide, 4 3/4" high, 11" deep) and includes built-in power supply. The kit prices reflect the recent deep price cuts by the semiconductor companies on microprocessor products. During the promotional period until the end of the year a complete kit (except erasable ROM's) is $293 or $420 for an assembled and tested unit. For the hardware nuts on limited budgets, the CPU board only is $180 for a kit or $275 assembled. Available software includes PROM monitor, assembler, debug package, and editor. We do not know whether the software is included or priced spearately.

MITS
6328 Linn N.E.
Albuquerque, NM 87108

The first kit we know of to use the MOS Technology microprocessor is being offered by Microcomputer Associates and is called, appropriately, JOLT. It sells for $249 as a kit and uses the "little bit of everything on the CPU board" philosophy that will be the hallmark of second generation computer kits. On a single board approximately 4" by 6 1/8" and using only 11 chips, there are 576 bytes of RAM (512 for program, 64 for stack), 1024 bytes of ROM with a debug monitor masked into it, a serial I/O port, and two 8 bit parallel I/O ports. The secret to the small number of parts is use of the MOS Technology MCS6530 "kitchen sink" chip. This 40 pin DIP has 64 bytes of RAM, 1024 bytes of masked ROM, two programmable I/O ports that work like Motorola's PIA, and an interval timer. Although the JOLT literature doesn't mention the timer, it should be there nevertheless. Use of the timer is far better than timed loops since it doesn't tie up the CPU and is not affected by memory waits.

An unusual feature of their debug monitor is a claimed self-adjustment to any serial data rate between 10 and 30 characters per second. This feature may have been necessary however since the on-board clock is not crystal controlled.

Other items in the JOLT line include a RAM board, an I/O board, a power supply, and a universal wire-wrap board. The RAM card sells for $265 as a kit and includes 4K of one microsecond static RAM, probably 2102's. This leads us to believe that the CPU clock is significantly slower that the 1mHz maximum in order to accomodate write cycles into the slow RAM (see article on MOS Technology chips elsewhere in this issue). CPU speed was not mentioned in the literature we received. The I/O card costs $96 as a kit and provides four 8 bit programmable I/O ports and two interrupt requests. It uses two of the PIA chips now made by Motorola and soon to be made by MOS Technology. The power supply is $145 as a kit and provides +5, +12, and -10. The current ratings were not specified but enough power for CPU, 4K of memory, and an additional I/O card is claimed. The wire-wrap card is the same size as the others and sells for $25 in single quantities.

The JOLT boards are interconnected by means of flat cable and Scothflex connectors rather than edge fingers and edge sockets. An "accessory bag" at $40 is available with the flat cable and connectors necessary to connect two boards together. Each additional card would require an additional accessory bag or equivalent. Software beyond the ROM based debug monitor was not mentioned.

Orders sent before Nov. 10, 1975 consisting of CPU cards and other items will receive a 20% discount on the other items.

Microcomputer Associates, Inc. Dept. A.
111 Main St.
Los Altos, CA 94022

SPHERE is now selling their CPU board separately. It too has a little bit of everything except memory which is a whopping 4K bytes on the CPU board. This was accomplished by using only eight 4K dynamic chips tucked away in the corner of the board. Having the memory timing and refresh control on the same board as the CPU greatly simplifies dynamic RAM interfacing. Sockets for four 1702 PROMS are provided with two of them used by their console replacement debug monitor. A real-time clock, 16 bits of programmable I/O lines, and a serial teletype interface are also included on their 6.5" by 8 1/8" CPU card. The mail order price for the complete CPU board to hobbyists is $350 and is a limited time offer.

SPHERE Corp.
791 South 500 West #3
Bountiful, Utah 84010

Software is the other half of a complete system that is becoming available from multiple sources and at greatly reduced prices. MITS has just cut their 4K and 8K BASIC package prices to non-Altair owners. The 4K package now sells for $150, down from $350 and the 8K package is now $200, down from $500. This move clearly indicates MITS's intention to remain competitive in software as well as hardware. It probably also indicates that the development costs have been recovered. The high quality of their package is generally unquestioned, although some have commented on its speed.

Although there is a tremendous interest in BASIC, many hobbyists may prefer FORTRAN for solving really complex mathematical problems. Another advantage is that most commercial scientific application packages such as circuit analysis programs are written in FORTRAN. Additionally, FORTRAN provides much better argument passing capabilities to subroutines than does BASIC.

Mini-Software Inc. now has a FORTRAN compiler for the 8080 as well as a BASIC compiler. Both are two-pass compliers meaning that the source code is translated into a pseudo-assembly language first which is read back to be translated into object code.

The two phases of the compiler require 14K bytes each and also rely on user supplied I/O routines which would occupy the remaining 2K on a 16K system. One shortcut taken for simplicity is a requirement that all subroutines be compiled along with the main program. These sizes provide enough table area to compile programs as large as 1000 statements. An additional 8K for the compiler's tables would more than double the maximum program size.

The run-time package is 6K and the generated code is quite compact (GOTO 25, 2 bytes; IF(A.EQ.B)GOTO 15, 7 bytes) so it would be possible to run a non-trivial compiled FORTRAN program on an 8K machine. Maximum object code size is 32K and maximum data area size is 16K although these limits may be expanded with a slight reduction in speed. The object code is interpreted by the run-time routines rather than being executed directly.

Total compilation speed is about 30 statements per minute exclusive of I/O time which would be substantial without digital cassette or floppy disk. This is not bad considering that the compiler itself is written in FORTRAN and what is run on the 8080 is actually a compiled version of the compiler. The arithmetic package is a little strange. Integers are 40 bits (5 bytes) long and floating point numbers are 48 bits long. This affords roughly 11 digits of accuracy rather than the more common 7 digits. Floating add takes 2.5 Ms, floating multiply takes 24 MS, and floating divide takes 17 MS. At the expense of 5 significant bits, the multiply speed may be doubled.

For the most part the compiler accepts ANSI standard FORTRAN. Most restrictions are minor hassles regarding reserved words and the placement of blanks. The two major restrictions are no three dimensional arrays and no quoted character strings in FORMAT statements.

