ADEXER

ALTAIR DATAKEEPER EXERCISER

USERS MANUAL

BY MARTIN EBERHARD

14 OCTOBER 2014

Revision History

Adexer Rev.	Manual Rev.	Date	Author	Notes
1.00		6 Oct 2013	M. Eberhard	Created - untested revision
1.01		26 Feb 2014	M. Eberhard	More verbose. Add TW command. Test controller test functions.
1.02		27 Feb 2014	M. Eberhard	Add the following commands: CC, EU, FD, HD, HL, ST, SE E=, SE O=. Split up help screens. Test read, write, format routines
1.03		5 Mar 2014	M. Eberhard	Add DC command. Failed to get Track Offsets working
1.04		7 Mar 2014	M. Eberhard	Fix various bugs, Add format and verify options to DC command. Rename FD to be FS, and create FD command. Add XD and XL commands. More verbose on SR, SW, CR, CW, etc. Track offsets still don't work.
1.05		19 Mar 2014	M. Eberhard	Fix bug in disk copy text. Allow platter to be specified in the XD and XL commands. Change ?R to ?D. Add /P options. Disabled track offsets until I can figure out how to fix them.
	X	19 Mar 2014	M. Eberhard	Original manual release
1.06		24 Mar 2014	M. Eberhard	Set up stack before any subroutine calls :-)
1.07		14 Apr 2014	M. Eberhard	Fix variable-clear loop in main
1.08		24 Apr 2014	M. Eberhard	Fix bugs with fill data & verify, in FD and FS commands. Fix bug when exiting XL command with short disk file.
	X	22 May 2014	M. Eberhard	Document CP/M version. Add Revision History.
1.09		10 Jun 2014	M. Eberhard	Fix Restore bug in XL
1.10	x	19 Aug 2015	M. Eberhard	Support 88-2SIO port B and 88-SIO for XModem file transfers
1.11	X	29 Aug 2014	M. Eberhard	Improve Debugging the Disk Drive section of manual, based on new experience. Default track verify mode off. Improve instructions in TI command. Add MS command.
1.12	X	10 Sep 2014	M. Eberhard	Add comments about loose voicecoil wires, etc. Improved accuracy on time measurements. Track offsets working!
1.13	X	8 Oct 2014	M. Eberhard	Add note about making spindle transducer alignment tool. Change references to match 1978 Pertec manual#104630. Fix bug counting errors. Perform a dummy read after seeking, to re-select the desired platter and head.

Contents

INTRODUCTION

Adexer, the Altair Datakeeper Exerciser, is a comprehensive program for testing and servicing the MITS Altair 88-HDSK Datakeeper Hard disk subsystem. It includes functions that allow fairly thorough testing of the Datakeeper controller, as well as functions that allow control and testing of the disk drive.

Adexer performs all of the functions necessary to tune the disk drive positioner servo and to align the read/write heads, thereby replacing the Pertec disk exerciser.

Adexer can also format, copy, and verify disks, as well as upload and download complete disk images, using the XModem protocol.

Adexer can be assembled to run either stand-alone in an Altair, or under CP/M, with a single assembly-time switch.

DOCUMENTATION REQUIREMENTS

You will need the following additional documents:

1. Pertec Operating and Service Manual. You can use any of the following. However, section references in this document refer to the first one.
   - Pertec Operating and Service Manual Number 104630, Models D3300/D3400 Disk Drives
   - Pertec Operating and Service Manual Number 104615, Models D3300/D3400 Disk Drive with Diablo Compatible Interface
   - Pertec Operating and Service Manual Number 103715, Model D3000 Disk Drive with Diablo Compatible Interface
2. Altair Hard Disk (88-HDSK) Preliminary Documentation

TOOLS

In addition to Adexer, you will need the following tools:

1. A decent 2-channel oscilloscope, at least 100 MHz
2. An accurate digital voltmeter
3. Screwdrivers, Allen Wrenches, wire cutters, needle-nosed pliers, small box wrenches, etc.
4. Soldering iron, de-soldering tool, good solder, etc.
5. A 50-pin loop-back cable. described later in this document
6. Several 12″ to 16″ jumper cables with hook-type ends
7. Mylar measurement shims, 0.05″
8. A 14-pin DIP header, to make an Emergency Unload Bypass Jumper
9. Lint-free cleaning wipes, cotton swabs, masking tape
10. 99% isopropyl alcohol and Windex

Section 1: Adexer

System Requirements

Adexer is designed to run in a “standard” Altair computer:

1. An Altair 8080-based computer (8800, 8800a, 8800b, or 8800Bt), or an equivalent computer with a 2 MHz 8080 CPU.
2. At least 16K-bytes of available zero wait-state RAM, starting at address 0. If you are running under CP/M, then you must have a 16K or larger version of CP/M.
3. In the stand-alone mode, an MC6850-based serial port, at address 10h. (This is the standard Altair Terminal port.) Compatible serial ports include the 88-2SIO, the 88-UIO, and the 8800b Turnkey Module. (If assembled to run under CP/M, then Adexer will use the CP/M Console port instead.)
4. A terminal or Teletype, connected to the above port.
5. A second serial port, required only for uploading and downloading disk images. If your Terminal port is port A an 88-2SIO, then port B on that board is suitable (and can also be used for loading Adexer). If your Terminal port is on an 88-UIO or 8800b Turnkey Module, then you can use an 88-SIO addressed at the standard address for an 88-SIO, 00h-01h.
6. Two MC6820-based parallel ports at addresses A0h and A2h. Compatible parallel ports include the 88-4PIO, or two 88-PIOs. (These ports are the standard Datakeeper interface.)
7. A complete 88-HDSK Datakeeper subsystem, with the controller, the disk drive, and all necessary cabling, connected to the above parallel ports.
8. At least one good, clean 24-sector disk cartridge for the Pertec drive.

LOADING ADEXER

You can load Adexer several ways:

1. Load the stand-alone version of Adexer from paper tape or cassette tape using the following Bootstrap Loader program.
   a. Enter the following program via the front panel. (Select the appropriate column to load from either 2SIO port.)

Octal Address	88-2SIO Port 0 Octal Data	88-2SIO Port 1 Octal Data	Mnemonic	Comment
000	076	076	MVI A,ARESET	ACIA Reset command
001	003	003		
002	323	323	OUT ACTRL	reset ACIA
003	020	**022**		
004	076	076	MVI A,AINIT	Initialize ACIA
005	025	025		
006	323	323	OUT ACTRL	
007	020	**022**		
010	041	041	LXI H,LADDR	End of checksum loader
011	302	302		
012	077	077		
013	061	061	LXI SP,STACK	for the returns
014	032	032		
015	000	000		
016	333	333	IN ASTAT	Get ACIA status
017	020	**022**		
020	017	017	RRC	Rx Data Available is bit 0
021	320	320	RNC	Loop if no data available
022	333	333	IN ADATA	Read ACIA data
023	021	**023**		
024	275	275	CMP L	Is it a leader byte?
025	310	310	RZ	Yes: loop back to 013
026	055	055	DCR L	Next addr, set Z if done
027	167	167	MOV M,A	Write data to memory
030	300	300	RNZ	Loop if not done
031	351	351	PCHL	Go execute the loaded code
032	013	013		Stack for above returns
033	000	000		

   b. Reset, and examine address 0.
   c. Set the sense switches (A15:18) to 00000000b to load from 2SIO Port 0, or 00000110b to load from 2SIO Port 1.
   d. Run.
2. Load the stand-alone version of Adexer from paper tape or cassette tape using the MBL or MBLe PROM. Follow the PROM loader’s directions for loading.
3. Load the CP/M version of Adexer by typing ‘ADEXER’ at the CP/M prompt.

Once Adexer is loaded, you can type ‘?’ to see an initial help screen.

GETTING STARTED

Most of Adexer commands require the following:

1. The Interface Card must be installed in the Controller. (Limited testing is possible with just the Controller’s Processor Card installed.)
2. Some of the Adexer testing functions require you to disconnect all disk drives from the controller. If no disk drives are connected to the Interface Card, then a jumper must be temporarily installed on the Interface Card from the + lead of capacitor C3 to the + lead of capacitor C8. (Be careful - the orientation of these two capacitors on the board is not the same.) This jumper applies +5 volts to the termination resistor network for the disk drive interface. (Normally, the disk drive provides this +5 volt supply.) With the termination resistor network powered, the four “Busy Seeking” signals become False, allowing the controller’s initialization routine to complete.
3. If no disk drives are connected to the Interface Card, then a another jumper must be temporarily installed on the Interface Card from Ready to Ground (Pin 27 to pin 28 on connector P2) (This makes the drive appear ready, so that Adexer commands that use the controller’s Read Status command will complete.)

ADEXER COMMANDS

Adexer commands are entered on a command line on the Altair’s Terminal. Adexer prints a prompt (‘%’) when it is expecting a command. Adexer prints results on the Terminal in a simple text form that is compatible with any kind of terminal - even a Teletype. Most commands that might run for a long time may be interrupted by typing ‘Q’ on the Terminal.

Many commands include variables, abbreviated with angle braces, from the following table.

Field	Abbrev.	Minimum	Maximum	Format
Unit	<U>	1	4	
Buffer	<B>	0	3	
Platter(1)	<P>	0	3	
Side (2)		0	1	
Sector	<S>	0	23	Decimal
Cylinder Address	<C>	0	405	Decimal
IV Byte Address (3)	<I>	0	255	decimal

Notes:

1. Platter 0 is the removable cartridge. For the normal 88-HDSK (with a Pertec 3422 drive), the single fixed platter is platter 1. For larger Pertec drives, the extension platters are 2 and 3.
2. Side 0 is the top side of the platter; side 1 is the bottom.
3. Not all IV Byte addresses exist. Reading a nonexistent IV byte returns FFh. Writing a nonexistent IV byte has no effect.

ADEXER STATE COMMANDS

?S Help with Adexer State Commands

SE Display current settings and state. Note that this command will overwrite Controller buffer 3. <LIST>

SE A={0/1} Set Automatic Seek Mode Off/On. This affects the SK and ST commands. When off, you must press the space bar seek to each listed track. When on, seeking is automatic. The default is off.

SE E={0/1} Set Hex Load Echo Mode (for HL command) Off/On. If Echo is on, then about 5 mS delay will be required between characters during a hex load, at 9600 baud.

SE O={-/0/+/G} Set Track Offset. Subsequent disk read operations (DC, DV, SB, SR, and XD) and seek commands (SK and ST) will be done with the track offset slightly outward if O=-, or inward if O=+. These track offsets are useful for margin testing, and for reading a marginal platter. O=0 selects no offset. O=G sets the read amplifier for reduced gain, which is useful for margin testing. For verify operations, the offset applies only to the source disk. The default is 0.

Track offset setting does not affect disk writes - these are always performed with the offset set to 0.

