

# HDCA

HARD DISK CONTROLLER

---

## REFERENCE MANUAL

MORROW



**Disk Jockey HDCA Controller  
Technical Reference Manual**

**Revision 2  
February 1983**

---

Copyright (C) 1983 by Morrow Inc.

All rights reserved.

---

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without prior written permission of Morrow Inc.

---

**DISCLAIMER**

No representations or warranties, express or implied, are made with respect to the contents hereof, including, but not limited to, the implied warranty of merchantability or fitness for a particular purpose. Further, Morrow Inc., reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation to notify any person of such revision.

---

Morrow  
600 McCormick St.  
San Leandro, CA 94577

## **IMPORTANT WARRANTY INFORMATION**

### **LIMITED WARRANTY**

Morrow, Inc. warrants its products to be free from defects in workmanship and materials for the periods indicated below. This warranty is limited to the repair and replacement of parts only.

This warranty is void if, in the sole opinion of Morrow Inc., the product has been subject to abuse or misuse, or has been interconnected to other manufacturer's equipment for which compatibility has not been established in writing.

Circuit boards - Parts, including the printed circuit board, purchased as factory assemblies, are warranted for a period of ninety (90) days from the original invoice/purchase date.

Electro-mechanical peripherals - Peripheral equipment such as floppy or hard disk drives, etc., not manufactured by Morrow, Inc., are included in the limited warranty period of 90 days from the original invoice date when sold as part of a Morrow system.

Exception - Expendable items such as printer ribbons, software media, and printwheels are not covered by any warranty.

Software/Firmware - Morrow, Inc. makes no representations or warranties whatsoever with respect to software or firmware associated with its products and specifically disclaims any implied or expressed warranty of fitness for any particular purpose or compatibility with any hardware, operating system, or software/firmware. Morrow, Inc. reserves the right to alter or update any program, publication or manual without obligation to notify any person of such changes.

**LIMITATION OF LIABILITY: THE FOREGOING WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MORROW, INC. BE LIABLE FOR CONSEQUENTIAL DAMAGES EVEN IF MORROW, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.**

### **WARRANTY RETURN PROCEDURE**

Should a buyer experience a defect in either workmanship or materials during the warranty period, any Morrow Authorized Service Center will replace or repair the product at its expense only if the product is promptly returned to the dealer or Service Center with dated proof of purchase.

Should factory repair be necessary, the Service Center shall contact Morrow Customer Service for a Return Materials Authorization (RMA) number.

Disk Jockey Winchester Hard Disk (HDCA) Controller  
Technical Reference Manual  
Revision 2

Table of Contents

1. INTRODUCTION.....	1
1.1. General.....	1
1.2. Manual Organization.....	1
2. SYSTEM OVERVIEW.....	2
2.1. Hardware Features.....	2
2.2. Discus Systems.....	2
2.3. Software Features.....	3
3. INSTALLING THE HDCA - STANDARD ENVIRONMENT.....	5
3.1. Installation Overview.....	5
3.2. Installation Requirements.....	5
3.3. Examine the Board.....	5
3.4. Setting the Switch.....	6
3.5. Factory Set Jumper Settings.....	7
3.6. Seek Complete Jumpers.....	8
3.7. Seating (Inserting) the HDCA Board.....	9
3.8. Power-Up.....	9
3.9. Cable Connection.....	10
3.10. Disk Drive Connection.....	12
3.10.1. M26 Drive Connection.....	13
3.10.2. M10 and M20.....	14
3.11. Multiple Drive Connection.....	15
3.11.1. M26 Daisy Chaining.....	16
3.11.2. M10/M20 Daisy Chaining.....	16
3.11.3. HDCA Modifications and Connections.....	16
4. SYSTEM START-UP AND POWER DOWN.....	18
4.1. Standard Environment - Overview.....	18
4.2. The System Bootstrap.....	18
4.2.1. Implementing the Bootstrap.....	18
4.2.2. Bootstrap Software.....	19
4.3. Disk Initialization and Formatting.....	19
4.4. Running CP/M on the Discus System.....	19
4.4.1. Morrow Designs System.....	20
4.4.2. Installing a System on the Hard Disk.....	21
4.4.3. Cold Booting the Hard Disk (BOOTH.D.COM).....	22
4.4.4. Automatic Hard Disk Boot.....	22
4.5. Back-Up Procedures.....	22
4.6. System Shut Down.....	24
5. DISK DATA ORGANIZATION.....	25

Table of Contents, Cont.

6.	OPERATION IN A NON-STANDARD ENVIRONMENT.....	28
6.1.	Installable BIOS (IBIOS).....	28
6.2.	I/O Driver Overview.....	29
6.3.	Creating an I/O Driver.....	29
6.4.	The I/O Driver.....	30
6.5.	I/O Driver Specifications.....	31
6.6.	Installing I/O Drivers.....	32
6.7.	Interrupts.....	33
7.	SOFTWARE CUSTOMIZATION.....	35
7.1.	Overview.....	35
7.2.	Low Level Drivers.....	35
7.3.	HDCA Jump Table .....	36
7.4.	Disk Utility Subroutines.....	36
7.4.1.	HOME.....	37
7.4.2.	SETTRK.....	37
7.4.3.	SETSEC.....	38
7.4.4.	SETDMA.....	38
7.4.5.	READ.....	38
7.4.6.	WRITE.....	39
7.4.7.	SETDRV.....	40
7.4.8.	GETSTAT.....	41
7.4.9.	HALT.....	41
7.4.10.	INDEX .....	41
7.4.11.	READY.....	41
7.4.12.	WRITE FAULT.....	42
7.4.13.	TIME OUT (TIMOUT).....	42
7.4.14.	COMPLETE (COMPLT).....	42
7.4.15.	OP DONE (OPDONE) .....	42
7.4.16.	RETRY.....	43
7.4.17.	SEEK DONE (SDONE).....	43
7.4.18.	SETHEAD.....	43
7.4.19.	SETKEY.....	43
8.	HARDWARE LEVEL REGISTERS.....	45
8.1.	Overview.....	45
8.2.	I/O Addressing.....	45
8.3.	I/O Register Map.....	45
8.4.	Read-Only Registers - Register 0.....	46
8.4.1.	HALT.....	46
8.4.2.	ILEVEL.....	47
8.4.3.	NREADY.....	47
8.4.4.	NFAULT.....	47
8.4.5.	TIME OUT (TIMOUT).....	47
8.4.6.	SEEK COMPLETE .....	48
8.4.7.	OPDONE.....	48
8.4.8.	NTRCKO .....	48
8.5.	Read-Only Registers - Register 1.....	49
8.5.1.	R0 and R1.....	49
8.5.2.	RETRY.....	49
8.5.3.	SDONE.....	50

## Table of Contents, Cont.

8.6. Read Only Registers - Register 2.....	50
8.7. Read/Write Registers - Register 3.....	50
8.8. Write-Only Registers - Register 4.....	51
8.8.1. WPROT.....	52
8.8.2. DSKCLK.....	52
8.8.3. RUN.....	52
8.8.4. FRENBL.....	52
8.9. Write-Only Registers - Register 5.....	53
8.9.1. COMMAND 0 .....	53
8.9.2. COMMAND 1.....	53
8.9.3. COMMAND 3.....	54
8.9.4. COMMAND 5.....	54
8.9.5. COMMAND 7.....	55
8.9.6. COMMAND 8.....	55
8.10. Write-Only Registers - Register 5.....	55
8.10.1. NHDSL.....	56
8.10.2. DIR.....	56
8.10.3. NSTEP.....	57
8.10.4. DRVSL.....	57

## Appendices

A. TROUBLESHOOTING GUIDE.....	58
B. SOFTWARE LISTINGS.....	59
B.1. The Bootstrap Software.....	59
B.2. Low Level Drivers.....	61
C. OPTIONAL PORT ADDRESSES.....	62
D. PERFORMANCE SPECIFICATIONS.....	64
COMPONENT LAYOUT/SHEMATICS	
PARTS LIST	
SUBJECT INDEX	

## List of Figures

3-1: HDCA DIP Switch Location.....	6
3-2: Factory Switch Settings (Port 50H).....	7
3-3: HDCA Jumper Locations.....	8
3-4: Seating the HDCA.....	9
3-5: 50-Pin Cable Connection (P1).....	10
3-6: 20-Pin Cable Connection (P2 -P5).....	11
3-7: Cable Connection.....	12
3-8: M26 Disk Drive (Spindle Lock and Actuator Lock).....	13
3-9: M10/M20 Disk Drive (Location of Head Lock).....	14
3-10: Multiple Drive Connection.....	15
3-11: HDCA Controller Board (Location for Required ICs).....	17
5-1: Disk Data Organization .....	25

## List of Tables

3-1: ICs Required for Multiple Drive Connection.....	16
5-1: Discus Systems Comparison .....	25
5-2: 512 Bytes/Sector Format.....	27
5-3: Sector Skew Table (Track Format).....	27
6-1: CP/M Jump Instructions.....	30
6-2: CP/M Devices.....	31
6-3: CP/M Drivers.....	32
7-1: Utility Subroutine Addresses.....	36
7-2: READ - Register A Error Bits.....	39
7-3: WRITE - Register A Error Bits.....	40
7-4: The "A" Status Register.....	41
7-5: The "B" Status Register.....	41
7-6: Sector Header Fields .....	43
8-1: HDCA I/O MAP.....	46
8-2: Controller Status Register.....	46
8-3: Auxiliary Status Register.....	49
8-4: Encoding Scheme for R0 and R1.....	49
8-5: Master Control Register.....	51
8-6: HDCA Command Summary .....	53
8-7: Drive Function Register.....	55
8-8: NHDSL Bits and Head Select Relationship.....	56
8-9: DRVSL Bits and Drive Select Relationship.....	57



## 1. INTRODUCTION

### 1.1. General

The Disk Jockey Winchester Hard Disk Controller (HDCA), when combined with a Morrow Designs Winchester-type hard disk drive and a power supply, provides the user with a complete Morrow Designs Discus M26, M20 or M10 mass storage system. The Discus M26 drive offers 26 MB of formatted storage capacity; the M20 drive is a 20 MB drive, and the M10 is a 10MB disk drive.

The HDCA controller plugs into any S-100 slot of the motherboard in an S-100 system, and is connected to the drive through a pair of flat cables. One of the cables carries clock and data information, and the other carries control and status information. The controller can accommodate up to four drives of the same type, so the M26 system can be expanded to a total capacity of 104 million bytes, the M20 to over 80 million bytes and the M10 to over 40 million.

Morrow Designs supplies the popular CP/M Version 2.2 with Discus systems, and also configures the HDCA to run Micronix, Morrow Designs' multi-user operating system. CP/M is the industry standard operating system for 8080/Z80 based systems.

### 1.2. Manual Organization

This manual was written as a technical reference guide for persons installing the HDCA in their computer systems. Section 2 provides an overview of the HDCA features; hardware, software and general features and capabilities are outlined. Section 3 covers installation of the HDCA in a "standard environment". Section 4 discusses system start-up in the standard environment, and procedures for back-up and system power-down. Section 5 provides general information concerning data organization on the hard disk, and sector format detail. Section 6 provides information to enable operation in a "non-standard" environment. Section 7 covers software customization, disk drivers, programming specifications and disk utility subroutines. Section 8 details the HDCA hardware functions.

Appendices include a troubleshooting guide, utility software listings, power requirements, component layout, schematics, parts list and a detailed subject index.

## System Overview

### 2. SYSTEM OVERVIEW

#### 2.1. Hardware Features

The HDCA controller board provides connection to a maximum of four (4) drives of the same type. A 50 pin connector is located in the upper right corner of the board. This connector, with cable installed, controls and selects the drive(s) in use. Four (4) other 20 pin connectors are provided to enable data transfers to and from the connected disk drives.

An 8 position DIP switch, located at 8C on the HDCA board, is set at the factory to respond to port addresses 50H - 53H. Jumpers provided may enable/disable interrupt drivers. The gold edge connector allows the HDCA controller to be installed in any S-100 bus computer system.

Refer to the component layout at the back of this manual, and the various figures in Section 3 for the location of these hardware features.

#### 2.2. Discus Systems

The HDCA is a major component of Morrow Designs' series of Discus systems. The Discus M26, M20, and M10 are complete mass storage subsystems, each consisting of three main components:

1. A Winchester-type hard disk drive with a formatted storage capacity of 26 million bytes (M26), 20 million (M20), or 10 million bytes (M10), respectively,
2. A universal power supply, which provides the necessary DC voltages to power the drive, and
3. A Disk Jockey Winchester hard disk (HDCA) controller that establishes a data channel between the disk drive and any S-100 based computer.

The M26 drive and power supply are housed in a cabinet which has been designed for either table top or rack-mount operation. The only extra hardware required to mount the M26 drive in a standard 19 inch RETMA equipment rack is a pair of rack-mount slides which attach directly to the cabinet.

The 8 inch M10 and M20 systems are currently supplied in a desk top unit only.