The complete FORTRAN system sells for $350 and BASIC sells for $400. Other major software products are an 8080 macro cross assembler for $100 and a FORTRAN cross compiler for $200. These cross products allow programs for the 8080 to be assembled or compiled on a large machine and run on a small 8080 system. A 50% discount is extended to non-commercial hobbyist users who prepay their orders.

Mini-Software
P.O. Box 7438
Alexandria, VA 22307

The IMP-16 microprocessor chip set made by National Semiconductor is perhaps the best microprocessor to use if a high performance, flexible hobby computer system is desired. Much emphasis has been given to the Intel 8008/8080 in the past and very little has been written on the IMP-16 even though it has been around quite awhile and is just as available as the 8080. There are many reasons for this and I hope that this series of articles will negate some of them and increase hobbyist awareness of this high-performance alternative.

Before continuing, a word should be said about National's PACE microprocessor. PACE is basically a one chip (instead of five) IMP-16. Improvements offered by the PACE are provisions for unlimited stack depth, five levels of vectored interrupt built-in, and reduced support IC package count. Sacrifices made include slower speed (2us state time instead of 1.4us), non-availability of extended instruction set CROM's and reduced flexibility or omission of some of the instructions. Nevertheless PACE is an excellent microprocessor that should be considered when an 8 bit machine is not enough but the full-bore system to be discribed is not quite necessary. As far as the hobbyist is concerned, PACE is about as far along as the Motorola 6800, good availability can be expected toward the end of this year. Conversely, the IMP-16 will not be discontinued any time soon, in fact, a bipolar chip set equivalent has been rumored.

Since adequate coverage of this topic will require a lot of space, it will be presented in four consecutive parts. This part will describe the system features and fundamental operation concepts. Hopefully , there will be enough detail to enable the reader to decide if this is the system for him. Part 2 will detail the system bus controller and timing generator. Logic diagrams of this portion of the machine will also be published. Part 3 will deal with the microprocessor chips themselves and the interface to the system bus described in Part 2. Again complete diagrams will be given. Part 4 will conclude the series with a presentation of an 8K by 16 dynamic memory board and a general discussion of I/O device interfacing with examples.

This implementation of the IMP-16 is original as far as can be determined. The CPU and memory design is part of a computerized music synthesis system being built by the author. Little circuitry was borrrowed from National's implementation, the IMP-16C. Weird or non-standard IC's were avoided as much as possible. Those that remain are listed in Appendix 1 along with the quantity required so that orders can be placed early.

Fundamentally, all system components communicate over a set of 44 lines which, for lack of a better name, will be called the PUNIBUS (Processor UNIfied BUS). This bus structure has much of the power found in Digital Equipment Company's UNIBUS used in PDP-11 computers but is simpler to interface to and is synchronous, having a definite cycle time of 700 ns. Another major difference is that it is intended to be a backplane bus with a maximum length of two feet or so. The length restriction allows ordinary TTL gates and standard tri-state devices to be used for bus driving. Thus in a system it is necessary that at least a portion of each I/O device interface be on a board that plugs into the system backplane. In actual practice (details and examples in Part 4) several simple device controllers or a complete complex one can fit on a single board.

The bus cycle time of 700 ns was chosen to be compatible with the IMP-16 microcycle time of 1.4 us and with 4K dynamic RAM chips. The same timing generator is used for bus timing and CPU timing thus reducing circuitry and insuring synchronization between the CPU and the bus. Timing signals are provided on the bus for operation of all popular 22 pin 4K dynamic RAMS. The newer 18 pin and 16 pin 4K RAMS can also be supported by changing one of the two 8223 PROM's that are used to generate the timing signals. Alternatively, you can use those cheap 1103's and 5260's that are available everywhere. As designed, the bus controller has no provisions for waiting on slow memory (slower than 400 ns access). This was done to reduce parts count and because adequately fast dynamic RAM's cost less per bit than the slower static RAM's anyway. The only real problem created by the lack of a wait is the inability to use most eraseable PROMS for program storage.

At the beginning of every bus cycle a decision is made as to which device will get the bus cycle. The CPU, memory refresher, and up to 5 direct memory access devices can be supported. If more than one device is requesting, the cycle is awarded on the basis of priority. The normal arrangement gives the CPU highest priority, memory refresher lowest priority, and the 5 DMA's are in between the extremes. This arrangement of priorities has several interesting properties. Since the CPU cycle is synchronized with the bus cycle and it has the highest priority, the CPU never has to wait. Conversely, the CPU will not hog the bus either because it will use at most 25% of the available bus cycles. The memory refresher is connected so that it is always requesting cycles. If neither the CPU nor any DMA devices are requesting the bus, then the cycle goes to the refresher. Adequate refreshing is assured (at least 64 cycles every 2 milliseconds) if as little as 2.3% of cycles are unclaimed. This allows totally transparent dynamic memory operation with the CPU running full speed and over one million words per second aggregate DMA rate. The bus could support up to 4 IMP-16 CPU's simultaneously. One CPU would be the master which would never wait and which would control the bus while the others would have lower priorities and may have to wait.

Input/output devices are addressed as memory locations and are connected to the bus as if they were memory. All of the memory reference instructions in the IMP-16 set can therefore be used for I/O operations. The regular IMP-16 I/O instructions are not used. I/O devices are assigned to addresses FF00 to FF7F hex and the distinction between input and output is governed by whether a memory read or a memory write is executed by the program. An interesting side-effect is that any DMA device also has access to the I/O system.

A tabulation of the bus signals appears in Appendix 2. Note that data and addresses are time multiplexed over a single set of 16 lines. Note also that timing signals are provided to identify what is on the bus or what should be gated onto the bus. The multiplexed bus not only saves connections but also saves IC's since fewer bus drivers and receivers are needed. Memory boards using 4K RAM's do not require address registers because they are built into the RAM chips. I/O device address decoding circuits will however require the use of a flip-flop to remember that it was addressed after the address disappears from the bus. Regulated supply voltages of +5, +15, and -15 are provided on the bus. Virtually any digital or linear IC can be operated from these voltages with perhaps a zener diode added. Two lines are reserved for +5 and three for ground to insure a noise-free +5 supply and ground. Ten amps from the +5 volt supply and 2 amps each from the 15 volt supplies should be enough to power a medium sized system with 16K of memory and 150 chips of I/O interfacing.