SE P=<P> Set current platter number. <P> is between 0 and 3. Platter 0 is the removable cartridge. Platter 1 is the fixed disk in a normal Datakeeper. Platters 2 and 3 are extension platters for larger Pertec drives. The default is 0. This command actually selects the specified platter by performing a sector read, which overwrites controller buffer 3. The sector read is not checked for any errors.

SE S={0/1} Set the current platter side, where 0 means the top side. The default is 0. This command actually selects the specified side by performing a sector read, which overwrites controller buffer 3. The sector read is not checked for any errors.

SE T={0/1} Set track verify mode Off/On. This affects the SK and ST commands. When On, the disk is read and the track number found on the disk is compared to the expected track number, generating an error report if not the same. When Off, disk header errors are ignored. The default is off.

SE U=<U> Set the current unit. <U> is between 1 and 4. The default is 1.

SE V={0/1} Set Verbose Seek Mode Off/On. Displays the track number after each seek when on. The default is off.

SE X=<H> Set XModem Transfer Port. 0 selects port B of an 88-2SIO (at addresses 22h and 23h), and 1 selects an 88-SIO (at addresses 00h and 01h). The default is set by the Altair's front panel sense switches <A10:A8>. If these switches are 010, then the default is the 88-SIO. Otherwise, the default is the 88-2SIO port B.

Memory Buffer Commands

?B Help with Memory Buffer Commands

BD Buffer Display. Displays all 256 bytes of Adexer's memory buffer.

BE {0-FF} Buffer Edit starting at the specified address. Adexer displays the contents of the specified Adexer buffer address in hexadecimal. You can change the contents of that address by typing a new hex value and then the space bar. If you want to leave the value unchanged, then just press the space bar. Adexer will display the contents of the next buffer location, allowing you to edit that location. This continues until you type ‘Q’.

BF [<H>] Buffer Fill with <H>. Fills the buffer with <H>, a 2-digit hexadecimal value. <H> defaults to 0.

CC [<B>] Controller Buffer Compare. Reads controller buffer <B>, and compares it to Adexer’s memory buffer. Reports the number of mismatches. <B> defaults to 0.

CR [<B>] Controller Read & display. Reads controller buffer <B> into Adexer’s memory buffer, and then displays the contents of Adexer’s memory buffer. <B> defaults to 0.

CW [<B>] Controller Write. Writes Adexer’s memory buffer to controller buffer <B>. <B> defaults to 0.

HD [<P>] Hex Dump Buffer. Displays Adexer’s 256-byte RAM buffer on the Terminal in Intel Hex format. <P> is an optional 2-digit hexadecimal Page Address (address high byte), which defaults to 0.

HL [<P>] Load Intel Hex file. Data is loaded from the Terminal into Adexer’s buffer only if the high address byte within the hex record matches the hexadecimal Page Address, <P>, which defaults to 00. Records from other pages are ignored. Type ‘Q’ at any time to abort.

Controller Commands

?C Help with controller commands

IC Initialize Controller. Note that this command will not reset the controller if it is hung waiting for input from the disk port, or is hung waiting to transfer data to the Altair. You must use the front panel Reset key switch when the controller is hung this way.

IR <I> Read the value of the specified Controller IV Byte. The IV Byte address is in decimal, because the Datakeeper documentation refers to IV Bytes using decimal addresses. (Although Adexer will allow an IV Byte address as high as 255, the controller has no IV Bytes above address 37.) The IV data value is printed on the Terminal in hex.

IW <I> <H> Write 2-digit hex value <H> to Controller IV Byte <I>. IV Byte addresses are decimal because the Datakeeper documentation uses decimal IV Byte addresses. (Adexer will allow an IV Byte address as high as 255, but the controller has no IV Bytes above address 37.) The IV Byte data value is hexadecimal since the Datakeeper documentation refers to IV Byte data in binary.

TB Test Controller Buffer Memory. This performs a reasonably extensive memory test on the controller's buffer memory, reporting any problems. This test can be run with just the Process or Card installed.

TF Test Controller FIFO. This performs a reasonable extensive test of the disk data FIFO, using its parallel-in and parallel-out ports, reporting any problems. This test requires the Interface Card to be installed, with the Ready line true.

TI Test IV Bus, and Read Channel. This performs a series of tests to validate the 3 boards in the controller:

Processor Data Bus Test. This writes a test pattern sequence to IV Byte A8 (IV address 02), and verifies the data by reading the data from the Error Flags port of the computer interface (Altair address A1h).

This portion of the test will run with only the Processor Card installed. All further tests require the Interface Card to be installed with no disk drives connected, and the Ready line held true.

Processor Address Bus Test. This reads from a sequence of IV bytes on the processor Card, and verifies the data for what should normally be at each address.

If the above two tests pass, then a 'Processor Card OK message is printed.'

Interface Card Presence Test. This test verifies the presence of the Interface Card by reading from IV Byte H (IV address 17) four times - each time with a different unit number specified in the Read command. Each time, the returned value contains the decoded unit number, if the Interface Card is functioning.

Disk Data Card Presence Test. This test verifies the presence of the Disk Data Card by writing to one output IV Byte (IV E, at address 37), and reading back the result. (This address was chosen because the equivalent address on the Interface Card is an Input IV Byte, and so will not read back what was written. This way, a stuck IV address line will get caught.)

IV Address/Data Bus Test. This test walks a test pattern through all of the IV output Bytes on the Interface Card and the Disk Data card that can be read back. The following IV Bytes are tested here: On the Interface Card: IV I and IV J. On the Disk Data Card: IV B, IV C, and IV E. Note that if the Interface Card Presence Test failed, then this test will not be run. If the Disk Data Card Presence Test failed, then the Disk Data Card IV Bytes will not be included in the test.

The remaining tests require a loop-back cable from P1 to P2 on the Interface Card. This loop-back connector should be a straight-through 50-pin connector, but with **no connection for pins 1 and 2**. (Pins 1 and 2 on connector P2 are the +5 volt pins for the Disk drive interface terminating resistors. But pin 1 is a signal output, and pin 2 is ground on connector P1.)

Note that this loop-back cable should not be installed when you reset the controller (either with the front panel key switch or with the RC command). If the loop-back cable is not removed before the controller is reset, then the controller’s initialization routine will hang waiting for the four ‘Busy Seeking’ signals to go False (high).

Disk Drive Loopback Test. This test writes a sequence of values to the output ports of the Interface Card, and reads them back from the input ports of the Interface Card, verifying the results. The loopback cable does not connect the signals in a simple manner - Adexer must unscramble the data in order to verify the results.

Several signals cannot get tested this way - either because they are outputs that don’t get connected to inputs by the loopback cable, or because the controller’s firmware rewrites certain output bits when reading IV bytes.

This test verifies the following output signals and input signals (Note that the IV bits in this table match the MITS documentation, and are reversed from the order in which they appear with the IR command.)

Output IV Byte:bit	Output Signal Name	Conn. Pin	Input IV Byte:bit	Input Signal Name
IV H:6	Select 2	3	IV K:7	Double Track
IV H:5	Select 3	5	IV K:5	Dual Platter
IV H:4	Select 4	7	IV K:3	Extension Stat
IV H:0	Start/Stop	15	IV L:5	Busy Seeking 3
IV I:7	Cyl Address 0	17	IV L:4	Busy Seeking 4
IV I:5	Cyl Strobe	19	IV L:3	Illegal Address
IV I:4	Cyl Restore	21	IV L:2	File Prot
IV I:3	Enable Write	23	IV L:1	Index Pulse
IV I:1	Offset Plus	29	IV M:7	Sector Count 0
IV I:0	Emerg. Unload	31	IV M:6	Sector Count 1
IV J:7	Cyl Address 0	33	IV M:5	Sector Count 2
IV J:6	Cyl Address 1	35	IV M:4	Sector Count 3
IV J:5	Cyl Address 2	37	IV M:3	Sector Count 4
IV J:4	Cyl Address 3	39	IV M:2	Sector Count 5
IV J:3	Cyl Address 4	41	IV M:1	Sector Count 6
IV J:2	Cyl Address 5	43	IV M:0	Sector Count 7

Read Channel Loopback Test. This uses 2 output signals of the loopback cable to shift a data pattern sequence into the Disk Data Card’s read channel hardware, and then verify it by reading the data from the FIFO. The data are shifted into the read channel hardware using FM encoding, the same as a real disk drive, though a lot more slowly. The data pattern includes a correct CRC value, so that the CRC checking (and generating) hardware is also checked.

In addition to testing the read hardware, this test verifies the following output signals and input signals:

Output IV Byte:bit	Output Signal Name	Conn. Pin		Input Signal Name
IV J:6	Cyl Addr 1	45		Read Clock
IV J:7	Cyl Addr 0	47		Read Data

This following signals are NOT fully tested by these tests:

Output IV Byte:bit	Output Signal Name	Conn. Pin	Input IV Byte:bit	Input Signal Name
IV H:7	Select 1	1		
IV H:3	Head Select	9	IV K:0	Malfunction
IV H:2	Platter Sel	11	IV L:7	Busy Seeking 1
IV H:1	Extension Sel	13	IV L:6	Busy Seeking 2
IV I:2	Offset Minus	27	IV L:0	Ready
		49		Write Data/Clock

Disk Positioner Commands

?P Help with positioner commands

RE Restore current unit to cylinder 0.

RR Repetitive restore current unit to cylinder 0. Type ‘Q’ to quit. (This is needed during alignment.)

SK [<C>] Seek Cylinder <C> on Current Unit. (The cylinder number must be specified in decimal.)

A dummy read is performed after seeking, to re-select the requested platter and side. This will trash controller buffer 3.

If track verify is enabled (with the ‘SE T=’ command), then the controller will read a sector header on the destination track, and will report an error if the sector header has a CRC error, or if the track number does not match the expected track number.

If a read track offset has been set (with the ‘SE O=’ command), then the requested offset (or reduced gain) will apply to the seek, and will affect track verify, if set.

SK <C1> <C2>...<Cn> Seek sequentially through the listed cylinders on the current unit. (Cylinder numbers must be specified in decimal.)If Automatic Seek Mode is off, then Adexer will seek each cylinder, and then wait for you to press the space bar before continuing to the next cylinder. If Automatic Seek Mode is on, then Adexer will seek continuously through the listed cylinders. Type ‘Q’ to quit.

Typing ‘M’ while seeking through a list of cylinders in automatic mode will switch to manual mode. Typing ‘A’ while seeking through a list of cylinders in manual mode will switch to automatic mode.

The controller always reads a sector header from platter 0, side 0 on each destination track after a seek. If track verify is enabled (with the ‘SE T=’ command), then Adexer will report an error if this sector header has a CRC error, or if the track number does not match the expected track number.

A dummy read is performed after seeking, to re-select the requested platter and side. This will trash controller buffer 3. If automatic mode is selected, then this dummy read occurs when the command terminates, or when switching to manual mode. If manual mode is selected, then this dummy read occurs after each seek.

If a read track offset has been set (with the ‘SE O=’ command), then the requested offset (or reduced gain) will apply to each seek, and will affect track verify success, if set.

