Preliminary documentation on hard disk controller and multi hard disks. November 23, 1982.
Updated, 6/23/83 Firmware 2.7 DRB
Updated, 1/06/84 Firmware 3.0 DRB

1.0 HARD DISK CONTROLLER

1.1 OVERVEIW

The CPU-floppy disk drive sub-system of a DSC-3/4 hard disk unit may be operated under CP/M in the normal way. However, a second board is necessary for the harddisk. This board is called the Hard Disk Controller (HDC).

The HDC has its own Z80 CPU with 32K of RAM buffer. It also has a 1K prom. The flow of data for a disk write is as follows:

A. Data exits the host CPU board through J3 (parallel port).
B. Data enters the HDC board through P1h (parallel port).
C. Data is converted to serial data by the HDC board.
D. Data exits the HDC board through P1d (drive interface).
E. Serial data is written on the harddisk.

1.2 THEORY OF OPERATION

Most of the basic operations of the HDC are stored in ROM as programmed sequences. These include:

A. Power up or hard reset initialization.
B. Host command and data communication.
C. Disk read and write operations.
D. Error detection.

NOTE---Error correction is handled by firmware in RAM.

1.2.1 Power Up or Hard Reset Initialization

The HDC begins execution upon power up at 0000h. If you executed a system reset, the HDC jumps to this location. The initialization procedure begins by setting the stack pointer, clearing the host interface status flip-flop.
Next, 1K of the software for operation with your CP/M or Oasis operating system is read into memory from track 0, head 0, sector 1 and stored in memory starting at 4100h. The HDC now waits in a loop for a command from the host computer.

When the first command for operating system firmware is received, the firmware does its initilization. This includes finding which volumes are present for a multidrive system, reading an additional 1 or 2 K or firmware from the disk and reading each disks bad sector table.

1.2.2 Host Command and Data Communication

The HDC samples the status of the host interface during each loop. If the HDC detects data on the interface, it reads the data and compares it with a "request to send" (51h). If it is a "request to send," the HDC sends a "clear to receive" (52h) to the host. The 8 byte command block is read into

memory starting at 4080h. The host can send commands to the HDC. The commands have been discussed previously.

1.2.3 Disk Read and Write Operations

A read command causes a disk read. If an error occurs, the location on the harddisk will be reread the number of times specified by byte 6 of the command block.

The BST is consulted to determine if the disk location is written on a replacement sector on the last two valid tracks. When the correct location is determined, the head is selected and the difference between the old track and the new track computed.

1.2.4 Error Detection

An error message is generated and the read operation stops if the track number generated is greater than the maximum number of tracks on the harddisk. If all is correct, step pulses are generated and the head is moved to the new position on the harddisk. The HDC program waits for the SEEK COMPLETE signal from the drive. It waits an additional 20 msec for the head to settle before starting the read.

The next task is to find the correct sector. A sector of data is stored on the harddisk in the following format:

14 bytes	sync	8 bytes	1024 bytes	4 bytes	8 + bytes
0's	0Fh	Header	Data	ECC	0's

The correct sector is found using a read and abort. The first 32 bytes of each sector are read. If the sector read is not the sector previous to the sector wanted, the read is repeated on the next sector. When the sector previous to the desired sector is found the HDC reads the entire next sector.

The hardware sequence for a read and abort or a full sector read is identical. The only difference is the HDC software aborts the read after 32 bytes.

1.3.5 HARD DISK READ

The hardware read sequence is started with an output command from the HDC CPU. This resets the byte counter and turns over memory buffer control to the HDC control logic. Data is read off the disk at the beginning of the next sector pulse. The control logic searches for the sync byte. When the sync byte is read, a bit counter starts and data is shifted into a shift register. After 8 bits have been shifted, the data is loaded into the memory buffer. The memory location of the buffer is incremented and more data is stored as each byte is assembled from the disk. This process continues until the HDC CPU takes back control of the memory buffer.

The control logic has several signals which the HDC CPU monitors while data is read off the disk. These signals are:

BYTE32 Monitored during a read and abort for the 32 bytes of header data for a sector.

BUSY This signal is checked to determine when

the disk read is finished.

ERROR Error "on", indicates that data read off the disk failed the ECC check.

BYTE2048 This signal is used to time the hardware portion of the error correction.

