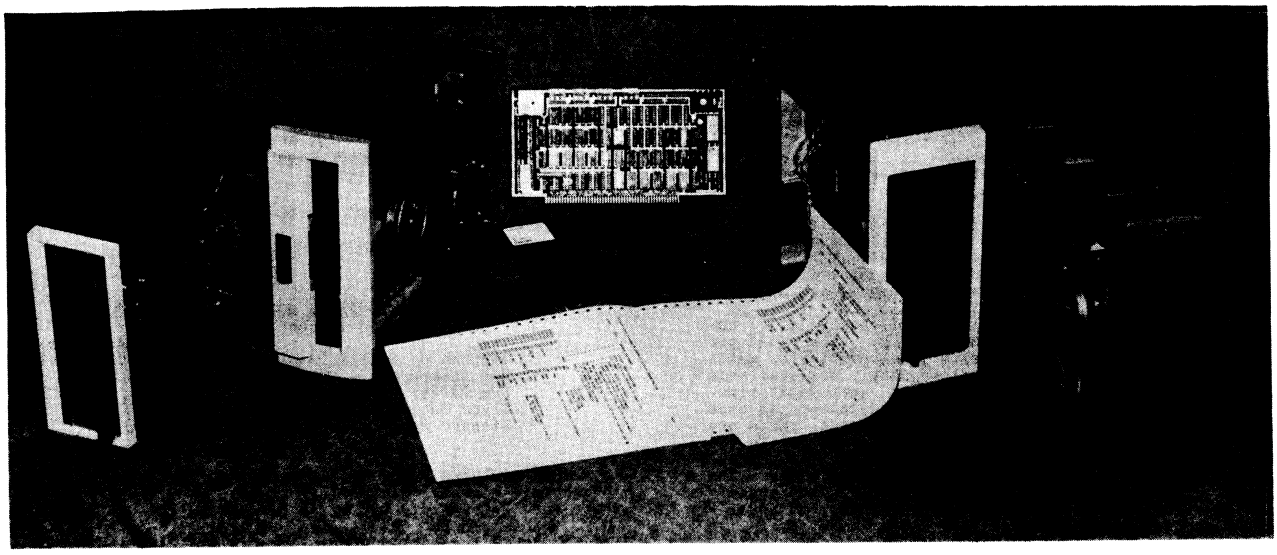
THE COMPUTER HOBBYIST

VOLUME I NUMBER 10 AUGUST 1976



The Complete System! TCH Super Simple Interface, Your Favorite Floppy Disk Drive, and Disk Read/Write Software.

TCH SUPER SIMPLE FLOPPY DISK INTERFACE Part 2.

Judging from the response to part 1 of this series there seems to be a great deal of interest in floppy disk systems. In this issue we will give a complete schematic of the interface along with a full technical description. Also basic programming concepts will be discussed but the complete listing of the 8080 support software will be held until part 3. A printed circuit board is being layed out and 1702/5203 PROM programming will be available but neither will be formally announced until part 3 is published. Please do not order boards or programming until they are formally announced.

Although the controller may be easily wire-wrapped, many readers would much rather have a printed circuit board to cut down on construction effort and minimize the possibility of wiring errors. Thus begins the delimma of choosing a board size, interface configuration, etc. Our usual policy is to make projects operable and equally applicable to any kind of computer from a 4004 on up. On the other hand, a vast majority of readers requested a plug-in Altair (or IMSAI or Polymorphic, etc.) compatible board to even further minimize expense and wiring effort.

After much debate, the following compromise emerged. The circuit supplied on the interface board and the board dimensions would be 100% Altair compatible. Jumpers would be available to adapt the card to nearly any manufacturer's floppy disk drive. Also sockets and addressing logic for two 1702 erasable PROM's would be provided so that the driver software could actually be a part of the interface. In order to provide universal applicability to any computer, the circuitry would be restricted to cover about 75% of the board area. The remaining space would have socket patterns for any additional logic necessary to interface the unit to another computer's bus. Alternatively, connection to a single input and output port (8 bits on input and 4 bits on output) can be accomplished directly. Packaging in the latter case would probably consist of mounting the controller card in a separate case along with the floppy drive and its power supply.

What this amounts to is a very convenient and inexpensive installation for the majority of readers. The remaining readers should not find installation any more difficult than if a "non-specific" board layout such as was done with the audio cassette interface had been used. Although microcomputer bus structures differ widely in details, essentially the same logic functions are required to interface to any of them. Board layout will be such that I/O pin assignments can be easily changed or the edge connector cut off and direct wire connections used instead. While specific instructions cannot be given for most computers (we have no way of testing them), the discussion and explanation of how the Altair interface works should give the reader a pretty good idea of how to connect the unit to another computer. Readers are encouraged to send in connection details and software used in their own non-Altair installations.

Here is a short summary of the interface characteristics. Jumpers are available to adapt the interface to any of the floppy disk drives listed in Appendix 1. Other drives may require the addition of a slight amount of circuitry to meet their signal requirements. The interface can handle either one or two floppy disk drives. For two drive operation, drives with a so-called "daisy chain" interface are required. This means that there is a DRIVE ENABLE line that enables the drive to accept commands and to return status. Thus, two drives may be connected in parallel except for the DRIVE ENABLE line. Two drives without this feature could also be supported but additional multiplexing circuitry would have to be constructed in the free IC matrix area. It should also be possible to support two drives of different make but additional circuitry may again be required.

The interface board contains a data separator and a sector separator. The former decodes the double frequency waveform from the drive read electronics into data and clock pulses. The latter separates the index hole pulse from the sector hole pulses. These circuits require adjustment however which can only be done with the aid of a good oscilloscope. If possible, the reader should try to acquire a disk drive with a built-in data separator and sector separator. This means that any needed adjustments were made at the factory. Also, many drives utilize complex separation circuitry that outperforms the simple one-shot circuits utilized in this interface.

CONTINUED ON 4

EDITORIAL by Hal Chamberlin

What is the first thing that comes into your mind when you see a commercial product price list that looks like this: 1-4 $X, 5-24 $.9X, 25-49 $.75X, 50-99 $.65X, 100-up $.5X? Group purchase, of course! The enticement of that "100-up" figure is directly proportional to the value of X. Unfortunately, the chances of success with a group purchase seem to be inversely proportional to X^2.

Before getting into the economic dynamics of group purchasing, let's try to figure out why the price list is the way it is. One obvious and often cited reason is that the proportional cost of paperwork is less if one order is for 50 units instead of 50 orders for one unit. This is certainly true, however it cannot explain the often hundreds of dollars per unit discount on big ticket items such as printers or disk drives. Further, many OEM manufacturers base their discounts on yearly contracts with multiple small quantity releases allowed spread throughout the year.

A related reason for discounts is the per-unit cost for a salesman's time. Great effort often must be expended in selling an industrial customer on the merits of a device against its competitors.

Another significant cost is field service support. With a large customer, a couple of phone calls may solve a design related problem on 50 units at once. With "oneseytwosey" customers, dozens of individual contacts with a variety of people would be necessary to service an equal number of devices.

The major portion of the larger discounts is, I believe, related to production scheduling. The cost per unit will be much lower if raw materials inventories match demand, the correct number of trained people are available (excess manpower is expensive but so is overtime at time-and-a-half), and the proper automated equipment is used full-time. Remember also that many component parts are bought from price lists that read much like the one in the first paragraph. The ideal situation is an exact knowledge of production requirements up to a year ahead. The chances of a wrong guess are much greater with today's volitile economy and the cost of a wrong guess is equally high with today's interest rates. Also note that list prices are maximums. If the order is very large or follow-on business has the potential of becoming large, further price reductions are likely through negotiation.