If Verbose Seek Mode is enabled (SE V=1), then Adexer will print the approximate time for each seek. Adexer subtracts the overhead of the Datakeeper controller’s firmware, as well as the average time for the controller to read a sector header after a seek. Because of the uncertainty for how long the controller will wait for a sector header, the error on this measurement is a little more than half of a sector time, or about 550 microseconds.

ST {I/O/A} Step in/out one cylinder at a time on the current unit. ‘ST I’ and ‘ST O’ step manually with each press of the space bar. ‘ST A’ steps automatically, and begins stepping inward.

When in manual mode, press space to step again in the same direction. Type ‘I’ to step inward, ‘O’ to step outward, ‘A’ to switch to automatic mode, and ‘Q’ to quit. Stepping inward past cylinder 405, or outward past cylinder 0 has no affect.

When in automatic mode, press ‘M’ to switch to manual mode, and ‘Q’ to quit. The stepping direction will change when the head reaches cylinder 0 or 405.

The controller always reads a sector header from platter 0, side 0 on each destination track after a seek. If track verify is enabled (with the ‘SE T=’ command), then Adexer will report an error if this sector header has a CRC error, or if the track number does not match the expected track number.

A dummy read is performed after each seek, to re-select the requested platter and side. This will trash controller buffer 3.

If a read track offset has been set (with the ‘SE O=’ command), then the requested offset (or reduced gain) will apply to each seek, and will affect track verify success, if set.

If Verbose Seek Mode is enabled (SE V=1), then Adexer will print the approximate time for each seek. Adexer subtracts the overhead of the Datakeeper controller’s firmware, as well as the average time for the controller to read a sector header after a seek. Because of the uncertainty for how long the controller will wait for a sector header, the error on this measurement is a little more than half of a sector time, or about 550 microseconds.

DISK READ/WRITE COMMANDS

?R Help with read/write commands

DC <SP> <DP> [/F] [/V] [/P] Copy source disk platter <SP> to destination disk platter <DP>, on the current unit. <SP> and <DP> may only be 0 for the removable cartridge and 1 for the fixed disk. The entire contents of both sides of the source platter are copied to the destination platter.

If a read track offset has been set (with the ‘SE O=’ command), then the requested offset (or reduced gain) will apply to all reads from the source disk.

The /F option causes the destination disk to be formatted before the copy. The /V option causes the disk to be verified once the copy is complete. The /P option will cause the heads to park and the drive to shut down once the copy (and verify) are complete. Any read or write error will abort the copy. If the disk is new, then you should use the /F option.

During each stage of the disk copy except the format stage, Adexer prints a pacifier dot for each completed cylinder.

Use the Write Protect switches on the disk drive is to prevent mistakes.

DV <SP> <DP> Verify Disk. Every byte of the source platter <SP> on the current unit is compared to the destination platter (DP> on the same unit. <SP> and <DP> may only be 0 for the removable cartridge and 1 for the fixed disk.

If a read track offset has been set (with the ‘SE O=’ command), then the requested offset (or reduced gain) will apply to all reads from the source disk only.

Errors are reported and counted. Adexer prints a pacifier dot for each verified cylinder.

FD [<H>] Format Disk, on the current unit and platter. Optional parameter <H> will be written to every sector on both sides of the disk, and then verified. If <H> is omitted, then no sector data is written, and no verify occurs.

This operation takes about 2 minutes without writing sector data, and about 7 minutes if sector data is written and verified. Use the Write Protect switches on the disk drive to prevent mistakes.

FS [<H>] Format Disk Side, on the current unit, platter, and side. Optional parameter <H> will be written to every sector on the disk surface, and then verified. If <H> is omitted, then no sector data is written, and no verify occurs.

This operation takes about 60 seconds without writing sector data, and about 3-1/2 minutes if sector data is written and verified. Use the Write Protect switches on the disk drive to prevent mistakes.

SB <S> [<B>] Read bad sector. Reads sector <S> from the current cylinder, head, and unit, into controller buffer <B>. (The sector number is specified in decimal and <B> defaults to 0.) This read skips the sector header information, allowing you to read sector with errors in the header.

If a read track offset has been set (with the ‘SE O=’ command), then the requested offset (or reduced gain) will apply to the sector read.

SR <S> [<B>] Read Sector. Reads sector <S> from the current cylinder, head, and unit, into controller buffer <B>. (The sector number <S> is specified in decimal and <B> defaults to 0.)

If a read track offset has been set (with the ‘SE O=’ command), then the requested offset (or reduced gain) will apply to the sector read.

SW <S> [<B>] Write Sector. Writes data from controller buffer <B> to disk sector <S>, at the current cylinder, head, and unit. (The sector number <S> is specified in decimal and <B> defaults to 0.)

Use the Write Protect switches on the disk drive to prevent mistakes.

TW [<B>] Write Track. Writes to every sector from controller buffer <B>, at the current cylinder, head, and unit. (<B> defaults to 0.) Note that the track must have a valid format before it can be written.

Use the Write Protect switches on the disk drive to prevent mistakes.

XD [<P>] [/P] Dump Disk in XModem format. The entire disk platter <P> is transmitted in XModem format out the transfer port (which can be specified with the ‘SE X=’ command). If no platter is specified, then the current platter will be transmitted.

/P will cause the heads to park and the drive to shut down once the dump is complete.

If a read track offset has been set (with the 'SE O=' command), then the requested offset (or reduced gain) will apply to all reads from the disk.

Data transmission starts with cylinder 0, side 0, sector 0, then sector 1, on through sector 23. Then the sectors of side 1 are sequentially transmitted. Then both sides of cylinder 1 are transmitted, on through cylinder 405.

The XModem receiver can select either normal XModem (with checksums) or XModem-CRC in the standard way.

Dumping a disk takes about 2 hours at 9600 baud. Adexer prints a pacifier dot for each transmitted cylinder.

XL [<P>] [/C] [/P] Load Disk in XModem format. The entire disk platter <P> is overwritten with a file that is received from the transfer port (which can be specified with the 'SE X=' command). If no platter is specified, then the current platter will be written.

Data is written to disk starting with cylinder 0, side 0, sector 0, then sector 1, on through sector 23. Then the sectors of side 1 are sequentially written. Then both sides of cylinder 1 are written, on through cylinder 405.

/C specifies XModem with checksum error detection - otherwise XModem-CRC will be received.

/P will cause the heads to park and the drive to shut down once the load is complete.

Loading a disk image takes about 2 hours at 9600 baud. Adexer prints a pacifier dot for each loaded cylinder.

Use the Write Protect switches on the disk drive to prevent mistakes.

Other Commands

? General help and help with these other commands

EU Emergency-unload the current Unit. This allows you to test the disk drive's emergency unload system, but is a bit violent, and should be done infrequently and with care.

MS Measure and report the average spindle revolution time on the current unit. Pertec specifies 25000 µSec per revolution +/- 1%, or 24750 µSec to 25250 µSec. The accuracy of this measurement is the accuracy of the 2 MHz crystal oscillator on your Altair's CPU board. When it was new, this oscillator was accurate to about 100 ppm. After 40 years, 200 ppm is a conservative estimate, which works out to be 5 µSec maximum error for this measurement.

This command works by reading sector 0, head 0 on the current track of the current unit 256 times in a row into buffer 3 (which will take about 11 seconds), while counting CPU cycles. The result is then divided by 256 to get the average time per revolution. All errors encountered during this read are ignored (and therefore the heads need not even be installed). Buffer 3 is overwritten.

SS Start/Stop current Unit. This pulses the Start/Stop signal to the current Unit, the same as pushing the Start/Stop button on the front of the Unit.

QU Quit. This jumps to the TURMON or UBMON PROM at address 0FD00h. If one of these PROMs is not installed, then don’t type this command.

SECTION 2: SUGGESTED TESTING PROCEDURES

This section gives some suggestions for how to use Adexer to diagnose and adjust a Datakeeper subsystem.

DEBUGGING THE CONTROLLER

POWER SUPPLY CHECKOUT

With all cards removed from the Controller, make sure that the 5V supply is working correctly. Once the basics of the controller are working - the power supply, the clock, etc. - you can use Adexer to bring the rest of the controller up.

PROCESSOR CARD RAM TEST

Install just the Processor Card in the controller, and connect it to the 4PIO in the Altair. Run the ‘TB’ command to test the 1K buffer RAM. This will also validate part of the IV Bus and most of the interface to the Altair - all except the CSTAT port.

PROCESSOR IV BUS TEST

Use the ‘TI’ command to test the CSTAT port, as well as the Processor Card’s IV Data/Address bus. Without the Interface Card installed, the ‘TI’ test will terminate after the Processor Card test.

INTERFACE CARD INITIAL VALIDATION

Power the controller down and install the Interface Card, with a temporary jumper from the + lead of capacitor C3 to the + lead of capacitor C8 and another temporary jumper from the Ready signal to Ground (Pin 27 to pin 28 on connector P2).Run the ‘TI’ test suite again. This time, the Interface Card should be detected, and it should run the IV Address/Data Bus test on the Interface Card. If this test passes, then (without turning off the controller’s power) remove the Ready jumper and install the loopback cable from J1 to J2 on the Interface Card. **MAKE SURE that pin 1 and 2 are NOT connected on the loopback cable**. Then proceed with the Disk Drive Loopback Test. Continue to the next step when this test passes.

DISK DATA CARD VALIDATION

Power the controller down again, and install the Disk Data Card. You will need to remove the loopback cable in the Interface Card, and reinstall the Ready jumper to get started. Run the ‘TI’ test suite again. This time the Disk Data Card should be detected. The IV Address/Data Bus test will test both the Interface Card and the Disk Data Card. If this test passes, remove the Ready jumper, and install the loopback cable (with the power still on). Then proceed with the loopback test. The final phase of the ‘TI’ test suite runs a comprehensive test of the read channel, which spans both the Interface Card and the Disk Data Card. If this test passes, the controller is in pretty good shape.

FIFO Validation

This test is particularly useful to help diagnose failures with the Read Channel test, part of the ‘TI’ test suite. It performs a comprehensive test of the data FIFO.

Debugging the Disk Drive

Disk drive debugging and alignment is per the Pertec Operating and Service Manual (Either Manual Number 103715 or Manual Number 104615), section IV. Adexer can perform all of the functions that would be performed by the Pertec Exerciser mentioned in this section.

The following is a brief outline of the restoration and alignment procedure, assuming that the drive needs significant restoration, not merely adjustment. Use this outline in conjunction with the Pertec Operating and Service Manual.