When BYTE32 returns to "0", control is returned to the HDC CPU. If the CPU is doing a complete sector read, it moniters the BUSY line. When the disk has completed the read, BUSY returns to "0" and the HDC CPU checks the ERROR line. If ERROR is "0", there is no error in the read and control returns to the HDC CPU. Data is moved from the HDC data buffer to the host computer RAM and the harddisk read is complete.

1.2.6 Read Error Correction

If ERROR is "1", the HDC CPU does not take over memory control from the HDC control logic. Instead, the CPU monitors BYTE2048 and takes control when it becomes active.

The time between BUSY inactive and BYTE2048 active is the amount of time the error correction hardware needs to generate the required correction information. This information is a bit stream approximately 1K bytes long. This bit stream is loaded into memory and located after the data buffer. The error correction software can correct errors in the data field with this error information. This error correction scheme has the ability to correct a six bit error burst. However, a maximum of 4 bits are corrected because this reduces the probability of a false correction to zero, assuming burst errors of

length 11 or less or two errors of length 2 or less.

When the data is corrected, it is transmitted to the host computer and the HDC waits in a loop for another command.

1.2.7 Hard Disk Writes

A "write command" writes a 1K byte block of data to the harddisk. The HDC will retry a write error the number of times specified by bits 0 through 6 of byte 6 of the command block. The HDC will also perform a read after write to verify data written to the harddisk if bit 7 of byte 6 of the command block is set to "1".

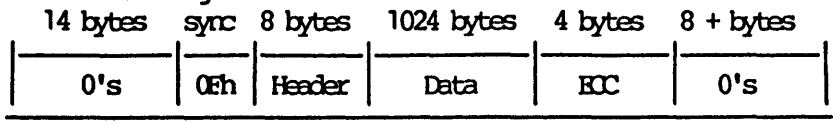
A write sequence begins with a transfer of data from the host CPU to the HDC. This data is stored in the HDC I/O buffer beginning at E017h. Next, the specified disk location is compared to entries in the BST. If a match occurs, a specific sector from the spare tracks is used for the write. The correct head is now moved to the track for the write. This is done in the same manner as a read from the hard disk.

The data preamble is now formatted in memory starting at E000h. It has the following format:

		<---------- 8 byte HEADER ---------->					
14 bytes	sync	1 byte	1 byte	1 byte	2 bytes	2 bytes	1 byte
0's	0Fh	track number	head number	sector byte	spares bytes	0's	header chk sum

A sector of data to be written to the harddisk has

the following format:



The correct sector to be written to on the harddisk is found in the same manner as for a read from the harddisk.

The hardware write sequence begins with an output command from the CPU. For the Format Write Command (command 4), sector 1 is written using INDEX as the reference point. This is necessary for an unformatted harddisk because the previous sector may not have been created yet. If this was the case, the normal reference for reading and writing a sector will not exist. All other sectors are written in the normal manner.

The sector pulse following the CPU output command clocks WRTGATE to "1" and resets the ECC shift registers (to zero). WRTGATE write enables the harddisk. Control of the data buffer memory is given to the controller logic. The bit counter is already counting in sync with the disk generated clock. All disk write functions are synchronous with this clock. With the bit counter counting, data is read from memory starting at E000h. It is shifted out of the shift register at 7.1 mbits per second. The ECC hardware is generating an ECC code while data is being written. After the data preamble and data have been written to the disk, a four byte ECC code is written. The rest of the sector is filled with 0's from the buffer until the next sector pulse. The sector pulse clocks WRTGATE to "0" which disables writes to the harddisk. This completes the hardware write sequence.

1.2.8 Write Error Correction

During the time the HDC logic is shifting data to the disk, the HDC CPU is monitoring BUSY and WRTFLT. If a fault condition develops during a disk write, a WRTFLT signal is generated. This results in the HDC CPU regaining control of the RAM, issueing a Fault Reset write retries are attempted if the command block indicates retries are to be attempted. The HDC will also do a read after write check in the command block requested it.

1.3 BOARD DESCRIPTION

The HDC has the following subsections:

1. A Z80 CPU operating at 4Mhz under ROM or RAM control.
2. 16K of CPU executable memory, 16K of I/O buffer memory, and a 1K prom.
3. Parallel port interface to DSC-3 or DSC-4.
4. HDC board to harddisk drive interface port.
5. HDC control program. This program is run by the Z80 CPU on the HDC board. The program performs 3 major functions:

    A. Operate the 16K buffer area.
    B. Control head select and seek. Moniter the drive status signals.
    C. Generate check bits on write and verifiy on read. Perform error correction if necessary
    D. Communicate with the host CP/M

Host CPU to HDC Interface

The parallel port interface is the means of communication between the host CPU and the HDC CPU. The two boards are connected by a 26 conductor ribbon cable.