The standard operating system software supplied with the Discus systems is the popular CP/M version 2.2, but the HDCA is also configured to run Micronix, Morrow Designs' multi-user Unix-like operating system.

## System Overview

### 2.3. Software Features

The HDCA software is configured so that all that is necessary to get the system running is a central processing unit (CPU), memory board(s) and a terminal. All the "tools" needed to customize the system software are also included. Each Morrow Designs system diskette (included with the controller) includes:

- \* CP/M 2.2 operating system
- \* Source to CP/M BIOS
- \* CP/M transient commands
- \* Disk formatting program (with source file)

The HDCA is configured for operation with CP/M Version 2.2 and Micronix, a multi-user operating system developed by Morrow Designs. Users running Micronix, version 1.3 or greater, require Revision 4a of the HDCA controller. Earlier revisions of this board may be upgraded by requesting and following the procedures described on ECN #1 for the HDCA-3, dated 4-8-82.

The user communicates with the Discus system on any of three levels:

1. Through CP/M, in a standard environment, or
2. Through low level software drivers (included on the disk), in a non-standard environment, or
3. Through direct commands to the HDCA controller, in custom applications.

The protocol for using the system through CP/M are covered in detail by the CP/M manual furnished under separate cover, and in Appendix B of this manual. The details of using the system through the low level software drivers are covered in Section 7 and Appendix B of this manual. Direct commands to the HDCA are explained in Section 8, and general information for operation in a non-standard environment is discussed in Section 6.

The HDCA disk controller is supplied with software making its integration into an existing CP/M system easy. The board is supplied with cables, software and a diskette. If you have a Morrow Designs system or a Morrow Designs Floppy disk controller, the diskette supplied you will boot up directly with all the software configured for the hard disk system you have purchased. If you are using another brand of floppy disk controller, Morrow Designs supplies a diskette with assembly language hard disk driver routines which can be incorporated into your CP/M CBIOS. For details on this refer to Section 6, Operation in a Non-Standard Environment.

## System Overview

For those users who are trying to incorporate the HDCA system into a non-Morrow Designs CP/M 2.2 system, and are not familiar with software customization, Morrow Designs supplies an installable IBIOS package for a nominal fee called "Install". This program will automatically install the Morrow software into your CP/M 2.2 CBIOS.

## Installing the HDCA - Standard Environment

### 3. INSTALLING THE HDCA - STANDARD ENVIRONMENT

#### 3.1. Installation Overview

The HDCA has been configured at the factory to work in the following operating environment:

- \* CPU does its own power-on jump.
- \* Board is set for interrupt drivers.
- \* Controller is set to respond to port addresses 50H - 53H.
- \* Controller anticipates connection to one drive.

**NOTE:** Information concerning multiple drive connection is included in this section. If your system does not meet the other requirements listed, please refer to Section 6, OPERATION IN A NON-STANDARD ENVIRONMENT.

#### 3.2. Installation Requirements

Installation of the HDCA controller board requires the following:

1. Setting the 8-position DIP switch.
2. Installing jumper options.
3. Seating the board in the system's S-100 bus slot.
4. Connecting the cable from the HDCA to the disk drive.
5. Connecting cables to additional drives.
6. Unlocking the drive head(s).

Each step is covered separately within this section.

**WARNING:** NEVER INSERT OR REMOVE A BOARD WITH SYSTEM POWER ON!  
THIS WILL DAMAGE THE BOARD.

#### 3.3. Examine the Board

Before installing the HDCA, examine the board for cracked, broken or missing components. If shipping damage exists, note the condition on the waybill, notify the carrier, and check the Warranty Return Procedure at the front of this manual for instructions on returning the board to Morrow Designs.

# Installing the HDCA - Standard Environment

## 3.4. Setting the Switch

Each HDCA controller board contains one (1) 8-position DIP switch, labeled SW8C and positioned at board location 8C.

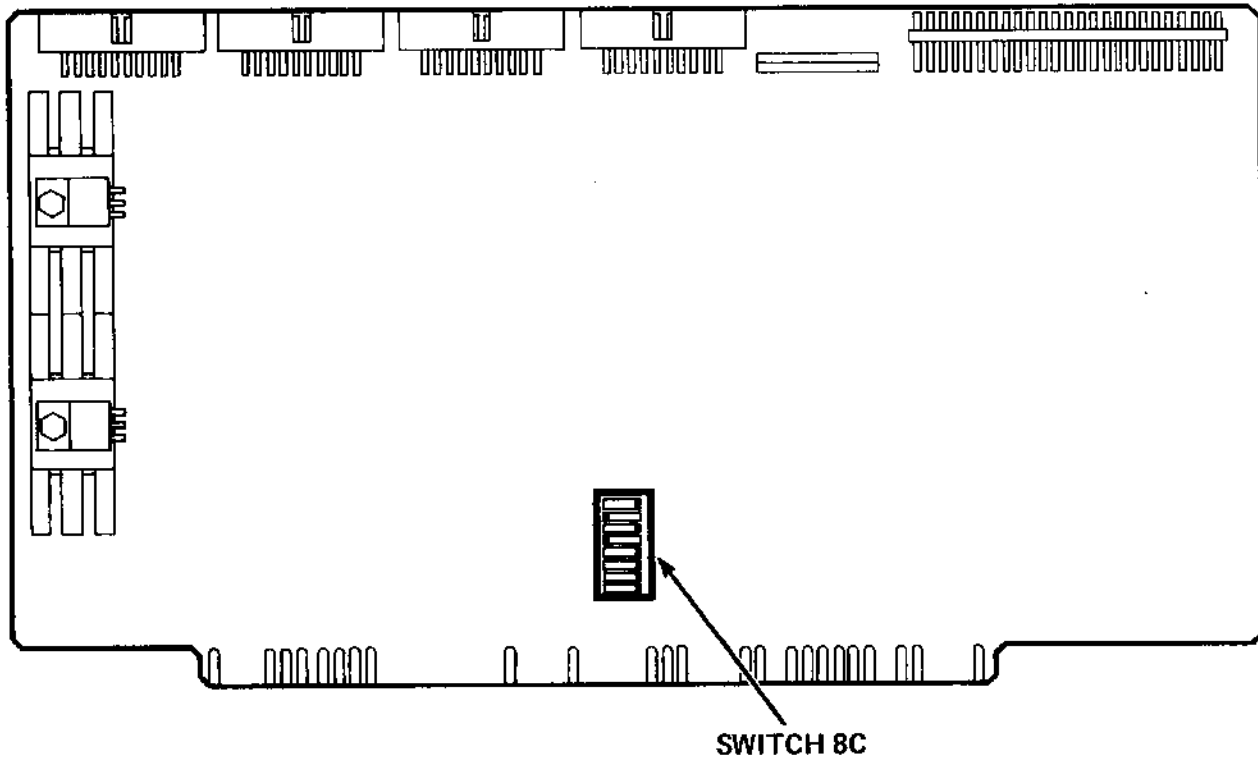


Figure 3-1: HDCA DIP Switch Location

## Installing the HDCA - Standard Environment

As shipped from the factory, the HDCA controller is set to respond to port addresses 50H - 53H. All Morrow Designs software assumes the HDCA controller to be located at these port addresses. (See Appendix C for other available addresses.) Switch at location 8C is used for port addressing and is set as illustrated in Figure 3-2.

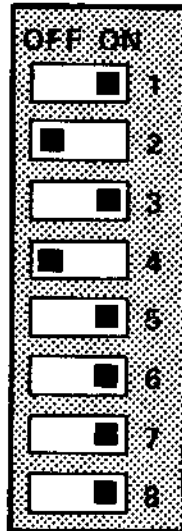


Figure 3-2: Factory Switch Settings (Port 50H)

### 3.5. Factory Set Jumper Settings

In area 1D of the HDCA controller (as viewed in Figure 3-3) there are two jumper pads marked A and B. As shipped from the factory these pads are connected to the pad marked VI0. These jumpers are installed to test the interrupt circuitry on the board. Most CP/M based systems do not use interrupts, and these jumpers will have no effect if left installed (or removed for that matter). Users whose systems use interrupts should refer to Section 6, Operation in a Non-Standard Environment, for details on these jumpers.

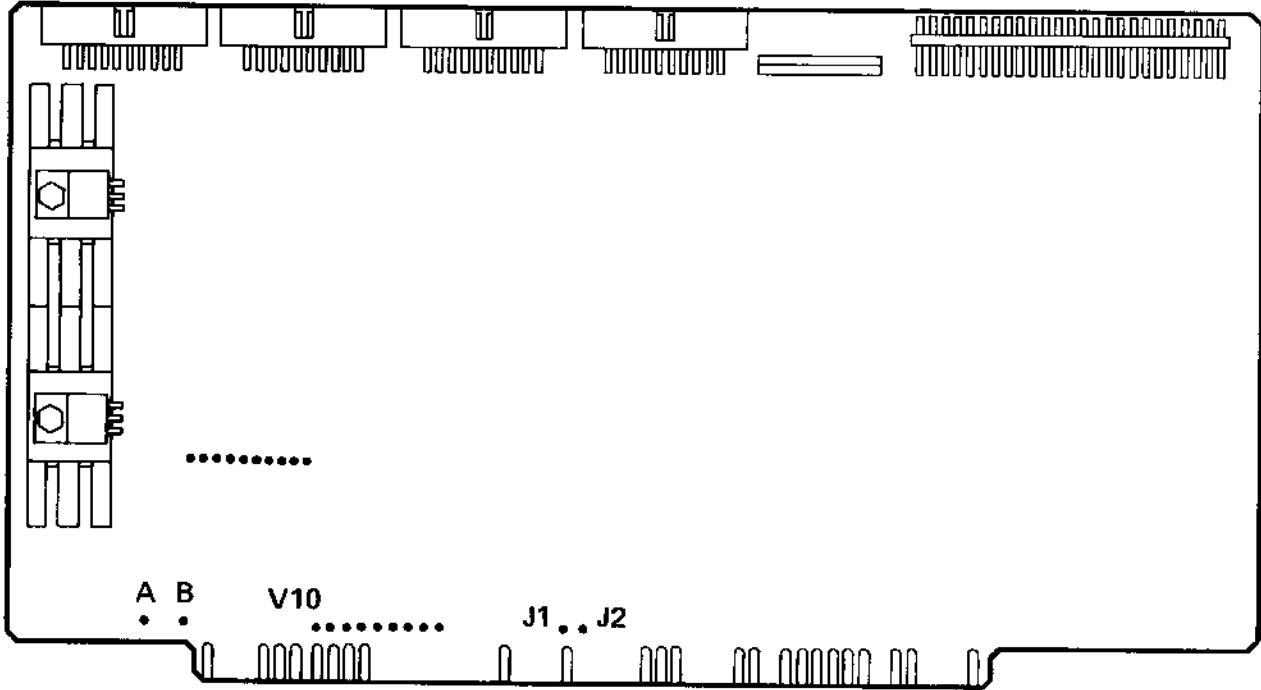


Figure 3-3: HDCA Jumper Locations

### 3.6. Seek Complete Jumpers

The HDCA controller board can communicate with up to four (4) disk drives. These disk drives report status back to the HDCA controller via the 50 pin and 20 pin flat cables. The 20 pin flat cable reports back the SEEK COMPLETE status from the drive, indicating the drive heads have settled on a track on which data may be read or written.

Since four drives are possible, four different SEEK COMPLETE lines return to the controller from these drives. It is necessary for all drives to have completed a seek before a further command can be issued. For this reason, SEEK COMPLETE jumpers must be installed in the non-active drive connectors on the HDCA board. As shipped from the factory, the HDCA board will have three shunts installed for this purpose. They are between pins 7 and 8 of connectors P3, P4 and P5. These jumpers must be in place for the the system to operate correctly. Refer to Figure 3-3 above for the location of these jumpers.



## Installing the HDCA - Standard Environment

### 3.7. Seating (Inserting) the HDCA Board

The HDCA controller board plugs into an S-100 bus slot in a system with an 8080, 8085 or Z-80 (1.75MHz to 4MHz) CPU. Grasping the board firmly with both hands, place the bottom of the board (the side with the gold S-100 edge connector) in the system's S-100 bus slot. You should be facing the component side of the board (see Figure 3-4). Gently rock the board from side to side to seat the HDCA securely in the S-100 slot. Push firmly on the top of the board to ensure that it is properly seated.

