TARBELL

DOUBLE DENSITY FLOPPY DISK INTERFACE

owner's manual

Tarbell Electronics

950 DOVLEN PLACE • SUITE B • CARSON, CALIF. 90746
(213) 538-4251 • (213) 538-2254

Tarbell Double Density Floppy Disk Interface

TABLE OF CONTENTS May 8,1980

INTRODUCTION TO THE TARBELL DOUBLE DENSITY FLOPPY DISK INTERFACE

The Tarbell Double Density Interface is an IBM soft sector floppy disk interface using the Western Digital 1791/1793 Floppy Disk controller chip and built to the IEEE S-100 Standard.

This interface is very similar to the now popular Tarbell Single Density Interface in function, but with many added new features.

These features include operation as either single or double density, or both, direct memory access (DMA), and extended memory addressing capability, with processor speeds of either 2 or 4 Mhz. The on-board BOOTSTRAP feature allows full system memory of 64K by using the PHANTHOM line on the bus.

Software available to run the double density controller is the widely used CP/M(R) disk operating system, and the new MP/M multi-tasking operating system. Both operating systems use an enhanced I/O system utilizing the new direct memory access capability, and automatic density select features that this product offers.

The capacity of the Disk Operating System running under CP/M in Double Density will be 476 Kbytes for an empty disk, or 243 Kbytes Single Density for an empty disk. The break down is as follows:

Double Density:

The disk will appear to CP/M as 77 tracks of 51 sectors, each sector containing 128 bytes. Because the first two tracks are used by CP/M for the operating system, there will only be 75 tracks available for Directory and data storage. This total space then equals approx. 476 Kbytes.

Single Density:

While running Single Density, the disk will appear to CP/M as 77 tracks of 26 sectors, each containing 128 bytes. The first two tracks are used for the CP/M operating system, leaving 75 tracks available for Directory and data storage, about 243 Kbytes of disk space.

It should be noted that the capacities listed above are realized as a result of using the CP/M operating system. If this operating system were not used, then the disk could hold more data, because the disk may be formatted with sectors of 256,512,1024 bytes in length. For further information about this, consult the 1791/1793 data sheet for sector lengths that may be used.

Note: CP/M and MP/M are trademark and tradenames of Digital Reaserch Post Office Box 579, Pacific Grove, California 93950.

SECTION 2: GETTING CP/M RUNNING WITH THE INTERFACE

One of the major problems confronting the implementers of new micro-computer systems, has been the lack of input/output (I/O) standards. The emergence of Digital Research's CP/M(r) disk operating system as a standard I/O environment has contributed greatly to aleviating this problem. Now the problem is reduced to implementing CP/M on the target hardware system, which consists of tailoring the BIOS part of CP/M to the situation. Unfortunately, since we can't assume any particular console interface at the factory, there is no way to make the system generation completely automatic.

Because of all the different possible system configurations, and because we try to update our hardware and software as quickly as possible, it has been difficult to create and maintain a set of documentation that is useful and correct for getting our double-density floppy disk interface working under CP/M. These instructions represent a major rewrite effort in this direction. We hope that most of the faults in the earlier instructions have been corrected in this set.

These instructions explain how to get the Tarbell Double Density Floppy Disk Interface going with Digital Research's CP/M 1.4 or 2.x disk operating system. It is important not to try and make more than one change in your system at a time. For example, if you wish to go from our single density interface operating under CP/M 1.4, to our double density interface operating DMA under CP/M 2.2 with a different memory size, DON'T try to do it all at once. First the single-density to double density, then to DMA, then to 2.2, then to different memory size.

Be sure that the title of the instructions you are going to use, matches the situation you have. If it doesn't, and you can't seem to find one that does match, call or write to us, and we'll try to help. If you don't think you are capable of carrying out the required instructions yourself, we can generate a customized system for you. Just have us send you an I/O tailoring questionaire. The cost is usually about $50.

INSTALLATION NOTES

1. If you have a revision D board (revision letter is at bottom of board), and the board has been modified according to earlier addendums, make a cut on the trace to the right of U41, which is the second trace down. The traces above and below this trace should already be cut.

2. The MWRITE option available on our board is only for those computers where the MWRITE is not generated directly from the bus signals PWR and SOUT. It is not intended as a substitute for the normal MWRITE line, which must be implemented somewhere on the bus (usually on the CPU or front panel).

3. Install the double density board as close to the CPU as possible, between the CPU and memory board(s).

INSTRUCTIONS FOR GETTING THE TARBELL DOUBLE DENSITY INTERFACE OPERATING WITH CP/M 1.4 WHEN YOU HAVE CP/M 1.4 ALREADY GOING ON A TARBELL SINGLE-DENSITY INTERFACE

1. First make sure that your situation matches the title above. If it doesn't, find another sheet that does match.

2. Check the option jumpers on your double-density interface board against the manual to make sure the board is addressed for E0 through F8 (hex), and that all other options are correct.

Use your current single-density interface, operating under CP/M 1.4 to do the following steps:

3. Use the FORMAT91 program on the public domain #2 disk (provided with the interface) to format at least two disks. DON'T use any of your old format programs to do this. When it says "READY TO FORMAT?" be SURE to get the public domain disk out of there before typing Y. Test the disks using the DISKTEST program.

4. Put one of the newly formatted disks in drive B. Put a disk with your normal CP/M 1.4 system and system programs in drive A. Now perform the following steps:

a) logged into drive A, type SYSGEN. Answer source as drive A, destination as drive B. Reboot.
b) type PIP with no arguments, then the following steps.
```
*B:=A:DDT.COM
*B:=A:ASM.COM
*B:=A:SYSGEN.COM
*B:=A:ED.COM
```
c) while still in PIP program, remove your system diskette from drive A, and insert into drive A the Public Domain #2 diskette that came with the double-density interface. Then continue as shown below:
```
*B:=A:ABIOS24.ASM
*B:=A:DBOOT24.ASM
```

5. Now take out the public domain disk #2 and put it aside. Take the newly formatted disk out of drive B and put it into drive A. Boot up on it. It should come up normally, since a copy of your system was just put onto it.

6. Using ED.COM, edit the ABIOS24.ASM to change the EQU's for your memory size, console, printer, drives, etc. Leave the DMACNTL and DUBSID EQU's set to FALSE. Set the MSIZE EQU to the same size as the CP/M 1.4 system you are now running on this disk. Be sure to set the console port numbers correctly. Exit from the editor. Rename the file to ABIOSxx.ASM, where xx is your MSIZE.

8. Assemble ABIOSxx with ASM.COM. Print the .PRN file if desired, then erase it.

9. Using ED.COM, edit DBOOT24.ASM. Set the MSIZE EQU to the

size used above. Leave the DOUBSID, DOUBDEN, and DMACNTL EQU's set to FALSE. Exit from the editor. Rename the file to DBOOTxx.ASM.

10. Assemble DBOOTxx.ASM with ASM.COM. Print the .PRN file if desired, then erase it.

11. Use SYSGEN to put a copy of your current CP/M 1.4 system onto the disk as a file. When it asks for source, answer A. When it asks for destination, press carriage-return to reboot. Then do a SAVE 32 CPMxx.COM, where xx is your system size.

12. Use DDT to bring in the CPMxx.COM file and to overlay the BIOS and BOOT hex files onto it. Type DDT CPMxx.COM . Then type IABIOSxx.HEX . Then type Rbias where bias is in the table below:

xx	bias	xx	bias	xx	bias	xx	bias
20	D480	24	C480	28	B480	32	A480
36	9480	40	8480	44	7480	48	6480
52	5480	56	4480	60	3480	64	2480

Now type IDBOOTxx.HEX . Then type R900 . Then do Ctl-C.

13. Next enter SYSGEN. When it asks for source, press return to skip. When it asks for destination, type A. At this point you may write this system onto more than one disk. After you are finished writing onto the disk(s), DON'T press return to reboot.

14. You can now shut off your computer, remove the single density interface, and put the double-density interface in. Then turn your computer back on.

15. The system you have just written onto one or more disks should now boot up correctly on the double-density interface. If it doesn't, check over the BIOS and BOOT .PRN files to make sure all EQU's were set correctly. Check your board to verify again that all the jumper options are right. If you still can't get it going, read section 2-3 of these instructions.

16. If the system does come up correctly, congratulations! You are now running the double-density interface in non-DMA mode. If you want to operate double-density next, see section 2-2 of these instructions. If you want to try operating in DMA mode, go to step 6 in this section, changing the DMACNTL EQU to TRUE in both the BIOS and the BOOT .ASM files. The rest of the instructions are the same.

17. Finally, if you notice any errors in this documentation, PLEASE call or write about it.

HOW TO MAKE THE TARBELL DOUBLE DENSITY INTERFACE OPERATE IN THE DOUBLE DENSITY MODE ASSUMING YOU HAVE THE DOUBLE DENSITY INTERFACE OPERATING IN THE SINGLE DENSITY MODE.

1. Check your situation against the title above. If it doesn't match, look for other instructions that do. In order to operate in double density mode, you will either need to be operating at 4 or above 4 Mhz (Z80 or 8085), or you need to be operating in DMA mode. To set DMA mode, see step 16 of the instructions in section 2-1.

2. Format some disks double density with DFORMAT, and test them using DTEST.

3. If you boot up on a single density system which was created using the auto-select I/O section (ABIOS or 2ABIOS), all you have to do is put the formatted double-density diskette in drive B. Files may be transferred to the new double density disk using PIP. Don't try to use the COPY utility to copy from single density to double density or vise-versa.

4. If you want to put a system from the first two tracks on the single density disk onto a double density disk, SYSGEN alone will not work. This is because the first sector of the first track contains a byte which has to be DD (hex) for double density, and your single-density disk doesn't have that byte. To perform this operation correctly, follow these steps:

a) On your single density disk, edit the file called DBOOTxx.ASM to change the DOUBDEN EQU from FALSE to TRUE. It is important that the MSIZE match your current CP/M system size (xx).
b) Assemble the new file: ASM DBOOTxx
c) Do a SYSGEN, answering source on A, skip the destination and reboot. Enter SAVE 34 CPMxx.COM where xx is system size.
d) Then overlay the CPMxx.COM system image with the new DBOOT:
```
DDT CPMxx.COM
IDBOOTxx.HEX
R900
```
e) Then press control-C to return to CP/M.
f) Then do another SYSGEN, this time skipping the source, and answering B to the destination. (This assumes you still have your double-density disk in B.)

5. Now you can take the double density disk out of drive B and put it into drive A and boot up on it.

WHAT TO DO IF YOUR TARBELL DOUBLE DENSITY FLOPPY INTERFACE IS NOT WORKING

1. Recheck the jumper options on the interface board against your manual in section 6. Note that manuals of boards rev B and earlier have an error in the board addressing section. People with these manuals can get a new manual free by sending us the cover of their old manual.

2. Recheck the EQU's in the BIOS and BOOT .ASM files to make sure that all are set correctly.

3. If you have a friend with a working Tarbell Double Interface, try using your interface in his computer. If his works and yours doesn't, there is probably something actually wrong with your interface. If so, you might want to consider sending it back to Tarbell for repair. If your interface does work in your friend's computer, the problem might be in your software, or in some other component of your system. Just because the other components of your system work under other circumstances, doesn't mean that there is nothing wrong with them.

4. Another thing to check is the diskette that you're using. Is it formatted correctly? How do you know it is?

5. Do you have dynamic memory in your computer. If so, how is it refreshed? It is possible that the way it is refreshed interferes with our interface, or that the way our interface works interferes with the memory's refresh circuitry.

6. Does your CPU board fully implement the new IEEE S-100 standard? In particular, does it use pin 67 (the phantom line) for anything besides phantom? Does it implement the control-disable, data-disable, and status-disable lines? Does it implement the PSYNC, PHOLD, and PHLDA lines? Neither the SDS SBC-100 or SBC-200 CPU boards meet this requirement.

7. Does the memory which occupies address 0000 in your system have a phantom line on pin 67?

8. Do you have other boards in your system that use the XRDY and PRDY lines (pins 3 and 72) besides the Tarbell interface and the CPU? If so, it might be best to disconnect those lines completely.

9. Since the Tarbell Double Density Floppy Disk Interface uses lines on your motherboard that aren't normally used, some of these lines could be shorted or open, or the connector pins could be dirty.

10. Check your system power supply, with a scope if possible, to make sure that all your voltages are steady, clean, and the right level, both on the drives and the motherboard. It is very important that on the drive power supply, the 24 volt, 5 volt, and -5 volt returns be connected together at the power supply end.

11. If you are having problems with the bootstrap, it's possible that C17 is not a high enough value to reduce the effects of ringing on the bus. You might try 220, 390, 470, or 680 pf capacitors, in that order. The symptom is that the bootstrap flip-flop gets reset before it has a chance to read a complete sector. This can be seen by looking at pin 19 on the 8257. This DRQ line should be a series of short pulses that should happen over a period of about 2 ms. If they don't last that long, you may have this problem.

12. If the DRQ line mentioned above never goes high at all, that means the interface is never receiving a valid data byte. This could be caused by a variety of factors, including a bad data separator component, bad 1793, bad drive, etc.

13. If the interface is picking up excessive errors after warming up, it could be the 1793. We are now testing these IC's more carefully.

*** NOTE ***

If you decide to send the interface back for repair, be sure to include a copy of your receipt, showing the date you bought it. Note that only the parts are warrantied on kits.