So to justify that low 100-up price the supplier is looking to process one order, unsolicited by his sales force, single party responsibility for field service, firm commitments on quantities and delivery dates, and a high probability of repeat business. A successful group purchase will have to give the supplier at least some of these expectations.

Now let's look at the other side of the coin, the group purchase organizer. There is a big difference between drumming up an order for 100 type 2102 memory chips at $1.60 each and getting together 50 orders for LSI-11 CPU's at $650 each. About $32,340 worth of difference. That's big money requiring professional legal and accounting help. Further, no one wants to be left holding that size of bag and the vendor will make sure that he doesn't. All too often the organizer gets cold feet when the orders start coming in and the deal starts taking shape. The vendor starts asking embarassing questions regarding the factors mentioned earlier, particularly field service.

Now let's assume the existence of an organization big enough to tackle a group purchase of a big-ticket item, such as the Southern California Computer Society. For an example I will use the recent LSI-11 group purchase. One quantity price, $990; 50 quantity price, $653; with SCCS overhead, $660. Except for one item, the planning and execution was perfect; big organization, professional help, frank discussions with DEC at the outset, and extensive publicity. Incidentally, individual orders were firmed by a $100 deposit requirement, refundable if the deal fell through.

The sequence of events was roughly as follows: Summer 1975 announcement of plan; first surge of inquiries and orders built confidence; October 31 deadline and not enough orders; order rate trails off short of goal; no communication with initial participants; passage of time (4 months); commercial broker offers LSI-11's for a substantial discount though not as great as SCCS; no communication; some participants threaten or execute withdrawal (deposit checks were not cashed); SCCS admits failure, privately; LSI-11's procured at higher price from broker and distributed to remaining participants.

Note that two failure modes were operating. One, a simple shortfall of interest, which is so obvious as to be often overlooked. The other is the passage of so much time that market conditions had changed and respondants took advantage of the changed market. Lack of communication did little to ease the wait and maintain confidence. Worst of all, the failure of this superbly organized group purchase will greatly discourage both organizers and participants in future efforts.

But let's not let the preceeding discussion discourage group purchases altogether. They can be successful if certain "rules of thumb" are followed. First, don't make it too large. A circle of a half-dozen friends is ideal, certainly no larger than the local club. Remember, the organizer will be responsible for defective unit returns, etc. and should be within easy reach of all participants. Second, get it organized and executed quickly. Time from announcement to order placing should be a very few weeks maximum. If more time is needed to publicize and gather orders, the deal is probably too big. Third, it should be understood by all participants that orders are binding. This policy may reduce interest but will prevent a possible fiasco when collection time rolls around.

So where does that leave the big deals, the ones that can save substantial amounts of money? The LSI-11 effort could not have been any smaller (the $990 price was for 1-49 units). If the price differential is large enough, brokers and distributors will step in, accept the responsibilities, invest a quantity of money, and pocket a portion of the differential for their efforts. This functioning of the free enterprise system is good for it benefits everybody involved.

THE COMPUTER HOBBYIST
Founded October, 1974

Stephen C. Stallings - managing editor
Hal Chamberlin - Contributing editor
Jim Parker - 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. and U.S. possessions. For Canada and Mexico the rate is $7.50 for 12 issues. Subscriptions in all other countries are $11.50 for surface mail and $16.00 for air mail. All back issues are available and cost 65 cents each for persons in the U.S.A., U.S. possessions, Canada, or Mexico. For all other countries back issues cost $1.00 each by surface mail or $1.50 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.

Advertising 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

LETTERS TO THE EDITOR

Dear TCH,

I am thrilled by your idea of a super simple floppy disk. But don't stop at PC cards!! Why not consider a group purchase of disk drives? $450 is a very attractive price! Or perhaps the Southern California Computer Society (being so large) would be willing to co-ordinate it.

Keep up the superb work.

Judson B. Ellmers

Dear Sirs:

I read with interest the floppy disk article in the last TCH. I think before I would be interested in PC boards and programming, I would be interested in finding out if there is a current group purchase effort ongoing. Just the drive is still a big ticket item for most people, and I suspect most potential users would jump at the chance to save $100 to $300. If SCCS or some other club does not step in and volunteer, why not consider organizing such a purchase?

Elmer Beachley

CONTINUED ON 3

Gentlemen,
In response to your article on the floppy disk interface, I think its a great idea! A price of $650 for the drive however is beyond my range. If we could coordinate a bulk-rate deal from the manufacturers to shave off $200 or so, I would seriously consider going that route.
As it stands, though, your tape interface will have to do.

Bob Walden

Gentlemen:
You wanted feedback on the floppy disk article. Here it is:
1. The article was great!
2. How about a dual floppy?
3. I would like a group purchase so we all could have a $400 floppy
4. How about a (nearly) complete kit (floppy, PC board, Altair interface)

Like most of your readers, I wish you could write more sooner, however you don't print garbage so it is well worth the wait. Keep up the good work.

Ken McGinnis, MD

Dear Sirs:
With reference to requests for feedback on floppies - yes, I am interested in boards, PROM's, etc. for a floppy I/F. Also how about a group buy of drives from somebody? Also some words (at least) about dual or even quad floppy disk systems, i.e. what kind of increased controller complexity, power requirements, etc.
Floppies are the only way to go for mass storage for any reasonably sophisticated application, so keep it coming!

Dave Warner

Gentlemen:
I definitely am interested in your floppy disk interface. I will want PC boards if they can be made available (for Altair preferably). I would also be interested in PROM programming assuming your locations would fit in my system. Also, any chance of a group purchase on drives to get the individual cost down?

Dr. Kelley

Gentlemen:
I'm very interested in board for floppy disk controller. I suggest you include a wire-wrap area on the board so it can be customized. I have 2 drives (Memorex 651, the original of all the others) and will want to do multiplexing.

Del Blevins

Gentlemen:
Yours is undoubtedly the best hobbyist computer publication! Now follow up with some ideas that will popularize and standardize your floppy disk.
1. Make the drives available through SCCS group purchase
2. Make the interface electronics on an Altair plug-in PC board
3. Push your floppy in the other hobbyist computer mags to make it a (semi) standard. This is where the tape unit ran out of steam!

Gary Alevy

Dear Sirs:
I am interested in the floppy disk interface cards if they become available, also PROM programs. Can a group order on the drives be arranged; if so I'll sign for 4!!
Keep up the great work.

Jerry Hewett

Gentlemen:
I vote yes for floppy disk PC boards & PROM programming.

B. R. Brunson

This is only a sample of the boxfull of mail we received concerning the super simple floppy disk interface. We took most of the suggestions received into consideration when designing the final version of the interface which is described in the feature article of this issue.
The group purchase is another matter however (see editorial). One new entrant (General Systems International, GSI) into the disk drive field greatly reduces the advantage of a group purchase by virtue of their flat discount curve. Single quantity price on an excellent drive (we bought one and it is indeed excellent) is $550. Also, at the recent Computerfest in Cleveland, one manufacturer's booth was selling brand new, completely compatible drives for $350 each and another was selling them for $400!

CONTINUED ON 10

 CONTINUED FROM 1

An on-board crystal oscillator is provided for non-Altair installations or where an accurate 2mHz source is not available. The oscillator uses a 4mHz crystal which is cheaper and smaller than a 2mHz unit. If a 2mHz clock is available, the parts associated with the oscillator may be omitted.