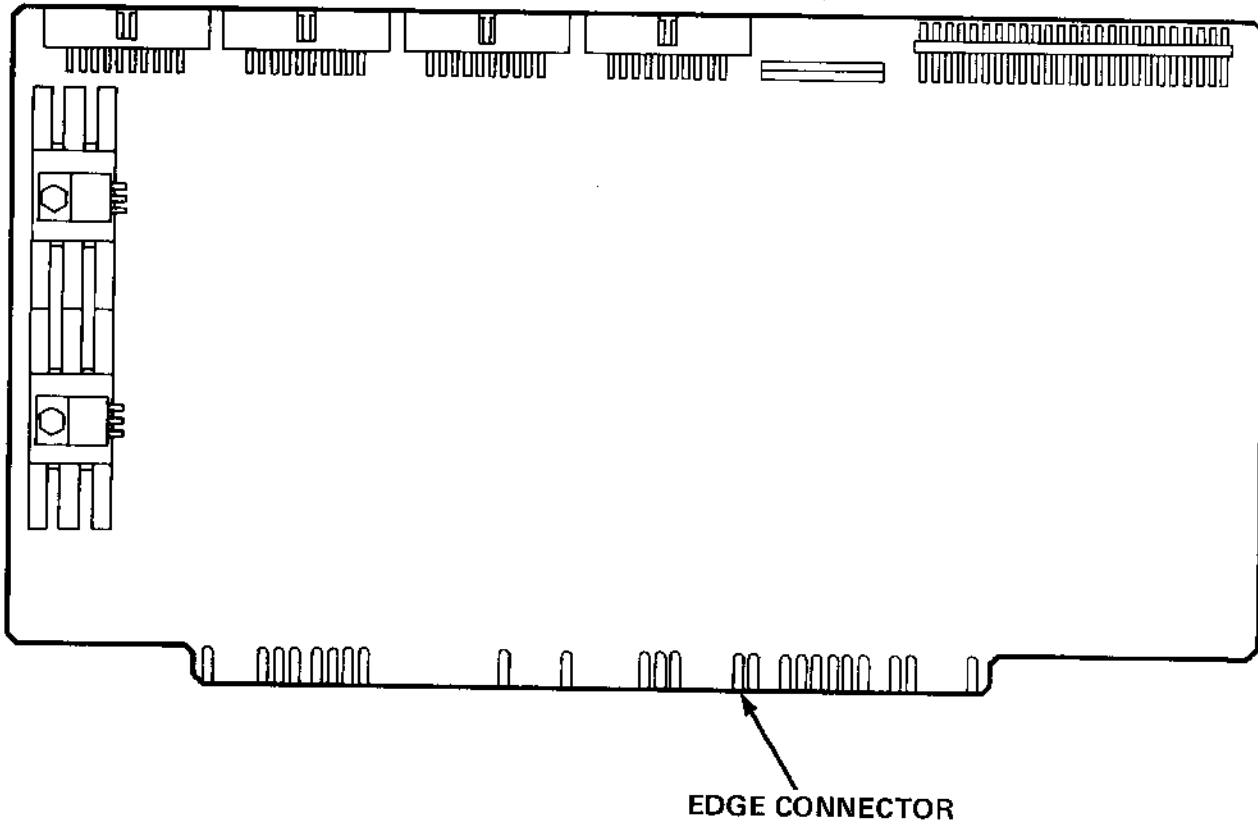


Figure 3-4: Seating the HDCA

### 3.8. Power-Up

Apply power to the system and observe the HDCA closely to be sure that no components are becoming excessively hot. Power the system down.

**NOTE:** If you see or smell smoke, immediately turn the system power OFF. Refer to the Warranty Return Procedure at the beginning of this manual for instructions on returning the board to Morrow Designs.

# Installing the HDCA - Standard Environment

## 3.9. Cable Connection

The HDCA controller is connected to an M10, M20 or M26 system using two 6 foot flat cables. These cables are supplied with your system. The 50 pin flat cable (the wider of the two cables) is the control cable responsible for controlling and selecting drives. The control cable may be daisy chained for multiple drive configurations (see Section 3.11, Multiple Drive Connections). The narrower 20 pin cable is responsible for transferring data to and from the disk drive during data transfers.

The 50 pin cable should be connected with the RED stripe to the right as viewed in Figure 3-5.

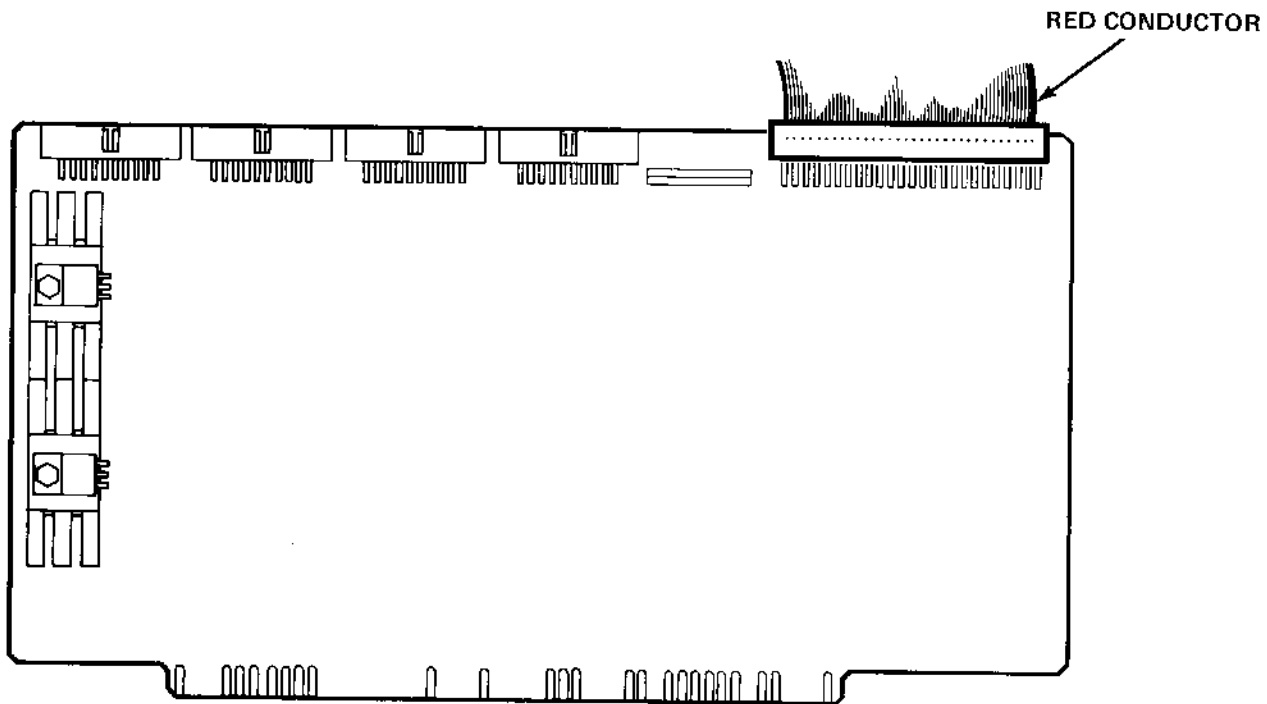


Figure 3-5: 50-Pin Cable Connection (P1)

## Installing the HDCA - Standard Environment

The 20 pin cable should be connected to the 20 pin header adjacent to the 50 pin connector with red stripe to the right as viewed in Figure 3-6.

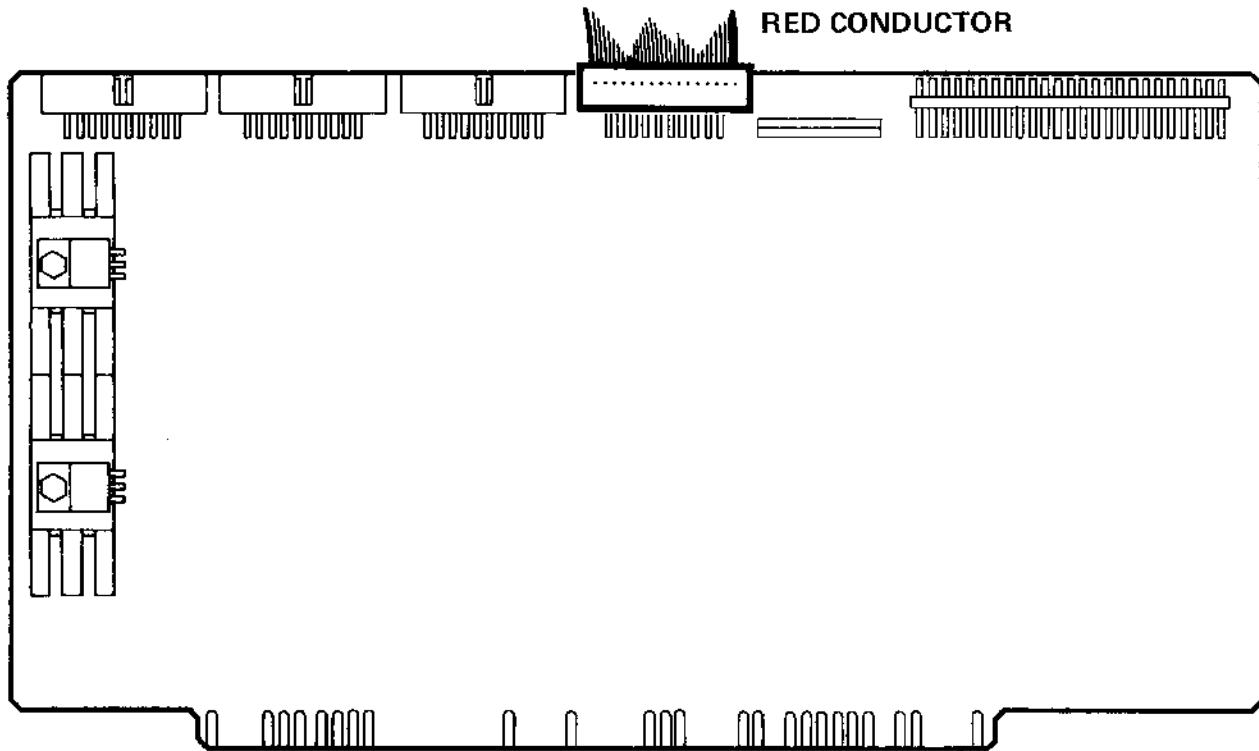


Figure 3-6: 20-Pin Cable Connection (P2 -P5)

## Installing the HDCA - Standard Environment

These cables must come out over the top of the HDCA controller board, and out the back of the computer. The other ends of the cables should be connected to the disk drives with the cables DOWN. Make sure that the cables are not twisted.

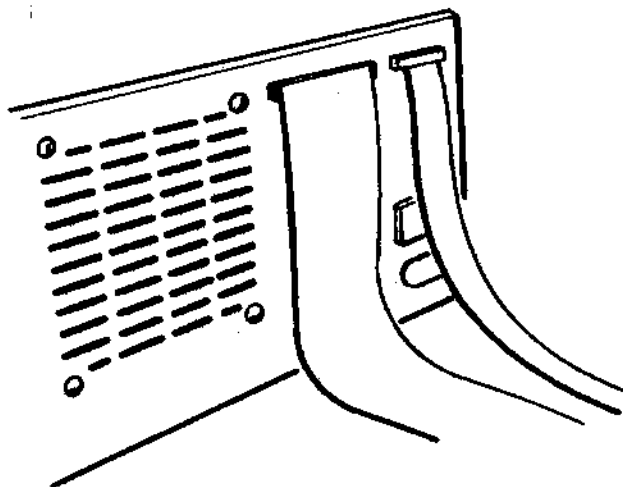


Figure 3-7: Cable Connection

### 3.10. Disk Drive Connection

**NOTE:** Before attempting to use any Morrow Designs software for the HDCA systems, be sure that the heads of the drives are unlocked. In the case of the M26, the spindle lock **MUST** be removed before power is applied (see Figure 3-8). After power is applied, the head (actuator) lock may be removed (see Figure 3-8). The M10 and M20 drives should be powered up and then the heads unlocked. Also note that the M26 drive takes as long as 2 1/2 minutes to become ready upon power-up. The M10/M20 drives require (on the average) 45 seconds to become ready.

Procedures for unlocking the disk drive heads and spindle lock (M26) follow.