Before you ever power-on the disk drive

1. Throughout this process, be careful not to drop screws, washers, etc. into the disk drive. The magnet is very strong, and will try to yank your tools, or suck the screws out of your grip. Small parts can also fall down into the power supply, requiring significant disassembly to retrieve.
2. Cleanliness matters. Start off by cleaning all accessible areas of the drive using a vacuum cleaner with a crevice tool, and using a rag and Windex. As you disassemble portions of the drive, clean the areas that become accessible. In particular, the areas inside the absolute air filter and inside the disk chambers need to be perfectly clean.
3. Make sure that the absolute air filter is in good condition. You can clean the absolute filter gently with compressed air, blown backwards through the filter. It is a good idea to replace the foam pre-filter as a matter of course. This pre-filter can be manufactured from readily-available filter foam.
4. Tip the drive on its side and remove the spindle-belt cover plate. (Remove 3 screws, and slide the plate forward.) Inspect the belt and all the bearings, and repair/replace parts as necessary, following Section 6.19. If there is any doubt, remove the belt and spin each bearing separately: the

motor, the fan, the spindle, and the idler. Also check that the pulley surfaces (where the belt rides) are clean.

5. Check the static discharge contact - see section 6.20.
6. With the drive right-side-up, check that the spindle spins freely, by turning it counterclockwise. Note that spinning the spindle also turns the blower fan.
7. Clean the spindle and surrounding area, following Section 6.4.3. In particular, be sure to remove any loose chrome plating that may have come off the spindle.
8. Inspect the foam gasket that surrounds the area (inside the disk cartridge chamber) where the heads travel. If the foam does not feel soft and compliant, then it must be removed and replaced. Commonly, this gasket turns to powder when you touch it. Be careful not to let this powder fall onto the fixed disk. Use a vacuum cleaner with a crevice tool to suck up all of the crumbly foam, and then carefully peal the adhesive backing off of the disk chamber. Fabricate and install a new gasket. (You can find very similar adhesive foam material in the insulation section of your hardware store.)
9. If the foam gasket in the previous step was not in great shape, then you will probably need to replace the gasket below the fixed disk's chamber too, on the left-rear side. Be extremely careful about cleanliness (particularly crumbling gasket material), dropped tools, etc. while working within the fixed disk chamber! Do not touch the fixed disk. While you are in this chamber, clean the area by the blower fan, accessible through the exposed hole on the left side. Also use this opportunity to inspect the fixed disk surfaces - use a bright light and a mirror. (You will otherwise inspect and perhaps replace the fixed disk later.)
10. Clean all accessible surfaces of the drive, including the base casting, the wiring harnesses, the dist covers, etc.
11. Inspect and repair the brushes, per Section 6.28. (Note that some of these drives do not have brushes and that in a late Product Improvement Bulletin, Pertec actually recommended removing the brush and brush motor subassembly, and closing off the holes where the brushes entered the disk chamber .)
12. Check and adjust the Cartridge Interlock mechanism, following the procedure in Section 6.16.
13. Check and adjust the clearances between the magnetic index transducers and the rotating surfaces, per section 6.18. The lower transducer can be adjusted with a screwdriver and a

0.007” feeler gauge. The upper transducer requires a special tool, which is probably not available. You can fabricate this tool by disassembling a disk cartridge, removing the metal hub, and cutting a piece off of the flange to give you access to the transducer when the hub is set into the disk drive’s spindle. I built mine with a CDC cartridge, and for this cartridge, the correct transducer gap is 0.015”. (You might buy a good disk cartridge for e.g. a DEC computer that has the wrong number of sector marks for the Datakeeper. Use the platter from this cartridge to replace your fixed platter, and cut up the hub to make an alignment tool. These cartridges can be found for as little as $50 if you are patient.)

14. Remove the Read/Write board, and then remove all 4 read/write heads from the disk carriage assembly. This will allow you to move the positioner carriage without loading the heads onto the media. (Note that you must realign the heads after you reinstall them.) See Section 6.22 for head removal instructions.
    a. Before removing the top 2 heads, note (and perhaps photograph) the adjustment of the two set screws. You will want to put them back to approximately the same setting later on, to make alignment easier.
    b. Mark each head’s connector, so that you can put them back in their original positions.
    c. Note there are two versions of the red aluminum bracket that holds the heads. On older brackets, the screws pass through two standoffs, each about ¾” long. Don’t let these standoffs fall into the disk drive when you remove the brackets. Newer brackets have an integral standoff.
    d. Note that on some drives, the lower two heads are held in with a single screw and washer, rather than with the assembly shown in Section 6.22.
15. Clean and inspect the read/write heads, following Section 6.4.1, as well as Section 6.21. Set the clean heads aside in a safe place.
16. With the heads removed, slide the carriage assembly back and forth along its track. (This is almost always necessary for a 35+ year old disk drive.) If the drive has been sitting for a long time, then it is likely that the voicecoil is stuck in the retracted position. There is a bumper inside the magnet that sticks to the inside of the voicecoil. (This is actually a good thing - typically, people fail to secure the voicecoil

when shipping, and this sticky bumper can save your heads and media!) Gently pull on the voicecoil until it comes free.

If the voicecoil assembly does not feel perfectly smooth when you move it, then you will need to remove it to restore its bearings, with the following procedure.

a. Throughout this process be extremely careful not to damage the glass positioner scale that is attached to the positioner carriage assembly, or the sensor assembly that is attached to the chassis.

b. Remove the positioner lamp assembly by removing the top screw that holds just the lamp assembly to the rest of the sensor assembly. You can leave the wires connected, but do not power the drive on with the lamp assembly lying on the chassis, as it will short out and damage the power supply.

c. Remove the wiring harness clamps from the rear of the magnet. (On some drives, these are screwed to the magnet; others have adhesive clamps.)

d. Remove the stick-on wire clamps that hold the ground wire for the Read/Write board (if it exists), and then remove the 3 sets of screws, nuts, and washers that connect the flex cable to the voicecoil. Note their arrangement, so you can put them back the same way. Be extra careful - the magnet will try to steal the small parts from the grip of your tools.

e. Remove the one cap screw that holds the velocity sensor rod to the carriage. (The velocity sensor is a metal rod, with a magnet on its tip, about 6” long that is attached to the lower right side of the carriage, and slides in a coil that is mounted beneath the lower platter.) Be very careful not to flex this rod, as the magnet on its tip is brittle and will break.

f. Slide the voicecoil carriage all the way toward the spindle. Remove the 4 machine screws that hold the magnet to the chassis. Carefully remove the magnet rearward from the drive. This magnet is very strong! set it down someplace far away from e.g. floppy disks.

g. Remove the machine screw that holds down the spring for the retainer bar on left side of the carriage assembly. Remove the two screws that hold down the right side retainer bar, remove the retainer bar, and gently slide the carriage assembly backward, out of the drive.

h. Always be mindful of the glass slide on the carriage assembly as you work on it! This is fragile, and is also probably still aligned, more or less.

i. On some drives, each bearing can be removed from the carriage assembly by removing a C-clip. On other drives, the bearings are stamped onto the carriage and cannot be removed. In the latter case, the bearings must be cleaned and repacked in place on the carriage. Remove each bearing, clean and repack it, and reinstall it.

   i. Remove both C-clips that retain the bearing dust covers, and lift out both dust covers. (Just remove the outside covers if you cannot remove the bearings from the carriage.) The dust cover C-clips can be removed with a tool that comes to a sharp point. Keep a finger over the bearing as you pop the C-clip's tip put, or the C-clip will fly across the room.

   ii. Flush the bearing with solvent until it is completely clean and spins freely. If you are cleaning the bearings while they are attached to the carriage, then be careful not to let the solvent runoff get onto the positioner scale. You still want to use plenty of solvent to get the bearing clean, but protect the scale.

   iii. Completely re-pack the bearing with suitable grease, spinning the bearing as you do so, to work the grease into the bearing race. Use your thumb to press the grease into the bearing, until it oozes out the other side.

   iv. Reinstall the dust covers and C-clips.

   v. Clean the bearing outside surfaces, removing all grease. Spin the bearing to force out any excess grease, and continue to clean the bearing until no more grease comes out when it is spun. If you repacked the bearings while they are attached to the carriage, then use a thin rag to thoroughly clean the excess grease from the spaces between the bearings and the carriage.

j. Take this opportunity to clean the disk drive chassis in all the areas that are now accessible with the magnet and carriage removed. In particular, thoroughly clean the metal rod on which the carriage assembly rides, as well as the underside of the two metal retainer bars that hold the carriage in place. Any dirt or defects on these mars will result in rough movement of the carriage.

k. Carefully clean (without damaging) the positioner sensor surface, which will be inaccessible once the carriage assembly is replaced. Use a bright light to inspect your work.

l. Thoroughly clean and inspect the entire positioner carriage assembly, paying close attention to clean (without damaging) the positioner scale.

m. Carefully inspect the voicecoil itself, looking for any loose windings. The slightest looseness of any winding will result in oscillations in the positioner servo. If any are loose, carefully repair the area with epoxy glue, working it into the windings, and preventing any surface buildup that would interfere with its travel through the magnet.

n. Reinstall the positioner carriage assembly, making sure the retainer bars are installed in the correct orientation.

o. Slide the positioner carriage assembly back and forth, making sure it does not have any interference or roughness.

p. Reconnect the velocity transducer (the long metal rod that slips into the coil beneath the fixed platter). Carefully adjust the position of this rod. When adjusted perfectly, the magnet drags slightly against the coil when it moves. (If it drags too much, it will interfere with the voicecoil servo stability. If it does not touch at all, then it can develop a resonance, causing an audible ringing, and throwing the servo stability off.)

q. Thoroughly clean the magnet, paying particular attention to the central cylinder and the gap between this cylinder and the front bar. The magnet probably has many little flecks of metal that are magnetically attracted to the magnet. These must be removed prior to reassembly.

r. Check the condition of the rubber end-stop on the tip of the magnet, which will go inside the voicecoil. If this rubber end-stop is not in good shape, replace it.

s. Slide the voicecoil all the way toward the spindle, and carefully reinstall the magnet. Be especially careful as you slid it onto the voicecoil. Make sure the magnet does not interfere with the voicecoil before tightening its mounting screws thoroughly.

t. Reinstall the voicecoil flex cable screws, nuts, and washers, reconnecting the voicecoil wires as you do so.

The magnet will make this task a lot more difficult than expected, for example, pulling split washers off if the screws after you have positioned them.

u. This step is critical: Use a 0.005” Mylar shim, and follow Section 6.7.1(11) through 6.7.1(21) to set the the reticule-to-scale gap correctly. Be especially careful to adjust the gap to be the same at both extremes of positioner movement.

v. Reinstall the positioner sensor lamp assembly. Do not over-tighten.

17. Carefully inspect the surfaces of the fixed disk, as well as any disk cartridge that you plan to use. Clean this disk following Section 6.4.2. Any dirt that cannot be removed, scratches from previous head crashes, or blemishes caused by corrosion of the underlying aluminum substrate will require you to replace the disk - the slightest amount of dirt or other surface blemish will cause a disk head crash.

    a. You can get a replacement fixed disk by removing the disk from a disk cartridge. Pretty much any 14” disk cartridge will do - for example, the relatively common DEC RL01 and RL02 cartridges are suitable. Use gloves to keep oil from your fingers off of the disk.

    b. It is also possible to replace the disk inside an Altair-compatible cartridge with a disk from some other type of 14” cartridge (e.g. RL01 or RL02 cartridge). If you do so, take care that the new disk is exactly concentric when clamped in place in your cartridge. This was done with a fixture at the factory, but can be done with careful measurement and patience.