Also provided on the interface board is a "power on reset" circuit. This prevents the interface from coming up in a write mode for example. When installed in an Altair, the console reset switch will also reset the interface to an idle state when actuated.

The key interface element is the 2048 bit buffer. Associated with the buffer is an 11 bit "buffer pointer" register. The contents of the buffer pointer point to a location in the buffer itself. Commands are available to increment the pointer and to reset it to the beginning of the buffer. Additional commands can be used to write a ONE or a ZERO into the buffer location pointed to by the pointer. When a disk sector is read or written, the bits stream either to or from the buffer by means of hardware controlled automatic incrementing of the pointer after every bit transfer. Two ordinary, cheap 2102 1K RAM's are used in the buffer. Only about 1300 bits are actually needed for a full sector including overhead.

The input port associated with the disk interface consists of 8 "status" bits. Bit 0 indicates the state of the buffer bit currently pointed to by the buffer pointer. Bit 1 is a ONE if the head is currently positioned at track 0; it is a ZERO if the head is at any other position. Bit 2 is a ONE when the index pulse from the disk is detected. Bit 3 likewise senses sector pulses. If the disk drive has the write protect hole sensor and the write protect hole on the disk is not covered up, bit 4 will be a ONE indicating that the disk is write protected. Otherwise it will be a ZERO indicating that writing is permitted. The write protect sensor is usually a $10 to $25 option. Some drives have a "watchdog monitor" circuit that senses illegal commands or drive malfunctions. If this circuit is triggered, it sets a flip-flop in the drive that locks out further operation until it is reset. Bit 5 is connected to this flip-flop and reads as a ONE if it had been set. The bit will always be ZERO if everything is OK or if the drive lacks this feature. Bit 6 is a ONE if the disk drive is ready. The drive is ready only if power is applied, a diskette is loaded, and the door is closed. Most drives include a speed sensor circuit so that ready status is not posted until the disk is up to speed.

All of the preceeding bits except bit 0 indicate the status of the currently selected drive in a dual drive system. Bit 7 is a "select readback" which indicates which disk drive is currently selected. If drive 0 is selected, it is a ZERO, otherwise it is a ONE if drive 1 is selected.

The output port assigned to the disk interface works in a somewhat strange manner. Only the 4 low-order bits (bits 0-3) are used. These 4 bits can specify one of 16 possible commands to the interface. To execute a command, the program would send the appropriate 4 bit command code to the disk interface output port address.

Command codes 0 and 1 are used to store a 0 or a 1 respectively into the buffer at the current location. After the bit is stored, the pointer is automatically incremented to point to the next buffer position. Command code 2 causes the head motor to step out toward the edge of the disk one track position. Command code 3 likewise does a "step in" operation. Code 5 sets the head load flip-flop which presses the head against the disk in preparation for a data transfer. Code 4 resets this flip-flop. Codes 6 and 7 reset and set respectively the "above track 43" flip-flop which controls the signal amplitude during write. Less amplitude is needed for the inside tracks where data is packed more densely. Not all drives actually need this signal however.

Command code 8 is called execute read. When issued, the interface latches the command until the leading edge of the next sector pulse is detected. At this time the bit stream from the disk is directed to the buffer which stores it. The transfer is stopped when another sector pulse is seen. Command code 9, execute write, works in a similar manner except that the bit stream is read from the buffer and written onto the disk. In both cases, hardware controls the exact point where reading and writing is started and stopped thus eliminating concern over exact software response times.

Command code 10 resets the buffer pointer to the beginning of the buffer. Code 11 increments the pointer, Neither command alters the information stored in the buffer. Command code 12 is used to reset the fault latch present in some disk drives. It should only be issed in response to a fault indication in bit 5 of the disk interface status byte. Code 13 is a spare which is not used by the printed interface circuitry. Code 14 causes drive 0 to be selected and code 15 causes drive 1 to be selected. Although drive selection doesn't matter in a single drive installation, the software should still explicitly select drive 0 to allow for compatibility with future expansion.

Now lets run through some disk drive control sequences in order to better understand interface operation and programming requirements. A fundamental task is positioning the head to a desired track number. Positioning to track 0 is easy. First the track 0 status bit should be examined. If it is a ONE, the head is already at track 0 and the operation is finished. If not, a command code 2 should be issued to step the head one position toward track 0. Since the stepping motor in the drive does not respond instantly, a 10 millisecond delay should be allowed before testing track 0 again or commanding another step. The best way to provide this delay independent of CPU speed or external hardware is to wait for two sector pulses to pass the sector sensor. After the delay, the program would loop back and test for track 0 again and take another step if necessary.

If a track number other than 0 is desired, it is necessary to know what track the head is currently at. Then the current track is subtracted from the desired track. If the result is negative, step out commands (code 2) are needed. If the result is positive, step in commands (code 3) are needed. A zero result of course means that the head is already at the desired track. The number of step commands needed is the absolute value of the difference. As before, 10 milliseconds (two sector times) should be allowed between step commands. Also, an additional 10 milliseconds (20 total) should be allowed after the last step for the motor assembly to stop vibrating.

Whenever a new program is loaded, part of the initialization procedure should be to move the head to track 0 and store 0 in the memory location used to hold the current head position. This will insure that the current head position assumed by the program agrees with the actual head position. In a two drive system, a current head position will have to be maintained for each drive.

After the head is at the correct track, it is still necessary to do the read or write operation in the proper sector. The most straightforward approach is to wait until an index pulse is detected and then count the appropriate number of sector pulses. When the head is scanning the sector just before the desired sector, the execute read or execute write command should be issued. After two more sector pulses have passed, the operation is complete. Note that sectors are numbered consecutively with the index pulse occurring in the middle of sector 0. This means that to read or write in sector zero, 31 sector pulses will have to be counted which is nearly a full disk revolution.

Of course the head must be loaded against the disk before a read or a write can be done. In order to minimize disk and head wear, the head is normally lifted during idle periods. Head load times (the time between the head load command and when reading or writing can be done) vary but 40 milliseconds (8 sector times) is generally adequate. This time may be overlapped with other preparatory operations such as stepping or waiting for the desired sector to come around. However, adequate time must be insured in cases where read or write preparation may require less than 8 sector times.

Before a write operation can be performed, the correct bit pattern must be set up in the buffer. The sector data format to be used is quite similar to that employed in the TCH audio cassette system. First there are 128 bits of ZEROES used in this case to synchronize the data separator during read and to allow for variation in sector sensor placement. Following the leading ZEROES is an 8 bit "start of data" ID pattern which is a constant 10010011. Eight bits is generally adequate in floppy disk applications since there is much less garbage at the beginning of a sector to reject than in an audio cassette application. Following the ID are 1024 data bits organized as 128 bytes. The most significant bit of the byte is written first. After the data are 16 bits of cyclic redundancy check (CRC). The CRC is computed and written in the same way as in the audio cassette system (see issue #5). The difference is that the CRC register is initialized to sector number (lower byte) and track number (upper byte) instead of all ZEROES. Trailing ZEROES fill out the remainder of the sector. The disk software is responsible for creating this data pattern in the buffer prior to issuing the execute write command. Note that the buffer pointer must be reset to the beginning of the data pattern before the write is executed.

When reading a sector back, the raw data bits from the entire sector are stored in the buffer starting from the current buffer position. Generally about 1300 bits are transferred. The program should reset the buffer pointer and start scanning the buffer contents until the 8 bit start of data ID pattern is located. If more than a couple hundred bits are scanned without seeing the ID, an error should be signalled since either the ID has been obliterated or the sector has never been written on. It is a good idea to clear the buffer before reading because otherwise a blank sector will leave the buffer contents unchanged and the read routine will process data left over from the previous operation.