3.10.1. M26 Drive Connection

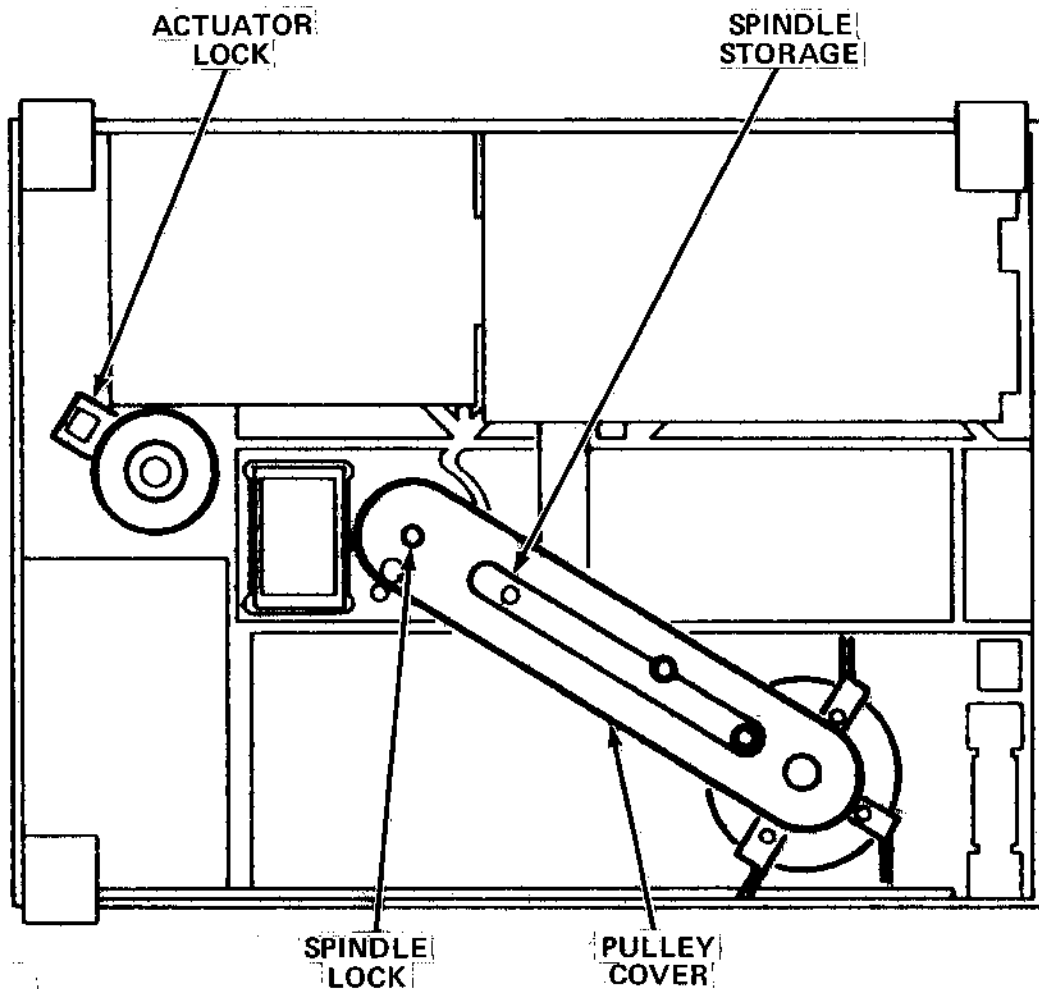


Figure 3-8: M26 Disk Drive (Spindle Lock and Actuator Lock)

**NOTE:** The procedure outlined below must be followed carefully to ensure correct operation of the drive. Make sure that the power is OFF when removing the spindle locking screw, and ON when removing the actuator lock. Failure to follow these instructions may result in media damage and/or extensive damage to the drive mechanism.

1. Using a screwdriver, remove the two 6-32 x 5/16" cover screws on the pulley cover.
2. Remove the cover, then unscrew the 4-40 x 9/16" spindle locking screw in the direction indicated by the yellow label in the drive pulley area (refer to Figure 3-8). This screw **MUST** be removed **BEFORE** applying power to the drive.

## Installing the HDCA - Standard Environment

3. Place the external star washer and the spindle screw in the designated storage slot (again, refer to Figure 3-8). Replace the pulley cover.
4. Power up the drive.
5. Allow the drive sufficient time to attain full speed operation (this may require up to 2.5 minutes after power-up).
6. With disk drive power ON, remove the "U-shaped" actuator lock (see Figure 3-8). Retain this lock for re-installation prior to transporting the drive.

### 3.10.2. M10 and M20

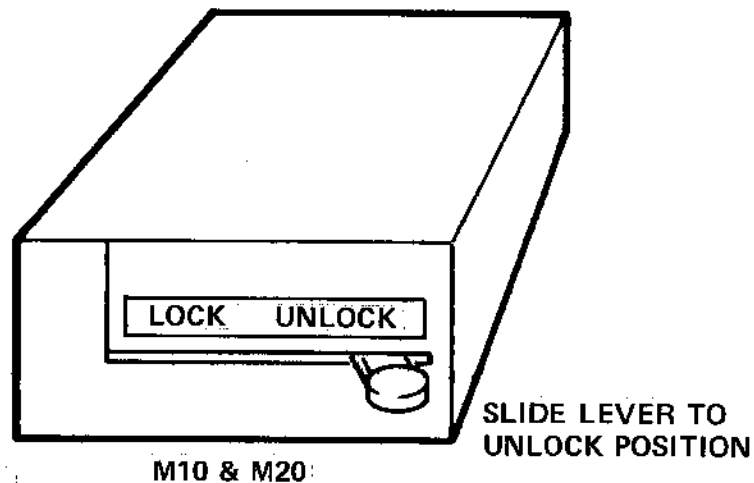


Figure 3-9: M10/M20 Disk Drive (Location of Head Lock)

1. Locate the disk drive head lock, as illustrated in Figure 3-9.
2. Lift up, and slide the lever from the LOCK position to the UNLOCK position.

# Installing the HDCA - Standard Environment

## 3.11. Multiple Drive Connection

More than one hard disk (of the same type) may be connected to the HDCA board for a total of up to four (4) drives per controller. Drives are "daisy chained" on one 50 pin flat cable; each has a separate data cable (20 pin flat cable). The last drive in the daisy chain must be terminated, while all others must have their termination pack removed.

Procedures for removal of the termination pack(s) are covered in the disk drive installation guide provided under separate cover. The following sections, however, provide an overview of these procedures.

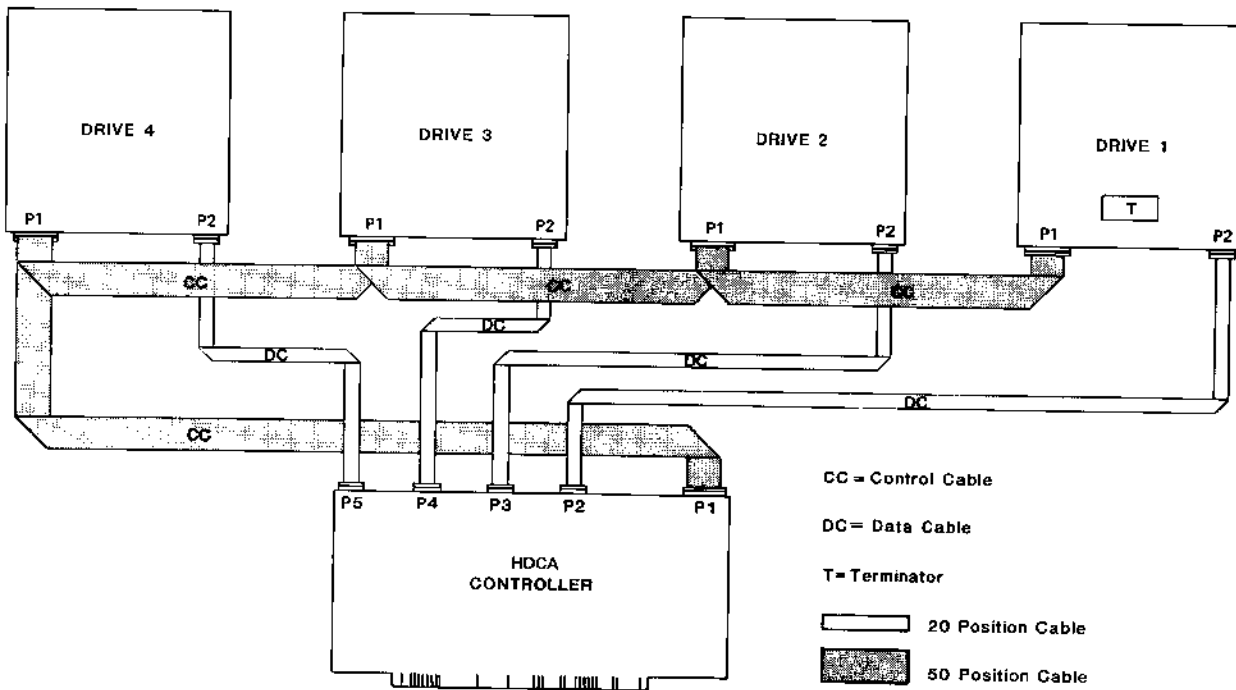


Figure 3-10: Multiple Drive Connection

## Installing the HDCA - Standard Environment

### 3.11.1. M26 Daisy Chaining

The termination resistor pack location on the M26 drive is at board location 2H on the main signal interface board. The drive select jumpers for the M26 are located on the same board at location 1H (next to the 50 pin card edge connector). Only one jumper should be installed in the drive select area. If the drive is to be drive 1, the DS1 jumper is installed. If the drive is to be installed as drive 2, then the DS2 jumper must be installed, etc.

### 3.11.2. M10/M20 Daisy Chaining

The M10/M20 Fujitsu drive resistor pack is located at board area 7A near the 50 pin card edge connector. Only the last drive in the daisy chain should have a resistor terminator installed. The drive select for these drives is controlled by a DIP switch (SW3) located at 6B. For the drive to respond as drive 1, Paddle 8 should be on and Paddles 7, 6 and 5 should be off. If the drive should respond as drive 2, Paddle 7 should be on and Paddles 8, 6 and 5 should be off.

**NOTE:** Only one paddle may be on for correct drive selection. Refer to page 3-9 of the Fujitsu M2301B/M2302B Microdisk Drives CE Manual for more detailed information.

### 3.11.3. HDCA Modifications and Connections

The HDCA is shipped from the factory configured for connection to one drive. The separate 20 pin data headers (P3, P4, P5) are supplied for multiple drive environments, but the data driver chips have NOT been installed. These chips may be obtained from the Morrow Designs Customer Service department for a nominal charge. Revision 3 and earlier models of the HDCA controller use 8 pin IC parts 9612 and 9613 for buffering the data lines. The HDCA Revision 4 and later models of the HDCA controller use 8 pin IC parts 9638 and 9637 for buffering the data lines. When ordering, please specify the type of controller you have (the revision is clearly marked on the component side of the PC board). Two 8 pin chips must be installed for each drive added to the controller. The location of these chips are illustrated in Table 3-1.

Table 3-1: ICs Required for Multiple Drive Connection

Drive #	HDCA 20 pin Connector	ICs installed
1	P2	5AA, 5AB
2	P3	4AA, 4AB
3	P4	2AA, 2AB
4	P5	1AA, 1AB



## Installing the HDCA - Standard Environment

**NOTE:** Parts for HDCA Revision 4 and higher (later revisions) require 9638 ICs at the AA (Top) locations. The bottom locations (AB positions) require that a 9637 IC is installed.

HDCA Revision 3 and earlier boards require 9612 ICs at the AA (Top) location. The bottom locations in the HDCA Revision 3 and earlier revisions require that the bottom locations (AB positions) have a 9613 IC installed (see Figure 3-11).

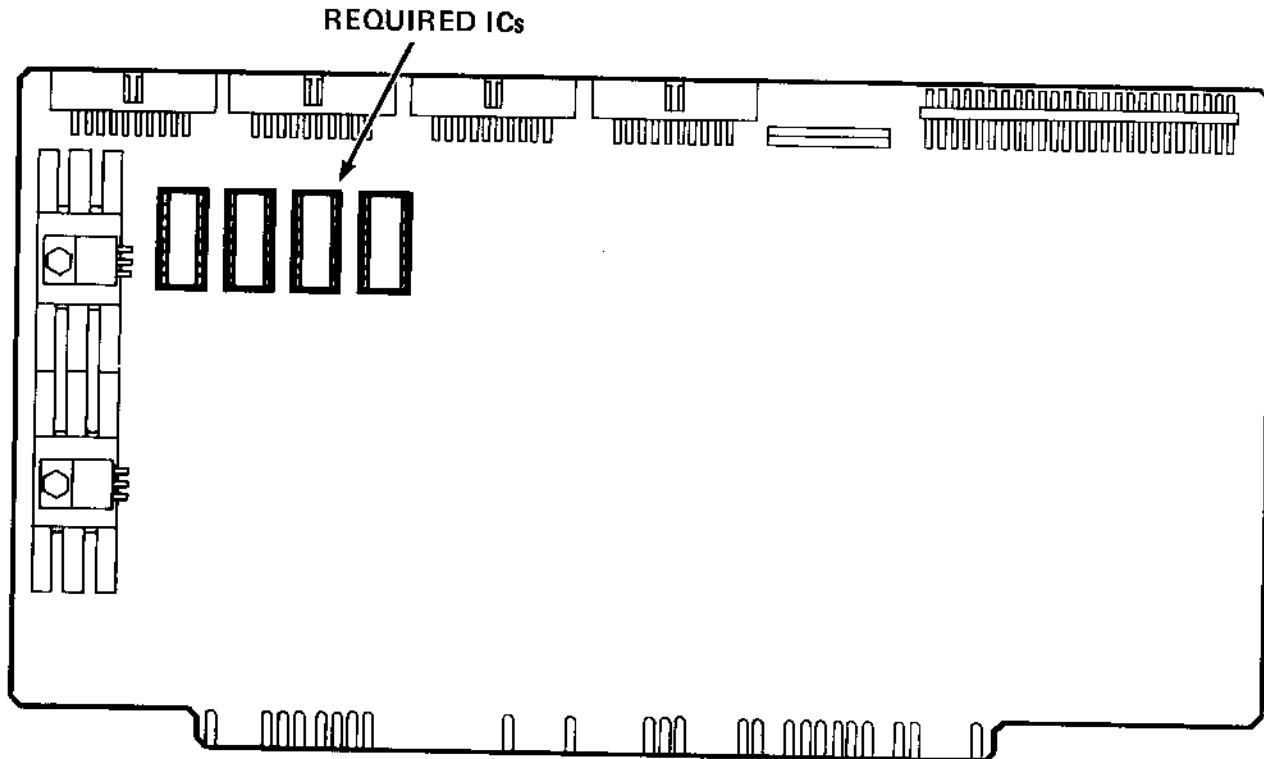


Figure 3-11: HDCA Controller Board (Location for Required ICs)

#### 4. SYSTEM START-UP AND POWER DOWN

##### 4.1. Standard Environment - Overview

The standard environment for an HDCA assumes the user is already using a Morrow Designs floppy disk system (DJ 2D/B or DJ/DMA) with console I/O through either the DJ boards, or a Morrow Designs MultiI/O board. It also assumes the HDCA will be addressed at I/O ports 50H-53H (see Figure 3-2).