The host CPU must provide the following I/O signals to operate the interface:

INSTATUS/ An active low signal samples the interface status bits. (gates the status bits onto the host CPU data bus)

INDATA/ When INDATA is active low, data is gated from the interface chip (74LS374 on the HDC board) onto the host CPU's data bus.

OUTDATA/ When OUTDATA is active low, data is gated from the host's data bus to the interface chip on the HDC board.

RESET/ When RESET is active low, rom is reset and restarts from location 0.

NMI/ When NMI is active low, a non maskable interupt occurs and the interupt processing restarts the rom at location 0.

These signals have the following port numbers on a DSC 3/4:

INSTATUS/	8H
INDATA/	1H
OUTDATA/	1H
RESET/	
NMI/	0H

1.4 SOFTWARE FOR HANDSHAKING

The CPU and HDC boards work asynchronously and hand shaking is necessary to communicate. This program sample is the handshake code which is executed before a transfer of data. Both subroutines must be initialized for the same sized block of data. They also must use the same buffer address in memory.

1.4.1 Host CPU Write Block Subroutine

```
; This subroutine writes a block of data to the PP interface
; Enter the subroutine with registers set as follows:
;    DE = Block size or byte count
;    HL = Address of data buffer
;
writPP:
     mvi  C,outdata
     mov  B,E
     mov  A,D
     ora  B
     rz             ;if byte count = 0, finished
wrtloop:            ;write data to parallel port
     in   instatus
     rrc
     jrc  wrtloop   ;loop if HDC has not received data
     outi           ;send data to port, HL=HL+1, B=B-1
     jrnz wrtloop   ;if B=0, fall through
     dcr  D
     jrnz wrtloop   ;loop if D has more
     ret            ;finished writting to parallel port
```

1.4.2 Read Block Subroutine

```
; This subroutine reads a block of data from the PP interface
; Enter the subroutine with registers set as follows:
;    DE = Block size or byte count
```

```
;    HL = Address of data buffer
;
readPP:
     mvi  C,fb
     mov  B,E
     mov  A,D
     ora  A
     rz             ;if byte count = 0, finished
rdloop:             ;read data from the parallel port
     in   fc
     rrc
     rrc
     jrnc rdloop    ;loop if host CPU hasn't sent data
     ini            ;rec data from port, HL=HL+1, B=B-1
     jrnz rdloop    ;if B=0, fall through
     dcr  D
     jrnz rdloop    ;loop if D has more
     ret            ;finished reading from parallel port
```

1.5 HDC TO HARDDISK INTERFACE BOARD

The HDC to harddisk interface provides all of the necessary signals for the operation of the harddisk. The interface connects these signals to the HDC logic.

The following interface signals are necessary:

A. WRTGATE/ active low enables harddisk for writing data
B. WRTDATA write serial data to the harddisk
C. RDCLK clock signal to read to and write from harddisk
D. RDGATE/ active low enables harddisk for reading data
E. RDDATA serial data input line from the hard disk
F. SECTOR/ active low pulse marks beginning of each sector

All other signals are connected via input and output ports. These ports are FE and FF.

The signals for each port are as follows:

A. IN FE clears the index status flip-flop. The index status flop-flop is set on for every INDEX pulse from the harddisk. The INDEX pulse occurs once for every revolution of the disk. One revolution of the harddisk takes 20.24 msec. INDEX indicates sector 0 for a track.

B. IN FF IN FF samples all of the disk drive status information. The data bits are defined as follows:

Bit	Description
D0	A logical one indicates that the drive is ready for operation.
D1	A logical one indicates seek complete. (SKCOMP)
D2	A logical one indicates the R/W arm is on track 0.
D3	A logical one indicates a WRTFLT. This means that a fault condition exists on the harddisk. This condition is reset by an active FAULT CLEAR. Read and write operations to the harddisk are inhibited until FAULT CLEAR goes active then inactive.
D4	A logical one indicates that INDEX status flip-flop has been set by an index pulse.

C.OUT FF Outputs harddisk command data to the drive

Bit	Description
D0	head select bit 1 or complement of bit 0 of volume to select
D1	head select bit 2 or complement of bit 1 of volume to select
D2	head select bit 3
D3	head select bit 4

The correct head is selected using the binary number given by the above bits. D0 is the low order bit and D3 is the high order bit.