INSTRUCTIONS FOR GETTING THE TARBELL DOUBLE DENSITY INTERFACE OPERATING WITH CP/M 2.x WHEN YOU HAVE CP/M 2.x ALREADY GOING ON A TARBELL SINGLE-DENSITY INTERFACE

1. First make sure that your situation matches the title above. If it doesn't, find another sheet that does match.

2. Check the option jumpers on your double-density interface board against the manual to make sure the board is addressed for E0 through F8 (hex), and that all other options are correct.

Use your current single density interface, operating under CP/M 2.x to do the following steps:

3. Use the FORMAT91 program on the public domain #2 disk (provided with the interface) to format at least two disks. DON'T use any of your old format programs to do this. When it says "READY TO FORMAT?" be SURE to get the public domain disk out of there before typing Y. Test the disks using the DISKTEST program.

4. Put one of the newly formatted disks in drive B. Put a disk with your normal CP/M 2.x system and system programs in drive A. Now perform the following steps:

a) logged into drive A, type SYSGEN. Answer source as drive A, destination as drive B. Reboot.
b) type PIP with no arguments, then the following steps.
```
*B:=A:CPM.COM
*B:=A:DDT.COM
*B:=A:ASM.COM
*B:=A:SYSGEN.COM
*B:=A:ED.COM
```
c) while still in the PIP program, remove your system diskette from drive A, then insert into drive A the Public Domain #2 diskette that came with the double-density interface. Then continue as shown below:
```
*B:=A:2ABIOS24.ASM
*B:=A:2DBOOT24.ASM
```

5. Now take out the public domain disk #2 and put it aside. Take the newly formatted disk out of drive B and put it into drive A. Boot up on it. It should come up normally, since a copy of your system was just put onto it.

6. Using ED.COM, edit the 2ABIOS24.ASM to change the EQU's for your memory size, console, printer, drives, etc. Leave the DMACNTL and DUBSID EQU's set to FALSE. Set the MSIZE EQU to the same size as the CP/M 2.x system you are now running on this disk. Be sure to set the console port numbers correctly. If you have Shugart 800 drives, don't set the step rate any faster than 10 ms. Exit from the editor. Rename the file to 2ABIOSxx.ASM, where xx is your MSIZE.

8. Assemble 2ABIOSxx with ASM.COM. Print the .PRN file if desired, then erase it.

9. Using ED.COM, edit 2DBOOT24.ASM. Set the MSIZE EQU to the size used above. Leave the DOUBSID, DOUBDEN, and DMACNTL EQU's set to FALSE. Exit from the editor. Rename the file to 2DBOOTxx.ASM.

10. Assemble 2DBOOTxx.ASM with ASM.COM. Print the .PRN file if desired, then erase it.

11. Use SYSGEN to put a copy of your current CP/M 2.x system onto the disk as a file. When it asks for source, answer A. When it asks for destination, press carriage-return to reboot. Then do a SAVE 34 CPMxx.COM, where xx is your system size.

12. Use DDT to bring in the CPMxx.COM file and to overlay the BIOS and BOOT hex files onto it. Type DDT CPMxx.COM . Then type I2ABIOSxx.HEX . Then type Rbias where xx is MSIZE and bias is in the table below:

xx	bias	xx	bias	xx	bias	xx	bias
20	D580	24	C580	28	B580	32	A580
36	9580	40	8580	44	7580	48	6580
52	5580	56	4580	60	3580	64	2580

Now type I2DBOOTxx.HEX . Then type R900 . Then do Ctl-C.

13. Next enter SYSGEN. When it asks for source, press return to skip. When it asks for destination, type A. At this point you may write this system onto more than one disk. After you are finished writing onto the disk(s), DON'T press return to reboot.

14. You can now shut off your computer, remove the single-density interface, and put the double-density interface in. Then turn your computer back on.

15. The system you have just written onto one or more disks should now boot up correctly on the double-density interface. If it doesn't, check over the BIOS and BOOT .PRN files to make sure all EQU's were set correctly. Check your board to verify again that all the jumper options are right. If you still can't get it going, read section 2-3 of these instructions.

16. If the system does come up correctly, congratulations! You are now running the double-density interface in non-DMA mode. If you want to operate double-density next, see section 2-2 of these instructions. If you want to try operating in DMA mode, go to step 6 in this section, changing the DMACNTL EQU to TRUE in both the BIOS and the BOOT .ASM files. The rest of the instructions are the same.

17. Finally, if you notice any errors in this documentation, PLEASE call or write about it.

GETTING THE TARBELL VERSION OF CP/M 1.4 OR 2.X RUNNING ON YOUR TARBELL DOUBLE-DENSITY FLOPPY DISK INTERFACE WITHOUT A CURRENTLY RUNNING CP/M SYSTEM OF ANY KIND

1. First make sure that your situation matches the title above. If not, you may find that another set of instructions will get your system going sooner.

2. You need to have the following hardware installed:

   a) An assembled and tested Tarbell Double Density Interface
   b) At least 24k bytes of random access memory, of which at least the first 32 bytes can be disabled by phantom line pin 67 going low.
   c) A Z-80, 8085, or 8080 CPU board which conforms to the IEEE S-100 standard.
   d) A console interface of some type, preferably not memory-mapped video, which supports an alphanumeric keyboard and a CRT display or teleprinter. If possible, this interface should be addressed for status on port 0, data on port 1, with bit 0 of the status low meaning keyboard ready, and with bit 7 of the status low meaning CRT display ready. If these port and status requirements are met, the Tarbell CP/M 1.4 or 2.x disks for the DD controller should boot up with no further work. Just put the disk in, push reset, and run. Skip to step 8 if so. If not, you will need to fulfill the requirements of substep (e) below and continue.
   e) Either a front panel or a ROM monitor (any ROM should be outside the 24k RAM), which allows depositing bytes into specified RAM addresses and executing at an address.

3. If possible, have a friend make a copy of your original CP/M disk, and don't use it except to make further copies. Then use the copy for the following steps.

4. Turn the computer on, then the CRT-keyboard, then the drive power.

5. Put the CP/M disk into the disk drive (on most drives, the label on the disk should face the door of the drive). Close the door. Push reset (and run if you have one) buttons on the computer.

6. The head should load against the disk and move in one track. If it doesn't do this, something is wrong with the hardware setup, and you should try a few times more. If it still doesn't do it, FIRST remove the diskette, then shut down the system. Something is either wrong with the hardware or the diskette. If so, have someone look at it or call Tarbell. If it does load and step ok, go onto the next step.

7. Either stop the computer from running, if you have a front panel, or jump into your ROM monitor, if you have one.

8. Look at the BIOS (Basic Input Output System) listing that

came with our CP/M. Find the label BOOT. After the LXI SP instruction, you will see a series of NOP's. This area is reserved for initializing console interfaces that require it. Using either front panel or ROM, deposit the initialization routine required, if any, at the address indicated by the listing. There should be a copy of any required initialization routine in the manual on your console interface. Assembly language code for the initialization of some common console interfaces can be seen in the following lines on the page.

9. Still looking at the BIOS listing, find the label CONST. Examine the code there for our "standard" interface. Put the code here to do a status check on your console interface. Notice that if your status bits are true when high, instead of low like ours, you will need to change the RNZ to an RZ. Other changes which might be required are the port number after the IN, and the mask after the ANI. Check your console interface manual for examples and instructions.

10. The next routine is labeled CONIN. Deposit the code to read a byte from your console keyboard into register A. Notice that you might need to make similar changes, such as JNZ to JZ, mask, and port numbers.

11. The last routine to change is labeled CONOT. Deposit the code to write the byte in register C to your console. Again, you might need to replace our JNZ with a JZ and make port number and mask changes. Be sure to end each of these routines with an RET instruction.

12. This should be all the patches you need to make to the CP/M system residing in memory, to get going temporarily. Now examine the content of address 5A00 (hex), which should be a C3 (hex for JMP) and execute (run) at that location.

13. Our BIOS should give you an opening message. If so, you're on the air, so go to step 14. If not, the system may not have loaded properly, and something may be wrong with the diskette or hardware setup. In that case, refer to section 2-3.

14. If you haven't already done so, copy the system and files onto another disk. In order to do this, keep your system disk in drive A and put a blank disk into drive B. Then type: COPY ALL. This will copy your original disk onto the blank disk. Note that the system you are running is only in memory, and the system on the disk hasn't yet been modified. Leave the new disk in drive B until you press return to reboot. Then take the original disk out of drive A and never use it again except to copy it. Now remove the copy you made from drive B and label it exactly the same as the original. You will find that it is important to keep the disk labels current, as it is easy to get confused and make a mistake. Put the new copy into drive A for further work. Then press Ctl-C.

15. The next thing to do is edit the BIOS and BOOT .ASM files and overlay them onto your system. Use the method described in the

Tarbell CP/M 1.4 or 2.x User's Guide, as this will properly document all your changes and allow you to make use of memory larger than 24k.

16. The latest ABIOS (Auto-density select Basic Input Output System) is always available from Tarbell for $15. Just ask for Public Domain Disk # 2.

CONTENTS OF PUBLIC DOMAIN DISK #2 May 2, 1980

This is a disk which is regularly updated with our latest ABIOS and 2ABIOS for the Tarbell Double Density Floppy Disk Interface. Other utilities are also maintained on this disk, such as format and test routines. We also had room to include the source for the FORTH language from the Forth Interest Group. Their name and address are included on the .ASM file. Following is a short description of each file. For further information, see the comments in the file itself, or the Tarbell CP/M User's Guide. The latest version of this disk is always available within 1 week from Tarbell for $15. If you are having problems, it's always wise to see if there is a newer version of this disk available than the one you have.

1. DBOOT24.ASM

THIS IS THE SAME AS THE NORMAL SBOOT PROGRAM USED IN SINGLE DENSITY, WITH THE EXCEPTION OF A DENSITY CONDITIONAL STATEMENT. AT THE PRESENT TIME SETTING THIS EQU TO EITHER TRUE OR FALSE WILL MAKE NO DIFFERENCE TO THE BOOTING OPERATION. THE REASON IT IS THERE IS FOR DOING AUTO-DENSITY SELECTION DURING BOOTING.

2. DDUMP.ASM & DDUMP.COM

THIS IS A MODIFIED DUMP.COM FROM SAM SINGER AND FROM THE CP/M USERS GROUP. THIS PROGRAM WILL ALLOW YOU TO VIEW ONLY TRACKS 2 - 76 OF A DOUBLE DENSITY DISK. THE ONLY LIMITATION IN THE PROGRAM IS THAT IT WILL NOT DUMP BY GROUP NUMBERS. ALL OTHER FEATURES ARE USEABLE.

3. DFORMAT.ASM & DFORMAT.COM

THIS PROGRAM IS THE DOUBLE DENSITY FORMAT PROGRAM. IT WILL FORMAT THE DISK IN MANY WAYS DEPENDING ON WHICH EQU'S YOU SET TRUE OR FALSE.

For example:

SETTING TRK1SD = TRUE,
FORMATS TRACK 0 AND 1 SINGLE DENSITY.
TRACK 2 - 76 DOUBLE DENSITY.

OR

SETTING TRK1DD = TRUE,
FORMATS TRACK 0 IN SINGLE DENSITY.
TRACKS 1 - 76 DOUBLE DENSITY.

*** NOTE ***

ONLY ONE (1), TRK1SD OR TRK1DD MAY BE TRUE AT A TIME. THE SKEW FACTOR IS EASILY CHANGEABLE IN THE DFORMAT PROGRAM TO ALLOW FOR OPTIMIZING DISK SPEED.

AS SIDE SELECT IS ONLY SUPPORTED BY CHANGING YOUR DRIVE SELECT IF YOU HAVE DOUBLE SIDED, FORMATTING SIDE B WITH THE DFORMAT PROGRAM MAY OR MAY NOT WORK FOR YOU.

4. DTEST.ASM & DTEST.COM

THIS PROGRAM IS USED TO TEST A DOUBLE DENSITY DISK FOR ERRORS. WHEN THE PROGRAM FIRST COMES UP IT WILL ASK YOU FOR A "TITLE:". YOU

MAY TYPE IN ANYTHING YOU WANT SUCH AS <FORMATTED WITH 62.5 NSEC,187.5 NSEC> AND THEN A CNTL-P, CARRIAGE RET, OR IF YOU DON'T WANT TO TYPE ANYTHING, JUST TYPE A CARRIAGE RET. THE TITLE ALLOWS YOU TO KEEP A RUNNING TAB ON THE ERRORS AND USING CNTL-P WILL TURN ON THE LIST DEVICE FOR MAKING A HARDCOPY LISTING. THE NEXT QUESTION WILL BE STARTING TRACK. YOU MUST ANSWER THIS WITH A TRACK NUMBER OF 0 OR GREATER. THE REST OF THE PROGRAM SHOULD BE CLEAR. THIS PROGRAM READS A TRACK AT A TIME AND KEEPS A RUNNING TAB OF ERRORS FOUND. DURING THE READING OF THE TRACK, IF A SECTOR IS BAD IT WILL DISPLAY THE SECTOR NUMBER AND THE NUMBER OF RETRYS IT TOOK TO READ IT. IT SHOULD BE NOTED THAT IT WILL DO 11 RETRYS MAX, AND THEN GO ON TO THE NEXT SECTOR. IF IT TAKES MORE THAN 10 RETRYS, THEN YOU SHOULD REFORMAT THE DISK AND CHECK IT AGAIN, AS OUR DBIOS ONLY DOES 10 RETRYS BEFORE INDICATING A FAILURE. RETRYS ON THE ORDER OF 1 TO 5 IS TYPICAL, IF THEY OCCUR AT ALL, WITH THIS INTERFACE. THIS PROGRAM DOES NOT WRITE ON THE DISK, IT IS READ ONLY.