18. Use a label maker and scissors to make labels for each of the potentiometers on the servo board, and stick them to the potentiometers. This will be a big help for many of the following steps.

POWER SUPPLY CHECKOUT

1. Remove J212 from the Servo Board, to isolate the unregulated power supply.

2. Power up the disk drive, and check the unregulated voltages on P212, the cable coming from the power supply. Correct any problems in the unregulated supply.

    a. Pin 4 to pin 1 should be about 21V AC, peak-to-peak.

    b. Pin 4 to pin 9 should be about +10V DC

   c. Pin 4 to pin 11 should be about +20V DC
   d. Pin 4 to pin 6 should be about -20V DC
3. Power down, and then reconnect J212.
4. Disconnect J106 from the Logic Board, and install the following jumpers in J106, the cable to the Servo Board:
   a. Pin 9 to pin 12
   b. Pin 5 to pin 6
   c. Pin 3 to pin 4
5. Follow Section 6.6.2 to adjust the +/-5V and +/-10V regulators. Be careful with probe tips in the power supply area of the Servo Board, as the cans of most of the large transistors are 'hot', and shorting them will damage the power supply.
6. When the power supply is working correctly, power down, remove the jumpers from J106, and reinstall J106 on the Logic Board. Power back up, and repeat the power supply adjustments of Section 6.6.2. Take the time to set these adjustments exactly.

Servo Adjustments

1. Fabricate an Emergency Unload Bypass Jumper using a 14-pin DIP header. Solder a wire from pin 1 to pin 8 to complete the fabrication.
2. Unplug J205 from the Servo board, and install an Emergency Unload Bypass Jumper in the socket on the Logic Board.
3. Follow Section 6.6.3 to adjust the spindle servo. (This will work with the heads removed, even though it "reads" sector zero to measure the speed.) You can use a 'junk' removable cartridge for this step, since the heads are removed. Note that the Figure 6 is drawn incorrectly: the signal is a negative-going pulse, not a positive-going pulse.

   Use the MS command to verify the spindle speed. The spindle speed is derived from the drive's crystal oscillator. If the adjustment of Section 6.6.3 is correct, and the spindle speed is still out out of tolerance (outside the range of 24750 uS to 25250 uS) then the problem is in the crystal oscillator or in the clock divider circuits on the Logic Board.
4. Follow Section 6.7.1 to set the drive up for static tests. If you haven't done so already (when reinstalling the positioner carriage assembly), pay particular attention to getting the reticule-to-scale gap correct (0.005"), step 6.7.1(11) through 6.7.1(21). Note that it is not necessary to start the

drive (get the spindle turning) because the heads are removed. Be especially careful to adjust the gap to be the same at both extremes of positioner movement.

5. If this is a major disk drive rebuild (i.e. the existing potentiometer settings are not valid), then set the potentiometers to their initial settings per Section 6.7.1.1. It is not always easy to set these potentiometers to their midpoint using the procedure described in this section. If you are in doubt, use an ohmmeter.
6. Follow Section 6.8 to perform the static adjustments. Again, it is not necessary to spin-up the drive for these adjustments, as the read/write heads are not yet installed.
7. Power-off the disk, remove the Emergency Unload Bypass jumper, and re-install J205.
8. Remove the servo lamp, and look closely at the servo sensor through the servo scale. Slide the carriage it that the inside edge of the pattern that is printed on the scale lust aligns with the edge of the glass on the servo sensor beneath it. Adjust the angle of the receiver post (upon which the servo sensor is mounted) until the printed edge on the scale is exactly parallel to the edge of the servo sensor. (This will make the quadrature adjustment in the next step easier.) Reinstall the servo lamp when done.
9. Section 6.8.3.3 (Quadrature) is among the more difficult adjustments, and it is critical to get this one right. Note that you can get a false setting that appears correct on the oscilloscope, when the receiver post is significantly out of adjustment. (The previous step should prevent this error.) When you are done, the waveforms should look like figure 6-5, and the receiver post should be very close to exactly perpendicular to the positioner scale. If not, you have found one of the false settings.
10. Section 6.8.7 requires a specialized tool, a voice coil polarity tester. However, if this drive ever worked, then the polarity of the voicecoil will still be correct.
11. Reinstall the read/write board for the following tests. Do not install the heads yet.
12. Perform Section 6.9, dynamic positioner adjustments, with the heads still removed. For this section, you will need to use Adexer, with a working Datakeeper Controller.
13. For Section 6.9.2.2 and 6.9.3.2, use Adexer’s SK function:
    a. %SE A=1 {enables automatic mode}

   b. %SK 0 1   {one-track repetitive seek}

14. For Section 6.9.3.3, you will want the hex screw loosened pretty far, but do so without letting the receiver post rotate. Then set a large screwdriver in the slot on the top of the receiver post. Use very light twisting ‘taps’ to tweak the shaft, while watching the ‘scope. Adjust until the waveform looks like Figure 6-5, and is symmetrical - the step on the top of the waveform is in the same position as the step on the bottom of the waveform. Take your time and get this exact. Then gently tighten the hex screw, while watching the scope (to make sure that tightening the hex screw doesn’t mess up the adjustment).
15. For Section 6.9.4.1, use the SK function
    a. %SE A=1   {enables automatic mode}
    b. %SK 0 134 {134-track repetitive seek}
16. For Section 6.9.7.1(3), use the SK function for single-track seeks, as above.
17. For Section 6.9.8.1(2) use Adexer’s RR function.
18. For Section 6.9.9, use the SK function to seek between the various track combinations in table 6-4. Use the ST function for step 12 of this table: %ST A.
    a. If you notice a DC offset at TP 20 as the heads move toward the spindle, then you probably have not disconnected and disabled the temperature compensation as required in the setup for these adjustments.
    b. If the servo oscillates (rings) in any position during this test, then you may have a loose winding on the voicecoil. Inspect it carefully in place - but you will probably have to disassemble things, remove the voicecoil, and inspect/repair it out of the drive.
    c. Another possible cause of ringing, particularly at inner tracks, is the velocity transducer magnet position - either dragging too much against its coil or vibrating because it is not touching the coil at all. Check and adjust.
19. Continue with all of the tests and adjustments through Section 6.11
20. Power down and install all 4 heads, following the procedure of Section 6.22.2. Be extra careful that the positioner magnet does not steal your screws as you try to install them. Set the adjustment screws to approximately the same position they were in, when you removed the heads.

21. Double-check the dynamic positioner adjustments of Section 6.9, with the added mass of the heads. (They should still be pretty close to perfect.)

HEAD ALIGNMENT

1. Power down and reinstall the Read/Write board. Install new adhesive wire clamps to hold the ground wire across the top of the magnet.
2. Double-check the cleanliness of both sides of the fixed disk and insert a clean removable cartridge.
3. Perform the alignment steps in Section 6.12.
4. For step 6.12.1.1(4) and 6.12.2(4), use the following Adexer functions to write a track of zeros:
   a. %FB 0 {fill RAM buffer with 0}
   b. %CW 0 {write RAM buffer to controller buffer 0}
   c. %SK 0 {Seek to cylinder 0}
   d. %SE P=0 {select removable cartridge}
   e. %SE S=0 {select top side}
   f. %TW 0 {write controller buffer 0 to all sectors}
5. For steps 6.12.2(3) and 6.12.3.1(4), use the following Adexer functions to write a track of all ones:
   a. %FB FF {fill RAM buffer with FF hex}
   b. %CW 0 {write RAM buffer to controller buffer 0}
   c. %SK 1 {Seek to cylinder 1}
   d. %SE P=0 {select removable cartridge}
   e. %SE S=0 {select top side}
   f. %TW 0 {write controller buffer 0 to all sectors}
6. Section 6.14 requires a CE alignment disk. You simply can't align the heads without a CE disk. Without a CE disk, you can adjust the drive so that it works correctly, but your removable cartridges will not be interchangeable with those from other disk drives.
7. During the alignment process, be aware that whenever you seek (or perform pretty much any other disk function), the Datakeeper controller will also select the top surface of the removable cartridge. If you are aligning any other head or looking at any other read channel, you will need to follow each SK operation with a SE H=1 or SE H=0 (whichever

head you are aligning), to re-select both the correct head and the correct platter.

8. Most CE disks do not have sector notches on them - they only have the index notch. This means that the sector separator cannot work, and you cannot use TP2 to trigger the ‘scope during alignment, as specified in the Pertec alignment instructions, section 6.14. Instead, trigger the scope on the negative edge of TP21.
9. Perform the radial alignment procedure of section 6.14.6. Note that the alignment may be off by several tracks. If you do not see recognizable waveforms at cylinder 146, use Adexer’s SK function to look at nearby cylinders, to see which way your alignment is off. Alignment is a delicate procedure - move the heads a little at a time, and observe the results on the ‘scope.
10. Perform the circumferential adjustment of 6.14.7.

Other Adjustments

11. Section 6.13 requires a temperature probe. A modern thermal imaging probe works fine.

Section 3: Supplemental Datakeeper Documentation

This section contains useful information about the Datakeeper. Much of this information can also be found in the Altair Hard Disk (88-HDSK) Preliminary Documentation, though that document contains mistakes that have been corrected here.

Disk Controller Commands

The Altair issues commands by first writing the low command byte to the ADATA port, and then writing the high byte to the ACMD port. Writing the high command byte initiates the command.

Command	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Seek	0	0	0	0	Unit		X	C8	Cylinder Address C7:C0							
Write Sector	0	0	1	0	Unit		Buffer		Head			Sector				
Read Sector	0	0	1	1	Unit		Buffer		Head			Sector				
Write Buffer (1)	0	1	0	0	X	X	Buffer		Byte Count (0 means 256)							
Read Buffer (2)	0	1	0	1	X	X	Buffer		Byte Count (0 means 256)							
Read Status (3)	0	1	1	0	Unit		X	X	IV Byte Address							
Set IV Byte (4)	1	0	0	0	X	X	X	X	IV Byte Address							
Read Unformatted	1	0	1	0	Unit		Buffer		Head			Sector				
Format	1	1	0	0	Unit		X	X	Head			X	X	X	X	X
Initialize	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X

Notes:

(1) After issuing both command bytes, wait for ADPA (in ADSTA) to become high, and then write Byte Count bytes to the ADATA port without handshaking.

(2) After issuing both command bytes, wait for CDA (in CDSTA) to become high, and then read Byte Count bytes from the CDATA port without handshaking.

(3) After issuing both command bytes, wait for CRDY (in CREADY) to become high, and then read the IV Byte value from the CDATA port. Note that reading any IV Byte will set the Select bits in IV Byte H according to the Unit field of this command, and will also set the Extension Select bit to False, and the Platter Select and Head Select bits to True, in IV Byte H.

(4) After issuing both command bytes, wait for ADPA (in ADSTA) to become high, and then write the IV Byte value to the ADATA port.