D4 This bit indicates the direction of the head movement when step pulses are applied. A logical zero defines the direction as towards the edge of the disk. A logical one defines the direction as towards the center of the disk.

D5 This is the step pulse line which causes head movement. The harddisk control program provides the tep pulses to the harddisk read/write heads. A step pulse is a low level outpudt followed by a high level followed by a low level output.

D6 This is the FAULT CLEAR bit. This is an active high signal. When this bit is toggled high then low bits 0 and 1 are interpreted as the complement of the volume to select by the disk multiplexor.

D7 Not used.

D. IN FB, OUT FB Data from and to host

E. IN FC Samples host port status.

Bit	Description
D0	One if data to host is busy (full).
D1	One if data availible from host.

F. OUT FC Command to hard disk, bits as follows:

D0 One for read request.

D1 One for write request.

D2 One to give ram to disk interface.

D4 One for write enable

D6 Sets bit 14 in ram address when disk has ram.

D7 Sets bit 15 in ram address when disk has ram.

G. IN FD Samples status of controller hardware, bits as follows:

Bit	Description
D4	Set if byte 32 has been read.
D5	Set if busy with write.
D6	Set if bit 2048 processed thru crc circuitry.
D7	Set if CRC error occurred.

1.6 DESCRIPTION OF FIRMWARE

There are two levels of HDC firmware. Level one resides in ROM. It is 1K bytes from 0000 to 03FFh. It controls all of the basic disk operations and communication with the host CPU. Level 2 resides in RAM from 4000h to 7FFFh max and is loaded from the harddisk. A cold boot or hard reset loads 1k of level 2 firmware
from the harddisk, track 0, head 0, sector 1.

Level one Firmware

The ROM program contains the routines for:

A. System initialization
B. Disk seek
C. Disk read. A physical sector = 1024 bytes
D. Disk write. A physical sector = 1024 bytes
E. Host system communication

Host to HDC communication always starts with:

A. Attention and syncronization handshake
B. 8 byte command block sent from the host to the HDC

Data may be transmitted in either direction, depending on the command type. Commands -1 through 5 are supported in ROM firmware. Commands 6 and higher are passed to level 2 RAM firmware. RAM firmware must be loaded correctly. It is loaded automatically at system initialization. It can also be loaded with commands 0 or 3, described below, from track 0, head 0, sector 1, or from the host.

For all commands, a "1" in the retry byte specifies

no retries (ie. this byte specifies total number of retries).

The following table summarizes the 7 commands available in level one firmware.

-1. Boot Volume

Byte 0 = 0ffh command type
Byte 1 = volume to boot (0-3)
Byte 2-7 unused

Firmware is loaded from the selected volume and all reads and writes will be directed to the selected volume until a new volume is selected.

-0. Load level 2 firmware

Byte 0 = 0 command type
Byte 1 = track
Byte 2 = head
Byte 3 = sector
Byte 4 thru 7 - not used

For commands 1 and 2, the TAG0 and TAG1 bytes can be used by the host system for any purpose. The TAG is stored with the 1024 byte data record.

1. Absolute sector read

Byte 0 = 1 Command type
Byte 1 = track
Byte 2 = head
Byte 3 = sector
Byte 4 = TAG0
Byte 5 = TAG1

Byte 6 = number of retries + 1
the value of the high order bit of the retry byte needs to be a "1" if you want error correction. This should almost always be the case.
Byte 7 = 0

2. Absolute sector write

Byte 0 = 2 Command type
Byte 1 = track
Byte 2 = head
Byte 3 = sector
Byte 4 = TAG0
Byte 5 = TAG1
Byte 6 = number of retries + 1
the value of the high order bit of the retry byte needs to be "1" to request a "read after write check". The remaining 7 bits of the retry byte determine the number of retries.
Byte 7 = 0

* Command 2 must be followed by a 1K block of bytes.

3. Load level two firmware from host (1K byte)

Byte 0 = 3 Command type
Byte 1 thru 7 = 0 (not used)

* Command 3 must be followed by a block of bytes.