The IMP-16 chip set has two levels of interrupt built in. The lower level can be masked off and is called the I/O or general interrupt. The interrupt request bus line is connected to this interrupt. The higher level cannot be disabled and is called the control panel interrupt. It is normally connected to a console interrupt button and causes a return to the system monitor.

Built into the CPU board are two sockets for 8223 bipolar PROM's. This gives 32 words of permanent storage at locations FFE0 through FFFF which is adequate for a bootstrap loader. Whenever the power is turned on or the console reset button is actuated, the CPU clears its registers and branches to location FFFE in the loader. The loader program then reads in the operating system software from an input device such as a paper tape reader, cassette, floppy disk, eraseable ROM or even (shudder) a bank of switches or keyboard. The floppy disk is particularly convenient, start-up or recovery would require less than a second.

The operator's control panel consists of three push-buttons and two lamps. One button is the system reset button, one is used to restart the system after a HALT instruction, and the last is the control panel interrupt button. The lamps monitor the running and interrupt enable status of the CPU. Effective communication with the machine then requires a debug-monitor software package, an ASCII keyboard, and a display or teleprinter; items that should be present in any high-performance system. It is interesting to note that even in National's complete microcomputer system, the IMP-16L, the extensive console provided is implemented with software in a ROM.

One of the major attractions of the IMP-16 over other microprocessors is the availability of extended instruction set CROM's. One of these is called simply the "extended instruction set CROM". Included are 15 by 16 bit multiply, 31 by 16 bit divide (both of which take an average of 150 microseconds), 32 bit add and subtract, byte manipulation using byte memory addressing, and a shift and count instruction, potentially useful for floating point. The microprogrammed multiply and divide routines can be corrected to obtain the more common 16 X 16 and 32/16 operations with a few additional regular instructions. Another CROM which has just been announced is called the "POWER I/O" CROM. Included are high-speed (approx. 90K words/sec) microprogrammed block I/O transfer instructions, a block move from one area of memory to another, masked memory search, and machine status save/restore in a microprogram controlled stack in memory. One possible drawback to using these instructions on long blocks of data is that interrupts cannot be recognized

until they are finished resulting in possible millisecond long interrupt latencies. Up to 4 CROM's may be used in a system. Others will doubtless come out as the masking charge is about the same as for a ROM and National has a microprogram prototyping kit available. Of particular interest, of course, would be a floating point CROM. (A "Power Math" CROM is rumored which supposedly handles all of the time-consuming functions of floating point software.)

Backing the IMP-16 is a substantial body of essential software. Although National asks $200 for a package including source listing and object tapes, the material is not copyrighted. We understand that the $200 pays mostly for "field support" functions. National asks only that they not be "bugged" if the software was not bought from them. TCH is punching source decks from the listings for modification. The modified software will be available at the end of this series. The code is well organized and thoroughly commented which makes understanding and further modification much easier. Input/output is generally done in only one subroutine thus alteration for different I/O schemes is easily accomplished.

The assembler requires 4K words to load and has many features not often found in micro- or even minicomputer assemblers. Conditional assembly, local symbols, relocatable or absolute code generation are all provided. Symbols require three or four words so an 8K system would allow about 1150 symbols. The relocating, linking loader allows each module of a massive program to be assembled seperately. When the entire program is to be run, it links the segments together and relocates them so that no memory "holes" appear between segments. The editor is essential when preparing large programs unless you have a keypunch and card reader. It works like the typical minicomputer editor in that a source file and keyboard commands are accepted as input and a destination file is produced as output. A memory buffer is used to allow random access to a portion of the text for editing. The debug program is quite extensive and is just the thing for replacing the typical computer console. It offers memory search, memory move and fill, and up to 5 separate breakpoints for program tracing. Many other typical functions are provided and the command structure makes expansion into a full scale operating system feasible. The CPU diagnostic program is very elaborate and very thorough. If it will run, you are assurred of a properly functioning system. If not, the error analysis usually pin-points the problem.

And last but not least are physical and packaging considerations. The board size used in the music system from which this design was taken is 8" by 8" with a 100 contact edge finger similar to that used in the Altair. The 64 square inches of space available is sufficient to place the CPU, bus controller, and bootstrap ROM on the same board. In the memory department there is enough space for 8K words using 22 pin 4K RAM's or probably 16K words using 16 or 18 pin 4K RAM's. One could also expect to fit a floppy disk controller or display generator on a single board. Using the 100 contact edge connector, up to 54 uncommitted pins are available for external world connections to each board. The mother board for the system is built in sections of 6 sockets each. The socket patterns are such that wire-wrap edge sockets will fit with 1.25 inches between boards. The wide spacing allows effective use of wire-wrap boards using even long pin, high profile sockets. A card file for standard rack mounting will hold two sections of the mother board for a total of 12 card positions, adequate for all but the largest systems.

Existing boards for the system include a wire-wrap board and a memory board. The wire-wrap board has 24 14-pin patterns, 44 16-pin patterns, and 10 24-pin patterns. Two pair of the 24 patterns can be combined to hold one 40 pin DIP each. One big advantage in using the board is the power and ground planes, bypass capacitor pads, and lands to the power and ground pins of the IC's. The memory board has a capacity of 8K 16 bit words using popular 4K RAM's such as the TMS-4030. It is complete and just plugs into the PUNIBUS. Jumpers select the block of 8K assigned to each board. A system is already operating at TCH and is essentially bug free. If you just can't wait for the next three parts to be published, blueprints of the CPU and memory board are available now for $4.00 but without the descriptive writeups. The memory board, wirewrap boards for CPU and interface construction, backplane boards and a card file with 12 card capacity are also available. Write the author in care of TCH for details.

APPENDIX 1

Integrated circuit list for IMP-16 CPU card.
* Denotes less common item. ** A real pain.