If your system meets the above criteria, Morrow Designs provides an 8 inch diskette which will "boot up" in your system, and the hard disk will be directly accessible. If, however, your system is made up of various manufacturers' components, you should refer to Section 6, Operation in a Non-Standard Environment.

##### 4.2. The System Bootstrap

The HDCA controller contains all of the logic necessary to control the drive, and to transfer data to and from the disk. The HDCA appears to the system as four I/O devices, and has no memory in the address space of the CPU. Therefore, some other element of the system has the responsibility for the initial load of the operating system and/or utility software from the disk.

This initial loading operation is referred to as the bootstrap operation. The purpose of this section is to discuss several ways that the bootstrap operation can be implemented.

##### 4.2.1. Implementing the Bootstrap

The code necessary to perform a bootstrap is roughly 100 bytes long. The least expensive, but most troublesome method of bringing up the system is to enter this code into memory and execute it. Fortunately, Morrow Designs can supply several alternatives:

1. The program is supplied on a CP/M compatible floppy diskette as BOOTH.D.COM (a CP/M command file). In this case, the program has a starting address of 100H (Hex), in accordance with the CP/M operating system requirements.
2. When CP/M 2.2 is supplied with a Discus I or Discus 2D system, the CP/M CBIOS software contains a module that anticipates connection of a Discus M26, M20 or M10 system. The computer treats these attached systems as drives E, F, and G (M26 and M20) or drives E and F (M10). Drives A, B, C, and D are reserved for floppy disk drives.
3. The MPZ80 EPROM in a Decision 1 system has the program built-in.

## System Start-Up/Power Down

### 4.2.2. Bootstrap Software

A sample program to load the disk system software into memory is provided in Appendix B.

### 4.3. Disk Initialization and Formatting

Once you have followed the procedures outlined in Section 3, the Discus system is ready to be powered up.

1. Apply power to the computer.
2. Apply power to the drives observing the note in Section 3.10.
3. Reset the computer.
4. Bring up CP/M 2.2 on your system. The system diskette included with your system contains a formatting and diagnostic program called FORMATHD.COM.
5. Although all drives shipped from Morrow Designs have passed a lengthy diagnostic and are formatted, the user is advised to run the format program to ensure that the controller, drive and the cables are functioning correctly.
6. From the CP/M prompt type:       FORMATHD   [CR]
7. The program will prompt you with information and instructions for formatting the drive.
8. Choose the format data and header option for the drive you have purchased. Users with an M10 will be asked for the manufacturer of their drive. M10s are currently manufactured by the Fujitsu Corporation.
9. After the options have been selected, the disk drive heads should restore to track 0 if they are not already positioned at track 0 (there should be a motor-like noise when this occurs). From this point, the heads will begin stepping a cylinder at a time, printing a dot on the console for every track formatted.
10. When the program is finished, return to CP/M.
11. If the heads do not move, or the program reports errors, refer to the Troubleshooting Guide in Appendix A.

### 4.4. Running CP/M on the Discus System

There are two separate procedures for bringing the system up under CP/M. The procedure used is dependent upon whether you are currently using a Morrow Designs system or Morrow Designs floppy subsystem, or running on a non-Morrow Designs floppy CP/M system.

## System Start-Up/Power Down

The following steps outline the procedure for installing the CP/M operating system on a Morrow Designs hard disk Discus system:

1. Boot the floppy disk.
2. View the directory of disk E, F, and G.
3. Execute MOVCPMnn \* \* to install the largest system possible.
4. Execute SYSGEN to place the system on the first drive addressed by the HDCA controller (from memory).
5. Copy the system files from the floppy to the hard disk, using the CP/M utility, PIP.
6. Boot the hard disk, noting the address changes in the header (i.e. A, B, C).

### 4.4.1. Morrow Designs System

In a Morrow Designs system, it is only necessary to install the system diskette supplied with your Discus system and perform a cold boot. You must be sure that the diskette is configured for your operating environment (DJ2D/B or DJ/DMA floppy controller, I/O through the DJ or Multi I/O board).

The operating environment is displayed as a header (on the console) when the system is booted.

Custom MOVCPM programs are included on the system diskette. The diskette label will include the program name MOVCPM; the size of the drive will be indicated next to the word MOVCPM. For example, if you are using an M26 disk drive, the label will display MOVCPM26. A sample system diskette label is provided below:

```
48K CP/M version 2.2, CBIOS revision E.3
MOVCPM; AB: DJDMA 8, CD: DJDMA 5, EFG: HDC3 M26
MOVCPM26; ABC: HDCA M26, DE: DJDMA 8, FG: DJDMA 5
Multio (9600) as console, Multio (1200) as list
(c) 1980 Digital Research, (c) 1982 Morrow Designs
```

The label also lists drive and controller designations for system boot, and these also appear on screen. Example:

System header at floppy disk boot:

```
Morrow Designs 48K CP/M 2.2 E.3
AB:DJDMA 8", CD:DJDMA 5 1/4", EFG:HDC3 M26
```

A>

The CP/M system should now be up and running, and the version will have the HDCA drivers for your Discus system resident in the

## System Start-Up/Power Down

CBIOS. Note that in the example above, drives E, F and G are hard disk drive surfaces (logical drives) controlled by the HDCA. By now typing:

```
DIR E:          [CR]
```

the contents of the hard disk system will be displayed. Since the drive has just been formatted, CP/M should respond "NO FILE". Typing:

```
DIR F:          [CR]
```

will cause the same response after the heads reposition themselves. The M10 drive is separated into two logical drives. Both the M20 and M26 are separated into 3 logical drives for CP/M. If you have an M20 or an M26, you may complete the initial check by typing:

```
DIR G:          [CR]
```

After successfully completing this simple test, you need to install the operating system on the hard disk (Section 4.4.2), and then copy (using the CP/M utility PIP) system files from the floppy to the hard disk. Upon completion, you will be ready to boot the hard disk.

### 4.4.2. Installing a System on the Hard Disk

For convenience, speed and versatility, the CP/M operating system should reside on the hard disk (as on a floppy disk). With the system diskette supplied from Morrow Designs, simply type:

```
MOVCPMnn * *          [CR]
```

where nn is the number that corresponds to the size of your hard drive (i.e. "10" if you have an M10 system, "20" if your system is an M20, "26" for M26 systems. Check the label on the boot diskette for the appropriate MOVCPM program. The asterisks (\* \*) tell MOVCPM to configure the largest system possible in your particular operating environment, in this case 64K.

An image of the system with the hard disk drivers (as drives A and B for M10, or A, B and C for M20/M26) is now resident in memory. At this point simply type:

```
SYSGEN              [CR]
```

When prompted, indicate the source is from memory and the destination is the "E" drive, or use the header to find the correct drive address; place the new system on the first drive addressed by the HDCA controller. Your hard disk will now have a copy of CP/M on the first cylinder of the drive.

## System Start-Up/Power Down

Now, copy over the files from the floppy to the hard disk using the CP/M utility PIP. Type:

```
PIP E:=A:*. * [V] [CR]
```

### 4.4.3. Cold Booting the Hard Disk (BOOTH.D.COM)

Morrow Designs supplies the BOOTH.D.COM (as well as a listing of this program in Appendix B of this manual) to allow the user to "Boot" from the Discus Hard Disk system. Simply type:

```
BOOTH.D [CR]
```

and the hard disk will be cold booted. The disk drive addresses will now change and are described on the boot header (message displayed on the screen). For example:

System header after SYSGEN and BOOTH.D:

```
Morrow Designs 64K CP/M 2.2 E.3  
ABC:HDCA M26, DE:DJDMA 8", FG:DJDMA 5"
```

```
A>
```

A "DIR" should display the system files copied from the floppy boot diskette.

### 4.4.4. Automatic Hard Disk Boot

BOOTH.D.COM is "burned into" (resident in) the CPU's monitor PROM for users wishing to boot up automatically from the hard disk. This requires resetting all paddles on Switch 16D of the MPZ80 CPU board to the ON position, except Paddle 6, which remains in the OFF position. Once these changes have been made, there is no need to use the boot diskette to start up the system. The user needs only to power up the computer and press the RESET button. The system will automatically boot up from the hard disk. Further details on this subject are described in Section 2 of the MPZ80 CPU Technical Manual and the Decision I installation guides.

## 4.5. Back-Up Procedures

With the advent of hard disk storage for the microcomputer world, users should approach the use of hard disk with more of a traditional view of the back-up procedures. The very fact that one device can store 26 million characters means that a potential problem will occur if the information becomes inaccessible or accidentally erased. People familiar with mini computer systems typically back-up all daily transactions (files which have been altered within a 24 hour period). In addition, total disk back-ups are usually performed on a weekly basis.

## System Start-Up/Power Down

A Winchester-type sealed disk drive, such as the Discus M26, M20 and M10, has many advantages and also some limitations. Large capacity, low maintenance and high reliability are the most obvious advantages. The major disadvantage is the fixed nature of the magnetic media. The media of a floppy disk drive can be removed and placed in a secure environment. This is not the case with the present generation of Winchester disk drives. The media is sealed inside the unit and cannot be removed. Thus, as long as the drive is connected to a computer, the data written on the magnetic surface of the disk is never really secure.

The present solution to this data security problem is to backup the data periodically on some storage device which has removable media. A floppy disk system, such as the Discus 2D or the Discus 2+2, is the most practical device for backing-up data stored on a hard disk drive. By using the facilities provided by CP/M, it is quite easy to transfer files from the hard disk to a floppy disk.

The CP/M PIP utility can be used to provide back-ups of the files on your hard disk drives onto floppy disks. Files may be PIPped onto 8 inch floppy disks routinely (daily for files that are crucial and updated frequently). The CP/M command to do this follows:

```
PIP A:=E:[FILENAME] [V]
```

which will PIP the file of given name from the hard disk (assumed to be the E drive for illustrative purposes) to a floppy (assumed to be the A drive for illustrative purposes). To restore the file back onto the hard disk, simply reverse the command:

```
PIP E:=A:[FILENAME] [V]
```

The [V] option will verify a good back-up copy is made and/or restored. Refer to Section 7 of the CP/M manual, Transient Commands, for a more detailed description of the CP/M PIP command.

Morrow Designs floppy subsystems can store 620 Kbytes single sided and 1.2 Mbytes double sided, so the user is advised to keep the size of each hard disk file within the back-up size limitations of the floppy disks. These back-up diskettes should be labeled as to the contents and last date of back-up for ease in restoring the system.

Another word of caution on back-ups is to ALWAYS back-up all the files on the hard disk system BEFORE physically moving the drives. Also, note that repair depots will generally not guarantee preservation of ANY data on the drive during routine maintenance and service. Whenever the product is to be serviced, the user should back-up all of the important and/or lengthy files.

It is also advisable to keep back-up diskettes for the back-up diskettes, should they accidentally be misplaced.

## System Start-Up/Power Down

**NOTE:** The HEADS of the drives (and the spindle on the M26) should be locked before any hard disk is moved. This will minimize possible damage to the hard disk media and loss of data. Refer to Section 3.10 for specific instructions.

### 4.6. System Shut Down

Circuitry modifications on Revision 4a of the HCDA controller protect the drive from being written to at power down. Users with an HCDA controller labeled Revision 4 or earlier, however, are advised to press the RESET button on the computer front panel before shutting off power to their system.



# Disk Data Organization

## 5. DISK DATA ORGANIZATION

To understand the significance of the disk utility software, it is necessary to say a few words about how data is organized on the disk. Figure 5-1 illustrates the concept of data organization.