4. Format write (write sector 1 from INDEX)

Byte 0 = 4 Command type
Byte 1 = track
Byte 2 = head
Byte 3 = sector
Byte 4 = TAG0
Byte 5 = TAG1
Byte 6 = number of retries + 1

If high order bit = 1, read after write check Byte 7 = 0

5. Get contents of HDC memory

Byte 0 = 5 Command type
Byte 1 = 0 (not used)
Byte 2 = low 8 bits from HDC
Byte 3 = high 8 bits from HDC
Byte 4 = low 8 bits of byte count
Byte 5 = high 8 bits of byte count
Byte 6 = not used
Byte 7 = not used

Note---ROM command -1 is only implemented in firmware version 2.2 or later.

Special Commands
Commands 6-9 are used for special tests and formatting.

Level two firmware

Digital Microsystems, Inc. currently supports four operating systems, CP/M 1.4, CP/M 2.2, HiNet, and HIDOS.

The following table summarizes the 9 commands for CP/M 1.4 or 2.2 .

10h. Warm start (No longer supported or used)

Byte 0 = 10h Command type
Byte 1 = 0 for track 0
Byte 2 = 1 for head 1
Byte 3 = 1 for sector 1
Byte 4 = number of logical sectors required
Byte 5 thru 7 not used

11h. CP/M read (128 byte)
Byte 0 = 11h Command type
Byte 1 = sector (1 to 128)
Byte 2 = track (low order byte)
Byte 3 = track (high order byte)
Byte 4 thru 5 not used
Byte 6 = number of retries + 1
Byte 7 = 0

12h. CP/M write (128 byte)
Byte 0 = 12h Command type
Byte 1 = sector (1 to 128)
Byte 2 = track (low order byte)
Byte 3 = track (high order byte)
Byte 4 thru 5 not used
Byte 6 = number of retries + 1
Byte 7 = 0

Command 12h must be followed by a 128 byte block of bytes.

13h. CP/M drive select
Byte 0 = 13h Command type
Byte 1 = partition number
Byte 2 = volume number (new to multi disks)
Byte 3 thru 5 not used
Byte 6 = ascii 'M' (indicates host knows of multi disk drives) (byte 2 will be ignored if anything but "M")
Byte 7 not used

14h. Buffer Flush
Byte 0 = 14h Command type
Byte 1 thru 7 not used

for HardHelp 4.0 to select partition

G164,CAB

S2400 partition #

G

15h. 1k read
Byte 0 = 15h Command type
Byte 1 = sector (1 to 128)
A 1k disc sector will be returned which includes the CPM sector specified. Most programs should insure that CPM sector mod 8 is 1.
Byte 2 = track (low order byte)
Byte 3 = track (high order byte)
Byte 4 if true,(FFh) force a pre-read
if false,(0h) no forced pre-read
Byte 5 not used
Byte 6 = number of retries + 1
Byte 7 = 0

16h. 1k write (NOT IMPLEMENTED)

17h. Assign a partition
Byte 0 = 17h Command type
Byte 1 thru 7 not used
Byte 8 thru 15 = Name
Byte 16 thru 21 = Password (If PW is all 0's it will match regardless of actual password)
On return:
Byte 0 = size byte or 0ffh if denied
Byte 1 = partition number
Byte 2 = control byte
Byte3 = volume number (newtomulti-disks)
Byte 4 thru 7 unused

18h. Get Volume Information
Byte 0 = 18h Command type
Byte 1 = ROM version
Byte 2 = ROM reversion
Byte 3 = firmware version
Byte 4 = firmware reversion
Byte 5 thru 7 unused

On return:
Byte 0 = 18h Command echo
Byte 1 = rom version number (all versions binary)
Byte 2 = rom revision number
Byte 3 = firmware version number
Byte 4 = firmware revision number
Byte 5 thru 6 unused

status

If status = 0 there follows 128 bytes, 32 bytes for each volume. Each volume's info is as follows:

Byte 0 = volume present (true or false)
Byte 1 = current track for volume
Byte 2 = tracks on volume
Byte 3 = sector per track
Byte 4 = head mask (disk type)
Byte 5,6 = location of partition offset table
Byte 7,8 = location of bad sector table
Byte 9,10 = location of bad sector dirty flag
Byte 11 = error in volume open during firmware init (0 = no error)
Byte 12 thru 21 = volume label, 10 chars of ascii
Byte 22 thru 31 not used

19h. Write CRC error to last sector on disk
Byte 0 = 19h
Bytes 1 thru 7 unused

1.6. x Completion of a command

Commands 1,2,11h,12h,13h,14h,15h, and 16h, result in the HDC returning a status block upon completion of the command or an interruption of the command due to an error. The status block will be a copy of the command block, with the following changes:

For commands 1 and 2, if bytes 1, 2, or 3 have been changed, the HDC has substituted a spare sector for a defective sector. The track byte will be 200 or 201 for a Shugart drive and 241 or 242 for a Fujitsu drive.