QUAN	TYPE
2	7400
* 1	74LS00
2	7402
2	7404
* 1	74LS04
* 2	74S04
1	7410
1	7413
* 1	74LS20
* 1	74LS21
1	7432
1	7442
* 1	74LS54 (Needed only for extended CROM)
4	7474
1	7493
2	74125 or 8093
* 1	74148
9	74173 or 8551
1	74174
* 1	74S174
* 1	8097 or 8T97
2	8223 (Programmed with CPU timing)
2	8223 (Programmed with IPL routine)
**2	8334 (9334)
* 5	8833
* 2	MH0026

Integrated circuit list for 8K by 16 memory card.

QUAN	TYPE
3	7404
1	7437
1	7474
4	74173 or 8551
* 2	8098 or 8096
* 1	MH0026
1	741
32	TMS4030 or 2107A or 2107B

APPENDIX 2 PUNIBUS SIGNAL SUMMARY

PIN	SIGNAL NAME	DESCRIPTION
1	+5	Regulated power
2	GROUND	
3	+15	Regulated power
4	-15	Regulated power
5	BUS DATA 14	Data bus
6	BUS DATA 15	Data bus most significant bit
7	BUS DATA 12	Data bus
8	BUS DATA 13	Data bus
9	BUS DATA 10	Data bus
10	BUS DATA 11	Data bus
11	BUS DATA 8	Data bus
12	BUS DATA 9	Data bus
13	BUS DATA 6	Data bus
14	BUS DATA 7	Data bus
15	BUS DATA 4	Data bus
16	BUS DATA 5	Data bus
17	BUS DATA 2	Data bus
18	BUS DATA 3	Data bus
19	BUS DATA 0	Data bus least significant bit
20	BUS DATA 1	Data bus
21	BUS I/O ADDR	Low when data bus has value in range of X'FF00 - X'FF7F
22	GROUND	
23	BUS ADDR ENAB	High when controlling device should gate address onto bus
24	BUS CE	Chip enable clock for 4K RAM's
25	BUS WRITE ENAB	Strobes data from bus during memory write or output
26	BUS MDR STROBE	Strobes data output register on 8K memory board
27	BUS CLOCK	17.1459 mHz signal 2 X frequency of middle C
28	BUS DATA OUT ENAB	Gates data onto bus during memory read or input
29	BUS DATA OUT STB	Strobes data from bus during memory read or input
30	GROUND	
31	BUS DATA IN ENAB	Gates data onto bus during memory write or output
32	BUS INT REQ	Wire-or I/O interrupt request
33	BUS RESET	low during power-on or reset
34	BUS WRITE CYCLE REQ	Wire-or direct memory access write request
35	BUS GRANT 4	Binary code of DMA
36	BUS GRANT 2	device in control
37	BUS GRANT 1	of the bus
38	reserved	
39	reserved	
40	reserved	
41	BUS REQ 1	Highest priority
42	BUS REQ 2	Wire-or lines for requesting DMA
43	BUS REQ 3	cycles
44	BUS REQ 4	Grant code corresponds to request
45	BUS REQ 5	number
46	POWER OK	Pulled down by power supply when voltages are OK

DEAL - A CARD DEALING SUBROUTINE FOR THE 8008

by Steven Roberts

The Random Number Generator proposed by Jim Parker in TCH Volume 1, Number 5, provides the basis for a card dealing subroutine which, in turn, can provide the basis for a number of interesting games. This routine returns to the calling program an ASCII string naming the card dealt, and in register B, a six bit card code to simplify logical operations involving the card's identity. In addition, a card list is created which is used by the routine to verify that a newly chosen card has not been dealt since the last shuffle; this list and its associated pointer may be used by the calling program to determine the number and identity of cards dealt. When all 52 cards have been dealt (as determined by cardlist pointer = maximum cardlist address), the word "EMPTY" is placed in the ASCII buffer (CRDBUF) and returned to the caller. Shuffling is accomplished simply by resetting the list pointer. Upon doing so, the word "SHUFFLE" is placed in the ASCII buffer.

In addition to the card code in reg. B and the contents of the ASCII buffer, the user has access to additional information. The card code is found also in the memory location CRDCOD (which contains a 377 octal if all 52 cards have been dealt). POINTR yields the number of cards dealt, and they may be found, in card code format, in the 52 locations defined as CRDLST. The format of the returned ASCII string is shown by the sample run, and the organization of the CRDCOD byte is shown in the program listing.