Field	Minimum	Maximum
Unit	0	3
Buffer	0	3
Head	0	7
Sector	0	23 (17h)
Cylinder Address	0	405 (195h)
Byte Count (1)	0	255 (FFh)

Note 1: the value 0 means 256 bytes to be transferred.

Error Codes

The Controller responds to commands by writing an 8-bit code to the CREADY port. The error bits are interpreted as follows:

Bit	Meaning (1)	Applicable Commands
0	Drive not ready	All except Initialize and Set IV Byte
1	Illegal Sector	Seek, Read Sector, Read Unformatted, Write Sector, Format
2	CRC error in sector data	Read Sector (2)
3	CRC error in sector header	Seek, Read Sector, Read Unformatted, Write Sector, Format
4	Header has wrong sector	Seek, Read Sector, Read Unformatted, Write Sector, Format
5	Header has wrong cylinder	Seek, Read Sector, Read Unformatted, Write Sector, Format (3)
6	Header has wrong head	Seek, Read Sector, Read Unformatted, Write Sector, Format
7	Write Protect	Seek, Read Sector, Read Unformatted, Write Sector, Format

Notes:

1. All bits are 1 after controller power-on.
2. Read Unformatted will usually return a CRC error in sector data.
3. Occurs spuriously when one of these commands is issued for a different unit than was specified for the previous seek. The write logic ignores such spurious errors.

IV Byte Functions

The IV Bytes are input and output ports for the 8X300 processor in the Datakeeper controller. These IV Bytes are visible to the Altair, via the Read Status command, and can also be written by the Altair, via the Set Byte Command. Note that the Read IV Byte command overwrites bits 6:0 of IV Byte H. (See above.) See the MITS documentation for further details.

IV Bus Bit:			**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Altair Bit:			7	6	5	4	3	2	1	0
IV Byte	**Addr**	**Mode**								
X	1	Out	Buffer Number		All dir	A10 dir	A13 Dir	A12 Dir	A9 Dir	A8 Dir
A8	2	Out	Buffer Address							
A9	3	In	ACMD							
A10	4	Out	P2:CB1	P2:CA1	X	X	X	X	P1:CB1	P1:CA1
A11	5	In	P2:CB2	P2:CB1	X	X	X	X	P1:CB2	P1:CA2
A12	6	Out	CDATA							
A13	7	In	ADATA							
H	17	Out	Start/ Stop	Exten. Select	Platter Select	Head Select	Select 4	Select 3	Select 2	Select 1
I	18	Out	Emerg. Unload	Offset Plus	Offset Minus	Enable Write	Cyl. Restore	Cyl. Strobe	X	Cyl. Addr 8
J	19	Out	Cylinder Address 7:0							
K	20	In	Malfunc	X	X	Exten. Status	X	Dual Platter	X	Double Track
L	21	In	Ready	Index Pulse	File Protect	Illegal Address	Seeking Busy 4	Seeking Busy 3	Seeking Busy 2	Seeking Busy 1
M	22	In	Sector Pulse	Sector Count 6:0						
A	33	In	FIFO Output Data 7:0							
B	34	Out	FIFO Input Data 7:0							
C	35	Out	Load Pulse 3	Load Pulse 2	Load Pulse 1	Load Pulse 0	Bit Counter Data 3:0			
D	36	In	X	X	X	X	X	DRDST	DTRCMP	TRR
E	37	Out	CRCAPE	DSTRAN START	DISRMD	X	CLRn	X	TRAS	SDSELn

ALTAIR - DATAKEEPER INTERFACE SIGNALS

This table shows the interconnections between the 88-4PIO in the Altair and the Datakeeper controller.

Altair	88-4PIO Port		8X300	Input		Output		C1 Handshake (4PIO Input)			C2 Handshake (4PIO output)			
Port Name	**Altair Address**	**4PIO Function**	**IV Byte**	**Data Meaning**	**Signal Effect**	**Data Meaning**	**Signal Effect**	**Signal Name**	**Operation**	**IV 4 bit Output**	**Signal Name**	**Operation**	**Controller Function**	**IV 5 bit Input**
CREADY	160 A0h	P1A Status/ Control		Bit 7 = **CRDY** Controller is ready for a command at ADATA & ACMD (1)		88-4PIO Initialization		P1:CA1 **CRDY STROBEn**	Pulsed by Controller to set CRDY	7				
CSTAT	161 A1h	P1A Data	2	Error Codes	Clears CRDY Pulses ERRACKn	88-4PIO Initialization					**ERRACKn**	Pulsed by 4PIO on read from P1A Data	Never viewed by Controller	7
ACSTA	162 A2h	P1B Status/ Control		Bit 7 = **CMDACK** Controller has received command (1)		88-4PIO Initialization		P1:CB1 **CMDACK STROBEn**	Pulsed by Controller to set CMDACK	6				
ACMD	163 A3h	P1B Data	3	No Meaning	Clears CMDACK	Command high byte, writing this byte initiates a command	Sets CMDRDYn				P1:CB2 **CMDRDYn** Command Ready	Set by 4PIO with write to CMD HIGH. Cleared by Controller with CMDACK STROBE.	Controller waits for CMDRDY before reading ACMD and initiating command	6
CDSTA	164 A4h	P2A Status/ Control		Bit 7 = **CDA** (CDATA Available) Data is available at CDATA. (2)		88-4PIO Initialization		P2:CA1 **CDA STROBEn**	Pulsed by Controller to set CDA	1				
CDATA	165 A5h	P2A Data	6	Controller Data (Data/Status from Ctlr)	Clears CDA Pulses CDACKn	88-4PIO Initialization					P2:CA2 **CDACKn** Ctlr Data Ack	Pulsed by 4PIO on read from CDATA	Controller waits for CDACK before sending next data byte to CDATA	1
ADSTA	166 A6h	P2B Status/ Control		Bit 7 =**ADPA** (ADATA Port Available) Altair may write to ADATA (3)(4)		88-4PIO Initialization		P2:CB1 **ADPA STROBEn**	Pulsed by Controller to set ADPA	0				
ADATA	167 A7h	P2B Data	7	No Meaning	Clears ADPA	Altair Data, also command low byte	Pulses ADSTRBn				P2:CB2 **ADSTRBn** Altair Data Strobe	Pulsed by 4PIO on write to ADATA	Controller waits for ADSTRB before reading ADATA (but not on commands)	0

Notes:

1. CMDACK is really only useful to see if the controller has hung - if CMDACK is active, and CRDY is not, then the controller is still executing the command.
2. The controller sets CDA only at the beginning of a block transfer from the controller. No handshaking is performed between bytes of a block transfer - the 8X300 just keeps up.
3. The controller sets ADPA only at the beginning of a block transfer to the controller. No handshaking is performed between bytes of a block transfer - the 8X300 just keeps up.
4. The controller does NOT set ADPA prior to receiving a command. The Altair may send both command bytes (low byte at ADATA first) if CRDY is true.