If the retry byte has been decremented, errors were encountered. If the byte is decremented to 0, the error is permanent. (see status byte).

Commands 11 and 15, (reads) will have a data block following the return status only if no error occured. For command 15h, a 1K read, byte 5 will be true (FFh) if the sector came from the buffer and false (0h) if the sector was read directly from the hard disk.

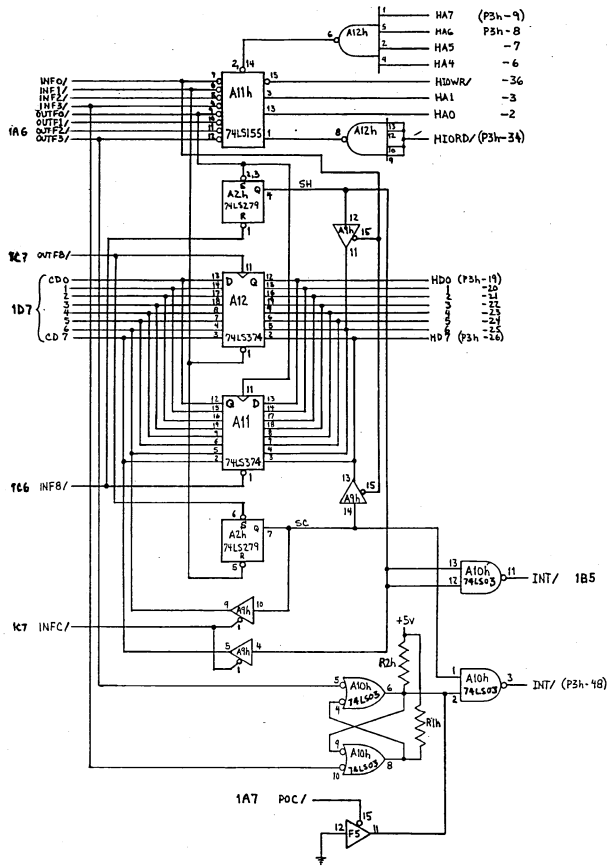
Byte 7 will be "0" for a non-error return. A nonzero in byte 7 (the last byte) indicates one of the following errors:

Hex Value of byte	Problem
11	Write fault: Possible drive problem
12	Drive not ready
13	Internal timeout HDC problem
14	Read after write compare error
15	Unprocessed CRC error. ROM detected error but firmware error correcting code was not present.
16	Volume selected (command 13h) not present.
17	No index mark on disk. Indicates no power to disk or disk not present.
18	Timeout while seeking a track.
19	Timeout while seeking track 0.
20	Error in firmware init process
21	HDC cannot find sector header
22	Bad track, sector, or head assignment in header
23	Header checksum error
24	Track invalid (too big)
25	Head invalid for drive
26	Sector invalid for drive
27	Invalid volume selected (command 13)
28	Bad firmware version
29	Duplicate volume label
30	CPM mapping error
31	Error occurred in flush (see note 2 below)
32	Partition invalid (too big) in select
40	Data CRC error
41	CRC error, correction successful
42	Non-correctable CRC error
80	Command error
91	Error saving 17K in format(HARDHELP)
92	Error restoring 17k in format(HARDHELP)

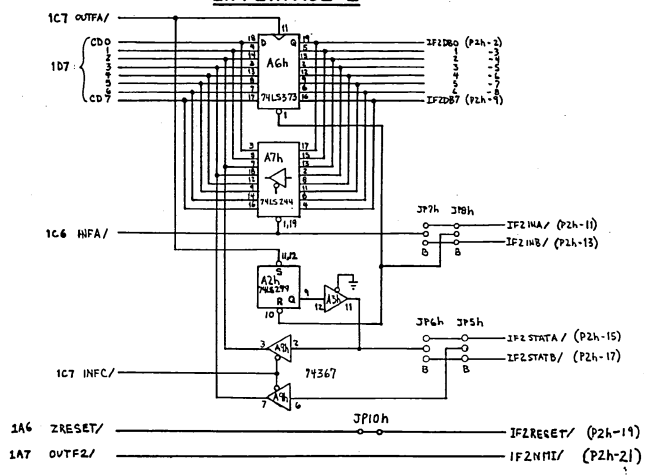
NOTE: A 41h in byte 6 and a 00h in byte 7 of the status block indicates a CRC error occurred and was corrected.