5. FORMAT.ASM & FORMAT.COM

IF YOU ARE USING OUR OLD SINGLE DENSITY FORMAT PROGRAM, YOU WILL NOT BE ABLE TO READ THEM ON THE NEW INTERFACE IN SINGLE DENSITY. THIS IS BECAUSE THERE IS A BYTE IN THE INNER RECORD GAPS THAT THE 1771 WILL READ BUT THE 1791/1793 WON'T. THIS FORMAT PROGRAM FIXES THAT PROBLEM FOR BOTH THE 1791/1793 AND WILL STILL ALLOW YOU TO USE IT WITH YOUR PRESENT 1771 CONTROLLER ALSO. YOU SHOULD DESTROY AND OLD COPIES OF THE OLD FORMAT PROGRAM YOU HAVE, AND USE THIS ONE FROM HERE ON OUT.

*** NOTE ***

THIS FORMATS SINGLE DENSITY ONLY, 26 SECTORS OF 128 BYTES AND ONLY RUNS ON THE NEW CONTROLLER BOARD.

6. FORMAT91.ASM & FORMAT91.COM

This program will only run on the single-density interface. It will format disks in standard IBM single-density format, to read correctly on the double density interface.

7. DFRAND.ASM

This is another format program, which only runs on the double-density interface, and which formats disks double-density in a random format. This is very useful to use in conjunction with the DTEST program, while setting up precomp. It gives a more realistic representation of the way that data may be present on the disk. Do NOT use this program to format disks that are to be used next with CP/M, as the directory needs to be filled with E5's.

8. MACRO.LIB & SKEW.LIB

THIS LIBRARY IS NECESSARY IF YOU HAVE DIGITALS MACRO ASSEMBLER AND WISH TO CHANGE AND ASSEMBLE DDUMP.ASM AND DTEST.ASM. THESE PROGRAMS USE MACROS.

CONTENTS OF PUBLIC DOMAIN DISK #2 May 2, 1980

9. STAT.COM (FOR CPM V1.4 ONLY)

THIS IS AN UPDATED VERSION OF THE STAT PROGRAM FOR THE ORIGINAL DISTURBUTION. IT FUNCTIONS THE SAME AS THE OLD ONE. THE ONLY IMPROVEMENT WAS TO MAKE IT DISPLAY THE CORRECT CAPACITY OF A DOUBLE DENSITY DISK. IT WILL STILL WORK SINGLE DENSITY.

10. ABIOS24.ASM

THIS IS THE AUTO-DENSITY SELECT VERSION OF THE BIOS FOR CPM V1.4. THIS BIOS WILL AUTOMATICALLY SELECT THE DENSITY OF THE DISK YOU ARE USING IN EITHER DRIVE, AND WILL ALLOW YOU TO CHANGE THE DENSITY AT ANY TIME. IF YOU ARE GOING TO CHANGE THE DENSITY OF THE "A" DRIVE, YOU MUST HAVE A DISK WITH THE SAME SYSTEM SIZE AS THE ONE YOU REMOVED. FILE TRANSFERS FROM SINGLE TO DOUBLE OR DOUBLE TO SINGLE IS COMPLETELY AUTOMATIC. YOU MUST SET DOUBDEN = TRUE IN THE DBOOT24.ASM FILE BEFORE YOU USE THE AUTO-DENSITY CAPABILITY OF ABIOS24.ASM, AS THIS IS THE ONLY WAY THE PROGRAM KNOWS IT IS LOOKING FOR A DOUBLE DENSITY DISK IN ANY DRIVE.

11. 2ABIOS24.ASM

THIS IS THE AUTO-DENSITY SELECT VERSION FOR CPM V2.x AND THE NEW INTERFACE. THIS BIOS MUST BE USED WITH 2DBOOT24.ASM TO BRING UP THE SYSTEM. PLEASE NOTE THAT 2ABIOS24 AND 2DBOOT24 ARE ONLY FOR CPM V2.x AND WILL NOT RUN ON CPM V1.4 OR CONVERSELY. THIS BIOS IS SHIPPED READY TO RUN IN THE AUTO SELECT MODE. YOU MAY CHANGE THIS AFTER YOU GET A RUNNING SYSTEM SO THAT IT WILL NOT RUN AUTO SELECT BY SETTING DOUBDEN = FALSE. YOU MAY ALSO DEFEAT THE DMA PORTION BY SETTING DMACNTL = FALSE, WHICH WILL ALLOW THE BIOS TO RUN IN PROGRAM DATA TRANSFER.

*** NOTE ***

YOU MUST SET DMACNTL = TRUE AND DOUBDEN = TRUE IF AUTO DENSITY SELECT IS NEEDED AND YOU WILL BE RUNNING DOUBLE DENSITY at 2 MHz.

12. 2DBOOT24.ASM

THIS IS THE SECONDARY COLD START LOADER FOR CPM V2.x FOR USE WITH 2ABIOS24.ASM. SEVERAL EQU'S APPEAR IN THIS LOADER. DMACNTL - SETTING THIS TRUE WILL ALLOW THE PROGRAM TO BOOT IN THE SYSTEM USING DMA CONTROL. IF FALSE, BOOTS SYSTEM UNDER PROGRAM DATA TRANSFER. DOUBDEN - SETTING THIS TRUE PUTS THE SPECIAL ID BYTE INTO THE DISK DURING GENERATION OF A DOUBLE DENSITY SYSTEM DISK THAT WILL BE BOOTED IN FROM DRIVE 'A'. SETTING THIS FALSE ALLOWS BUILDING A SYSTEM ON A SINGLE DENSITY DISK. THIS BYTE IS HOW THE SYSTEM KNOWS WHETHER OR NOT A SINGLE OR DOUBLE DENSITY IS ON LINE.

**** NOTE ****

IF YOU HAVE TROUBLE READING A SINGLE DENSITY DISK ON THIS CONTROLLER, YOU MAY HAVE A DISK WITH THE WRONG SECTOR FORMATTING. TO FIND OUT, TAKE ANOTHER DISK AND USE THE NEW FORMAT.COM FILE ON THIS DISK TO REFORMAT IT. THEN USING YOUR OLD CONTROLLER, TRANSFER ALL THE PROGRAMS YOU WISH TO SAVE FROM THE DISK THAT WOULD NOT RUN ON THE NEW

CONTROLLER BOARD TO THE NEWLY FORMATTED DISK. WE REALIZE THAT THIS IS ALSO A REAL HASSEL TOO, BUT IT IS A NECESSARY EVIL. BESIDES, THE NEW FROMATTED DISK WILL STILL WORK WITH THE OLD CONTROLLER BOARD.

*** NOTE ***

IF ALL ELSE FAILS, EVEN AFTER READING THE DIRECTIONS, FEEL FREE TO CALL ME HERE AT TARBELL ELECTRONICS. AND IF YOU WOULD LIKE TO DISCUSS ANYTHING ABOUT THE BOARD OR SOFTWARE, CALL ME.

THANKS FOR INPUTS AND OUTPUTS ON THIS PRODUCT AND HOPE YOU WILL FIND THIS PRODUCT BOTH INFORMATIVE AND FUN TO WORK WITH.

GERALD.W.MULCHIN
ENGINEERING DEPT.
TARBELL ELECTRONICS
213 538 4251

GENERAL THEORY of OPERATION

The internal operation of a floppy disk operating system is probably the most complicated part of a micro-computer system. The hardware and software interact very closely, and therefore a very strong knowledge of 8080 assembly language and common logic operations is desirable to understand this section. Knowledge of the IEEE S-100 standard is also desirable, as there are some changes that have been made to the S-100 bus with this new specification. The IEEE S-100 Standard has been enclosed with this manual in the appendix, along with the data sheets for the rest of the integrated circuits used on the interface.

The heart of the Double Density interface is the 1791/1793 Floppy Disk Controller Chip. This chip is very similar to the 1771 in operation, but now includes all the functions necessary to run Double Density. For your convenience, the 1791/1793 data sheet is included in the appendix.

The interface may be broken down into subsections as follows:

1. 1791/1793 disk controller chip.
2. 8257 Direct Memory Access Controller chip.
3. Phase Locked Loop circuit.
4. Write Precompensation circuit.
5. General Drive and Computer interface.

The 1791/1793 controls the actual reading or writing to the floppy disk, and in which density this will happen. The floppy controller chip contains 4 internal registers which are programmed by the computer through it's data and address lines. the four registers are as follows:
1. status register
2. track register
3. sector register
4. data register

Review of the data sheet will help in understanding each function of these 4 registers.

DDEN is used to control in which density the floppy disk controller chip will be operating. Low equals Double Density, High equals Single Density.

EARLY and LATE control the Write Precompensation of the data being written to the disk. These two signals along with the TRK > 43 line control the amount of shift in time the bit that is being written to the disk is subjected. Precompensation during writing is a must because of the bit packing on the medium of the disk.

This interface also has provisions for precompensation on tracks < 43, and it is recommended that a small amount be used. The amount of precompensation is a switch selection, and will be explained in the jumper options.

READ GATE is an output used for synchronization of the data seperator circuit during read operations. A high on this pin indicates

that a field of 'ones' (or zeros) has been found in the inner record gaps of the disk.

The 8257 Direct Memory Access Controller chip controls the actual transfer of computer data between the disk and computer. This device can be thought of as a high speed semi-intelligent cpu in its operation. It contains 16 registers of which only 3 are used by this floppy disk interface board. One is a command register, and the other two are byte pair data registers. During read and write operations, the 8257 must be initialized with the byte transfer count, the address of where the data is to go, and the type of transfer that is to occur, such as read or write. The 8257 can transfer up to 16384 bytes of data, and put it into any memory locations within a normal 8080's addressing range. The important aspect about the 8257 is that it will transfer data to or from memory without the need of any cpu intervention other than being initally programmed by the cpu. After it has been programmed, it alone transfers the data, and in fact removes the cpu from the S-100 bus during these transfers. In-depth information about the 8257 is available in the INTEL data catalog and in the appendix of this manual.

The Phase Locked Loop circuitry on this interface is necessary for reliable data recovery, especially while running under double density. It's function is to remove effects of data fluctuations during READ operations which may be the result of drive speed or power line changes, and general system noise. The Phase Locked Loop is used during Single and Double Density operation. This is accomplished by logic on the interface and is controlled by the DDEN line and an internal hardware latch. Tri-state switching is used to select the master oscillator timing clock. When in Single Density, the master clock is equal to 8 Mhz. Double Density selection causes the master clock to change to 16 Mhz. A close examination of the schematic will show these logic changes during density selects.

The write precompensation is in general terms really a count down circuit, controlled by the EARLY, LATE, and TRK > 43 lines during writing operations. This applies only when operating in double density, as the EARLY and LATE signals are not active during single density. The count down circuit is clocked by the 16 Mhz master oscillator circuit. The amount of precompensation is determined by the preset value jammed into the 74LS161 from the EARLY or LATE pins of the floppy chip. After this value is loaded, the 74LS161 counts until it reaches zero, at which time it writes the data to the disk. The effect of this is to delay the time the bit would have been written to the disk, until it is actually written. The smallest amount of precompensation that may be realized from this circuit is:
1 / 16 Mhz = 62.5nsec.
Jumpers provided on the interface have been selected for optimum drive performance. Selection is based on manufacturer's recommendations for the drive in use, and our testing here at Tarbell Electronics.

The computer interface for this board is based on the IEEE S-100 bus interface specification. It is not radically different from the old S-100 bus design, and should run with most S-100 products on the

market. If you have any questions about your interface cards, check the section on S-100 compatible products in our manual.

The Disk Interface occupies an address range from E0-EF hex for the DMA controller portion and F8-FD hex for the disk controller portion. A jumper is provided to allow the address range to be moved to 60-6F hex and 78-7D hex respectively, to avoid address conflicts with exsiting computer boards you may have in your system. This is explained in the jumper options section. The address ports are used as follows:

E0 hex 8257 address register (must be two bytes to this port)
E1 hex 8257 word count register (must be two bytes to this port)
E8 hex 8257 command register (1 byte)
F8 hex disk command port (input)
F8 hex disk status port (output)
F9 hex disk track port (input/output)
FA hex disk sector port (input/output)
FB hex disk data port (input/output)
FC hex wait control port (input)
FC hex drive select port (output)
FD hex DMA end of operation port (input)
FD hex extended address port (output)

The extended address port (FD hex), allows the DMA controller to transfer data to and from memory beyond the normal 64k range of the 8080 cpu. This function is avalible for both read and write operations using the disk interface. The extended address lines are provided on the S-100 bus as described in the IEEE S-100 bus interface specification. The extended lines are A16-A23.