SECTION 4: BOOT LOADER

Following is a listing for an 88-HDSK boot loader ROM.

```
;=================================================================
; Hard Disk Boot Loader (HDBL)
; By Martin Eberhard
;
; HDBL is a 256-byte PROM program that loads the boot file from
; an 88-HDSK hard disk, and executes the successfully loaded
; code. Progress and error messages are printed on a "standard"
; 6850-based Altair Terminal port at address 10H and 11h, such
; as port A of an 88-2SIO, the serial port on a Turnkey Module,
; or the serial port of an 88-UIO.
;
; The standard 88-HDSK system uses a Pertec D3422 disk drive,
; which contains 2 platters - one is in a removable cartridge,
; the other is a fixed platter. However, The 88-HDSK controller
; can actually support up to 4 platters, supporting the Pertec
; D3462 disk drive, which has one removable platter, and 3
; fixed platters.
;
; Altair software normally boots only from the removable
; cartridge of a D3422 disk drive. HDBL allows you to boot also
; from the fixed platter. You can select from which platter to
; boot using Sense Switch 3 (A11 on the front panel). 0 (down)
; selects the removable cartridge, 1 (up) selects the fixed
; platter. This switch was chosen to minimize collisions with
; the normal use of sense switches <A11:A8>. 0000 through 0110
; select the input device for MBL and for the various checksum
; loaders on tape (and values above 0110 will generate an
; error). So if you have the sense switches (which may be
; inaccessible on an 8800b Turnkey board) set forloading from
; e.g. cassette, then the machine will still boot from the
; removable cartridge when booting via HDBL.
;
; There are 24 256-byte sectors per track, and these are
; numbered 0 through 23 on each track. Each platter has 2
; sides, numbered 0 and 1. Data on each platter is organized as
; a sequence of Disk Pages, where each Page is one sector.
; Pages are numbered sequentially starting at 0 (on track 0,
; side 0), through the 24 sectors on track 0, side 0, and then
; on to track 0, side 1, where sector 0 is page 24.  Page 47 is
; the first sector on track 1, side 0, and page numbering
; continues this way through all the tracks.
;
; Page 0 (which is track 0, side 0, sector 0) is the Pack
; Descriptor Page, containing various information about the
; particular disk platter. Bytes 40-43 of this Page are the
; "Opsys Pointers." Bytes 40 & 41 are the Page number of the
; starting boot Page, Bytes 42 & 43 are the number of Pages to
; load during boot. HDBL assumes that the boot file is to be
; loaded into memory starting at address 0000, and executed
; there.
;
; During loading, the INTE (Interrupt Enabled) LED on the front
; panel will be off. Any error during loading will cause the
; INTE LED to light and a "LOAD ERR" message to be printed
; on the Terminal. The error code is stored in memory at
```

```
                ; address 0. HDBL will then hwng in a loop until Reset.
                ;
                ; Because HDBL may be running stand-alone, the Terminal port
                ; gets initialized during HDBL initialization. But if control
                ; came from UBMON (with an "L" command) or from TURMON (with a
                ; "J 176000" command) then the Terminal port's ACIA will still
                ; be transmitting the last two character-echos of the command
                ; when HDBL begins. HDBL will stall 6.5 uS (one character time
                ; above 1800 baud) before initializing the Terminal port, so
                ; that the ACIA will have time to finish transmitting this last
                ; characters before it gets reset.
                ;
                ; Newer Turnkey Modules (and older ones with the 88-SYS-CLG
                ; modification) will disable the PROMs when an IN instruction
                ; reads port FFh (the Sense Switches). Some of these boards
                ; will disable the PROM when *any* IN instruction is
                ; executed. For this reason, HDBL copies itself to RAM, and
                ; runs from there, before executing any IN instructions.
                ;
                ; This code is written to assemble with ASM by Digital Research
                ;
                ;===============================================================
                ; Revision History
                ;   1.00  15SEP2013  M.Eberhard
                ;     Created
                ;   1.01  25MAR2014  M.Eberhard
                ;     Fixed bug with loading from alternate platters, code
                ;     squeeze, improve comments
                ;   1.02  26MAR2014  M.Eberhard
                ;     Further compression. Print platter number in signon msg.
                ;     Jump to XENTER instead of XMON on error, so ACIA reset
                ;     doesn't hose the last chr of error code.
                ;   1.03  05JUN2013  M.Eberhard
                ;     Use sense switch All to select boot platter. Eliminate
                ;     selection of boot drive.
                ;   2.00  15AUG2014 M.Eberhard
                ;     Copy code to RAM before execution, for Turnkey Module
                ;     compatibility. Remove RAMCOD option. Eliminate return
                ;     to UBMON/TURMON on load error (Must reset on error.)
                ;
                ; (Remember to update the Version String below)
                ;===============================================================
 0000 =         FALSE   equ   0
 FFFF =         TRUE    equ   not FALSE

                ; PROM address and Entry point for HDBL. UBMON assumes FC00h.

 FC00 =         HDBL    equ   0FC00h      ;Beginning of HDBL PROM

                ; RAM address for moved code. Exactly one of these
                ; should be used.

 BF00 =         RAMPAG  equ   0BF00h      ;beginning of RAM page (48K system)
                ;RAMPAG equ   0F700h      ;beginning of RAM page (62K system)
                ;RAMPAG equ   0FB00h      ;beginning of RAM page (63K system)
                              ;(e.g. Turnkey Module's RAM)

 3D00 =         ROF     equ   HDBL-RAMPAG ;RAM relocation offset

                ; Sense Switch assignment for selecting the boot platter & unit
```

```
 00FF =          SSWTCH  equ   0FFh        ;Sense Switch address
 0008 =          PSWTCH  equ   008h        ;mask for platter Switches
                                     ;Code assumes 008h (bit 3)

                 ; Terminal port equates - same for 88-2SIO port 0, Turnkey
                 ; Module, and 88-UIO (all based on the Motorola 6850 ACIA)
                 ; Note: transmitting with 2 stop bits is also compatible with a
                 ; receiver that is programmed for 1 stop bit.

 0010 =          ACCTRL  EQU   10h         ;ACIA Control output port
 0010 =          ACSTAT  EQU   10h         ;ACIA Status input port
 0011 =          ACTXD   EQU   11h         ;ACIA TX Data register
 0011 =          ACRXD   EQU   11h         ;ACIA RX Data register

 0003 =          ACRSET  EQU   00000011b   ;Master reset
 0001 =          ACRDF   EQU   00000001b   ;RX Data register full
 0002 =          ACTDE   EQU   00000010b   ;TX Data register empty
 0011 =          ACINIT  EQU   00010001b   ;/16, 8-bit, No Parity, 2Stops

                 ; 88-HDSK ports (The interface board is actually an 88-4PIO.)

 00A0 =          CREADY  equ   0A0h  ;IN: Ctlr ready for command (bit7)
 00A1 =          CSTAT   equ   0A1h  ;IN: error flags, reset CREADY
 00A2 =          ACSTA   equ   0A2h  ;IN: Command Ack (bit 7)
 00A3 =          ACMD    equ   0A3h  ;IN: reset Command Ack
                               ;OUT: Command high byte/initiate
 00A4 =          CDSTA   equ   0A4h  ;IN: data/stat availablr at CDATA
 00A5 =          CDATA   equ   0A5h  ;IN: Disk data or status from Ctlr
 00A6 =          ADSTA   equ   0A6h  ;IN: ADATA Port Available (bit 7)
 00A7 =          ADATA   equ   0A7h  ;OUT: Command low byte

                 ; 88-HDSK ACMD:ADATA Commands

 0000 =          CSEEK   equ   00h   ;Bits 15:12 = 0000b
                               ;Bits 11:10 = Unit #
                               ;Bits  9:0  = Cylinder #

 0030 =          CRDSEC  equ   30h   ;Bits 15:12 = 0011b
                               ;Bits 11:10 = Unit #
                               ;Bits  9:8  = Buffer #
                               ;Bit   7:6  = Platter #
                               ;Bits    5  = Side #
                               ;Bits  4:0  = Sector #

 0020 =          CSIDE    equ    020h        ;Side select for CRDSEC
 00C0 =          CFPLTR   equ    0C0h        ;platter mask for CRDSEC
 000C =          CUNIT    equ    00Ch        ;Unit mask for CSEEK & CRDSEC

 0050 =          CRDBUF  equ   50h   ;Bits 15:12 = 0101b
                               ;Bits 11:10 = not used
                               ;Bits  9:8  = buffer #
                               ;Bits  7:0  = # bytes to transfer
                               ;(00 means 256)

                 ; 88-HDSK CSTAT error bits

 0001 =          ERDNR   equ   01h   ;drive not ready
 0002 =          ERBADS  equ   02h   ;illegal sector
 0004 =          ERSCRC  equ   04h   ;CRC error during sector read
 0008 =          ERHCRC  equ   08h   ;CRC error during header read
 0010 =          ERSWRG  equ   10h   ;header has wrong sector
```

```
0020 =          ERCWRG  equ   20h   ;header has wrong cylinder
0040 =          ERHWRG  equ   40h   ;header has wrong head
0080 =          WPROT   equ   80h   ;Write Protect
007F =          ERMASK  equ   7Fh   ;all the actual error bits

                ; 88-HDSK Constants

0028 =          OSOFF   equ   40    ;Page 0 offset to opsys pointers
0018 =          HDSPT   equ   24    ;Sectors per track
0000 =          DBUFR   equ   0     ;Default controller buffer: 0-3
                              ;Code gets longer if <>0

                ; ASCII characters

000D =          CR      equ   0Dh
000A =          LF      equ   0Ah

                ;=============================================================
                ; Start of HDBL PROM
FC00              org   HDBL
                ;=============================================================
FC00 F3           di                ;front panel INTE light off

                ;-------------------------------------------------------
                ; Copy code to RAM. This will provide 6.5 mS of delay.
                ; This will allow a UART that is running at 1700 baud or
                ; faster to complete transmission. (This also leaves 18
                ; bytes in RAM for the stack.)
                ;-------------------------------------------------------
FC01 2100C0       lxi   h,RAMPAG+100h           ;last RAM address+1
FC04 16FC         mvi   d,(HDBL/256)            ;PROM code page

FC06 2B         COPLUP: dcx   h           ;(5+1)
FC07 5D           mov   e,l         ;(4+1)
FC08 1A           ldax  d           ;(7+1)
FC09 77           mov   m,a         ;(7+1)
FC0A 7D           mov   a,l         ;(4+1)
FC0B D612         sui   RAMCOD and 0FFh   ;(7+2)ends with a=0
FC0D C206FC       jnz   COPLUP            ;(10+3)

                ;54 cycles per pass X (256-18) /2 = 6.426 mS

                ;----------------------------------------------------
                ; Set up system stack immediately below RAM code image
                ;----------------------------------------------------
FC10 F9           sphl

                ;-----------------------------
                ;go to loaded code (with a=0)
                ;-----------------------------
FC11 E9           pchl

                ;============================================================
                ; All of the following code gets copied to RAM and run there.
                ; On Entry:
                ;   a = 0
                ;============================================================
FC12 67         RAMCOD: mov   h,a         ;set load initial page
FC13 6F           mov   l,a         ;hl=0

                ;-------------------------------------
```

```
                ; Initialize 88-HDSK interface board
                ; (Actually ports 0 and 1 of an 88-4PIO)
                ; On Entry:
                ;   a = 0
                ;  hl = 0
                ; On Exit:
                ;  hl = 0
                ;--------------------------------------
FC14 D3A0         out   0A0h        ;Select port 0Ah DDR
FC16 D3A2         out   0A2h        ;Select port 0Bh DDR
FC18 D3A4         out   0A4h        ;Select port 1Ah DDR
FC1A D3A6         out   0A6h        ;Select port 1Bh DDR
FC1C D3A1         out   0A1h        ;Port 0Ah is an input port
FC1E D3A5         out   0A5h        ;Port 1Ah is an input port

FC20 2F           cma
FC21 D3A3         out   0A3h        ;Port 0Bh is an output port
FC23 D3A7         out   0A7h        ;Port 1Bh is an output port

FC25 3E2C         mvi   a,2Ch       ;set up input port handshakes
FC27 D3A0         out   0A0h
FC29 D3A4         out   0A4h
FC2B D3A6         out   0a6h        ;output port 1Bh handshakes

FC2D 3E24         mvi   a,24h       ;set up port 0Bh handshakes
FC2F D3A2         out   0A2h

FC31 DBA1         in    CSTAT       ;clear Controller Ready bit

                ;-------------------------------------------
                ; Reset and initialize the Terminal port ACIA
                ; On Entry & Exit:
                ;  hl = 0
                ;-------------------------------------------
FC33 3E03         mvi   A,ACRSET
FC35 D310         out   ACCTRL
FC37 3E11         mvi   A,ACINIT
FC39 D310         out   ACCTRL

                ;----------------------------
                ; Print HDBL version message
                ; On Entry & Exit:
                ;  hl = 0
                ;----------------------------
FC3B CDE5BF       call  PRINTF-ROF  ;print the following string
FC3E 0D0A484442   db    CR,LF,'HDBL 2.0','0'+80h

                ;-------------------------------------------------
                ; Read the Pack Descriptor Page (Disk Page 0)
                ; to get the Opsys Pointers:
                ;   Bytes 41:40 = Initial Disk Page number
                ;   Bytes 43:42 = Disk Page count (Byte 43=MSB=0)
                ; On Entry:
                ;  hl = 0
                ;-------------------------------------------------
FC49 062B         mvi   b,OSOFF+3   ;byte count to end of pointers