After the ID has been located, the CRC should be initialized to the track and sector number just read. Then groups of 8 bits are taken from the buffer, assembled into bytes, stored into memory, and combined with the CRC until 128 bytes have been processed. The two CRC bytes should then be combined with the CRC and the CRC checked for for a ZERO result. If the CRC is ZERO, everything is OK. If it is non-zero, either a data transfer error has occurred or the track and/or sector just processed is not the one desired. Errors of the latter type should never occur if the drive is in good mechanical condition.

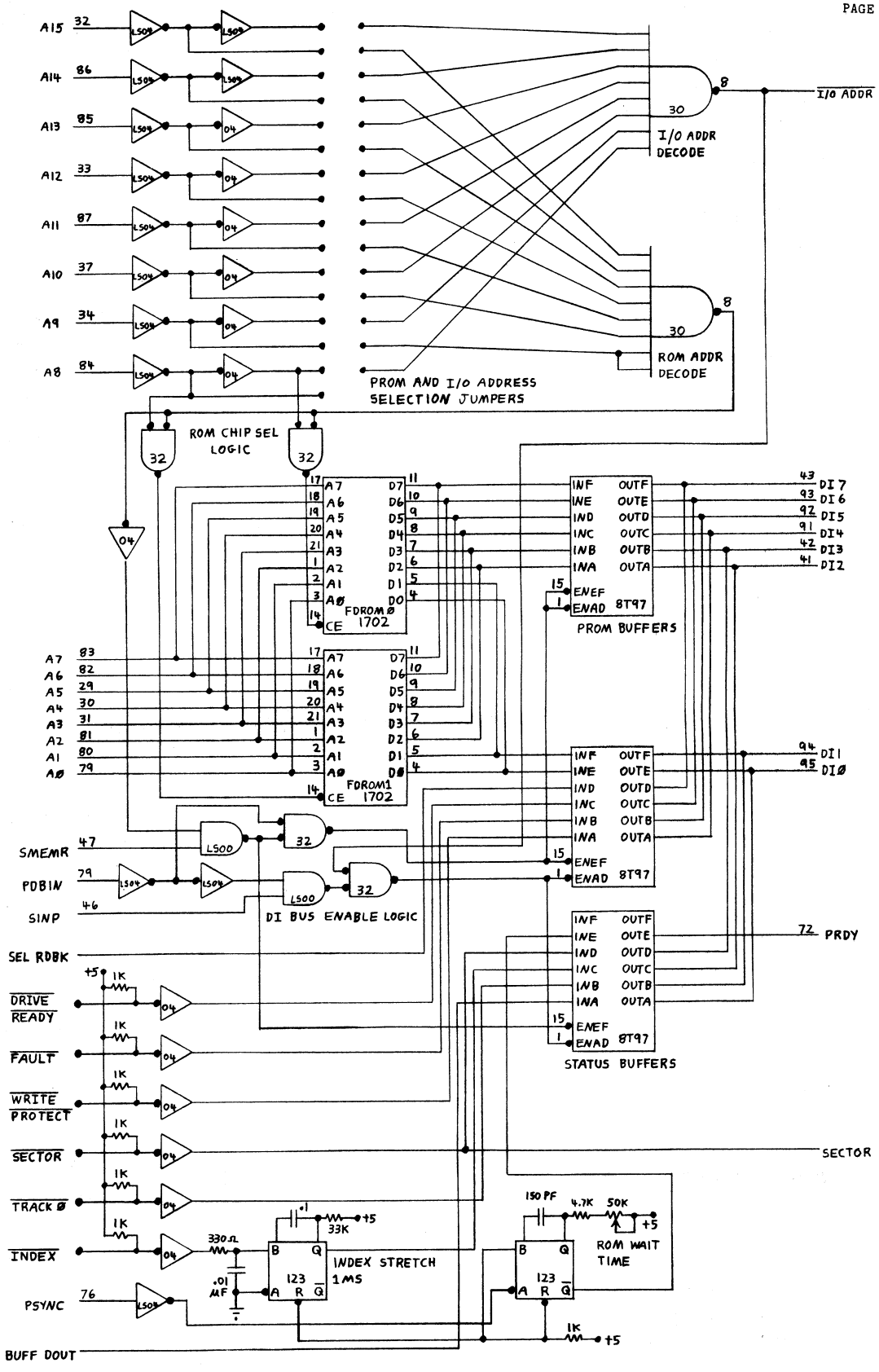


FIGURE 1. ALTAIR BUS INTERFACE

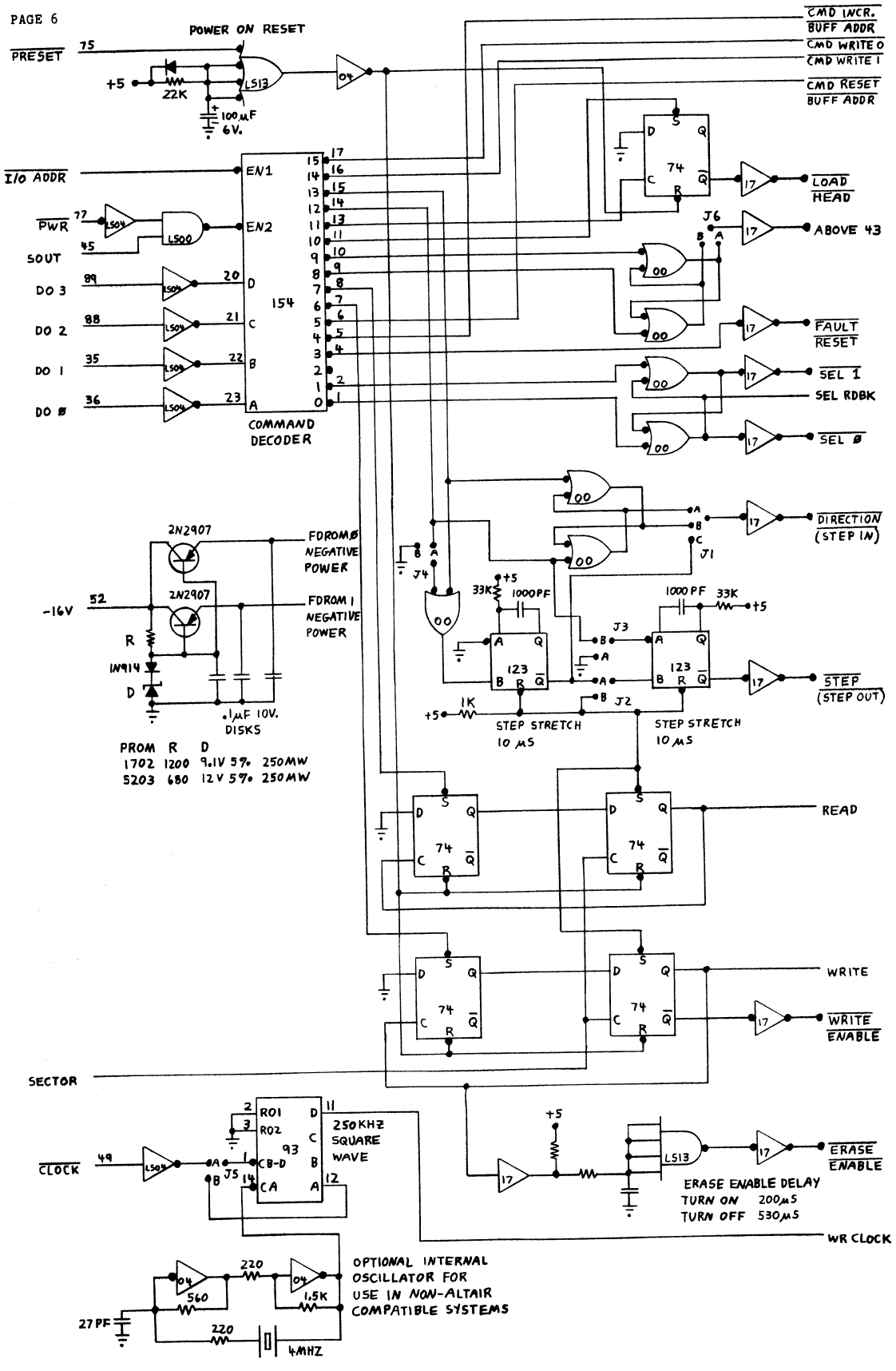


FIGURE 2. COMMAND LOGIC

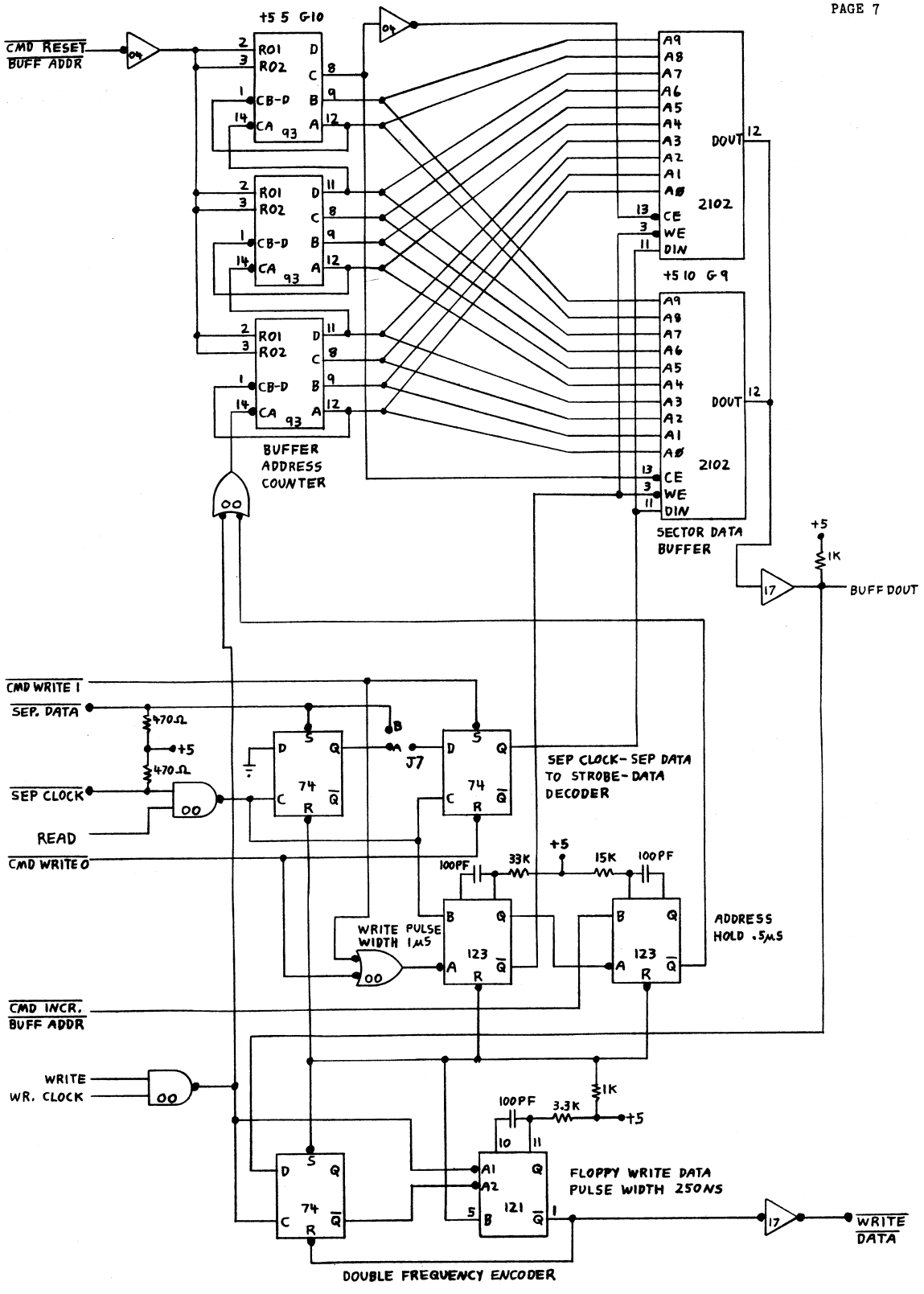


FIGURE 3. BUFFER CONTROL LOGIC

The preceeding discussion covers fairly well the basic programming considerations for the interface. Obviously, what has been saved in hardware complexity has shown up as added software complexity. The interface board has sockets and address decoding for two 1702 type erasable PROMS for a total of 512 bytes. The basic disk software can easily fit into these PROM's and relieve the user of detailed disk programming. This is particularly important when bringing up the system after power on. An "initial program load" routine in the PROM can automatically read track 0 sector 0 from the disk into memory and branch to it. Only three bytes need to be toggled in for a branch to the IPL routine. The remainder of the disk ROM software provides a read and a write subroutine which takes care of all of the operations described. The user need only load the track number, sector number, and memory address into registers and call the appropriate routine.