To use the extended address function, user written software must supply an 8 bit value corresponding to the bank of memory you want to access, out to port FD hex before any read or write operations occur with the disk interface. During DMA operation, this 8 bit latch is enabled, placing its contents onto the extended address lines. The output of this latch is normally tri-state until the DMA controller becomes active. Also, this latch may be programmed at any time with any value you wish, except during an actual transfer by the DMA controller. You could even change this latch value between byte transfers if you wish. With optional decoding on memory boards, possible memory capacities can be theoretically 8 banks of 64k (512kbytes), to 256 banks of 64k (16 Megabytes).

The interface is set up for Shugart 8" compatible drives, which means drives with a Shugart interface such as Shugart 800/801 and Siemens FDD 100-8,120-8, and 220-8 will interface directly to the controller board. There are many drives which fall into this catagory and we will be supplying updates as we go along for your convenience. At this time we at Tarbell Electronics have run the Shugart 800/801 and all 8" drives made by Siemens. This includes both single and double density. Persci drives model 270/277's are now supported with this interface. See the section on jumpers for an explanation and changes which must be made to support Persci drives.

DETAIL THEORY of OPERATION

This section of the manual will deal in depth with the theory of operation of the double density board. The explanation will be broken down into sections within the board design.

The following logic notation will be used:
XX' = The logic name XX not, or the inversion of XX.

1. RESET circuit

The floppy disk interface receives it's reset from line 75 (PRESET) on the S-100 bus whenever a power on sequence is initiated or whenever a RESET on the front panel of the computer is pushed. This line is active low (0) whenever a reset is pushed. The result of pushing a reset is to cause the floppy disk controller chip to do a restore of drive 0 to the home position, reset the DMA controller chip to the idle state, and the latch (U46), which is used for drive select, density select, and side select, to be cleared to the following condition, (a.) select drive 0, (b.) select single density, and (c.) select side 0.

The reset line from the bus is first inverted by U9 (7404) to provide an active high (1) signal for the DMA controller chip U29 (8257). This signal is inverted again to provide an active low (0) reset for floppy controller chip, U31 (1791/1793), and for U46 (74LS174) an 6 bit latch. This active low reset signal also turns on the bootstrap circuit by pulling the preset line of U35 low.

2. BOOTSTRAP Circuit

The bootstrap circuit is enabled during a power up sequence or from the front panel reset switch. The reset causes a preset of flip-flop U35, a 7474 D-toggle flip flop, causing it's output, pin 5, to go high (1). This output is tied to one of three (3) inputs of U33, a 7411 3-input AND gate. The two other inputs come from the S-100 bus interface pins 47 (SMEMR), and 78 (PDBIN). The output of U33 is tied to the input of a 7406 (U87), which drives the PHANTOM line (67) low. U33 also also drives a hex inverter U34, which drives the chip select line on the Bootstrap Prom low, enabling the data outputs of the prom on to the data bus. When run is enabled on the computer, the contents of the prom are read onto the data bus and into the cpu as instructions. Because this prom is only 32 bytes long, the method of disabling it when it has completed all its instructions, is to look at address line 5 of the S-100 bus. This address line is fed to one of the inputs of U47, a 7421 4-input AND gate. The other three (3) lines in to U47 are PHASE 1 (25),PSYNC (76), and SMI (44). When the computer has read all 32 bytes of the prom, and an attempt to read the 33rd byte is tried, address line 5 will go high (1), which causes one input of U47 to go high. The three other inputs of U47 will also go high, causing its output to go high. This output line then causes the output of U26, a 7404 inverter, to go LOW (0), resetting U35. When this flip flop is reset, its output, pin 5, will go low, disabling

both the PHANTOM line and the Bootstrap Prom chip select. This action releases the the bus data lines back the their normal operating condition. During power up of your system, if you wish to disable the bootstrap circuitry, and have a front panel, you may raise address-data switch 5 and hit examine. This will turn off the bootstrap circuit, allowing you to access all computer memory. The bootstrap may be defeated entirely by using jumpers E8 to E9 on the disk interface. Using jumpers E9 to E10 enables the bootstrap during each reset. Below is a listing of the bootstrap prom for both the standard Tarbell disk ports, and for our non-standard disk ports. Our standard prom is model # 100 and our non-standard prom is model # 101. Model 101 is for people who wish to run the disk interface at addresses 60 to 7D hex.

STANDARD PROM BOOTSTRAP LISTING

```
0000                    ORG     0       ;PROM RUNS AT LOC ZERO.
                ;
00F8 =          DISK    EQU     0F8H    ;BASE ADDRESS OF DISK PORTS.
00F8 =          DCOM    EQU     DISK    ;COMMAND PORT.
00F8 =          DSTAT   EQU     DISK    ;DISK STATUS PORT.
00FA =          SECT    EQU     DISK+2  ;SECTOR PORT.
00FB =          DDATA   EQU     DISK+3  ;DATA PORT.
00FC =          WAIT    EQU     DISK+4  ;WAIT PORT.
007D =          SBOOT   EQU     007DH   ;START OF SBOOT.
                ;
0000 DBFC       BOOT:   IN      WAIT    ;WAIT FOR HOME.(caused by reset)
0002 AF                 XRA     A       ;CLEAR ACCUM.
0003 6F                 MOV     L,A     ;CLEAR REG L.
0004 67                 MOV     H,A     ;CLEAR REG H.
0005 3C                 INR     A       ;SET A = 1.
0006 D3FA               OUT     SECT    ;START AT SECTOR 1.
0008 3E8C               MVI     A,8CH   ;READ THE SECTOR.
000A D3F8               OUT     DCOM    ;ISSUE THE COMMAND.
000C DBFC       RLOOP:  IN      WAIT    ;WAIT FOR DRQ OR INTRQ.
000E B7                 ORA     A       ;SET FLAGS.
000F F21900             JP      RDONE   ;DONE IF INTRQ.
0012 DBFB               IN      DDATA   ;ELSE,GET A BYTE FROM DISK.
0014 77                 MOV     M,A     ;PUT IT INTO MEMORY.
0015 23                 INX     H       ;BUMP POINTER.
0016 C30C00             JMP     RLOOP   ;LOOP TILL DONE.
0019 DBF8       RDONE:  IN      DSTAT   ;READ DISK STATUS.
001B B7                 ORA     A       ;SET THE FLAGS.
001C CA7D00             JZ      SBOOT   ;IF ZERO, GOTO SBOOT. (7DH)
001F 76                 HLT             ;ELSE, DISK ERROR
```

DETAIL THEORY of OPERATION

NON-STANDARD BOOTSTRAP PROM LISTING

```
0000                    ORG     0       ;PROM RUNS AT LOC ZERO.
                ;
0078 =          DISK    EQU     078H    ;BASE ADDRESS OF DISK PORTS.
0078 =          DCOM    EQU     DISK    ;COMMAND PORT.
0078 =          DSTAT   EQU     DISK    ;DISK STATUS PORT.
007A =          SECT    EQU     DISK+2  ;SECTOR PORT.
007B =          DDATA   EQU     DISK+3  ;DATA PORT.
007C =          WAIT    EQU     DISK+4  ;WAIT PORT.
007D =          SBOOT   EQU     007DH   ;START OF SBOOT.
                ;
0000 DB7C       BOOT:   IN      WAIT    ;WAIT FOR HOME.(caused by reset)
0002 AF                 XRA     A       ;CLEAR ACCUM.
0003 6F                 MOV     L,A     ;CLEAR REG L.
0004 67                 MOV     H,A     ;CLEAR REG H.
0005 3C                 INR     A       ;SET A = 1.
0006 D37A               OUT     SECT    ;START AT SECTOR 1.
0008 3E8C               MVI     A,8CH   ;READ THE SECTOR.
000A D378               OUT     DCOM    ;ISSUE THE COMMAND.
000C DB7C       RLOOP:  IN      WAIT    ;WAIT FOR DRQ OR INTRQ.
000E B7                 ORA     A       ;SET FLAGS.
000F F21900             JP      RDONE   ;DONE IF INTRQ.
0012 DB7B               IN      DDATA   ;ELSE,GET A BYTE FROM DISK.
0014 77                 MOV     M,A     ;PUT IT INTO MEMORY.
0015 23                 INX     H       ;BUMP POINTER.
0016 C30C00             JMP     RLOOP   ;LOOP TILL DONE.
0019 DB78       RDONE:  IN      DSTAT   ;READ DISK STATUS.
001B B7                 ORA     A       ;SET THE FLAGS.
001C CA7D00             JZ      SBOOT   ;IF ZERO, GOTO SBOOT.
001F 76                 HLT             ;ELSE, DISK ERROR
```

3. ADDRESS SELECTION

The double density controller board is selected based on the low order 8 address line values presented to the interface during an input or output instruction. The normal address range for this board is from E0 hex to FD hex. There are provisions for changing the address range from 60 hex to 7D hex by means of jumpers E4,E5,and E6.

The base address for the DMA controller (8257) is E0 hex, with all possible 16 address from E0 to EF hex available for use by the DMA controller chip. U33, a three input AND gate, is used to determine the base address of E0 hex by tying address lines A7,A6,and A5 to its three inputs. You will notice that an inverter, U34, is in between one input of U33, and address line A7. By selecting the proper jumper at E4,E5, and E6, you may select a base address of either E0 hex or 60 hex. By using the inverter, U34, you will select the base address of 60 hex. By not using the U34, but using the jumper which bypasses U34, you will select a base address of E0 hex. The output of U34 is next

AND'ed with the inversion of A4 through U26. This gate (U48) is used to allow selections from E0 hex to EF hex for I/O operations with the DMA controller. The output of U48 drives one of the inputs to U43 for selecting the DMA controller chip select line. The other input for U43 (7400), comes from U45 (7432). U45 determines whether the current I/O operation is an Input or an Output with the interface board. The signals SINP and SOUT are two status lines from the CPU card used to determine the I/O operation. Because these two signals will not occur together, they may be OR'ed together through U43 to provide just one general signal for I/O operation decoding.

Disk I/O ports for the 1791/1793 floppy controller chip are decoded by U47, a 4 input AND gate. One input to U47 comes from U33's output, which is the address decode for the base address of Ex hex or 6x hex. Address lines 4 and 3 are tied to two of the other inputs of U47. Address line 2 is inverted through U26 (7404) and is the fourth input to U47. The Output of U47 goes high whenever any of the following address appear on the address bus,F8,F9,FA,and FB hex.

Ports FC and FD hex are decoded by U27 (74LS138). The output of gate U33 is used for the enable input of U27. The other two enables of U27 are an enable when low function and are derived from the inversion of address lines 3 and 4 through two inverters U26. The output of inverter U26 pin 10 also goes to one input of gate U48. If address line 4 is high (eg. Port FC), The output of U26 pin 10 will be low (0),causing the output of gate U48 to be low, disabling the selection of any port with the base address of Ex hex. The 3 low address lines, A0,A1, and A2 determine which port is selected on the output of U27. Also, even though the low 3 lines of the address bus are used by U27, only two of the possible 8 ports are used by the disk interface board. Port decodes from U27 for F8,F9,FA, and FB are not used. Only Ports FC and FD hex are used.

4. DISK CONTROL circuit

The 1791/1793 Floppy disk formatter/controller chip performs all the functions necessary to read or write data to a floppy disk drive. Both single and double density storage capabilities are supported. The chip is compatible with the IBM 3740 (FM) data format,IBM System 34 (MFM), or may be operated non-standard by using the controllers variable length sector capability.

The floppy controller chip contains five (5) internal registers that can be read or be written to. These registers are used to write commands, read status, and read and write data to and from the floppy disk drive. These five registers are selected by providing the proper binary code on the A0 and A1 lines of the floppy chip in conjunction with either a read or write operation. The registers and their addresses are as follows:

CS'	A1	A0	RE'= 0	WE'=0
0	0	0	Status Reg	Command Reg
0	0	1	Track Reg	Track Reg
0	1	0	Sector Reg	Sector Reg
0	1	1	Data Reg	Data Reg
1	x	x	Deselected	Deselected

The five internal registers of the floppy controller chip are accessed through the internal interface data bus. The internal data bus is buffered by U71 (8208) and U72 (8208). U71 and U72 are 8 bit bi-directional transcievers which are used as bi-directional data buffers between the computer S-100 bus and the floppy disk interface internal data bus. These transceivers allow data travel in either direction depending upon whether the TRANS/REC' pin (11) is high or low. To program the floppy controller chip, A0,A1 are selected for the desired operation. Write enable of U31 is made active low by processor signal SOUT and PWR' gated together by U23 (7400). The data that the CPU wishes to program the floppy controller with, is now placed on the DO0 - DO7 lines of the CPU card. This data is then presented to U71. Because we are not in the DMA mode of operation, the TRANS/REC' pin is already pulled low, causing the transceiver to be in the receive mode. The data is then passed through U71 onto the internal data bus and into the floppy chip DAL0 - DAL7 lines. Chip select, pin 3, of U31 is made active low by U48 (7408) output. One input of U48 comes from the DMA controller and the other input is from U43 which is the gating of the address port and the I/O mode desired. In this case, since we are programming the floppy controller, any port address in the range of F8 - FB hex gated together with SOUT will cause the data to be written to the controller chip.