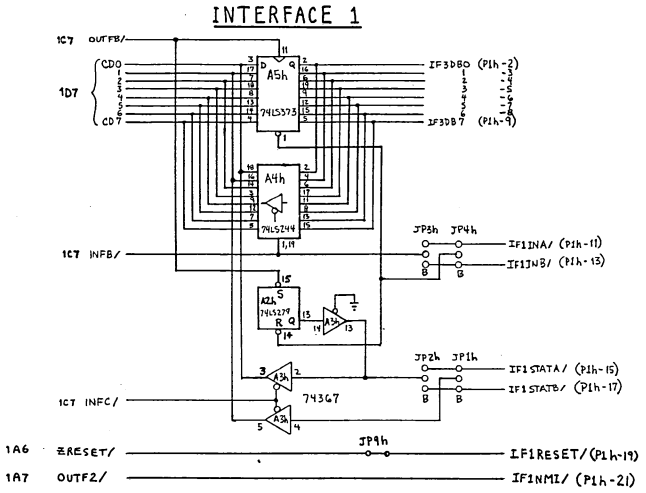
NOTE 2: Error 31 will occur in response to a command 13 (CP/M drive select). The status of the problem is as follows:

Byte 0 = Command generating error
Byte 1 = track
Byte 2 = head
Byte 3 = sector
Byte 4 = block number (low order byte)
Byte 5 = block number (high order byte)
Byte 7 = ACTUAL ERROR
Byte 8 = 31h