One problem with a completely ROM based disk support package is that the current track position for each drive needs to be saved somewhere between calls. One possibility is to have the calling routine pass the current track number as an argument and save it in read/write memory on return. However it would be much more convenient if the support package could somehow keep track of the track positions itself without using dedicated RAM locations. The solution employed is to use the buffer in the interface itself to save the current track positions. The track numbers are simply serialized into 8 bits each and saved in the first 16 buffer locations. The current track position values and the disk drives are initialized by calling a control subroutine in the package and requesting a seek to track 0 on each drive. More details on the floppy disk support package will be given in part 3 along with a complete assembled listing.

Now let us take a look at the interface schematic diagram. Several Altair interface concepts that have already been covered in detail in issues #8 and 9 will only be briefly described here. See the referenced issues for a more thorough discussion. Signals that go offpage and have only a name, mate with similarly named signals on other pages. Offpage signals which have a number shown are Altair bus signals. Signals which have large dots at their entry/exit points are generally connections to the disk drive cable.

The address decoding logic is shown at the top of figure 1. The high 8 address lines are double inverted to provide buffered true and complement forms of each bit. Jumpers are placed between the 7430 inputs and true or complement address bits to select the I/O and ROM addresses. The top 7430 decodes the input and output port address assigned to the disk interface. Note that the input and output port addresses must be the same. The other 7430 selects a pair of page addresses for the 512 bytes of PROM. The ROM page addresses must be an adjacent even-odd pair such as 374 and 375 octal. Odd-even pairs such as 375 and 376 cannot be properly decoded.