5. Direct Memory Access

The sequence of operations for a DMA access is as follows:

A. The Basic Input Output System (BIOS part of CP/M) program sends a series of initialization bytes to the 8257. These include the starting address in memory for the data, the number of bytes to transfer, and the type of command (read, write, verify) which is being requested. The port numbers for these and other transfers are listed on page 3-3 in this manual.
B. The BIOS then sends track number, sector number, and type of command (read or write) to the 1793.
C. The 1793 then causes the head to load against the media (we are assuming we're already on the correct track). When the 1793 finds a sector header that matches the track and sector number in it's registers, and is ready to write or read the first byte, it activates its DRQ line, which is fed to the 8257. This signifies that it is ready to transfer a byte.
D. Soon after receiving the input on its DRQ0 line, the 8257 makes its HRQ line go high. This is fed through an OR gate

to S-100 bus line PHOLD'. This notifies the CPU that there is a DMA request pending.

E. When the CPU finishes the cycle it is currently in, it sends back a PHLDA signal over the S-100 bus to the HLDA input on the 8257. This tells the 8257 that its DMA request has been granted, and that the CPU is in a HOLD state.

F. The 8257 then activates its AEN line, which indicates that it has a valid address on A0 to A7 and D0 to D7. This signal is ORed with the HRQ to keep PHOLD' active, and disables the CPU address and data lines using S-100 bus lines DO DSB' and ADR DSB'. AEN also is fed to (U41) and (U67), which gates the DD board's own status and control lines onto the S-100 bus (PWR', PDBIN, SMEMR, SINP, SOUT, PSYNC, MWRITE, SINTA, SHLTA, SM1).

G. A short time later, the 8257 activates its DACK0' line, which disables the CPU's status and control lines, using S-100 bus lines STAT DSB' and C/C DSB'. The time between step F and G provides the overlap time, during which both our DD board and the CPU board have control of these status and control lines.

H. At the same time as the DACK0' line is activated, the 8257 activates either its MEMR' line, or its I/OR' line, depending upon whether it's going to do a write or read respectively, to disk. Step I will explain what happens for a write operation, Step J for a read (from disk) operation.

I. For a write operation: The MEMR' line coming from the 8257 is used for several things. It is fed through U67 to S-100 lines PDBIN and SMEMR to the system memory to gate the memory onto the S-100 bus. It is also used to activate U72 so that the S-100 DI bus is fed into the DD board's internal data bus and therefore providing a path from the memory output to the 1793. A short time after the MEMR' signal, the I/OW' signal is fed from the 8257 to the WE' input on the 1793, which strobes the data into the 1793. The 1793 then writes the data onto the disk.

J. For a read operation: The I/OR' signal from the 8257 is fed to the 1793, gating the data in its register onto the data bus. This signal is also used to activate U71 in the outward direction, so that the data is fed onto the S-100 DO bus out to the system memory. (mod for rev-e) A short time after the I/OR' signal, the MEMW' signal is generated by the 8257. This is used to generate the S-100 bus line PWR', which strobes the data into memory.

K. After either step I or J above, the reverse sequence occurs, turning over control of the bus back to the CPU.

6. Write Precompensation

The Write Precompensation circuit consists of a programmable counter, (U36), for delaying or advancing write data pulses during disk writes, a dip switch (SW-1) for adjusting the precompensation value, and the necessary decoding logic for early, late, and TRK > 43 (U12).

DETAIL THEORY of OPERATION

The 1793 produces along with it's write data pulses, two signals called 'early' and 'late'. These two signals, along with TRK > 43, are decoded by a 1 of 4 decoder circuit (U12) to select the switch setting from SW-1 for loading into the counter (U36).

Loading the counter starts the counter down counting. If there is no early or late signal generated (no precomp needed), the programmable counter is loaded with a positive (0111) 7, which is the zero precompensation center reference value.

Values from (0110) 6 through (0000) 0 are increasing late times, and values (1000) 8 through (1110) -2 are increasing early times. The programmable counter is clocked with 16 MHz during double density. Therefore, the minimum resolution is 62.5 nsec.

After the counter has been loaded with the desired value of precompensation, the counter down counts until a carry is produced.The carry, delayed one (1) clock cycle by a D-toggle flip-flop(U35), is used to trigger a 250 nsec. one-shot. The output of the one-shot, after buffering, is the precompensated data which is written to the drive. The counter then remains stopped until a new data pulse is sent from the 1793, starting the whole process again.

7. Data Recovery

Data recovery is provided by an on board Phase Locked Loop circuit. This circuit allows for maximum data recovery reliability, while rejecting drive speed variations and bit crowding. One shot (U74) conditions the data being read from the disk and provides RDATA* to the 1793. It also provides data to the phase comparator (U55) which is part of the phase locked loop. The active filter network which consist of U55,R5,R6,R7,R8, C5,C6 and Q2 removes noise and jitter from the incoming data and provides a correction voltage to the voltage controlled oscillator (U53). This oscillator provides the master clock frequency necessary for a stable read clock signal (RCLK) for the 1793. This output is either divided by 2 or divided by 4, controlled by the DDEN line, to provide the necessary 500 KHz or 250 KHz RLCK signal for double density or single density operation.

ASSEMBLY INSTRUCTIONS

Before attempting any assembly operations, it is recommended that you read this section and the parts list section first. This will help put your mind into a comfortable state about what you are going to be doing with the board.

1. ( ) Locate and confirm that your kit contains all the parts listed in the parts list in this manual. Any errors found should be reported to Tarbell Electronics for corrective measures. Also, you should make any necessary changes which you receive from Tarbell Electronics, on your board at this time.

2. ( ) If you are using sockets, install them at this time.

3. ( ) Locate the 50 pin header connector and install it at the top of using the 2 2-56 x 3/8 mounting screws and hardware.

4. ( ) Locate Q1, an LM-323 votage regulator, and the heatsink. Mount the regulator and heatsink on the board using the 4-40 x 3/8 screws and hardware.

5. ( ) Locate Q2, an MPS6571 transistor, and mount it as indicated by outline on the board.

6. ( ) Locate the 220 uhy choke, L1, and mount it on the board. This part may appear to you as looking like an resistor. It is not a resistor but is in fact a RF. coil. Use caution if you are in doubt about which one L1 is.

7. ( ) Locate the crystal, Y1, and mount it on the board. Mount the crystal down close to the board. Then take a small piece of wire and push it through the hole that is to the left of crystal, and solder it. Then, take the other end of this wire and bend it over the top or the crystal and cut off all but 1/8" of it. Then solder the wire to the crystal can. Do not over heat the crystal can, or you may open the hermetic seal.

8. ( ) Using the parts list and using the board layout drawing, start mounting all the capacitors on the board. Be sure to observe the polarity of the tantalum capacitors C13,C14,C15. The little square block on the board is the end that the plus (+) lead of the capacitor goes into.

9. ( ) Using the parts list and board layout drawing, mount all the resistors on the board.

10. ( ) Locate the dip switch and mount it. Note, if the dip switch you have has only 8 positions, you will have to mount the switch on the board starting at position 2 at the location on the board. There should be one open location at the top of the switch that is not used after you mount the switch.

11. ( ) Do not mount the IC's at this time, as you will need to check the voltage regulators for proper output voltages.

ASSEMBLY INSTRUCTIONS

12.( ) After all the components are mounted, plug the board into your computer and check the voltage regulators for proper operation. CAUTION** Be sure your computer is turned off before you plug in the board. The output of Q1 should be approx. 5 volts +/- .5 volts. The output of CR1 should be approx. 12 volts +/- .75 volts.

13.( ) Using the parts list and board layout drawing, mount the IC's into their respective locations. Use ''GREAT'' caution with the two 40 pin IC's when mounting them as it is possible to bend or break some of the leads on the part.

14.( ) This completes the assembly of the unit. On the following pages are the jumper options for setting up the board for your system.

JUMPER OPTIONS

This board has very little jumpering required to get it running. The only jumpers that need setup are for address selection, XRDY or PRDY line, and Write Precompensation value.

1. Normal Board Setup

The normal configuration is as follows:
a. Board address set for base address of E0 hex.
b. PRDY line used for processor wait.
c. SW-1 set for 62.5 nsec on trk < 43 and 187.5 nsec for trk > 43.

NOTE **
The write precompensation may be the only adjustment you will have to make for your drives.

2. Address Selection

This board may be addressed in one (1) of two (2) address areas in your computer. Either an address range of 60 - 7D hex or E0 - FD hex may be selected. You should beware of existing computer boards in your system which use these address ports for I/O operations, as a conflict may exist. Our I/O section for CP/M uses an address range of E0 - FD hex, as does our Standard bootstrap prom.

PORT	JUMPER
E0 - FD Hex	E19 to E20
60 - 7D Hex	E19 to E18

3. XRDY or PRDY selection

Three possible configurations exist in the selection of the right line to use. These are, no front panel, front panel, and if you have a front panel, who made it.

No Front Panel	Jumper E1-E2 or E3-E2
Imsai Front Panel	Jumper E1-E2
Altair Front Panel	Jumper E3-E2

NOTE **
Be sure that other boards in your system do not use either the XRDY or PRDY lines improperly, as this would cause improper operation of this interface board. Things to beware of are memory boards which activate a tri-state driver in anticipation of memory wait states, or dynamic memory boards which require the use of this line for refresh operations, or memory boards which use slow memorys such as old, old 2102's.

4. Write Precompensation Selection

The selection of the write precompensation is based on the type of drive you are using and the amount of errors you are getting from the disk when you run our special disk test program. To properly set

the precompensation, first set the dip switch S1 for a value of 62.5nsec for < 43 and 187.5 nsec for > 43 as shown in the Precomp Table in Fig.1 Next, format a good certified double density disk with our DFRAND.COM program. Then read the disk for errors using our DTEST.COM program. If you find any errors, then change the precompensation switch settings to another value in the table and run the sequence again. The initial values of 62.5nsec for < 43 and 187.5nsec for > 43 were found to be midrange values here at Tarbell Electronics. These values are with the switch positions 5 and 8 on (if there is a 9 position switch), or positions 4 and 7 on (if there is an 8 position switch).

After you have found the proper settings for the precompensation switches, reformat the disk using the DFORMAT.COM program so that the disk will have the proper data in each sector required for CP/M operation. (E5 data fill)

SW - 1 settings

	Trk < 43					Trk > 43						
	--Early--		--Late--			----Early-----				----Late----		
Pos	1		2	3		4	5	6		7	8	9
	ns		ns		ns				ns			
*	62.5	①	62.5	0	0	62.5	0	1	1	62.5	0	0	0
x	x	x	x	x	x	187.5	0	1	0	x 187.5	0	1	0
x	x	x	250.0	0	①	250.0	0	0	1	x 250.0	0	1	1
	x	x	x	x	x	312.5	0	0	0	x 312.5	1	0	0
-	x	x	375.0	1	0	x	x	x	x	375.0	1	0	1
	x	x	x	x	x	x	x	x	x	437.5	1	1	0
	x	x	500.0	1	1	x	x	x	x	500.0	1	1	1

FIG 1

Note: x = not adjustable, 1 = on, 0 = off

5. Bootstrap

The on-board bootstrap may be enabled or disabled by jumpers E8,E9,and E10 as follows.

Bootstrap enabled - Jumper E9 to E10
Bootstrap disabled - Jumper E9 to E8

6. Persci drives

If you have a Double Density board Rev. D or Rev. E, and want to run a Persci model 277/270 with it, then do the following changes to the board.

1. On the double density interface:

JUMPER OPTIONS

a. on back side, cut the trace connecting E4 to E6 and jumper E4 to E5. (bypasses step one-shot)

b. Jumper E11 to E12. (1793 test pin)

c. Jumper E13 to E14. (seek complete line)

Now you must make the Persci 277/270 look like a Shugart compatible drive. Thanks to John Mock of Bits & Bytes of Fullerton, we now have the all the changes. These are changes which you must make to the drive and in some cases will involve making trace cuts on the Persci drive.

Cuts to make:
---

1. Trace to P1-16 (back side)
2. Trace between BL-BC (front side)
3. Trace between U11-12 and U11-13 (back side)
4. Trace between U11-10 and U11-11 (back side)
5. Trace from P1-2 to U1-8 (front side)
6. Trace to P1-30 (back side)
7. Trace to P1-32 (back side)

Jumpers to add:
---

1. U11-12 to P1-32 (back side)
2. U11-10 to P1-30 (back side)
3. H to M (for head load)
4. H to EL (for head load)

Persci jumpers: (board assy.200263-007 rev N/S)

1( ) A - B 2( ) J - Z 3( ) K - L 4( ) N - P 5( ) T - S
6( ) U - V 7( ) BK-BM 8( ) BA-BB 9( ) AK-AH 10( ) X - W
11( ) AD-AE 12( ) AP-AR 13( ) AS-AT 14( ) AM-AN 15( ) BE-BD

Jumpers E-D and G-F and AC-AB should be open.

Drive select:

Drive 1 (A & B)	Drive 2 (C & D)
U11 13 to 2	U11 12 to 2
U11 11 to 4	U11 10 to 4
	Remove U1

Our bios packages for CP/M use with the double density board must have some of the equates changed for proper operation with the Persci 277/270.

JUMPER OPTIONS

In the bios change HLAB EQU 0 to HLAB EQU 8 for head load at beginning of seeks.

Set DUAL to TRUE as there are two heads moving together.

These changes were tried on our Persci model 277/270 here at Tarbell Electronics which has a revision number of N/S. We are not sure about other revisions which Persci makes or about there operation in Double Density. You should probably consult the factory about any revisions you have which do not agree with our own. Operation with the Persci has been quite good in Single Density operation on the new controller. But Double Density has been poor. This is probably due to the fact that our Persci 277/270 is not rated for Double Density operation. This is probably due to head design. Adjusting the precompensation may help some what.