DSC-2 INTERFACE





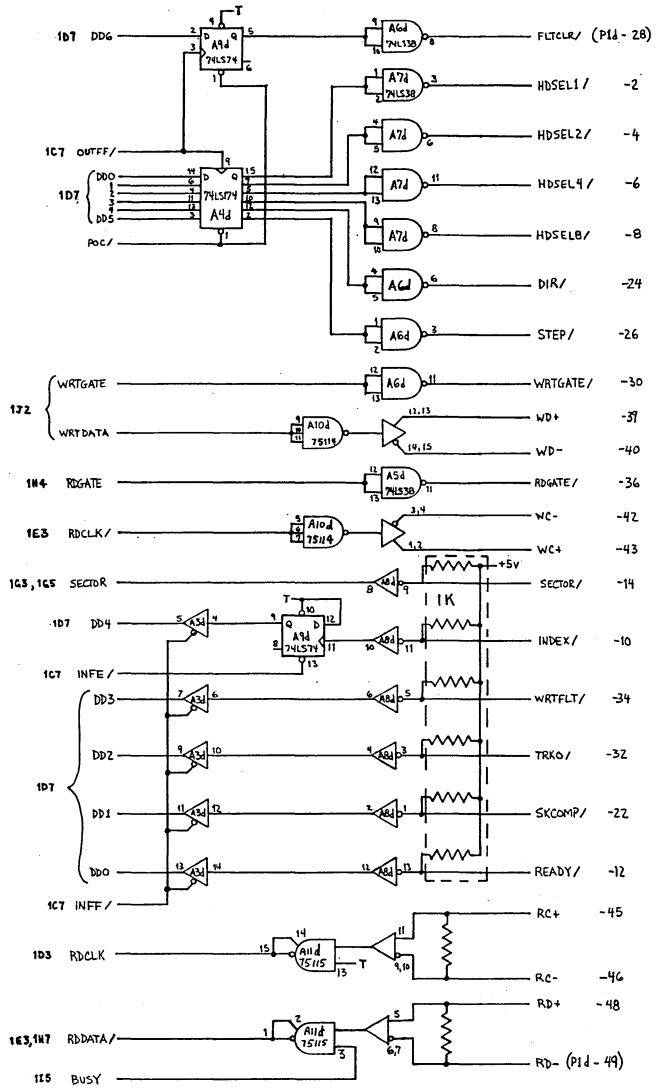
DIGITAL MICROSYSTEMS

SCALE:	APPROVED BY:	DRAWN BY D. COBB
DATE: 4-27-79		REVISED 1-28-81

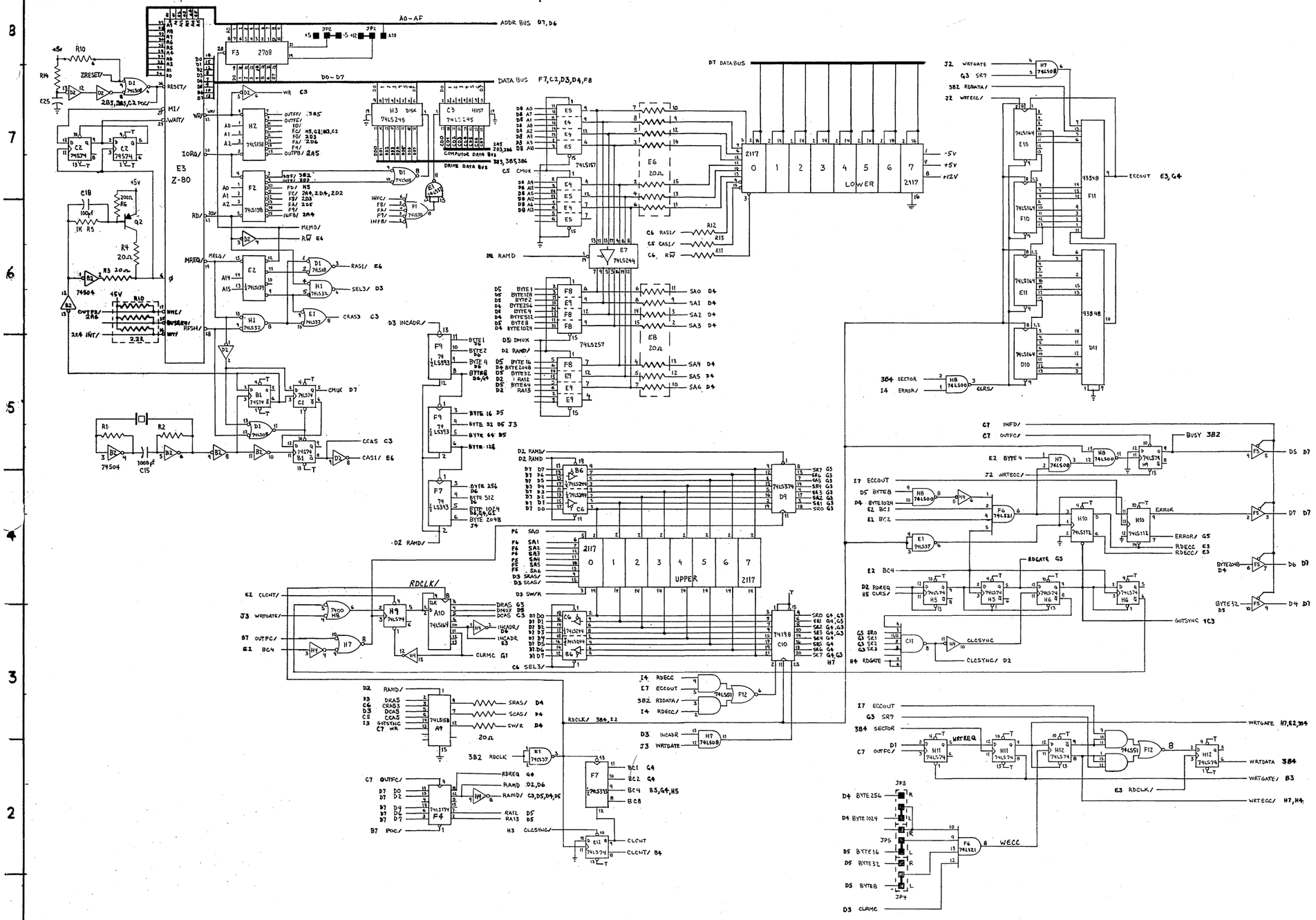
HARD DISK CONTROLLER

HOST INTERFACE

DRAWING NUMBER 2 OF 3



DIGITAL MICROSYSTEMS			
SCALE:	APPROVED BY:		DRAWN BY D. COBB
DATE: 4-27-79			REVISED 1-29-81
HARD DISK CONTROLLER			
DRIVE INTERFACE			DRAWING NUMBER 3 OF 3



DIGITAL MICROSYSTEMS	
SCALE:	APPROVED BY:
DATE: 4·13·79	DRAWN BY D. COBB
	REVISED 1-28-81
HARD DISK CONTROLLER	
CONTROL SECTION	DRAWING NUMBER 1 OF 3