Two 7432 NAND-NOT gates are used to enable one ROM or the other if the ROM is addressed. The least significant address bit determines which ROM is enabled. Both ROM's are disabled when not addressed thus reducing the power drawn from the negative Altair supply. Eight bits of 8T97 (or 8097 or 74367) are enabled by the coincidence of ROM addressed, SMEMR and PDBIN which gates the ROM data onto the Altair DIN bus. A wait delay single-shot is fired by PSYNC and gated onto the PRDY line by the coincidence of ROM addressed and SMEMR. The single-shot may be adjusted for 0 to 5 wait states which will accomodate PROM's as slow as 3 microseconds. Since the speed of surplus erasable PROMS is seldom known, the single-shot may be adjusted for the fastest reliable operation.

Also on this page is the disk status and input port logic. Status signals from the disk drive are usually in inverted form and require pullup resistors. The 1K resistor value shown is somewhat larger than that recommended by most drive manufacturers. If the disk cable is kept to less than 5 feet, a fair fraction of an amp can be saved by using 1K pullups. The sector separators on some drives put out a very short index pulse. This is stretched to approximately 1 millisecond by a one-shot to insure that the program sees it. In a dual drive installation, all of the output signals from the drive are tied together. The drive select line then determines which drive is enabled to send its status back to the interface. Eight more 8T97 sections gate the status bits onto the DI bus and are enabled by the coincidence of I/O address, SINP, and PDBIN.

The reset signal generator is shown at the top of figure 2. This circuit generates a reset when power is turned on or when the PRESET bus line is held low by the front panel reset switch. At power on time, the 100uF capacitor is not charged and holds the 74LS13 inputs low. The capacitor gradually charges toward +5 and eventually crosses the upper threshold of the Schmidt trigger thus releasing the reset. The diode prevents the capacitor from discharging through the gate inputs when power is removed.

A 74154 decoder converts the command code sent to the output port into one of 16 short pulses. Since the bus DO buffering also inverts the 4 command code bits, the decoder outputs are backwards. Code 0 appears as a pulse at the "15" output of the decoder, 1 at 14, etc.

The commands associated with write current amplitude switching and drive selection simply operate set-reset flip-flops constructed from NAND gates. The head load flop is a 7474 and can be set by command. It can also be reset by command or by the reset signal generator. Type 7417 high current buffers are used to drive the disk input signal lines which can require as much as 40MA each on some types of disk drives. The fault reset command is simply buffered and sent directly to the drive.

The logic associated with stepping the head to different tracks is more complex in order to accomodate a variety of disk drives. Many drives have "step in" and "step out" control inputs. For this type of drive jumper 1 is set to position C, jumper 2 to position B, jumper 3 to position B, and jumper 4 is set to position B. With the jumpers in these positions, the two single shots merely stretch the .5 microsecond pulses from the command decoder into 10 microsecond pulses required by the disk drive.

Other drive types have a "step" line and a "direction" line. To move the head, first the direction signal is set up and then the step signal is activated to make the head move in the selected direction. For this type of drive, jumper 1 is set to position A or B, jumper 2 to A, jumper 3 to A, and jumper 4 to position A. When in these positions, the flip-flop formed from two 7400's remembers which direction the step command was for and drives the direction line to the disk drive. The left single-shot delays the actual step pulse for 10 microseconds to allow time for the direction select to set up in the drive. The right single-shot gives a 10 microsecond step pulse to the drive after the delay. Jumper 1 positions A and B select the polarity of direction select to match the disk drive being used.

Four sections of 7474 flip-flops are associated with the execute read and execute write commands. When the command is decoded, the left flip-flop of the corresponding pair is set. This then conditions the right flip-flop to be set on the leading edge of the next sector pulse. When it is set, the left flip-flop is immediately turned off. The result is that the READ or WRITE signals generated by this logic are on for exactly the full sector time of the sector immediately following the one during which the command was issued. All of these flip-flops are forced off during reset to prevent spurious commands during power up.

The WRITE ENABLE signal to the disk drive is active while WRITE is true. This causes current to be passed through the read/write head which allows writing of new data while the strong magnetic field simultaneously erases old data.

In order to insure some blank space between data tracks to prevent possible overlap, the disk head has two "trim erase" gaps that run parallel to the recorded track along each edge. These gaps trail the read/write gap somewhat and are controlled by the ERASE ENABLE signal. It is necessary to delay ERASE ENABLE some so that only the newly written data is trimmed off. The 74LS13 and associated buffers provides the proper delay to compensate for the gap spacing. The 5% tolerance on the two resistors and capacitor is important to insure proper trim erasing. Also, the capacitor should be a mylar or other temperature stable type; ceramic types should not be used. Some drives have the the trim erase delay built-in and factory adjusted.

The write circuitry requires an accurate, perfect square wave at the bit frequency of 250kHz. In Altair systems, an accurate 2mHz signal is available from the bus. This signal is divided by 8 with a 7493 counter to provide a 250kHz square wave. The J5 jumper may be moved to position B and the components for the internal oscillator installed if a 2mHz source is not available. The crystal should be well within 1% of 4mHz.

The top of figure 3 shows the buffer and buffer pointer register which is an 11 bit counter. The buffer is made of two 2102 static memory chips configured as 2048 words of 1 bit each. The address inputs on the 2102's are connected to the least significant 10 bits of the buffer pointer. The eleventh bit selects one RAM or the other through the chip enable input. A 7417 is used to buffer the memory data output since more than one TTL load is being driven.

The buffer pointer counter is built with 7493 ripple counters. It may be reset with a reset buffer pulse from the command decoder. A 7400 acting as an OR-NOT allows either of two signals from the read/write logic to increment the counter.

The remaining circuitry is probably the most difficult to understand part of the interface. Two D-type flip-flops and two single-shots are used to decode data read from the disk and to control writing into the buffer. When reading from the disk, separated clock pulses clock both flip-flops. The clock pulses are gated by the execute read command so that they do not affect the circuit at other times. Separated data pulses, which are present only for ONES, can directly set the first flip-flop. If a ONE is read from the disk, the data pulse sets the first flop and the next clock pulse transfers the ONE to the second flop while simultaneously resetting the first flop. If a ZERO is read, the first flip-flop remains a ZERO which is transferred on the next clock pulse as before. The clock pulses also trigger a buffer write sequence generator made from two one-shots. The

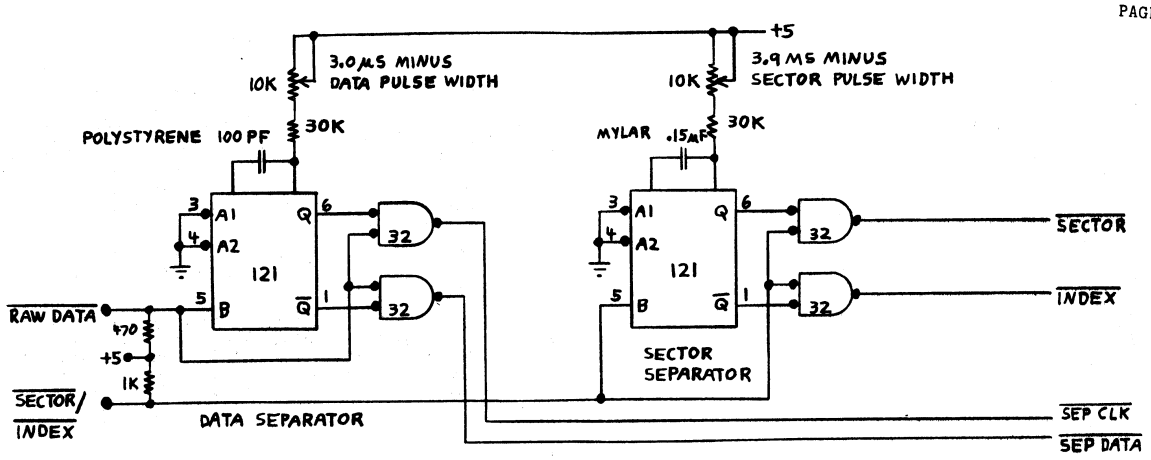


FIGURE 4. DATA AND SECTOR SEPARATORS

first one-shot connects to write enable on the 2102 RAM's and allows them to accept data from the second flip-flop described above. The second single-shot delays incrementing of the buffer pointer until well after the write operation. The timing used is slow enough to allow use of any kind or speed of 2102's.