For normal operation with Shugart compatible drives, install jumpers as follows:

(✓) E13 to E15
( ) E4 to E6 <-- normally etched on the board

7. Phase Locked Loop circuit

The Phase Locked Loop circuit has provisions for a second order low pass filter if needed. At this time, we have found no need to incorporate this feature. Therefore the following jumpers or components are used.

R7 install a jumper
C4 will be missing
R2 and R4 will be missing

8. Interrupt option

Interrupts from the Floppy controller may be used to generate a interrupt to the CPU if desired. The large hole (E23) just above pin 8 of the S-100 connector is the output for the interrupt line. This is an active low signal and is normally tri-state. This line may be connected to any of the S-100 vector interrupt lines on the bus. You will need a vector interrupt card if you plan on using interrupts other than RST 7. The vector interrupt lines are defined in the S-100 standard on pages 10-1 to 10-5 of this manual.

9. MWRITE option

If your memory boards require the MWRITE signal to write data into memory, and this signal is not generated on the cpu or front panel, then you will have to jumper E16 and E17 together. If your memory boards use PWR' for writing data into memory, then leave E16 to E17 open.

JUMPER OPTIONS

10. 8257 Clock option

An option has been provided on the board for running IC U29 at either 2MHz or 4MHz. This option is provided so that people who wish to use the AMD 9517 at full speed may do so. Otherwise, if you are using the 8257 supplied in our interface, then you will have to run it at 2MHz.

Jumpers are as follows:

device	jumpers	
8257	E23 - E25	2MHz clock
9517	E24 - E25	4MHz clock

** Note ** Software programming for the 9517 is different than the 8257. They are pin compatible only.

11. 1793 Read delay

The 1793 has a hold time restriction when reading data or status from it. A one-shot has been provided to stall the CPU during any reading from ports F8 - FB Hex (78 - 7B Hex). This option is provided allready connected on the board. This is the only WAIT, when running DMA operation, generated on the board. This option may be defeated by cutting the trace between E26 and E27 on the board. The only problem you may find if you cut this trace is that while running with a Z-80 at 4MHz, you may read the 1793 too fast. This is also a function of the 1793 you are using at the time. You may have to replace the jumper if you find you have any problems reliably reading the 1793.

12. Extra buffer chip

Located at U41 is an extra buffer for use as you may see fit. This buffer is an enabled Tri-state buffer of the non-inverting type. Jumpers are provided so you may use it.

E22 = buffer input
E21 = buffer output

TARBELL DOUBLE DENSITY FLOPPY DISK INTERFACE FULL WARRANTY

1. Any faulty component part purchased from Tarbell Electronics, which is returned within 6 months after the date of purchase will be replaced at no charge. Components returned under this part of the warranty should be with a letter explaining what is wrong with the part.

2. Any factory-assembled floppy disk interface, which does not work correctly, and is returned within 6 months after the date of purcnase, will be restored to proper operating condition or replaced without charge.

3. Any floppy disk interface kit, which in the opinion of the manufacturer has been assembled with reasonable care, and is returned for repair within 6 months after the date of purchase, will be repaired for a charge commensurate with the work required. (parts will be free) but in no case will exceed $100 without notification of the owner.

4. Any floppy disk interface not covered by the above condition will be subject to a charge commensurate with the work and parts required, but in no case will exceed $100 without notification of the owner.

5. Parts can be returned directly to the address below for replacement. Complete floppy disk interfaces should be returned to the place of purchase. If this is not possible, or if it is very inconvenient, it may be returned to the address below, with proof of purchase.

6. Tarbell Electronics assumes no responsibility for consequential damages to other connected equipment, or for time lost, or programs or data lost, because of interface malfunction or incorrect documantation.

7. If you are disatisfied with the operation of a factory-assembled Tarbell Double Density Floppy Disk Interface for any reason, your money will be cheerfully refunded, provided the unit is returned within the six month warranty period.

8. Tarbell Electronics does not warrant that the disk interface will work with all "S-100" computer systems, or with all floppy disk drives. Call the factory or ask your local dealer about any possible conflicts in your system.

9. This warranty does not cover parts, or interfaces built from parts, which are not traceable to Tarbell Electronics.

10. An interface which is assembled from a kit by a Tarbell dealer has only the parts covered by this warranty, not the labor. All interfaces which were sold as kits, will have a "K" marked on the solder side. The dealer may provide his own warranty in this case.

TARBELL DOUBLE DENSITY FLOPPY DISK INTERFACE FULL WARRANTY

Defective parts or interfaces covered under this warranty should be sent WITH PROOF OF PURCHASE (like a receipt) to:

Tarbell Electronics
950 Dovlen Place, Suite B
Carson, California 90746

PARTS LIST

Qty.	Descriptor	Desiginator
Resistors		
1	120 ohm 1/2 watt	R29
1	220 ohm 1/8 watt	R6
3	330 ohm 1/4 watt	R1, R3, R16
5	470 ohm 1/4 watt	R10, R11, R12, R13, R14
1	470 ohm 1/8 watt	R8
1	510 ohm 1/8 watt	R39
5	1 k ohm 1/4 watt	R19, R24, R22
1	1 K ohm 1/8 watt	R7
7	2.2kohm 1/4 watt	R15, R16, R30, R31, R38, R20
2	2.4Kohm 1/8 watt	R9, R17
2	4.3Kohm 1/4 watt	R40, R43
5	4.7kohm 1/4 watt	R17, R18, R23, R25, R26
2	5.1Kohm 1/8 watt	R41, R42
1	6.2Kohm 1/8 watt	R5
4	10 kohm 1/4 watt	R21, R27, R28
2	33 kohm 1/4 watt	R37, R35
1	4116-003-221-331	U8
Capacitors		
2	18 pf cap	C9
2	56 pf cap	C3, C4
2	100 pf cap +/- 5%	C7, C8
2	220 pf cap	C12, C17
1	680 pf cap	C1
1	.0015 mfd cap	C6
1	.0039 mfd cap	C5
28	.1 mfd cap	C11, C20, C21, C22, C23, C24, C25, C26, C27, C28 C29, C30, C31, C32, C33, C34, C35, C36, C37, C38 C39, C40, C41, C42, C43
1	4.7 mfd @ 10 volts	C10
2	10 mfd @ 16 volts	C13, C15
1	22 mfd @ 10 volts	C14
Misc.		
1	16 MHz crystal	Y1
1	MPS 6571 Transistor	Q2
1	1N4742 12 volt 1 watt zener	CR1
1	9 position dip switch	SW-1
Hardware		
1	TO-3 Heatsink	
2	4-40 x 3/8 screw	
2	4-40 x 1/4 nut	
2	# 4 washer	
2	2-56 x 3/8 screw	
2	2-56 x 3/16 nut	
2	# 2 washer	

PARTS LIST

Connectors

1	50 pin connector	J1
1	16 pin dip-socket	U52
2	40 pin dip-socket	U29,U31

IC's

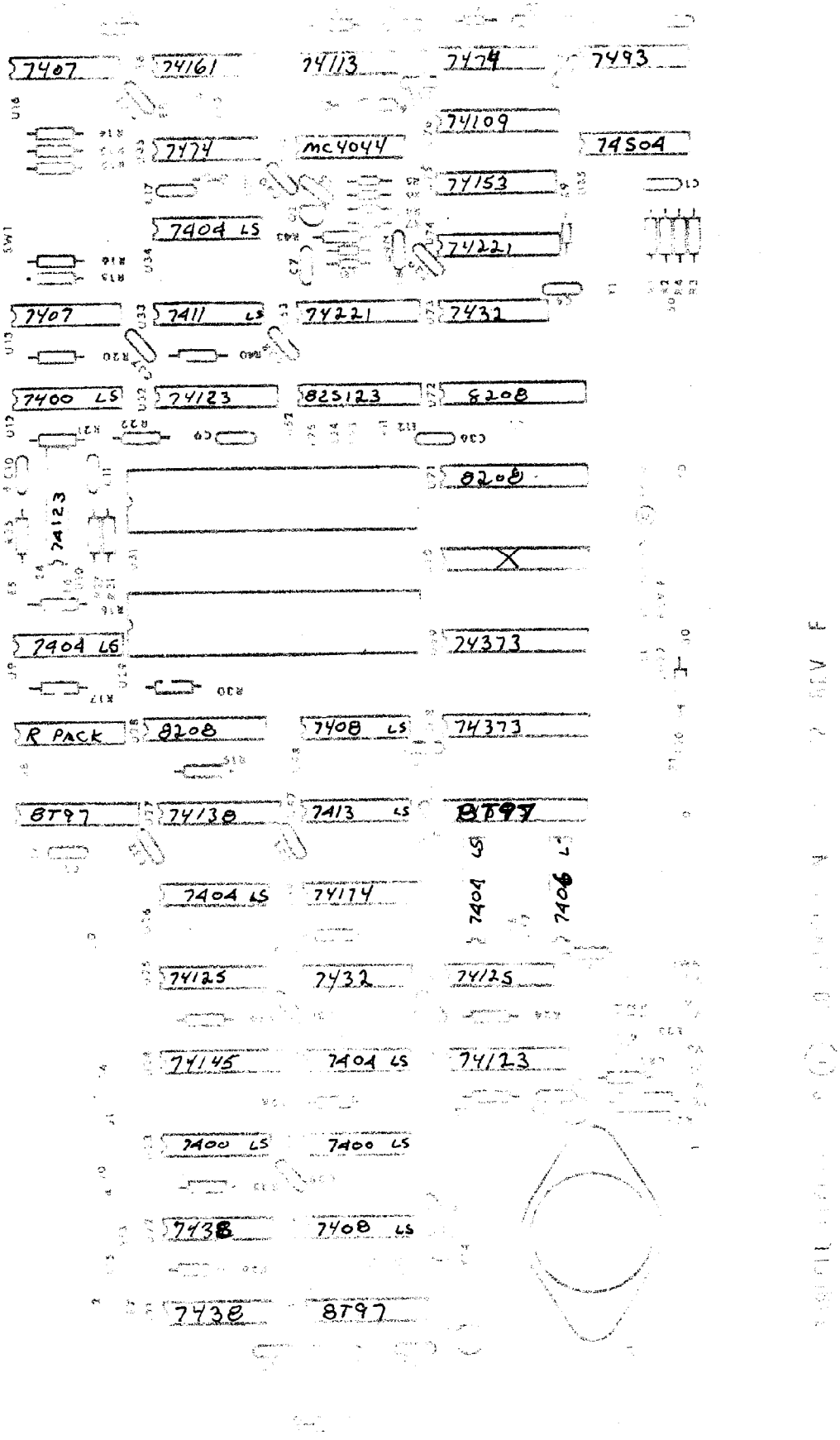
** Note ** Equiv. 74LS parts may be used where indicated in list.

3	7400/74LS00	Quad Nand gate	~~U12,U23,U45~~
5	7404/74LS04	Hex Inverter	~~U9,U26,U34,U44~~
1	7404	Hex Inverter	~~U66~~
1	74S04	Schottky Hex Inverter	~~U85~~
1	7406	O.C.Hex Inverter	~~U87~~
2	7407	Hex Buffer	~~U15,U16~~
2	7408/74LS08	Quad And Gate	~~U42,U48~~
1	7411/74LS11	3 Input And Gate	~~U33~~
1	7413/74LS13	4 Input Schmitt Trigger	~~U47~~
2	7432/74LS32	Quad Or Gate	~~U45,U73~~
2	7438	O.C.Quad Nand Gate	~~U21,U22~~
2	7474/74LS74	D Toggle Flip Flop	~~U35~~,(~~U77~~)
1	7493	4 Bit Binary Counter	~~U86~~
1	74LS109	Dual J-K Flip Flop	~~U76~~
1	74LS113	Dual J-K Flip Flop	~~U56~~
3	74LS123	Dual One Shot	~~U10,U32,U64~~
1	82S123/5331	Bootstrap Prom	~~U52~~
3	74125/74LS125	Quad Tri-state Buffer	~~U25,U65~~
1	74LS138/8205	Octal Decoder	~~U27~~
1	74145	1 of 10 Decoder	~~U24~~
1	74LS153	Dual 4 to 1 mux.	~~U75~~
1	74LS161	Preset Counter	~~U36~~
1	74174/74LS174	6 Bit Latch	~~U46~~
2	74LS221	Dual One-shot	~~U55,U74~~
2	74LS373	Octal Tri-state Latch	~~U68,U69~~
3	8T97	Hex Tri-state Buffer	~~U7,U67,U41~~
3	8208/8286/8304	Octal Transciever	~~U28,U71,U72~~
1	MC4044	Phase Comparator	(U55) (Motorola)
1	LM 323	5 Volt 3 Amp Regulator	Q1

LSI chips

1	8257	DMA Controller Chip	U29
1	1793	Floppy Controller	U31

** Note ** U70 is not used


7407
74161
74113
7474
7493
74109
7474
MC4044
74504
74153
7404 LS
74221
7407
7411 LS
74221
7432
7400 LS
74123
825123
8208
8208
74123
X
7404 LS
74373
R PACK
8208
7408 LS
74373
8T97
74138
7413 LS
8T97
7404 LS
74174
7404
7406
74125
7432
74125
74145
7404 LS
74123
7400 LS
7400 LS
7438
7408 LS
7438
8T97

IEEE Specification, IEEE Task 696.1/D2

The following is a list of the IEEE pin functions, signals, type of signal, active level, and a description of the signal.