The simplest use of DEAL is in the application used to generate the sample run; the program involves only a call of SHUFFL and then a call of DEAL and a print of CRDBUF for each card. Card games of any description could make use of the subroutine, however - BLKJAK and POKER are currently under development, and others will likely follow.

```
*       DEAL PROGRAM BY STEVEN ROBERTS
*       CALL SHUFFL TO SHUFFLE DECK
*       CALL DEAL TO GET EACH CARD
*       RETURNS A 10 CHARACTER ASCII STRING IN CRDBUF
*       AND A 6 BIT CARD CODE IN REG. B
*       BITS 2 - 3 SUIT CODE
*       BITS 4 - 7 RANK CODE

        ORG 0           PROGRAM ORIGIN

SHUFFL  LAI L(CRDLST)   POINTER VALUE FOR START OF CARD LIST
        SHL POINTR
        LMA             LOAD NEW POINTER VALUE
        SHL SUITS+38    POINT TO THE WORD "SHUFFLE"
        LEI 10
        LCI L(CRDBUF)
        JMP MOVLOP      GO MOVE IT TO CRDBUF

DEAL    CAL RAND        CALL RANDOM NUMBER GENERATOR
        NDI 077B        REMOVE BITS 6 AND 7
        LDA             SAVE THE REST
        NDI 017B        REMOVE BITS 4 AND 5 LEAVING CARD VALUE
        LCA             SAVE RESULT IN C
        SBI 015B        IS IT OUT OF RANGE?
        JFC DEAL        IF SO, REDEAL
        LAD             OTHERWISE, RESTORE THE 6 BIT CODE
        NDI 060B        REMOVE BITS 0 - 3, LEAVING SUIT CODE
        LDA             RETURN THAT TO D
        ORC             REASSEMBLE CARD CODE
        LBA             SAVE IT IN B
        SHL POINTR      ADDRESS CARD LIST POINTER
        LEM             POINTER TO E
        LAI L(CRDLST+52)  LOAD A WITH MAX POINTER VALUE
        CPE             IS POINTER AT MAX?
        JTZ EMPTY       IF SO, GO PUT "EMPTY" INTO CRDBUF
        SHL CRDLST      ADDRESS START OF CARD LIST FOR SCAN
SCAN    LAM             CARD TO A
        CPB             IS IT = TO NEW CARD CODE (IN B)?
        JTZ DEAL        IF SO, REDEAL
        LAL             OTHERWISE, MOVE CURRENT LIST ADDRESS TO A
        CPE             SAME AS POINTER?
        JTZ STORE       IF SO, CARD IS VALID (NOT DEALT BEFORE)
*                       GO STORE
        INL             OTHERWISE, INCREMENT ADDRESS AND CHECK
*                       ANOTHER
        JMP SCAN        LOOP UNTIL EITHER COMPARISON IS VALID
STORE   LMB             STORE NEW CARD, SINCE IT IS VALID
        INL             UPDATE POINTER
        LEL             SAVE IN E
        SHL POINTR      ADDRESS POINTER BUFFER
        LME             STORE IT
        LAI L(CARDS)    BASE ADDRESS OF ASCII CARD VALUES
        LLC             CARD VALUE TO C
        ADL             ADD TO BASE ADDRESS
        LLA             SAVE THAT IN L, NOW POINTING TO NUMERIC VALUE
        LAM             ASCII CARD VALUE TO A
        SHL CRDBUF      ADDRESS FIRST LOC. OF ASCII BUFFER
        LMA             STORE IT (A,2,3,....T,J,Q, OR K)
        INL             POINT TO SECIND LOCATION IN CRDBUF
        LMI ' '         CLEAR SECOND LOCATION
        LAD             SUIT CODE TO A
        RAR             ROTATE RIGHT
        SHL SUITS       ADDRESS START OF SUITS BUFFER
        ADL             ADD NEW SUIT CODE TO BASE ADDRESS
        LLA             PUT THAT IN L, NOW POINTING AT CORRECT SUIT
*                       NAME

MOVE    LEI 8           PREPARE FOR 8 ITERATIONS
        LCI L(CRDBUF+2) POINT WITH C TO SUIT FIELD IN CRDBUF
MOVLOP  LAM             PULL CHARACTER FROM SUIT NAME
        LDL             SAVE ADDRESS IN D
        LLC             CRDBUF ADDRESS TO L
        LMA             STORE CHARACTER
        LCL             CRDBUF ADDRESS BACK TO C
        LLD             SUIT NAME ADDRESS BACK TO L
        INC             INCREMENT BOTH
        INL             ADDRESSES
        DCE             DECREMENT COUNTER REGISTER
        JFZ MOVLOP      IF NOT 0, GO BACK AND MOVE ANOTHER CHARACTER
        SHL CRDCOD      OTHERWISE, ADDRESS CARD CODE BYTE
        LMB             SAVE IT
        RET             RETURN TO CALLING PROGRAM

EMPTY   SHL CRDBUF      POINT TO CARD BUFFER
        LMI ' '         CLEAR THE CARD BYTE
        INL
        LMI ' '         CLEAR THE SECOND BYTE
        LBI 377B        "EMPTY" CODE TO CARD CODE
        SHL SUITS+32    POINT TO THE WORD "EMPTY"
        JMP MOVE        GO MOVE IT INTO CRDBUF, RETURNING WHEN DONE

RAND                    RANDOM NUMBER SUBROUTINE FROM TCH # 5

        ORG 400B        ALL DATA IN SAME PAGE

CRDLST  DST 52          LIST OF CARDS DEALT IN 6 BIT CODE
SUITS   DEF 'HEARTS  SPADES  CLUBS   DIAMONDSEMPTY   SHUFFLE
CARDS   DEF 'A23456789TJQK'
CRDCOD  DST 1           CARD CODE RESULT OF DEAL
*                       BITS 2 - 3  SUITS  00=HEARTS, 01=SPADES
*                       10=CLUBS, 11=DIAMONDS
*                       BITS 4 - 7 RANK  0000=A, 0001=2, 0010=3
*                       0011=4, 0100=5, 0101=6, 0110=7, 0111=8,
*.                      1000=9, 1001=T, 1010=J, 1011=Q, 1100=K
CRDBUF  DST 10          ASCII BUFFER 10 CHARACTERS
POINTR  DST 1           POINTER TO LOCATION IN CRDLST
        END
```

SAMPLE RUN OUTPUT WITH ADDED CALLING ROUTINE AND PRINT ROUTINE

```
  SHUFFLE
8 DIAMONDS
8 HEARTS
4 HEARTS
4 SPADES
J DIAMONDS
4 DIAMONDS
9 SPADES
6 DIAMONDS
T SPADES
A SPADES
A CLUBS
8 CLUBS
7 CLUBS
K HEARTS
2 CLUBS
Q SPADES
9 DIAMONDS
3 HEARTS
6 HEARTS
T DIAMONDS
5 CLUBS
5 SPADES
3 CLUBS
A HEARTS
Q CLUBS
6 SPADES
7 HEARTS
K DIAMONDS
Q DIAMONDS
5 HEARTS
J SPADES
K SPADES
J HEARTS
6 CLUBS
T HEARTS
Q HEARTS
3 DIAMONDS
9 CLUBS
5 DIAMONDS
K CLUBS
9 HEARTS
3 SPADES
2 HEARTS
2 DIAMONDS
7 DIAMONDS
J CLUBS
8 SPADES
A DIAMONDS
4 CLUBS
T CLUBS
2 SPADES
7 SPADES
  EMPTY
```

Well, it has one! The MOS technology (NOT Mostek) MCS6501 and MCS6502 are out, cost only $20 and $25 each respectively in quantities of one, and are being shipped to anybody who wants one. The best part is that the $20 price is not a limited time come-on, it is the true, factory direct, single quantity price. In larger quantities they cost less! Better yet, the chip doesn't limp along like an 8008, its speed and programming ease equals or surpasses chips such as the 8080 and Motorola 6800. With the $25 MCS6502, you can forget about clock drivers because the chip has a built-in oscillator, just connect a 1mHz crystal or R-C and go. These things are going to take over the world!