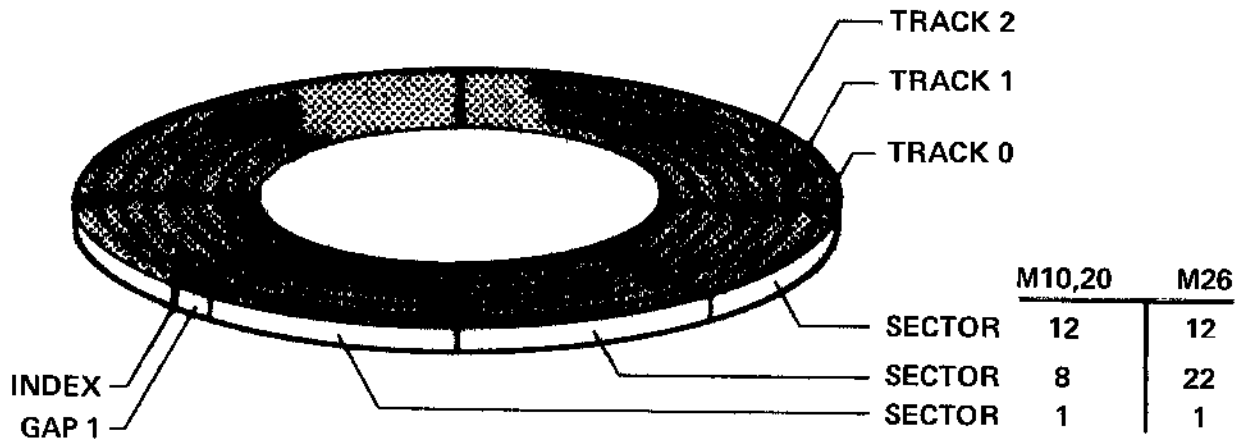


Figure 5-1: Disk Data Organization

Information on the disk is organized into concentric tracks. The number of tracks varies, depending on the hard disk drive you have ordered. Table 5-1 is a comparison of the Discus hard disk systems.

Table 5-1: Discus Systems Comparison

Discus System	# of Tracks	# of Platters	# of R/W Heads	# of Surfaces	Sectors/Track
M26	202	2	8	4	32
M20	244	4	8	8	21
M10	244	2	4	4	21

There are two rotating platters inside the M10 and M26 drives; the M20 has 4 platters. Both the lower and upper surfaces of these platters are used to store magnetic information. A read/write head floats over each surface.

## Disk Data Organization

The M26 contains eight heads, two for each surface. This decreases by half the amount of surface the stepper mode must be able to position the heads. When one head is at track 0 on the M26, its associated head for that platter is located exactly halfway in the middle of the surface.

M10 and M20 drives have only one head per surface. The M20 contains eight heads, and the M10 system contains four heads.

The disk read/write heads can be moved to any track by a series of step in or step out commands. A step in command moves the read/write heads one track toward the center of the disk. A step out command moves the heads one track away from the center of the disk. The numbering of the tracks is arranged so that track zero is farthest from the center.

Once the read/write heads have been moved to the desired track, the rotation of the disk will move a circle of magnetic material beneath (or above) each of the heads. Within these eight circles (four circles on the M10), data is recorded in distinct regions called sectors. A sector is the smallest amount of information that can be separately read from or written to the disk, and each one contains 512 bytes of data.

Sector, description of

The HDCA controller uses 512 byte sectors. Each sector is uniquely identified on the drive by a SECTOR HEADER. The header field records the following information:

- . track,
- . head,
- . key,
- . sector number and
- . data security information

The header field precedes the data field of a sector. During read or write commands, this header is read before data transfers take place. In addition, a 2 byte number which is a CRC (Cyclical Redundancy Check) check word is generated by the controller whenever a write to either the header or data is performed. This number is appended to both the header and data fields. When the header or data is read back, this value is checked by the HDCA circuitry to ensure that the data has been read correctly.

The fact that the M26 is physically larger than the M10 and M20 drives means that it can support more sectors per track (cylinder). The M10/M20 drives can accommodate 21 sectors on a track; the M26 supports 32 sectors per track. The sector format for Morrow Designs software is identified in Table 5-2.



## 6. OPERATION IN A NON-STANDARD ENVIRONMENT

### 6.1. Installable BIOS (IBIOS)

Since Morrow Designs cannot supply a system diskette configured for your particular hardware environment, the Installable BIOS (IBIOS) package can be obtained to simplify patching the HDCA drivers into your CP/M system. The package locates the HDCA drivers in RAM above your CP/M system, and changes the jump table to reflect the presence of these drivers. This is done automatically, but requires the CP/M 2.2 system to have some free RAM (normally at the top of memory). Each time you cold boot the system, however, you must run the INSTALL program. If you wish to permanently incorporate the drivers into your CBIOS, refer to the CP/M documentation for instructions on customization.

NOTE: Details on the IBIOS program are contained on the IBIOS diskette in the .DOC files. The user may TYPE the INSTALL.DOC file for a more complete description of the IBIOS package.

Assuming you are running a 60K CP/M system, and you have formatted your hard disk drive, proceed as follows:

1. Put the IBIOS diskette in the floppy drive and type:

```
INSTALL HDCA.PRL -E F000      [CR]
```

2. Type:

```
DIR E:      [CR]
```

The contents of the hard disk system will be displayed. Since the drive has just been formatted, CP/M should respond "NO FILE".

3. Type:

```
DIR F:      [CR]
```

This will cause the same response ("NO FILE") after the heads reposition themselves. The M10 drive is separated into two logical drives.

4. Both the M20 and M26 are separated into 3 logical drives for CP/M. If you have an M20 or an M26, you may complete the initial check by typing:

```
DIR G:      [CR]
```

5. After successfully completing this simple test, you may begin using the hard disk to store files. A good idea at this time is to PIP all the files off the floppy onto each of the Discus logical drives to further test the system (see Section 4.5). This completes the initial CP/M checkout.

## Operation in a Non-Standard Environment

### 6.2. I/O Driver Overview

The operating system must communicate with the disk and at least one other I/O device. The additional device is assigned by CP/M as the SYSTEM CONSOLE, and generally consists of a video display screen and a keyboard. The SYSTEM CONSOLE allows the user and the operating system to communicate with each other. Depending on overall requirements, other I/O devices (i.e. a printer or a modem) may be connected to the system.

I/O devices vary greatly in their electrical and mechanical characteristics. To communicate with them, allowances must be made for these variances. A flexible operating system must remain aloof to the peculiarities of the devices with which it communicates; therefore, each time a new type of I/O device is connected to the system, a communications problem is created. With a Morrow Designs hard disk system, the problem is resolved by a software module called a "driver".

The driver must accept commands from the operating system and translate them into a form that the I/O device will accept. The I/O device is usually connected to the system through an interface cable. This interface translates the computer's logical signals into proper electrical signals that are recognized by the I/O device. Generally, this interface is a circuit board that plugs into the bus of the computer, and is connected to the device by a cable.

### 6.3. Creating an I/O Driver

In order to create a driver, the user must have detailed knowledge concerning:

1. how information is passed back and forth between the device and the interface,
2. how the interface and the computer communicate, and
3. how the operating system passes information back and forth to the driver.

Drivers vary a great deal in length and complexity; however, regardless of size or intricacy, the following rule applies:

DO NOT attempt to write a software driver if you do not possess a clear and detailed understanding of how both the device and its interface function, as well as assembly language programming experience.

## Operation in a Non-Standard Environment

The creation of an I/O driver requires considerable skill in assembly language programming. Faulty or incomplete knowledge of the device or its interface usually results in a program that doesn't work or, worse yet, only partially works. A great deal of time can be wasted and equipment may be damaged in trying to make a bad driver function. If there is any question in the user's mind about how a driver should be written, he/she should consult with the dealer/personnel at the computer store where the system was purchased. In many cases, the driver software will be a stock item and can usually be installed before the system leaves the store.

### 6.4. The I/O Driver

This section of the manual discusses how information is passed back and forth between the operating system and the driver. Also presented are several sample drivers for the system console which interface to some of the more common S-100 computers.

The CP/M operating system maintains a table of jump instructions in high memory. Each entry in this table points to an I/O driver that the operating system may have occasion to use. There are 17 entries in this table; six of these are of concern to the user, and the 11 others point to disk drivers. The table, as it appears in memory, is presented in Table 6-1. Entries marked with an asterisk (\*) are of interest to the user, and will be discussed in detail in the following subsections of this manual.

Table 6-1: CP/M Jump Instructions

JMP BOOT	;arrive here from a cold start load
JMP WBOOT	;arrive here for warm start
*JMP CONST	;check for console character ready
*JMP CONIN	;read console character in
*JMP CONOUT	;write console character out
*JMP LIST	;write listing character out
*JMP PUNCH	;write character to punch device
*JMP READER	;read the reader device
JMP HOME	;move to track 0 on selected disk
JMP SELDSK	;select disk drive
JMP SETTRK	;set track number
JMP SETSEC	;set sector number
JMP SETDMA	;set data transfer address
JMP READ	;read selected sector
JMP WRITE	;write selected sector
JMP LISTST	;return list status
JMP SECTAN	;sector translate subroutine

At location 0 in memory, CP/M maintains a jump instruction to the second entry of this table - JMP TABLE+3 (to WARM BOOT). The value in location 2 is the page number on which the table starts. In almost all operating systems, the Jump Table starts at the beginning of a page, which means the value of location 1 will be three (3).

## Operation in a Non-Standard Environment

The Discus M26, M20 and M10 disk systems are shipped with two types of I/O configurations:

1. Preinstalled drivers which interface to a:
  - a. DISCUS 2D floppy disk controller
  - b. SWITCHBOARD I/O controller board
  - c. SOL computer
  - d. EXIDY computer
2. No installed drivers; each entry in the table marked with an asterisk is a "jump-to-self" instruction.

### 6.5. I/O Driver Specifications

All simple character I/O operations are assumed to be performed in ASCII, upper and lower case, with high order (parity bit) set to zero. An end-of-file condition for an input device is given by an ASCII control-z (1A hex). Peripheral devices are seen by CP/M as "logical" devices, and are listed in Table 6-2. The assignment of peripheral devices to physical devices occurs within the CBIOS (see the CP/M documentation).

In order to operate, the system needs only the CONST, CONIN, and CONOUT drivers; LIST, PUNCH, and READER may be used by application programs, but not by the CCP (see Tables 6-2 and 6-3). The

LISTST entry is currently used only by DESPOOL, and may be a simple RET instruction.

Table 6-2: CP/M Devices

CONSOLE	The principal interactive console which communicates with the operator, accessed through CONST, CONIN, and CONOUT. Typically, the CONSOLE is a device such as a CRT or teletype.
LIST	The principal listing device, if connected to the system, is usually a hard copy device such as a printer or teletype.
PUNCH	The principal tape reading device, such as an optical reader or teletype.

Table 6-3: CP/M Drivers

CONST	Sample the status of the currently assigned console device. Return OFFH in register A if a character is ready to read, and 00H in register A if no console characters are ready.
CONIN	Read the next console character into register A, and set the parity bit (high order bit) to zero. If no console character is ready, wait until a character is typed before returning.
CONOUT	Send the character from register C to the console output device. The character is in ASCII, with high order parity bit set to zero. You may want to include a time out on a line feed or carriage return if your console device requires some time interval at the end of the line (such as a TI Silent 700 terminal). You can, if you wish, filter out control characters which cause your console device to react in a strange way (i.e. a control-z causes a Lear Seigler terminal to clear the screen).
LIST	Send the character from register C to the currently assigned listing device. The character is in ASCII with zero parity.
PUNCH	Send the character from register C to the currently assigned punch device. The character is in ASCII with zero parity.
READER	Read the next character from the currently assigned reader device into register A. The character is ASCII, with with zero parity (high order bit must be zero). An end-of-file condition is reported by returning an ASCII control-z (1AH).

Note that a single peripheral can be assigned as the LIST, PUNCH, and READER device simultaneously. If no peripheral device is assigned as the LIST, PUNCH, or READER device, the user-supplied driver may give an appropriate error message so that the system does not "hang" if the device is accessed by PIP or some other user program. Alternately, the PUNCH and LIST drivers can just simply return, and the READER driver can return with a 1A (hex) in Register A to indicate an immediate end-of-file.

### 6.6. Installing I/O Drivers

In systems without installed drivers, a 512 byte section of memory is reserved for user-supplied drivers. This memory section immediately follows the jump table location.

In some instances it may not be necessary to install all six of the drivers listed in Table 6-3. For instance, if there is no



## Operation in a Non-Standard Environment

PUNCH device in the system, the PUNCH device driver would consist of a single RET(urn) instruction. Refer to the following procedure to install the necessary drivers:

1. Carefully study the Alteration Guide section of the CP/M 2.2 documentation.
2. Code and assemble the drivers individually.
3. Enter the assembled code into the reserved 512 byte memory area immediately following the jump table.
4. Alter the "jump-to-self" instructions to point to the newly installed drivers.
5. Test and debug the drivers.
6. When a driver functions correctly, execute the SYSGEN command (discussed in the CP/M documentation). SYSGEN automatically alters the operating system so the driver is loaded back into memory when the system boots.
7. Save a copy of the source code for the drivers on a disk file. If the operating system is made larger (or smaller), the drivers can be re-installed with a minimum of effort.