The following conventions will be used:

1. * = inverted signal or the NOT function.
2. M = master.
3. B = bus.
4. S = slave.
5. O.C. = open collector.
6. H = high or logic level 1
7. L = low or logic level 0

Pin	Signal	Type	Active Level		Description
1	+8 volts	B			Instantaneous min greater than 7 volts, instantaneous max less than 25 volts, average max less than 11 volts.
2	+16 volts	B			Instantaneous min greater than 14.5 volts, instantaneous max less than 35 volts, average max less than 21.5 volts.
3	XRDY	S	H		One of two ready inputs to the current bus master. The bus is ready when both inputs are true (H). see pin 72.
4	VI0*	S	L	O.C.	Vectored interrupt line 0.
5	VI1*	S	L	O.C.	Vectored interrupt line 1.
6	VI2*	S	L	O.C.	Vectored interrupt line 2.
7	VI3*	S	L	O.C.	Vectored interrupt line 3.
8	VI4*	S	L	O.C.	Vectored interrupt line 4.
9	VI5*	S	L	O.C.	Vectored interrupt line 5.
10	VI6*	S	L	O.C.	Vectored interrupt line 6.
11	VI7*	S	L	O.C.	Vectored interrupt line 7.
12	NMI*	S	L	O.C.	Non-maskable interrupt line.
13	PWRFAIL*	B	L		Power fail bus signal.
14	DMA3*	M	L	O.C.	Temporary master priority bit 3.
15	A18	M	H		Extended address bit 18.

IEEE Specification, IEEE Task 696.1/D2

16	A16	M	H		Extended address bit 16.
17	A17	M	H		extended address bit 17.
18	SDSB*	M	L	O.C.	Disable signal for 8 status lines.
19	CDSB*	M	L	O.C.	Disable signal for 8 control lines.
20	GND	B			Common with pin 100
21	NDEF				Not to be defined. Manufacturer must specify any use in detail.
22	ADSB*	M	L	O.C.	Disable signal for 16 address lines.
23	DODSB*	M	L	O.C.	Disable for 8 data out lines.
24		B	H		Master timing signal for bus.
25	pSTVAL*	M	L		Status valid signal.
26	pHLDA	M	H		Control signal used in conjunction with HOLD* to coordinate bus/master transfer operations.
27	RFU				Reserved for future use.
28	RFU				Reserved for future use.
29	A5	M	H		Address bit 5.
30	A4	M	H		Address bit 4.
31	A3	M	H		Address bit 3.
32	A15	M	H		Address bit 15.
33	A12	M	H		Address bit 12.
34	A9	M	H		Address bit 9.
35	DO1/DATA1	M/MS	H		Data out bit 1,bidirectional bit 1.
36	DO0/DATA0	M/MS	H		Data out bit 0,bidirectional bit 0.
37	A10	M	H		Address bit 10.
38	DO4/DATA4	M/MS	H		Data out bit 4,bidirectional bit 4.
39	DO5/DATA5	M/MS	H		Data out bit 5,bidirectional bit 5.
40	DO6/DATA6	M/MS	H		Data out bit 6,bidirectional bit 6.
41	DI2/DATA10	S/MS	H		Data in bit 2,bidirectional bit 10.

42	DI3/DATA11	S/MS	H		Data in bit 3,bidirectional bit 11.
43	DI7/DATA15	S/MS	H		Data in bit 7,bidirectional bit 15.
44	sM1	M	H		Status signal indicating current cycle is op-code fetch.
45	sOUT	M	H		Status signal indicating data transfer bus cycle to output device.
46	sINP	M	H		Status signal indicating data transfer bus cycle from input device.
47	sMEMR	M	H		Status signal identifying bus cycles which transfer data from memory to a bus master, which are not interrupt acknowledge instruction fetch cycles.
48	sHLTA	M	H		Status signal indicating a HALT instruction has been executed.
49	CLOCK	B			2 MHz (0.5%) 40-60% duty cycle. Not required to be synchronous with any other bus signal.
50	GND	B			Common with pin 100.
51	+8 volts	B			Common with pin 1.
52	-16 volts	B			Instantaneous max less than -14.5 volts,instantaneous min greater than -35 volts, average min greater than -21.5 volts.
53	GND	B			Common with pin 100.
54	SLAVE CLR*	B	L	O.C.	Reset signal to reset bus slaves. Must be active with POC* and may also be generated by external means.
55	DMA0*	M	L	O.C.	Temporary master priority bit 0.
56	DMA1*	M	L	O.C.	Temporary master priority bit 1.
57	DMA2*	M	L	O.C.	Temporary master priority bit 2.
58	sXTRQ*	M	L		Status signal which requests 16-bit slaves to assert SIXTN*.
59	A19	M	H		Extended address bit 19.
60	SIXTN*	S	L	O.C.	Signal generated by 16-bit slaves in response to the 16-bit request signal

					sXTRQ*.
61	A20	M	H		Extended address bit 20.
62	A21	M	H		Extended address bit 21.
63	A22	M	H		Extended address bit 22.
64	A23	M	H		Extended address bit 23.
65	NDEF				Not to be defined signal.
66	NDEF				Not to be defined signal.
67	PHANTOM*	M/S	L	O.C.	Bus signal which disables normal slave devices and enables phantom slaves. Primarily used for bootstrapping systems without hardware front panels.
68	MWRT	B	H		Gating of pWR*-sOUT. Signal must follow pWR* by no more than 30 ns.
69	RFU				Reserved for future use.
70	GND	B			Common with pin 100.
71	RFU				Reserved for future use.
72	RDY	S	H	O.C.	See comments for pin 3.
73	INT*	S	L	O.C.	Primary interrupt request bus signal.
74	HOLD*	M	L	O.C.	Control signal used in conjuction with pHLDA to coorodinate bus master transfer.
75	RESET*	B	L	O.C.	Signal used to reset bus master devices. This signal must be active with POC* and may be generated by external means.
76	pSYNC	M	H		Control signal identifying BSl.
77	pWR*	M	L		Control signal signifying the presence of valid data on DO bus or data bus.
78	pDBIN	M	H		Control signal that request data on the DI bus or data bus from the currently addressed slave.
79	A0	M	H		Address bit 0.
80	A1	M	H		Address bit 1.
81	A2	M	H		Address bit 2.

IEEE Specification, IEEE Task 696.1/D2

82	A6	M	H		Address bit 6.
83	A7	M	H		Address bit 7.
84	A8	M	H		Address bit 8.
85	A13	M	H		Address bit 13.
86	A14	M	H		Address bit 14.
87	A11	M	H		Address bit 11.
88	DO2/DATA2	M/MS	H		Data out bit 2,bidirectional bit 2.
89	DO3/DATA3	M/MS	H		Data out bit 3,bidirectional bit 3.
90	DO7/DATA7	M/MS	H		Data out bit 7,bidirectional bit 7.
91	DI4/DATA12	S/MS	H		Data in bit 4,bidirectional bit 12.
92	DI5/DATA13	S/MS	H		Data in bit 5,bidirectional bit 13.
93	DI6/DATA14	S/MS	H		Data in bit 6,bidirectional bit 13.
94	DI1/DATA9	S/MS	H		Data in bit 1,bidirectional bit 9.
95	DI0/DATA8	S/MS	H		Data in bit 0,bidirectional bit 8.
96	sINTA	M	H		Status signal identifying the bus input cycles that may follow an accepted interrupt request presented on INT*.
97	sWO*	M	L		Status signal identifying a bus cycle which transfers data from a bus master to a slave.
98	ERROR*	S	L	O.C.	Bus status signal signifying an error condition during the present bus cycle.
99	POC*	B	L		Power-on clear signal for all bus devices. During active condition, this signal must stay low for at least 10 msecs.
100	GND	B			System ground bus.

Compatible S-100 Products

As a service to our customers, we have decided to provide information about other S-100 products that will work correctly with our new double density interface.

The list below only represents those products that We here at Tarbell Electronics have tested ourselves, and does not imply that these are the only S-100 that the double density interface will work with, and it is hoped that our customers will provide us with additional information about other S-100 products that they have found works with our interface.

CPU's

Imsai 8080, Delta Products Z-80, Cromemco Z-80, S.D.Sales Z-80, Vector Graphics Z-80.

** NOTE ** S.D.Sales SBC-100 and SBC-200 will not work because there are no bus disable lines on their board.
Ithica Audio Z-80 cpu's have been found to be unreliable with our double density interface when running with the DMA mode of operation.

MEMORY

Tarbell 32K static, Seals 32K static, Spacebyte 16K static, Industrial Microsystems 16K static, Godbout 32K static, Measurement System & Control DM 6400 and DMB 6400 dynamic boards

I/O interfaces

Processor Tech 3P+S, MITS SIO2, Thinker Toys Switcher, Vector Graphics Bit streamer, Dynabyte Intelligent CRT, S.D.Sales VDB-8024.

Drives

Shugart SA-800/801, Siemens FDD 100-8, FDD 100-8D / 120-8 / 220-8, CDC 9606 2/3, Remex RFD 4000/4001, MFE model 700.

NOTE *****

In order to use the Delta Products CPU, you must disable the Power-on-jump to the prom. The following modification to the CPU card will allow you to use the bootstrap on the floppy interface.

1.( ) Remove IC10c, bend pin 11 out away from the socket, and put IC10c back in.
2.( ) Ground pin 13 of IC13c.

This modification now allows you to boot up our disk controller by pushing the front panel reset button.

The Measurement System & Control dynamic memory card is not guaranteed to run with an 8080 cpu and our double density interface at 2 Mhz. They will run ok at 2 or 4 Mhz using Z-80 CPUs.

Compatible S-100 Products

Some dynamic memory cards require PSYNC to control refreshing on their cards. As our double density interface does not supply PSYNC during a DMA cycle, there may be a problem in refreshing on these memory cards.

DISK TEST ROUTINES

1. Test routines

The following routines are provided so that you may test your interface for proper operation. These routines involve both the TYPE 1 and TYPE 2 commands for the floppy disk controller chip. *** Note *** These routines assume that you have a front panel on your computer. If you do not have one, you will have to modify these programs to work with a system monitor, or whatever.

```
0000                    ORG     0
                    ;
                    ; define disk i/o ports.
                    ;
00F8 =              DCOM    EQU     0F8H            ;DISK COMMAND PORT
00F8 =              DSTAT   EQU     DCOM            ;DISK STATUS PORT
00F9 =              TRACK   EQU     DCOM+1          ;TRACK PORT
00FA =              SECT    EQU     DCOM+2          ;SECTOR PORT
00FB =              DDATA   EQU     DCOM+3          ;DATA PORT
00FC =              WAIT    EQU     DCOM+4          ;WAIT PORT
                    ;
                    ;The following routine will allow you to
                    ;check for any internal shorts or opens
                    ;with in the floppy interface. The routine
                    ;reads the front panel switches, writes it
                    ;to the floppy disk data port, then reads
                    ;it back from the floppy disk data port, and
                    ;writes it back out to the front panel lights.
                    ;there should be a one to one match of the
                    ;switches to the data lights. If more than
                    ;one light comes on at a time, there is an
                    ;internal short on the floppy data lines. If
                    ;a light does not come on when a switch it
                    ;turned on, then there is an open line on that
                    ;data bit.
                    ;
                    ; TEST ROUTINE FOR CHECKING
                    ; DISK INTERNAL DATA BUS.
                    ;
0000 DBFF           BUSS:   IN      0FFH            ;GET FRONT PANEL DATA
0002 D3FB                   OUT     DDATA           ;SEND IT TO CONTROLLER
0004 DBFB                   IN      DDATA           ;GET DATA BACK AGAIN
0006 2F                     CMA                     ;INVERT FOR FRONT PANEL
0007 D3FF                   OUT     0FFH            ;SHOW IT ON FRONT PANEL
0009 C30000                 JMP     BUSS            ;LOOP AGAIN
```

DISK TEST ROUTINES

```
                        ;
                        ;The following routine allows testing the
                        ;seek operation of the interface. You will
                        ;need to put in blank disk into the drive
                        ;and close the door. Do not put in a disk
                        ;which contains any important data, as you
                        ;may wipe it out. Do not set the front panel
                        ;switches for anything higher than 3F hex,
                        ;as this will cause the head to slam up past
                        ;track 76, possibly causing damage.
                        ;
                        ; SEEK TEST ROUTINES
                        ; DON'T SET FRONT PANEL
                        ; SWITCHES FOR A HEX VALUE
                        ; HIGHER THAN 3F OR YOU WILL
                        ; SLAM THE HEAD BEYOND TRACK
                        ; 76.
                        ;
000C DBFF               SEEK:   IN      0FFH            ;READ FRONT PANEL
000E D3FB                       OUT     DDATA           ;SEND TRACK NUMBER
0010 3E13                       MVI     A,13H           ;GET SEEK COMMAND
0012 D3F8                       OUT     DCOM            ;SEND IT
0014 C30C00                     JMP     SEEK            ;LOOP AGAIN
                        ;
                        ;The following routine test for step in
                        ;and step out. You will need to mount a
                        ;disk into the drive and shut the door.
                        ;DO NOT PRESS RUN !!! for this test. Use
                        ;single step operation.
                        ;The program as listed will step in toward
                        ;track 76. To step out toward track 0, you
                        ;need to change the 43H to a 63H.
                        ;
                        ; STEP IN/OUT ROUTINE
                        ;
0017 3E43               STEP:   MVI     A,43H           ;STEP IN COMMAND
0019 D3F8                       OUT     DCOM            ;ISSUE COMMAND
001B C31700                     JMP     STEP            ;LOOP AGAIN
                        ;
                        ;The following routine checks for head
                        ;load command. You will need to mount a
                        ;disk and shut the door to run this test.
                        ;DO NOT USE A VALUABLE DISK !!! for this
                        ;test.
                        ;
                        ; TYPE 2 COMMAND
                        ; HEAD LOAD ROUTINE
                        ;
001E 3E8C               HEAD:   MVI     A,8CH           ;HEAD LOAD COMMAND
0020 D3F8                       OUT     DCOM            ;ISSUE COMMAND
0022 C31E00                     JMP     HEAD            ;LOOP AGAIN
```