The write 0, write 1, and increment buffer address command pulses sneak into this circuitry in order to accomplish their functions. Increment buffer address fires only the "address hold" single-shot. Write 0 and write 1 command pulses directly reset or set the second flip-flop and then trigger the write sequence generator.

Writing is not quite so bad. When execute write is active, the 250kHz write clock is gated into the write data encoder. Negative edges of write clock strobe the current buffer output data into a 7474 flip-flop and also increment the buffer pointer to read out the next bit. If a ONE was latched, the single-shot is fired to send a data pulse to the disk drive and the flip-flop is immediately reset to ZERO. If a ZERO was latched, nothing happens. Positive edges of the clock always fire the single-shot for interleaved clock pulses. The disk drive internally converts the write data pulses to magnetic flux transitions on the disk surface.

Figure 4 shows the optional data and sector separators. The components for these may be installed if the drive being used does not have internal data and/or sector separators. The two circuits are identical except for time delay values. A 74121 precision single-shot is used in each. The trailing edge of every clock pulse or sector pulse fires the single-shot which defines a "window" for detecting the data or index pulse. Using the trailing edge and a large timing resistor minimizes recovery time problems with the 74121. Type 7432 NAND-NOT gates route clock and sector pulses to one output and route data and index pulses to the other output under control of the single-shot. When adjusting the single-shots, a good scope is needed. For both circuits, the time should be adjusted so that the single-shot turns off midway between the data or index pulses and the following clock or sector pulses. This time is 3 microseconds minus the data pulse width for the data separator and 3.9 milliseconds minus the sector pulse width for the sector separator.

That about covers the hardware side of the disk interface. Next time a 512 byte ROM based software package for the 8080 will be discussed and formal announcement of available floppy disk related components will be made.

APPENDIX 1. FLOPPY DISK DRIVE CHARACTERISTICS

CHARACTERISTIC	INNOVEX 220	PERTEC FD400	GSI 105	SHUGART SA800	SHUGART SA801	REMEX RFD7400	CDC 9400	CALCOMP 140
STEP METHOD	STEP/DIR	STP I/O	STEP/DIR	STEP/DIR	STEP/DIR	STEP/DIR	STP I/O	STEP/DIR
J1	A	C	A	A	A	A	C	B
J2	A	B	A	A	A	A	B	A
J3	A	B	A	A	A	A	B	A
J4	A	B	A	A	A	A	B	A
UNIT SELECT	YES	NO[1]	YES	YES	YES	NO[1]	NO[1]	YES
READY SENSOR	YES	YES[2]	YES	YES	YES	YES	NO[3]	YES
READY GATED BY SELECT	YES	-	NO[4]	YES	YES	-	-	NO[4]
SECTOR SEPARATOR	YES	NO[5]	YES	NO	YES	OPT[6]	NO	OPT
DATA SEPARATOR	YES	NO	YES	OPT[7]	OPT[7]	OPT	YES	YES[8]
ABOVE 43 NEEDED	J6=B	J6=A	J6=A	NO	NO	J6=A	J6=A	J6=A
ERASE ENABLE NEEDED	NO	YES	YES	NO	NO	YES	NO	NO
WRITE PROTECT	OPT	NO	OPT	OPT	OPT	OPT[9]	NO	NO
FAULT SENSOR	YES	NO	YES	NO	NO	YES	YES	NO
MINIMUM STEP TIME MS	10	10	8	10	10	6	10	6
ADDITIONAL SETTLE TIME	10	20	12	8	8	24	10	10
HEAD LOAD TIME	30	40	40	35	35	50	60	16

NOTES:
1. Multiplexor (2 74157) will have to be constructed in wire-wrap area
2. Switch on door, delay will have to be constructed or operator will have to exercise care
3. Either ignore READY from interface or detect presence of sector pulses with retrig. one-shot
4. Multiplexor (1/2 7450) will have to be constructed in wire-wrap area
5. Gain of sensor may have to be reduced to resolve closely spaced holes
6. Option consists of aperture wheel, allows use of one hole disk for hard sector operation
7. Data separator is standard on model 1 drives
8. Separated data is in NRZ form, move J7 to position B
9. Manually operated switch when disk is inserted

APPENDIX 2. FLOPPY DISK INTERFACE COMMANDS AND STATUS

COMMAND CODES (octal)

0 Store 0, increment buffer pointer
1 Store 1, increment buffer pointer
2 Step head out towards lower track numbers
3 Step head in towards higher track numbers
4 Lift head away from disk surface
5 Lower head onto disk surface
6 Set high write current, tracks 0 - 43
7 Set low write current, tracks 44 - 76
10 Read next sector from disk into buffer starting at current buffer position
11 Write next sector from buffer to disk starting at current buffer position
12 Reset buffer pointer to zero
13 Increment buffer pointer
14 Reset fault condition
15 (spare)
16 Select drive 0 (primary drive)
17 Select drive 1 (secondary drive)

Status bits

0 State of bit at current buffer position
1 Head is at track 00 if bit is a ONE
2 Is a ONE during duration of index pulse
3 Is a ONE during duration of sector pulse
4 Is a ONE if disk in drive is write protected
5 Is a ONE if a fault condition detected
6 Is a ONE if disk drive is ready
7 Indicates which drive was last selected

CONTINUED FROM 3

Dear Sirs:

A number of us have read with great enthusiasm your article on a floppy disk interface. Yes we would be interested in PC boards for such, and are anxiously awaiting the next TCH!

I'm uncertain if you have been informed of our group. We call it the Mid Michigan Microcomputer Group - M^{3}Gr, and it consists of about 20 - 25 members. About 4 to 5 of us have 8008 machines, while there are about 10 8080's (Altairs). I have a MIL MOD-8 with MONITOR-8, TVT-1 with UART connected to the MIL CPU, and a total of 4K memory (2K monitor ROM, 2K RAM). It works very nicely.

We also are interested in your graphics display system. Do you have, or are you planning to manufacture boards of layouts for boards? Are there any updates to your graphics articles?

Thanks for the tremendous TCH! We are looking forward to the next issue. Thanks also for the inside on the "Kansas City Standard".

Richard F. Shultz

An improved (higher speed, better accuracy) graphics circuit is under development. A vector generator board, deflection amplifier board, and CRT power supply board are planned but a few months off. In the meantime, the old circuit slightly modified is now available from SUNTRONIX as a PC board that plugs into their surplus Sanders 720 display units. See the article in the September issue of 73 magazine.

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: 8080 TRACE program - Simplifies and speeds program debug. TRACE enables the user to scan all or selected portions of a program's execution for problem analysis. The PC, SW, and A-L contents are displayed; SW and A-L registers are sense switch selectable. Program listing and description for $7.50. ALTAIR ACR compatible tape included for $10. For additional information send SASE to: R. Rydel, 14021 Cricket Lane, Silver Spring, MD 20904.