Availability is always a big problem with any new product but as far as we can tell, it won't be any problem with these chips. MOS Technology seems to be trying a bold new marketing technique - direct from the factory mail-order sales. Among our staff members, 5 of the chips have been orderd and received from four separate orders. Only one order was placed on letterhead, the other three were personal and one of those was handwritten! All four orders were acknowledged within 10 days and a shipping date was given (Sept. 20 in all cases). By Oct. 6 all 5 chips were in-hand along with some manuals that were ordered. The invoices give a clue to their unusually efficient handling of factory orders. At this time only 5 different items are offered (MCS6501 $20, MCS6502 $25, hardware manual $5, programming manual $5, cross assembler manual $4). The invoices already had these items and prices preprinted so preparing them was essentially a check-off operation. All that seems necessary is a check or money order for the exact total price and a legible listing of the items desired. We sincerely hope that this marketing plan proves successful and that some other manufacturers adopt it. We can do our part by confining information requests to manual orders.

The MCS6501 has been advertised as bus compatible with the Motorola MC6800 with a similar but more powerful instruction set. After looking over the detailed interface specifications we would have to say that the bus is similar to the MC6800 but unlikely to be compatible in any but the simplest systems. The instruction set would be better described as "heavily influenced by the 6800". The user should probably consider the MOS Technology chips as new and different and not try to compare them with the Motorola chip.

TCH will be working with the built-in oscillator version, the MCS6502 and recommends that readers do likewise to avoid clock driver problems. The remainder of this discussion is applicable directly to that chip although all software remarks apply equally to all chips in the family. Interfacing to memory and peripherals seems simple but some problems do arise that tend to dirty up an otherwise clean design. The chip has a READY line but it only works on read cycles; write cycles ignore it and proceed at full speed. This means that although slow erasable ROM's may be used, high speed RAM is needed unless the clock itself is slowed down. One solution (the one used by TCH) is to use 4K dynamic RAM's, the slowest of which are fast enough to avoid problems and cost only $6 to $12 each.

Believe it or not the MCS6502 has three "no connections" on its 40 lead DIP package, a sin punishable by 60 days of 8008 system design! One signal that could have been provided is a "read enable" which would distinguish between active read cycles that actually fetch data and passive ones that occur when internal processing is taking place such as adding a base address to an index register. Memory refreshing could then be done during the passive cycles without slowing the system down. As it is, every non-write cycle must be considered as an active read cycle. The other two no connections could have been connected to selected status bits such as interrupt enable and overflow.

On the other hand a couple of unusual signals are provided that increase flexibility. One, called SYNC, signals instruction fetch read cycles. This can be used to implement a single instruction console control if desired. A better use is in memory refresh and direct memory access circuitry since a write cycle (which cannot be delayed) will never immediately follow an instruction fetch cycle. Another interesting signal is called SET OVERFLOW which will set the overflow status bit when pulsed. The only statement about possible uses given by the manual is that it is reserved for use with future MCS650X family products. This probably means that an extended arithmetic chip is planned that would signal overflow through this line. If overflow status was available as an output, then serial I/O might be implemented with no additional hardware. One other feature that many people might consider a drawback is a non-tri-state address bus, i.e., the address lines from the chip are always on. In a system with much memory or I/O however the address lines will be buffered by TTL. Tri-state buffers and the READY line can be used to provide a tri-state address bus if desired. According to the manual, substantial chip area was saved on the address buffers thus lowering costs.

Other architectural details of the MCS 6502 are similar to those of the MC6800. Input/output is accomplished with the regular memory reference instructions as if the I/O device ports were in fact memory locations. The rich repertorie of memory reference instructions makes this an even more effective method of handling I/O operations. If I/O registers are wired so that they can be both read and written by the CPU, then the modify memory instructions (increment, decrement, and shift memory) can also modify the I/O registers. This could be a powerful tool in handling high speed I/O devices such as digital cassettes or floppy disks, particularly in special purpose systems. A transfer vector containing three full addresses occupies the last six memory locations. One is the start address after a system reset and the other two are the service routine addresses for the regular and the non-maskable interrupts respectively. High memory should therefore be ROM so that system reset and console interrupt (using the non-maskable interrupt) will automatically jump to the debug program in ROM. As in the 6800, the first 256 bytes of memory are very easy to access and are termed the base page. Also, the stack pointer in the MOS Technology chips is only 8 bits long and can only point to locations in the second 256 bytes of memory, thus low memory should be the system RAM. Internal architectural improvements include a degree of "pipelining" which greatly reduces the number of cycles necessary to execute an instruction. Most instructions require only one more clock cycle than memory cycles for execution and some such as the immediates and jumps use no additonal cycles.

In some ways the MC6800 instruction set was trimmed down for the MCS6502 and in others it was beefed up or changed. The MC6800 has a total of eight bytes of internal registers whereas the MCS6502 has only six thus further reducing the chip area. One cut was in the stack pointer (16 bits to 8 bits) and the other cut was in working registers (two 8 bit accumulators and a 16 bit index to one 8 bit accumulator and two 8 bit indexes). It is this shifting around of registers and lengths that make the two machines very different when it finally comes to programming them.

Most programmers will have to make a major adjustment in their thinking about indexing in this machine. The roles of "base address" and "displacement" have been reversed from the norm seen in minicomputers and the Motorola chip. The indexed instructions on the MCS6502 supply a 16 bit base value as part of the instruction itself and then add the 8 bit index register contents to that base value to get the effective address. An immediate problem is seen in manipulating large tables (greater than 256 bytes) since manipulating the index registers alone is not enough to scan the whole table; the base value must be periodically changed. Fortunately, two indirect addressing modes are provided using pointers on the base page so that the base value in the instruction itself need not be modified for large table scans. Programming for small tables and short string moves however is a model of simplicity and efficiency.