DISK TEST ROUTINES

```
                        ;
                        ;The following routine reads a sector
                        ;from the disk and puts the data into
                        ;a buffer call RAMADD. This buffer can
                        ;be anywhere in memory you wish.
                        ;The routine saves the status of the disk
                        ;operation at the location called STAT.
                        ;If you have problems, check the bits at
                        ;this location against the status bits
                        ;shown in the data sheet for the floppy
                        ;controller chip, in Appendix A (1793).
                        ;You will need to mount a blank disk into
                        ;the drive and shut the door.
                        ;DO NOT USE A VALUABLE DISK !!! for this
                        ;test.
                        ;
                        ; READ A SECTOR ROUTINE
                        ;
1000 =                  RAMADD  EQU     1000H           ;CAN BE ANYWHERE
                        ;
0025 3ED0               READ:   MVI     A,0D0H          ;ISSUE FORCE INTERRUPT
0027 D3F8                       OUT     DCOM            ;SEND IT
0029 E3                         XTHL                    ;SOME
002A E3                         XTHL                    ;DELAY
002B 210010                     LXI     H,RAMADD        ;POINT TO MEMORY BUFFER
002E 3E01                       MVI     A,1             ;LOAD SECTOR NUMBER
0030 D3FA                       OUT     SECT            ;SEND IT TO SECTOR PORT
0032 3E8C                       MVI     A,8CH           ;GET READ COMMAND
0034 D3F8                       OUT     DCOM            ;ISSUE READ COMMAND
0036 DBFC               RLOOP:  IN      WAIT            ;WAIT FOR DRQ OR INTRQ
0038 B7                         ORA     A               ;SET FLAGS FOR OPERATION
0039 F24300                     JP      RDONE           ;DONE IF INTRQ
003C DBFB                       IN      DDATA           ;ELSE, READ A BYTE FROM DISK
003E 77                         MOV     M,A             ;PUT IT INTO THE BUFFER
003F 23                         INX     H               ;BUMP BUFFER POINTER
0040 C33600                     JMP     RLOOP           ;READ NEXT BYTE FROM DISK
0043 DBF8               RDONE:  IN      DSTAT           ;CHECK DISK STATUS
0045 324B00                     STA     STAT            ;SAVE THE STATUS
0048 C34800             HJMP:   JMP     HJMP            ;HARD LOOP
004B 00                 STAT:   DB      0               ;SAVE STATUS HERE WHEN DONE
```

```
                        ;
                        ;The following routine writes one
                        ;sector of the disk. If you have
                        ;a problem, check the status bits
                        ;at location WSTAT against the bits
                        ;in the data sheet for the 1793 in
                        ;the appendix.
                        ;This routine is similar in operation
                        ;to the read a sector routine above.
                        ;You must mount a blank disk into the
                        ;drive and shut the door.
                        ;DO NOT USE A VALUABLE DISK !!! as
                        ;this routine does write on the disk,
                        ;and you will lose any data on the
                        ;sector you write to.
                        ;
                        ; WRITE A SECTOR ROUTINE
                        ;
004C 3ED0       WRITE:  MVI     A,0D0H          ;FORCE INTERRUPT COMMAND
004E D3F8               OUT     DCOM            ;ISSUE IT
0050 E3                 XTHL                    ;SOME
0051 E3                 XTHL                    ;DELAY
0052 3E01               MVI     A,1             ;GET SECTOR NUMBER
0054 D3FA               OUT     SECT            ;TELL SECTOR PORT
0056 3EAC               MVI     A,0ACH          ;GET WRITE COMMAND
0058 D3F8               OUT     DCOM            ;ISSUE IT
005A DBFC       WLOOP:  IN      WAIT            ;WAIT FOR DRQ OR INTRQ
005C B7                 ORA     A               ;SET FLAGS FOR OPERATION
005D F26700             JP      WDONE           ;JUMP WHEN INTRQ
0060 7E                 MOV     A,M             ;GET DATA FROM BUFFER
0061 D3FB               OUT     DDATA           ;PUT IT TO DISK
0063 23                 INX     H               ;BUMP BUFFER POINTER
0064 C35A00             JMP     WLOOP           ;LOOP FOR MORE DATA
0067 DBF8       WDONE:  IN      DSTAT           ;CHECK DISK STATUS
0069 E6FD               ANI     0FDH            ;MASK OFF NON ERROR BITS
006B 327100             STA     STATW           ;SAVE STATUS BITS
006E C36E00     WJMP:   JMP     WJMP            ;HARD LOOP WHEN DONE
0071 00         STATW:  DB      0               ;SAVE STATUS BITS HERE
```

2. Port and bit explanation

The explanation for the disk controller ports are covered in the 1793 data sheet and will not be covered here. The extended ports FC and FD Hex will be explained.

An input from port FC Hex is a combination hardware-software port which allows the disk controller to stall the CPU with a wait state to allow the disk controller chip time to access the data bus. This is generated by the gating of DRQ and INTRQ from the floppy chip onto either the XRDY or PRDY lines of the bus. Bit 7 of port FC is also used in the software during programmed data transfers to tell the software read/write and seek routines when an interrupt from the floppy chip has occurred. When this bit is a '1' an interrupt has occurred, and the status port of the 1793 must be read to clear it back to a '0'.

An output to port FC Hex is also the drive select, density select, and side select port. A break down of the port is as follows:

LSB	(bit 0)	x	not used
	(bit 1)	x	not used
	(bit 2)	x	not used
	(bit 3)	1/0	0 = single density, 1 = double density
	(bit 4)	1/0	binary value for drive select } 4 drives max
	(bit 5)	1/0	binary value for drive select
	(bit 6)	1/0	0 = side 0, 1 = side 1 (side select)
MSB	(bit 7)	x	not used

An input from port FD Hex is the software port for checking the INTRQ of the floppy chip while running under DMA operation. This port polls bit 7 of this port looking for a low (0), which tells the CPU that the DMA operation is complete. A one (1) means that the DMA is busy. All the rest of the bits in this port are not used.

An output to port FD hex may be used to change the extended addressing bits on A16 - A23 during a DMA read or write operation. This port is an latched 8 bit port and is enabled on to the S-100 bus only during DMA read or write operations. The bits which are placed into the latch, (U68), should be written only when the DMA controller is in-active, (not doing disk transfers). A typical example might be:

```
MVI     A,1             ;select bank # 1
OUT     0FDH            ;set extended port latch
```

```
        .
dma operation           ;transfer to/from new bank now
        .
        .               ;dma operation complete
        .
MVI     A,0             ;select bank # 0
OUT     0FDH            ;set extended port latch
        .
    etc
```

TROUBLE-SHOOTING

This section of the manual is to help you trouble-shoot the interface in case you are having problems.

1. Interface between computer cards.

Compare the section on the IEEE S-100 pin functions (section 10), to the pin function list in your cpu manual, to find out if there are any conflicts with your computer and the new bus specification. If there are, you should make any necessary corrections for compatibility.

2. Compatible interfaces

Review the section on S-100 compatible interfaces (section 11). The computer cards listed in section 11 only represents those products that we here at Tarbell Electronics have tested ourselves, and in no way defines any not in this list as not working with out interface. We would like to encourge our customers to send in any information with regard to other manufactures products that do or 'do not' work with our interface. If they don't work, maybe a reason as to why could be sent along also. Also, some Z80 cpu cards will not boot up correctly at 4Mhz with our board. We have found that in some cases cutting the phase 1 line (25) of the bus on the controller card may help.

3. Jumper selections

Review the section on jumper selection. If you are going to be running our software, then you must select the address jumper from the table in section 6 for the address range of E0 - FD hex. If you are going to use the on-board bootstrap prom, you "MUST" use the phantom line (pin 67) to disable your first memory card (addressed at 0000 hex). If you are going to use your own bootstrap program, then you must defeat our on-board bootstrap prom. If you have a revision B or below, you can turn off the on-board bootstrap by pulling out U35 pin 5 from it's socket, and then ground U33 pin 13 to U33 pin 7. For revision C and above, follow the instructions listed in the jumper options section.

4. XRDY or PRDY line

If you are getting lost data errors, besure to check which line you are using for the processor wait line. This is important during a cold start load, as the on-board prom bootstrap uses this line during the booting of the system. If you are using the wrong line, you will more than likely get lost data errors. If you are not sure which line to use, review the section on jumper options (section 6).

5. Disk Problems

Be sure that the disk you are trying to read has the correct format on it. It must be an IBM soft sector disk, 128 bytes/sector. Most disks shipped from the manufacturer that claim to be soft sector, are. But if you have reformatted the disk with the wrong format program, then you will have trouble reading it on this interface. If

you have any questions about this, review the documentation that is supplied with the interface. I can not over stress this point.

6. Drives

This interface is Shugart compatible only. That means drives which comply with the Shugart pin functions will more than likely work with no probblems. If you do have problems, consult the OEM drive manual or the factory for information. The drives that have been tested here at Tarbell Electronics are the Siemens, Shugart, and Persci drives. Also, be sure that the drive is set up for soft sector operation. We have recieved drives from the factory, set up for hard sector in the past. So look carefully at your drives for this possibility.

7. Review

Be sure and review section 2-3 in the front of this manual very carefully, as it can not be over stressed enough.

8. Symptoms vs Possible causes

This table is a list of possible troubles with possible causes to look at.

Symptom	Possible cause
1. Won't boot.....	Check pin 67. Must use Phantom line on first memory card. Make sure jump enables to system proms is defeated. Check for disk with correct format. Check C17, may have to change the value of this capacitor as outlined in section 2-3 of this manual. Check disk drive power supply lines for all voltage return lines tied to one commen place at power supply. Is there an operating system on the disk you are trying to boot from?
2. Won't run DMA	Check IEEE standard for compatibility with memory cards, cpu, and i/o cards in your system. If you are using a dynamic memory are there any problems with refreshing and control lines necessary to initiate the refreshing signals and our interface? Check for slow memory chips, e.g. old 2102's. Check our compatibility list.
3. Double density errors .	Are you using a good certified double denstiy disk? Have you adjusted the pre-comp settings for your drive using the DFRAND.COM program? Is your drive rated for double denstiy operation?

4. Won't run at all	Check voltage regulator on interface for correct voltage of +5 volts. Are any changes required to the board in the way of corrections to mistakes on the board? If not a factory assembled unit, are the changes on correctly?

DISK DRIVE SETUP

This interface is designed for Shugart compatiable drives. This includes Siemens drives also. Jumpers and power requirements for both drives are listed below.

Power requirements:

Shugart and Siemens

DC Power		AC power	
Pin 1	+24 volts DC	Pin 1	110 volts AC
Pin 2	+24 volt return	Pin 2	frame ground
Pin 3	- 5 volt return	Pin 3	110 volts AC
Pin 4	- 5 volts DC		
Pin 5	+ 5 volts DC		
Pin 6	+ 5 volt return		

Be sure all return lines are connected solidly together at the power supply end. If they are not, you may experience actual disk data errors.

Shugart Drive Jumpers

The following jumpers should be installed on each drive used:

Drive 0 (A)	Drive 1 (B)	Drive 2 (C)	Drive 3 (D)
A	A	A	A
B	B	B	B
C	C	C	C
Y	Y	Y	Y
T2	T2	T2	T2
DS	DS	DS	DS
800 (not 801)	800	800	800
DS1	DS2	DS3	DS4
T1			
T3			
T4			
T5			
T6			

Note : Some drives may have a jumper installed at X. If so, remove it.

DISK DRIVE SETUP

Siemens -------

Drive select:

Drive A	Drive B	Drive C	Drive d
Radial Sel 0	Radial Sel 1	Radial Sel 2	Radial Sel 3

Note :
Some drives may come with Radial Sel 0 etched on the board. You will have to cut this etch if you wish to use it for a drive other than Drive A.
If you are using more tan one drive, you must remove all the terminating resistor packs but the last one in the daisy chain.
Be sure that the drive is jumpered for SOFT SECTOR options.
If you have any doubts, consult the OEM manual for your drive.
Siemens model FDD 100-8D drives may come from the manufacture with jumper G installed. You must change this jumper to location H for proper operation with our interface.

Persci drives

Changes are required to the Persci drive to make it look like a Shugart compatible drive. These changes are listed in the Jumper section of this manual.