FOR SALE: (1) Altair 8800 kit - new, unbuilt sell for $400. (2) MOD-8 - debugged, running with Monitor-8 in ROM complete set of boards with 1702A programming station, all IC's socketed, $300. (3) CREED TTY - $100. (4) Digital Group cassette interface - $20. (5) TVT-1 plus KBD-1 (SWTP keyboard) with UART board modified to work like 32 character 16 line KSR 33! Works nicely with MIL MOD-8 microcomputer above. Send SASE if interested to: R. Shultz, 611 N. Dexter, Lansing Mich., 48910.

PAPER TAPE splices - prepunched for any kind of 8 channel tape. $3.50 per hundred, post paid. H. S. Corbin, 11704 Ibsen Drive, Rockville, MD 20852

FOR SALE: Core stacks - 9 4Kx40 20mil and 2 8Kx36 80mil, $1.00/Kbyte. 5262 2K RAM's, $1.00 eaqh. 1101's, 75¢ (16 only). PC edge connectors, .156" centers, non-standard solder terminals, gold palted: 22x1 - 10¢, 43x2 and 58x2 - $1.50. 5mHz Heath tube type scope, AC coupled, $75. Add shipping on all items. SASE for info. S. Wiebking, 919D Magellan Circle, Dallas, TX 75218. Ph. 214/328-4035.

FOR SALE: 1 inch computer tape at bargain prices! Scotch 861 and 871 on 3600' reels in original boxes. List price $38, my price $6/reel plus postage. Jim Stitt, 311 N. Marshall Road, Middletown, Ohio 45042.

FOR SALE: Univac model 490 computer. CPU with 32K core and power supply. C. E. Fry Jr., Box 507D Geyser Rd., Pittsburgh, PA 15205.

DO YOU KNOW of any minicomputer or microprocessor systems capable of synthesizing or recognising human speech? Any information on hardware or software handling of speech I/O will be appreciated. A summary of all information received will be sent to contributors requesting it. B.W. Klatt, R.R.1, Oliver, B.C., Canada.

FOR SALE: MITS RS-232 serial I/O board for the Altair 8800 (88-SIOA), assembled and tested, $80. MITS 256 word static memory, exapndable to 1K words (88-MCS), assembled and tested, $60. Expander motherboard (88-EC), $8. David Richards, Lawrence Berkley Labortory, University of California, Berkely, CA 94720. LBL ext-5776 or 529-0759 evenings.

FOR SALE: Factory assembled Altair 8800 and Processor Technology 3P+S I/O board - $650. M. A. Keane, 6242 Malvern, Troy, Mich. 48084.

THE BETA terminal owner's group of the Computer Hobbyist Group of North Texas is interested in establishing communications with owners of terminals that use the Univac 0769 series print mechanism. We are looking for Beta keyboards (microswitch # 53SW1-2). One of our members has some spare parts for BETA terminals for sale. Contact L. G. Walker, RT. 1 Box 272, Aledo, TX 76008, Ph. 817/244-1013.

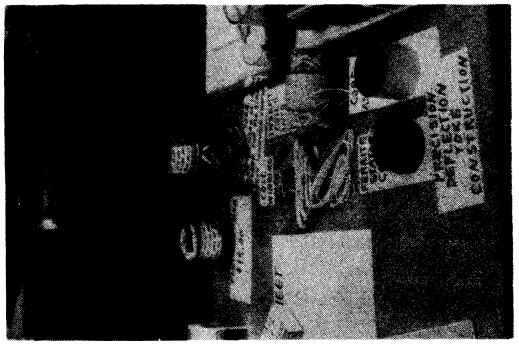
MAG MODULE - Info needed: Ampex 3227339-10F core memory "module" containing nineteen 3227xxx-10 series boards with unknown potted components. Need full schematics, operating voltages, timing; others welcome. Terry Ritter, 2524 Glen Springs, Austin, TX 78741, Ph. 512/441-0036.

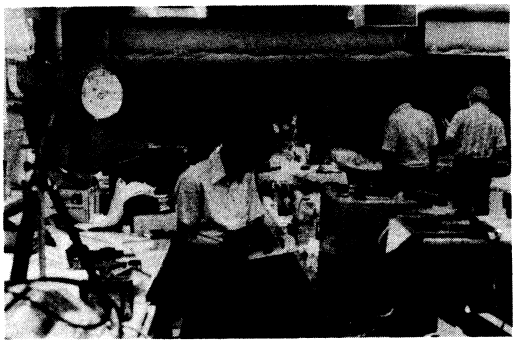
FOR SALE: Teletype DRPE Super high speed paper tape punch - 240 characters/sec standard 8 level 1" tape. New complete $300 plus shipping. New dot-matrix line printer 180 char/sec. Simple parallel interface - $2000. (prices shown are U.S. or Canadian) John Youngquist, Box 122, Fort Erie, Ontario, Canada.

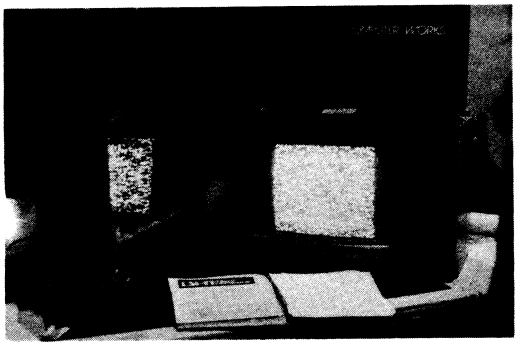
WILL TRADE Bicycling magazines, pre-1950 radio magazines, or comics (1950-1976) for microcomputer magazines or books, or a micro computer, or peripherals. Ian MacMillan, P.O. Box 128, Mount Royal, Quebec, Canada H3P 3B9.

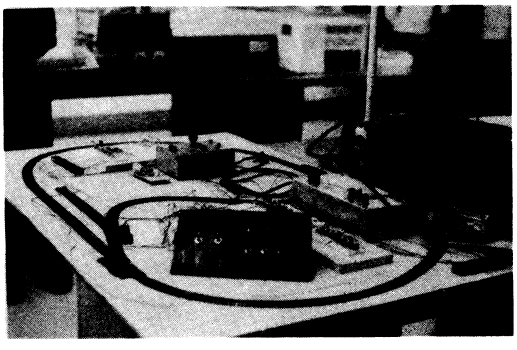
FOR SALE/TRADE - HP5601A medical monitor, HP7848A 8 channel chart recorder, ATL profile monitor 17LP7A tube, with precision yoke, rack mounting, separate power supply, 2HP 1120A 500mHz active probes & 1 HP1122A power supply, all accessable. Motorola SLN state of the art freq. std. 3x10^{-11}, 26VDC aerospace component, F.J.W. infra-red viewer with accessory IR source 6929 tube, very compact, uses one hearing aid battery! TT550/G paper tape reader - optical, 150-2400 baud, all IC circuitry, with literature. More weird & wounderful toys, write. I need - Selectric I/O, floppy disk, many 2102's, teletype #33, printer - mechanics only, YIG components. Ray Kajma Jr., 308 Flordia St., Farrell, PA 16121, Ph. 412/347-1736.

FOR SALE - Computer grade power supply. This is an insurance salvage deal, 20 units are available. Independent outputs - 5VDC @ 12 amps, 15VDC @ 2.8 amps, 15VDC @ 2.8 amps. All outputs are filtered, regulated, and variable. Modern L-shaped open-frame construction. $100 plus 10% for shipping and handling. M. D. Rivers, 28 Leyfred Terrace, Springfield, Mass., 01108.









Some scenes from the world's first ComputerFest
held in Trenton, N.J.