### 6.7. Interrupts

In the lower left hand corner of the board are two jumper holes labeled "A" and "B" (see Figure 3-3). "A" is driven by the OPDONE bit of the main status register and "B" is driven by the SDONE bit of the auxiliary status register (see Section 8). These two jumper holes can be connected to any of nine interrupt lines of the S-100 bus: VIO - VI7 or PINT. These signals have been brought in from the bus to jumper holes to the right of the "A" and "B" holes and just above the edge connector at the bottom of the board. These jumper holes are labeled on the silk screened legend. "A" and "B" can be connected to the same line or, if OPDONE and SDONE should have different priority levels, they can be connected to different interrupt lines. When one or more of these signals are connected to the interrupt lines, interrupts can be generated by the board when transfer operations complete and/or when head motion at the disk completes. Throughput to and from the disk can be greatly enhanced by using these two signals properly.

On Revision 3 of the HDCA, SDONE (see Section 8.5.3) interrupts were reset by issuing a new command to the controller. The current HDCA version now requires software to input from port base+2 (52H) to reset this interrupt. The OPDONE interrupt is still reset by issuing a new command to the controller, however.

## Operation in a Non-Standard Environment

In terms of performance, especially for overlapped seeks and commands, we feel it is much more desirable to be able to reset the SDONE interrupt independently from the OPDONE interrupt.

## Software Customization

### 7. SOFTWARE CUSTOMIZATION

#### 7.1. Overview

Transferring a sector of disk data between memory and the disk involves the following steps. Each step corresponds to a sub-routine call to the disk utility software, with the exception of error checking:

1. Specify the track number over which the read/write heads should be positioned during subsequent data transfers between the disk and memory. There are a total of 202 tracks, numbered 0 through 201 on the M26. There are a total of 244 tracks, numbered 0 through 243 on the M10/M20.
2. Check for error conditions.
3. Specify a head to be selected during subsequent read or write operations. There are a total of eight heads, numbered 0 through 7 on the M26/M20. The M10 has a total of four heads which are numbered 0 through 3.
4. Specify a sector number that will be involved in subsequent data transfers between the disk and memory. There are a total of 32 sectors, numbered 1 through 32 for the M26. The M10/M20 has a total of 21 sectors, numbered 1 through 21.
5. Check for error conditions.
6. Specify the starting memory address of the block of data to be transferred to or from the disk.
7. Perform the read or write operation.
8. Check for error conditions.

#### 7.2. Low Level Drivers

Some applications require a more intimate degree of control over the disk than is possible through the operating system. On the other hand, it may not be desirable or necessary to communicate directly with the controller through hardware registers. A set of low level software drivers is included with the system to meet the foregoing needs. These drivers can be loaded directly from the system tracks through an option in the bootstrap loader (see Appendix B). The source code has also been provided in a file called HDFIRM.ASM. This file can be accessed through the operating system if the user needs to modify it. The code has been assembled to run starting at F800 Hex. A change in the "origin" statement of the file will allow the code to be assembled for alternate starting addresses.

### 7.3. HDCA Jump Table

There are 11 standard I/O subroutines for the HDCA. These routines are provided in Appendix B. To use these low level drivers (subroutines), the user should branch to the appropriate address in a jump table in the first few words of the driver software. Since each subroutine ends with a RET(urn) instruction, a CALL instruction should be used to branch to it.

The jump table contains jump instructions that point to the true address of a utility routine within the software. Having a jump table allows the individual routines to be updated and moved, without having to change software that calls the routines.

Table 7-1: Utility Subroutine Addresses

Symbolic Name	Function
HOME	Recalibrate (seek to track 0)
SETTRK	Seek to a track
SETSEC	Select a sector
SETDMA	Set the DMA transfer address
READ	Read a sector of disk data
WRITE	Write a sector of disk data
SETDRV	Select a disk drive
DMASTAT	Read data transfer address (opposite of SETDMA)
GETSTAT	Read disk controller primary status port
SETHEAD	Select a read/write head
SETKEY	Set the "key" for a sector

Each subroutine, upon completion, executes a RET instruction. If the routine completes normally, it returns with the carry flag cleared to zero (0). If an error is detected, the carry flag is set to one (1). A map of the error conditions is placed in the CPU's A register. A program should always test the carry flag after a return from a disk utility subroutine, and branch to an appropriate error handling routine if the carry flag is set.

### 7.4. Disk Utility Subroutines

This section lists and describes the CP/M disk utility subroutines to help ease system integration. The source form of these routines is included on diskette, and also provided in Appendix B.

### 7.4.1. HOME

This subroutine positions the read/write heads to the outermost track of the disk platters: track 00. The disk drive has a sensor that reports when the read/write heads are physically positioned at track zero. A series of step out commands must be issued by the controller until this status line becomes active. This operation will always position the head to the same physical track. The HOME command is often called a recalibrate command and is a standard utility subroutine. Whenever the heads are moved to another track, the disk drivers must account for this change in position so that when read or write commands are issued, correct track information is passed to the controller.

One responsibility of the disk utility software is to always know over what track the heads are positioned. In general, when a drive is first selected, the track position of the heads is not known. Thus, the HOME routine should be called. In fact, if there is ever any doubt about the position of the heads, this routine should be called.

### 7.4.2. SETTRK

This routine will issue the proper commands to the drive to position the read/write heads over the track that is specified by the CPU's C register. The value in the C register should be between 0 and 201 (decimal) for the M26, and 0 to 243 for the M10/M20. A value outside of these bounds will cause the routine to abort with the carry flag set, and bit 6 of the A register set.

A test is performed to make sure the controller is not busy processing a data transfer command. Also, the status of the most recently selected drive is tested. If the controller is busy, or the drive is not ready, the carry bit is set and the routine aborts.

As before, the A register will indicate the type of error that was encountered. If bit 1 is set, the controller was busy. If bit 5 is set, the drive was not ready. If there are no error conditions, the routine issues a series of step pulses to the drive to move the read/write heads to the proper track.

This series of step commands is issued much faster than the heads can move. This does not pose a problem however, since the drive has the ability to buffer and collect pulses that arrive too rapidly. This ability enhances the performance of a multiple drive system; after a series of step commands is issued, it is possible to deselect the drive and select another. In this way, one drive can be moving its heads to a new track while another is transferring data. This type of operation is called overlapped seek. The logic of the SETTRK routine has been designed to allow as many overlapped seeks to occur as is practical. Care has been

taken so that waits encountered for head settle times are shared

whenever possible.

### 7.4.3. SETSEC

This routine allows the user to specify what sector will be involved in the next data transfer operation between the disk and memory. The sector number is passed by the C register of the CPU. It should be between 1 and 32 (decimal) for the M26, and between 1 and 21 (decimal) for the M10/M20. If the value in C is outside these bounds, the carry flag is set and the routine aborts.

### 7.4.4. SETDMA

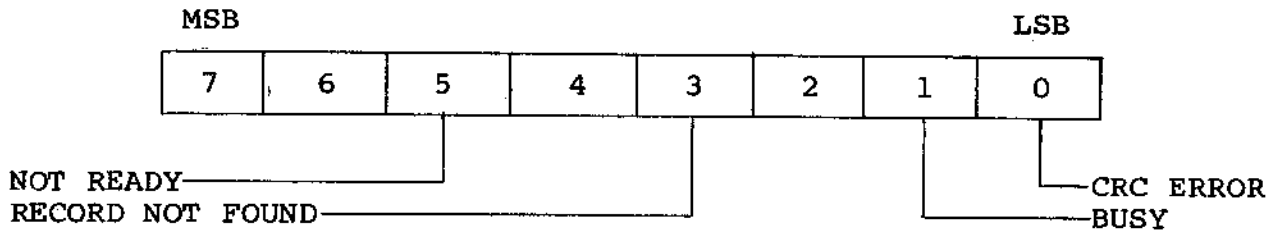
During disk transfer operations, blocks of data move to and from the disk. These blocks are 512 bytes long. When the SETDMA routine is called, the starting address (in memory) of a data block that will be involved in the next disk transfer operation is specified by the contents of the B-C register pair. The high order byte of the address is in the B register, and the low order byte is in the C register. This routine cannot produce an error.

### 7.4.5. READ

This subroutine transfers information from the disk to memory. If the controller is busy or if the drive is not ready, the routine aborts with the carry flag set. Error information is detailed in Table 7-2. The drive involved in the operation is one specified by the most recent call to the SETDVR routine (see 7.4.7). The position of the read/write heads is determined by the latest call to the SETTRK routine which involved the presently selected drive. The head number and sector number is given by the most recent calls to the SETHEAD routine (see 7.4.18) and SETSEC routine respectively. The starting memory address where the transfer will occur is specified by the most recent call to the SETDMA routine.

If the drive is ready and the controller is not busy, READ issues a series of commands to the controller. This will cause the controller to transfer information from the proper sector of the disk to its internal buffer. If any errors have occurred, the carry flag is set and the routine aborts without transferring data from the controller to memory. If the transfer is free of errors, the data is moved from the controller's buffer into memory, starting at the address specified by the last call to SETDMA.

Table 7-2: READ - Register A Error Bits



The "RECORD NOT FOUND" bit indicates that the external software has not selected the proper key to access the sector in question. If the "RECORD NOT FOUND" bit is set, the "CRC ERROR" bit should also be set. On rare occasions, error bits 0 and 3 will indicate that the header of the sector contains bad data or that a flaw exists in the magnetic media at this area of the sector.

The bad data can be corrected or the media flaw can be detected through the use of diagnostic software.

If CRC ERROR is 1 and the other bits are 0, an error was encountered in reading the data in the "data" area of the sector. When data is written to the disk, a binary polynomial is created from the serial stream as it is transferred to the disk. This number is divided by a fixed prime polynomial of order 16 until a remainder of less than 16 is produced. The data bits of this remainder are appended to the end of the data field. When a sector is read back from the disk, the same number is re-created by the serial stream, except that the remainder at the end is appended to the stream. Thus, when the original prime polynomial is divided into the new one, there will be a zero remainder only if there have been no read errors.

If there were read errors, the division will produce a remainder. If this remainder is non-zero, the CRC error bit in the A register is set. The hardware in the controller which implements the CRC logic may not, at first glance, appear to function as described above. The hardware takes advantage of the fact that the division can be done while the polynomial is being created. Normally, when the hardware detects a CRC error, the calling software will try to re read the data. If the data is still bad after ten tries, a hard error is reported. In this case, diagnostic software should be used to test the integrity of the magnetic media in this sector, and place it in the BAD SECTOR file, if necessary.

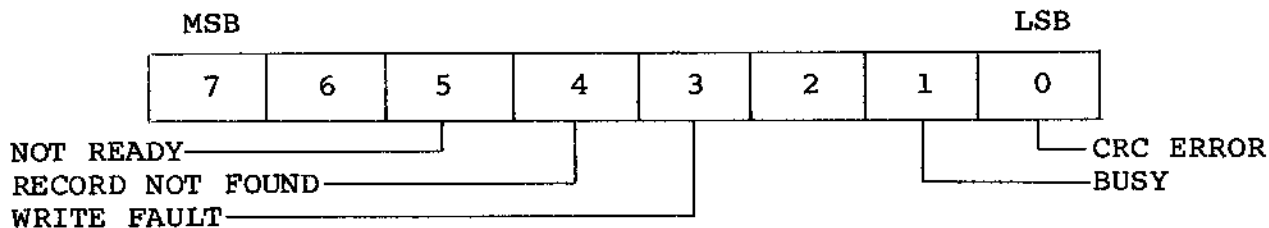
#### 7.4.6. WRITE

This subroutine transfers information from memory to the disk. If the controller is busy or if the drive is not ready, the routine aborts with the carry flag set. A map of the error bits is presented in Table 7-3.

The drive involved in the operation is the one specified by the most recent call to the SETDRV routine (see 7.4.7). The position of the read/write heads is determined by the latest call to the HOME routine which involved the presently selected drive. The head and sector number are given by the most recent calls to SETHEAD (see 7.4.18) and SETSEC routines respectively.

The most recent call to the SETDMA routine specifies the starting memory address of the transfer. If the drive is ready, and the controller is not busy, a block of data 512 bytes long is transferred to the controller's buffer from memory. WRITE then issues a series of commands to the controller which causes the controller to write the data in its buffer to the proper sector on the disk. If any errors occur, the carry flag is set and the A register is loaded with the proper error bits.

Table 7-3: WRITE - Register A Error Bits



For disk write operations, the "CRC ERROR" and "RECORD NOT FOUND" bits should always be set together. This type of error condition is discussed in the READ routine (7.4.5). The "WRITE FAULT" bit is an indication of an exceptional condition at the drive during the time the WRITE GATE signal is active. For details, refer to the disk drive documentation. This bit should never be set if the hardware is functioning correctly, and there are no faults in the cables which connect the controller to the drive(s).

#### 7.4.7. SETDRV

The value of the C register determines which one of four drives is to be selected. Only the two low order bits of register C are used for drive selection. The routine tests the "drive ready" status and delays approximately two minutes if the drive has not been selected previously. The reason for this delay is that the drive may require this amount of time to stabilize when power is initially applied. If the drive is not ready, the routine returns with the carry flag set.