FC4B CD82BF       call  GETPAG-ROF  ;Seek, read page hl into buffer
                                    ;set up to read b buffer bytes

FC4E E5           push  h           ;execution address on stack
FC4F EB           xchg              ;load address into de
```

```
                ; Read from the controller buffer and discard everything until
                ; we get to the opsys pointers. Load the opsys pointers into
                ; C & HL. Note: no testing any handshake here - just assume
                ; the controller can keep up. (The controller can send a data
                ; byte every 2.5 uS.) This only reads the low byte of the
                ; page count, since the high byte must be 0 anyway.

FC50 DBA5       PTRLUP: in    CDATA       ;read byte from controller

FC52 6C           mov   l,h         ;shift everybody over...
FC53 61           mov   h,c
FC54 4F           mov   c,a         ;...and put it away

FC55 05           dcr   b
FC56 C250BF       jnz   PTRLUP-ROF

                ; Announce 'LOADING FROM <platter>' on the Terminal

FC59 CDDCBF       call  LOADPF-ROF  ;CR,LF,'LOAD', then string
FC5C 494E472046  db    'ING FROM',' '+80h

FC65 DBFF         in    SSWTCH            ;read platter switch
                                    ;(disables PROMS in Turnkey bd)
FC67 E608         ani   PSWTCH            ;mask off all others
FC69 0F           rrc
FC6A 0F           rrc
FC6B 0F           rrc
FC6C C630         adi   '0'         ;make it ASCII
FC6E CDF3BF       call  PRINTA-ROF  ;and print it

                ;------------------------------------------------
                ; Read c Pages from disk, starting at Page hl, into
                ; memory starting at the address on the stack
                ; On Entry:
                ;   b = 0
                ;   c = page count
                ;  de = LDADDR
                ;  hl = initial Disk page number
                ;------------------------------------------------
FC71 CD82BF     PAGELP: call  GETPAG-ROF  ;Seek, read page hl into buffer
                                    ;set up to read b buffer bytes
                                    ;b=0 here always.

                ; Load 256 bytes of buffer data into memory at de (b=0 here)
                ; Note: no testing any handshake here - just assume the
                ; controller can keep up. (The controller can send a data byte
                ; every 2.5 uS.)

FC74 DBA5       BYTELP: in    CDATA       ;get a data byte
FC76 12           stax  d           ;write it to RAM
FC77 13           inx   d           ;next address
FC78 05           dcr   b           ;bump byte counter
FC79 C274BF       jnz   BYTELP-ROF  ;until done (b=0)

                ; Next Disk Page

FC7C 23           inx   h           ;Next Disk Page
FC7D 0D           dcr   c           ;bump Disk Page count
FC7E C271BF       jnz   PAGELP-ROF
```

```
               ;---------------------------------------------------
               ; Go execute loaded code, at the address on the stack
               ;---------------------------------------------------
FC81 C9          ret

               ;===Subroutine=========================================
               ; Seek and read disk Page hl into 88-HDSK buffer 0
               ; On Entry:
               ;   b=number of bytes to transfer (0 means 256)
               ; On Exit:
               ;   a,flags trashed, all others preserved
               ;   Controller has specified sector data in its buffer
               ;======================================================
FC82 E5        GETPAG: push  h           ;Save requested Page
FC83 D5          push  d           ;Save regs
FC84 C5          push  b           ;save byte count

               ;-----------------------------------------------------------
               ; Compute cylinder and sectorX2  from Disk Page number in hl
               ;  hl := hl / (2*HDSPT) (Quotient=cylinder)
               ;   h := hl MOD (2*HDSPT) (Remainder=sectorx2)
               ; This is fast only if the cylinder number is low. MITS
               ; usually put the boot image starting at cylinder 0, side 1,
               ; so this will be faster and shorter than the 'fast'
               ; division of previous HDBL rev. This will become slower
               ; if the boot image is above cylinder 20 or so. But we will
               ; always miss the next sector anyway, so each sector will
               ; require a full disk rev (25 mS), plenty of time
               ;-----------------------------------------------------------
FC85 01D0FF      lxi   b,-2*HDSPT
FC88 50          mov   d,b         ;de=FFFF=-1
FC89 58          mov   e,b         ;since loop goes 1 extra

FC8A 13        DIV1:   inx   d           ;compute quotient=cylinder
FC8B 09          dad   b           ;hl gets remainder
FC8C DA8ABF      jc    DIV1-ROF

FC8F 7D          mov   a,l         ;fix remainder, since
FC90 91          sub   c           ;..loop went 1 extra

FC91 EB          xchg              ;cylinder number to hl

               ;--------------------------------------------------
               ; Compute Sector & Side
               ; If sectorX2 > sectors/track then set CSIDE
               ; bit, and reduce sector number by sectors/track
               ;   hl= Quotient (cylinder)
               ;   a = Remainder (sectorX2, either for head 0 or 1)
               ;--------------------------------------------------
FC92 FE18        cpi   HDSPT       ;past end of side 0?
FC94 DA99BF      jc    SIDEOK-ROF  ;N: sector number is good

FC97 C608        adi   CSIDE-HDSPT ;Compute sector mod HDSPT,
                                   ;..and set side 1 bit

FC99 47        SIDEOK: mov   b,a         ;save sector # with side

               ;-------------------------------------------------
               ; Seek Cylinder
               ;    b = sector number, with side bit set correctly
               ;    hl = cylinder number<9:0>
```

```
                ;---------------------------------------------------
                 if CSEEK+DBUFR                  ;these are actually 00
                  mov   a,h          ;h<1:0>=cylinder<9:8>
                  ori   CSEEK+DBUFR ;combine with SEEK cmd
                  mov   h,a
                 endif

                                     ;hl=SEEK command with cyl #
FC9A CDB9BF       call  HDCMD-ROF    ;HCMD gets unit # from switches

                ;---------------------------------------------------------
                ; Get platter from sense switch, and combine with
                ; side and sector already in b
                ;   b<7:6> = 0
                ;     b<5> = side
                ;   b<4:0> = sector number
                ;   Sense Switch <A11> = platter number
                ;---------------------------------------------------------
FC9D 2630         mvi   h,CRDSEC+DBUFR    ;read command, high byte

FC9F DBFF         in    SSWTCH            ;read platter switches
FCA1 E608         ani   PSWTCH            ;mask off all others

FCA3 07           rlc                ;Shift to CFPLTR position
FCA4 07           rlc                ;..which are bits 7:6
FCA5 07           rlc

FCA6 B0           ora   b            ;combine w/ sect & side

                ;----------------------------------------------------------
                ; Read Sector from current track into controller's buffer 0
                ;   a<7:6> = platter
                ;     a<5> = side
                ;   a<4:0> = sector number
                ;----------------------------------------------------------
FCA7 CDBABF       call  HDCMDA-ROF  ;low command byte is in a

                ;----------------------------------------------
                ; Issue CRDBUF command to kick off read of 256
                ; bytes from the controller's buffer
                ; Note: this assumes the controller is ready.
                ; (and it is, because HDCMD left it that way.)
                ;----------------------------------------------
FCAA DBA5         in    CDATA        ;reset CDA in CDSTA
FCAC DBA3         in    ACMD         ;clear CMDACK in ACSTA

FCAE C1           pop   b            ;b=requested byte count
FCAF 78           mov   a,b
FCB0 D3A7         out   ADATA        ;..to controller

FCB2 3E50         mvi   a,CRDBUF+DBUFR    ;issue Read Buffer command
FCB4 D3A3         out   ACMD         ;..to controller

FCB6 D1           pop   d            ;(10)
FCB7 E1           pop   h            ;(10) 10 uS total from 'out'

                ; The 8x300 is ready to transmit data in 8 uS. This code takes
                ; 30 cycles (including the 'ret'), or 15 uS min to get around
                ; to reading the data - so there is no need to wait on CDSTA

                 if FALSE
```

```
                DATAWT: in    CDSTA       ;Wait for data port to be ready
                  rlc               ;msb=CDA
                  jnc   DATAWT-ROF
                 endif

                ;----------------------------------
                ; Controller is ready to transfer
                ; 256 bytes of data from its buffer
                ;----------------------------------
FCB8 C9           ret               ;(10)done with GETPAG

                ;===Subroutine===============================================
                ; Issue a disk command, and then wait for the controller
                ; to complete it
                ;
                ; Note: this just assumes the controller is ready, which is OK
                ; since the last command was either a seek (where HDCMD waited
                ; for the controller to become ready) or it was a CRDBUF, which
                ; ended with all bytes transferred - and the controller becomes
                ; ready very soon (1.5 uS) after the last byte is transferred.
                ; On Entry at HDCMD:
                ;   hl = complete command
                ; On Entry at HDCMDA:
                ;   a=low byte of command
                ;   h=high byte of command
                ; On Exit:
                ;   a,flags trashed, all others preserved.
                ;   The command is completed and the controller is ready.
                ;   Any errors will terminate the load, and print an error
                ;   message on the Terminal
                ;============================================================
FCB9 7D         HDCMD:  mov   a,l         ;low byte of command

FCBA D3A7       HDCMDA: out   ADATA       ;..to data port

FCBC DBA1         in    CSTAT       ;reset CRDY flag just in case
FCBE DBA3         in    ACMD        ;clear CMDACK in ACSTA

FCC0 7C           mov   a,h         ;command high byte
FCC1 D3A3         out   ACMD        ;issue command

FCC3 DBA0       HDWAIT: in    CREADY            ;Is the controller done?
FCC5 07           rlc               ;look at msb=CRDY
FCC6 D2C3BF       jnc   HDWAIT-ROF  ;N: keep waiting

FCC9 DBA1         in    CSTAT       ;reset CRDY flag
FCCB E67F         ani   ERMASK            ;and get A=error code
FCCD C8           rz                ;No errors: happy return

                ;       Fall into error exit

                ;===Error Exit===============================================
                ; Report a load error and store error code in RAM at 0.
                ; Hang here forever, with the INTE light lit.
                ; On Entry:
                ;   a=error flag bits
                ;============================================================
FCCE 320000       sta   0           ;save a=error flags
FCD1 CDDCBF       call  LOADPF-ROF  ;CR,LF,'LOAD', then string
FCD4 204552D2     db    ' ER','R'+80h
```

```
FCD8 FB           ei                ;INTE is error indicator light
FCD9 C3D9BF     FOREVR: jmp   FOREVR-ROF  ;N: die here, INTE light lit

                ;===Subroutine============================================
                ; Print inline string on the Terminal,
                ; preceeded by CR,LF,'LOAD'
                ; On Entry:
                ;    The string address is the "return address" on the stack.
                ;    The string is terminated by bit 7 set in its last chr.
                ;    The actual return address is the next address after the
                ;    last string character.
                ; On Exit:
                ;    Trashes a and flags, all other registers preserved.
                ;=========================================================
FCDC CDE5BF     LOADPF: call  PRINTF-ROF
FCDF 0D0A4C4F41  db    CR,LF,'LOA','D'+80h

                ;fall into PRINTF

                ;===Subroutine============================================
                ; Print inline string on the Terminal
                ; On Entry:
                ;    The string address is the "return address" on the stack.
                ;    The string is terminated by bit 7 set in its last chr.
                ;    The actual return address is the next address after the
                ;    last string character.
                ; On Exit:
                ;    Trashes a and flags, all other registers preserved.
                ;=========================================================
FCE5 E3         PRINTF: xthl              ;get string address, save hl

FCE6 7E         PRNTLP: mov   a,m         ;get string character
FCE7 E67F         ani   7Fh         ;strip end-of-string mark
FCE9 CDF3BF       call  PRINTA-ROF  ;and print it

FCEC BE           cmp   m           ;end of string?
FCED 23           inx   h           ;point to next chr
FCEE CAE6BF       jz    PRNTLP-ROF  ;No difference: keep going

FCF1 E3           xthl              ;restore hl, put return address
FCF2 C9           ret               ;..onto stack, and go there

                ;===Subroutine==================
                ; Print a the Terminal
                ; On Entry:
                ;   a=chr to print
                ; On Exit:
                ;    all registers preserved.
                ;===============================
FCF3 F5         PRINTA: push  psw         ;save chr to print

FCF4 DB10       PALOOP: in    ACSTAT            ;Wait for TX to be ready
FCF6 E602         ani   ACTDE
FCF8 CAF4BF       jz    PALOOP-ROF

FCFB F1           pop   psw
FCFC D311         out   ACTXD       ;and send chr
FCFE C9           ret

FCFF              end
```