There are several other points about the MCS6502 instruction set worth mentioning. Only two arithmetic instructions are provided, add with carry and subtract with borrow. In order to do a regular add or subtract it is necessary to clear the carry first with a one byte, two cycle instruction. If indeed only two arithmetic instructions were allowable then this was clearly a better choice than providing only single precision arithmetic capability on an 8 bit machine. There is a decimal mode bit in the processor status register which causes the arithmetic instructions to consider the operands as two digit BCD numbers rather than 8 bit binary numbers. Since the processor status is saved on the stack during interrupts, there is no problem in getting interrupted during decimal operations. However the interrupt service routine will have to issue a "clear decimal mode" instruction if the main program is expected to ever set it. There is one processor status bit unused in the MCS6502 which could have controlled whether or not the carry flag was included in the arithmetic instructions but according to the manual it is reserved for use in future family members. Jump conditions are somewhat limited as compared with the MC6800 but if speed is important it should be noted that the MCS6502 can do two conditional jumps in the time required for the MC6800 to do one although 4 bytes of memory would be used.

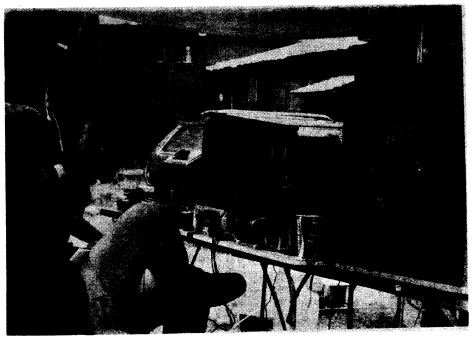
MOS Technology clearly intends to build a software compatible microprocessor family with the MCS6501 and MCS6502 as the first members. New products to be available soon are 3 different 28 pin microprocessors which are performance identical to the 40 pin versions except for maximum memory size (4K bytes) and bus control functions. These chips would be cheaper yet and should find their way into hobbyist peripheral controllers sooner than you think. (Imagine a printer controller using a more powerful microprocessor that the system it connects to!) Also available in November will be a 2mHz 6502 equivalent (twice as fast). Since the required memory access time will be 300NS or less for full speed operation, there should be renewed interest in 4K RAM's since they can easily go that fast. Future plans seem to call for a 16 bit chip in which all of the registers would be 16 bits long and the unused op-codes in the 8 bit machines would be for 16 bit operations. The unusual feature is that memory would still be 8 bits wide allowing the new chip to be a plug in replacement.

In summary, the speed and power of the MOS Technology chips far outstrips their lowball price. The manuals are excellent and should be obtained by anyone interested in further details.

MOS Technology
Valley Forge Corporate Center
950 Rittenhouse Rd.
Norristown, PA 19401

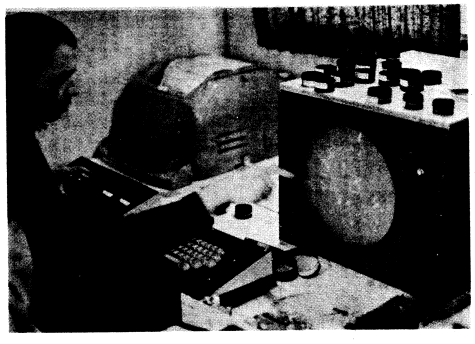
PHOTOS

CHAMBERLIN ON THE ROAD WITH TCH



Somebody is taking pictures of what?
SHELBY, N.C. HAMFEST

And this is how its done.
SHELBY, N.C. HAMFEST



And this is the intersection
where we got off of the interstate.
DAYTON, OHIO HAMFEST

And what else did you plug in backwards?
ARRL CONVENTION, RESTON, VIRGINIA

Several of our readers pointed out some errors in the 8080 version of the cassette ROM. Two of the errors prevent the program from working at all. The corrections are listed below:

LOCATION	WAS	SHOULD BE	COMMENTS
177512	167	161	MOV M,C instead of MOV M,A
177516	042	107	Jump address was wrong
177661	302	312	Read data bit on trailing
177670	312	302	edge of clock

All of the erasable ROM's we have programmed have these corrections. There are no known errors in the 8008 version of the program.

There is an operational problem we have experienced that audio cassette users should be aware of. Most of the cheaper recorders have a very small capstan diameter and rather soft pressure roller. The higher RPM of the small capstan cuts down on required flywheel size and the soft rubber reduces the need for precision mechanical alignment. If the recorder is allowed to sit for a long period of time with the mechanism engaged but the motor off, the capstan can put a dent into the rubber and a crease into the tape. The minor speed wiggle from a dented pressure roller is of no consequence but the crease in the tape can be severe enough to completely lift the tape from the head for several bit times. Since the tape normally stops in the gaps between data blocks, any creases will not be discovered until the tape is erased and reused or data is written on the flip side. Thinner tape should be less subject to creasing but causes other problems with inexpensive recorders. The only real solution is to manually disengage the recorder if it will not be addressed for awhile.

Although TCH has exhausted its supply of 6 volt 3PDT relays for the cassette boards, they are a perfectly standard Potter & Brumfield part. An exact replacement is P&B number KA-14 DY with a typical unit quantity price of $4.50. Similar relays with coil voltages up to 40 volts can be used. The relay supply voltage (+) should be connected to pin 12 of the I/O connection socket on the cassette board.

CLASSIFIED ADS

There is no charge for classified ads in TCH but they must pertain to the general area of computers or electronics, and must be submitted by a non-commercial subscriber. Feel free to use ads to buy, sell, trade, seek information, announce meetings or for any other worthwhile purpose. Please submit ads on separate sheets of paper and include name and address and/or phone number. Please keep length down to 10 lines or less.

FOR SALE: Wanlass Model 200-IC-1 power supply. 3.6 - 6.3 VDC, 25 Amp output, .25% line and load regulation with crowbar protection. Original cost $250. Also Heath Impscope Model EV-3. Both excellent condition. Reasonable offers accepted. David Milhouse, 2823 Griffa Ave., Columbus, Indiana 47201

WANTED: Altair got you down? Sell me your used 8800 and recover some of your hard earned cash. Get back into computing when the smoke clears. Ken Hopper, 4201 S. Bowman Ave., Indiannapolis, Indiana. 307/787-8661

WANTED: Any information on the COQAR System 4. I have some tapes and would be willing to copy and swap them. W. B. Llewellyn, 3523 N. Druid Hills Rd., Decatur, GA 30033

FOR SALE: R.P.C. 4000 Computer system. 8K by 32 bit drum processor with hardware multiply and divide, 60 CPS card reader, 30 CPS card punch, 10 CPS parallel entry teletype. Software includes FORTRAN, PINT, RUNE, PERT-2, ACT IV, and an assembler. All hardware documentation except assembler. $800 at Wilmington Del. You pick up or ship. Also 8K by 32 bit magnetic drum memory, $150. MITS "ALTAIR" computer equipment assembled for 55% of MITS cost. Paul Gumerman, 101 Stonecrop Rd., Wilmington, Del. 19810

WANTED: IMP-16 Assembler. Jim Gaudreault, 7909 14th Ave., Hyattsville, MD 20783. 301/434-2482

FOR SALE: Used 1101A. I have 48 used 1101A's. Each one has been tested and working. All are ceramic DIP. Will sell as a lot for $36 on a first come basis. These would be ideal for a Mark-8 project. Roberto Denis, 11080 NW 39Th St. Coral Springs, FL 33065. 305/752-7067

SALE OR TRADE: HP-55 programable calculator with all accessories for a basic minicomputer (8008 - 8080 - IMP-16 etc.) Contact: Tony Demase, 8108 Westmoreland Avenue, Pittsburgh, PA 15218. Ph. 412/242-5780

POWER SUPPLY: I have a quantity of 5V 6A highly regulated power supplies taken from keyboard terminals. Schematics included and plans for obtaining -5V, -9V, and -12V. $25 plus postage on 15 pounds. Edward G. Runyan, 1146 Nirvana Rd., Santa Barbara, CA 93101.

WANTED: Two used MARK-8's in good working condition for educational use. Units should be enclosed in professional appearing case. Will give $165 each. Indicate prices desired for any accessories. Jordan Kreindler, PhD, 1363 Pine Ridge Rd., Montgomery, Ala. 36109

SERVICE: You want to save $100 on an 8800 but don't want to put it together yourself? Or maybe you need a parallel I/O card but don't want to spend more than $30. You wish you could get that vectored interrupt card with your $40. Well for heaven's sake write: Steve Witham, 168 Painter Road, Media, PA 19063

FOR SALE: GE card reader - 300 CPM. Used, good condition $75 - you pay shipping (about 50lb.) Also complete set of PC boards for Mark-8 (from Techniques) - CPU board, front panel and address latch board have most IC's installed, but no CPU chip - $35 for the set. Donald Bailey, 6 Jay Drive, Concord, NH 03301

FOR SALE: DIAN 9030 teleprinter (RS-232 compatible), used but recently factory-rebuilt and in perfect condition. Prints and transmits full ASCII code at 110, 150, or 300 baud, using 5x7 dot matrix, on standard computer paper from 5" to 15" wide. Why pay MITS $1500 for a TTY when you can get much more in a DIAN for $1300 . . . or make an offer! Call 404/483-4570 after 6:00. Bud Pass, 1454 Latta Lane, Conyers, GA 30207

ALTAIR 8800: SAVE OVER $200 Assembled, tested, and completely operational. Includes 1K RAM board with 256 words installed, complete documentation including news letters. Quality built by electrical engineer. ONLY $500 postpaid. D. Schreiter, 9032 Whitehaven Drive, St. Louis, MO 63123

SURPLUS SUMMARY

For you people in the Illinois area, Wilcox Enterprises has come up with some interesting goodies. Included are IBM input/output Selectrics Model 731 for $750, Dura Business Machines Model 10 (this machine includes a Selectric printer, paper tape reader, and paper tape punch) $390, Flexowriters $150 and $250 for the mono and dual case machines respectively, CDC Typetronic system components including paper tape readers and punches for $75, and receive only Model 33 teletypes for $375. The prices are not steals but they are good. The catch? These units must be picked up, they will not be shipped. Actually this is a good approach since you get to examine before purchase.

Wilcox Enterprises
25 W 178 - 39. Street
Naperville, Illinois 60540

Another source for few of a kind but very worthwhile items is a guy called Gary Coleman. Available items range from tape decks, to modems, to keyboards, to IC's. For a list send a SASE to:

Gary Coleman
14058 Superior Road, Apt. 8
Cleveland, OH 44116

Looking for a CHEEP printer? Here is one way to get one. Richard Page informs us that he has several unusual model 28 receive only teletypes. Unusual? Well they run 115 words per minute, have the fractions character set (have 1/4, 1/2, etc. in place of :, ;, (,), ?, and !), have sprocket feed platens, print 5 characters per inch (36 characters per line), and feed at 2 lines per inch. The other unusual thing is that this well known, rugged machine is being sold for $50. Furthermore an industrious person can purchase and install the parts to convert it back to a standard model 28. For further details contact:

Richard Page
223 East Howard Street
Pontiac, Illinois 61764
815/844-5550

Here is a tip in reverse. Les Veenstra of Action Technical Services has asked us to tell you folks that he is completely out of those rebuilt 33RO's. Apparently people receiving back issue sets are still writing him!

Gary Fishkin of Rochester, NY has informed us that he has some digital cassette decks for sale. They are Interantional Computer Products DigiDeck model 63. He is selling the decks for $100 while they last. You can obtain a complete technical manual directly from ICP for $5 by writing to ICP, Box 34484, Dallas, Texas 75234. To order a deck write to:

Gary Fishkin
Box 349
25 Andrews Memorial Drive
Rochester, NY 14623

THE COMPUTER HOBBYIST
Box 295
Cary, NC 27511

ADDRESS CORRECTION REQUESTED

3rd Class
POSTAGE PAID
Cary, N. C. 27511
Permit No. 35