7.4.8. GETSTAT

The controller has two status registers. One is a full 8 bits wide while the other is only 2 bits wide. This routine reads the first status byte into the A register and the two bits of the second status register into the B register. The meaning of the various bits is detailed next.

Table 7-4: The "A" Status Register

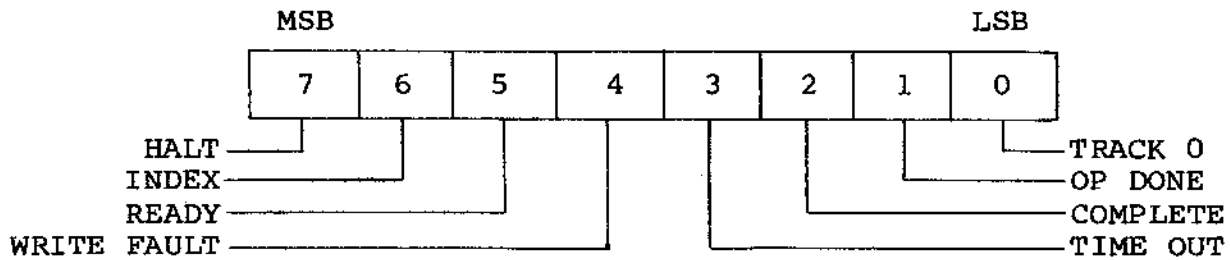
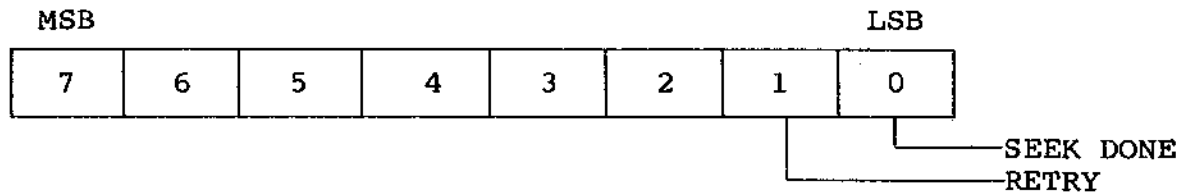


Table 7-5: The "B" Status Register



7.4.9. HALT

When this bit is a 1, the controller is halted and not presently executing a command. When this bit is a 0, the controller is either preparing to, or in the process of, executing a command.

7.4.10. INDEX

The level of this bit changes whenever an index pulse is transmitted from the presently selected drive.

7.4.11. READY

When this bit is a 1, the presently selected drive is "ready" and can respond to commands from the controller. When this bit is a 0, the drive is not ready and will not respond to controller commands.

#### 7.4.12. WRITE FAULT

When this bit is a 1, it indicates there was an exceptional condition present the last time the WRITE GATE signal to the drive was active. An example of an exceptional condition is that both READ GATE and WRITE GATE were active at the same time. It is possible that WRITE FAULT will be active when power is first applied to the drive. However, DRVSEL will always reset WRITE FAULT when the drive is initially selected. This bit indicates there is a hardware fault of some kind; either in the drive, the controller, or the connecting cables. Normally, this bit will be a 0.

#### 7.4.13. TIME OUT (TIMOUT)

Whenever the controller executes a command that is still in progress after eight disk revolutions, the TIMOUT bit is set to 1. When the controller starts to execute a command, a counter is enabled which is clocked by index pulses from the drive. If the command is still in progress after eight revolutions, the TIME OUT bit is set and the command in progress is terminated. If this bit is set after a transfer operation, it is an indication that the "key" field in the sector header on the disk does not match the key that the disk drivers have been given. When this bit is set, the RETRY bit in the B register (see 7.4.16) should also be set. Normally, the "key" field of the sector header has a zero value.

For the disk utility software discussed in this section of the manual, the default value for the key is 0. Unless a call was made to the SET KEY special routine to change the key, the TIME OUT bit should always be zero. The one exception to this rule is if there is a hard data error in the header field of the sector. Usually this will mean there is a flaw in the magnetic media, and this sector should be added to the BAD SECTOR file.

#### 7.4.14. COMPLETE (COMPLT)

When this bit is a 0, it is an indication that a drive in the system has received one or more step commands, and is in the process of moving its heads from one track to another. A drive does not have to be selected to affect this bit. When this bit is a 1, all the drives in the system have completed their seeks.

#### 7.4.15. OP DONE (OPDONE)

When this bit is a 1, it indicates that the controller has completed some kind of transfer command. Unlike HALT, this bit will be reset whenever a command is issued to the controller (even a NOP command). Once reset, it will remain 0 until another transfer operation is completed.

## Software Customization

### 7.4.16. RETRY

When the RETRY bit is set, a CRC error of some kind was made during the most recent transfer operation between the disk and the controller. If the CRC error was in the header area of the sector, the TIME OUT bit will also be set. If the error was in the data area of the sector, the TIME OUT bit will be 0. Once set, this bit will remain set until a transfer operation occurs in which there is not a CRC error.

### 7.4.17. SEEK DONE (SDONE)

This bit is set whenever the COMPLETE bit makes a transition from zero to one. It is reset when software reads port base+2 (52H).

### 7.4.18. SETHEAD

This routine selects one of eight read/write heads on the M26/M20, or one of four read/write heads on the M10. The least significant three bits of the C register determine which head will be selected. The heads are numbered 0 to 7 on the M26/M20; 0 to 3 on the M10 disk systems.

Once a head is selected it will remain selected, even if a different drive is selected. No errors are reported by this routine.

### 7.4.19. SETKEY

This routine is used to pass a new sector access key to the disk utility software. There are six bytes in the header field of a sector. These are detailed in Table 7-6.

Table 7-6: Sector Header Fields

HEAD	TRACK	SECTOR	KEY	CRC HIGH BYTE	CRC LOW BYTE
------	-------	--------	-----	------------------	-----------------

When a read or write command is issued to the controller, it scans sector headers until it finds the correct one. However, it demands more than just the correct sector number:

1. The number of the selected head must agree with the value of the first byte of the header field.
2. The track number that the heads are positioned over must agree with the value of the second byte of the header field.

## Software Customization

3. The value of the fourth byte of the header field must be a 0, or must agree with what was passed in the C register during the most recent call to the SETKEY routine.
4. The value of the two CRC error detection bytes must produce a 0 remainder for the controller hardware.

If all these requirements are satisfied, the controller will proceed to transfer a sector of data to or from the disk. Thus, the "key" field of the sector header allows for 256 levels of security for each sector of data. Except on track 0, this "key" field is normally a 0. By using the software described in the next section, this field can be altered and even read. The purpose of the SETKEY routine is to allow a user access to sectors which have a key different from zero.

## Hardware Level Registers

### 8. HARDWARE LEVEL REGISTERS

#### 8.1. Overview

The set of low level software drivers described in Section 7 is included with the HDCA to provide greater control over the disk than is possible through the CP/M operating system alone. For greater control over the disk, however, and the ability to communicate with the HDCA directly, users should refer directly to the I/O device registers on the HDCA controller. There are seven 8 bit registers. Three are read only, three are write only and one is read/write. These registers occupy four locations in the I/O address space of the system. They may appear anywhere in this space. The only restriction is that the lowest of the four addresses must be divisible by four (4).

#### 8.2. I/O Addressing

At location 8C on the circuit board, there is an 8 position DIP switch used to determine the starting address of the controller. One of the switches is not used, and another serves as a board enable. The other six are connected to a comparator which compares switch settings with I/O addresses on the bus. If there is a match and the board is enabled, I/O commands are allowed to access the controller. Figure 3-1 illustrates the layout of this DIP switch.

Use the settings illustrated in Figure 3-2 as an example of addressing the HDCA controller to respond to I/O addresses 50H through 53H.

Refer to Appendix C for a list of all possible I/O addresses available through this switch.

#### 8.3. I/O Register Map

Let A represent the address of register 0 of the controller. In boards with standard addresses, A = 50H. The addresses of the controller registers are detailed in Table 8-1.

Table 8-1: HDCA I/O MAP

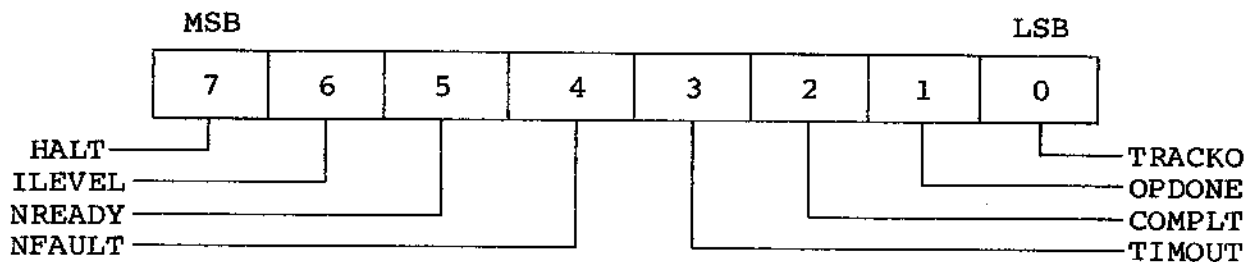
STANDARD I/O PORT ADDRESS	I/O READ FUNCTION	I/O WRITE FUNCTION
50H	HDCA PRIMARY STATUS REGISTER	HDCA CONTROL PORT
51H	AUXILLIARY STATUS REGISTER (LOWER 4 BITS ONLY)	CONTROLLER COMMAND PORT
52H	INTERRUPT CLEAR REGISTER	DRIVE CONTROL/SELECT PORT
53H	HEADER/SECTOR DATA WINDOW	HEADER/SECTOR DATA WINDOW

8.4. Read-Only Registers - Register 0

Function: (READ) - Primary Status Register  
 Location 50H standard HDCA

This is the main status port for the controller and drive, located at 50H in the standard system. It contains bits that identify the current status of the HDCA controller and the currently selected drive. The details of this register are presented in Figure 8-2.

Table 8-2: Controller Status Register



8.4.1. HALT

This is the "not busy" flag of the controller. When the Disk Jockey is not executing a command, this bit is a 1. When a data transfer command is strobed into the command register, the HALT bit is reset to 0. At this point, the controller is busy and will not respond to new commands until the HALT bit is again 1.

## Hardware Level Registers

Moreover, while HALT is 0, the CPU does not have access to the internal data bus and therefore cannot read from or write to the controller's data buffer.

A program that interfaces directly to the controller should monitor this bit to determine when a command is completed. The Main Status Register interfaces directly to the S-100 DI (Data Input) bus to allow the system to have access to the status port, regardless of the state of the controller.

### 8.4.2. ILEVEL

This bit changes state with each index pulse from the currently selected drive. Drives that are not selected or not ready cannot transmit index pulses. Thus, ILEVEL only toggles when the selected drive is ready.

### 8.4.3. NREADY

This bit is a 0 only when the currently selected drive is powered up and ready to receive commands or transfer data.

### 8.4.4. NFAULT

Each drive in the system (when selected), sends a negative logic signal to the controller called WRITE FAULT. NFAULT monitors this line. WRITE FAULT is active (at a 0 level) if an illegal logic condition existed during a data transfer to the drive. An example of an illegal logic condition is READ GATE and WRITE GATE were active at the same time. This could happen, for example, if the 50 conductor cable between the controller and the drive were installed upside down at one end.

Other conditions may also occur internal to the drive which can cause WRITE FAULT to be active (see the drive manual for details). Occasionally, WRITE FAULT is active when a drive is first powered up. The utility software, as a matter of course, resets WRITE FAULT on drives that it selects for the first time. Under normal conditions, NFAULT is 1.

### 8.4.5. TIME OUT (TIMOUT)

This bit is the latched output of a counter which is clocked by index pulses from the currently selected drive. The counter is enabled when the controller is busy. If a command is in progress after 16 revolutions of the disk, TIMOUT is set to 1 and the command is terminated. This insures that the controller will never "hang" trying to complete a command. Typically, this bit is set when the controller is asked to search for a sector header image that does not exist on the current track. TIMOUT is reset whenever a new command is sent to the controller.

## Hardware Level Registers

### 8.4.6. SEEK COMPLETE

Each drive in the system (which is ready) sends a negative logic signal to the controller called SEEK COMPLETE. This signal is present even if the drive is not selected. When a drive receives a head step command (or a burst of head step commands), logic inside the drive sets the SEEK COMPLETE line false. While the heads are moving to a new track this signal remains false. SEEK COMPLETE goes active again just as the heads have stopped (but not settled). When all the SEEK COMPLETE lines from the drives are active, COMPLT is 1. If any drive's heads are in motion, COMPLT is 0.

### 8.4.7. OPDONE

This bit is set to a 1 whenever the controller finishes a data transfer command. It is reset whenever any command is issued to the controller.

### 8.4.8. NTRCK0

This bit is 0 when the heads of the currently selected drive are positioned over track zero. If the heads are over any other track, NTRCK0 is 1.



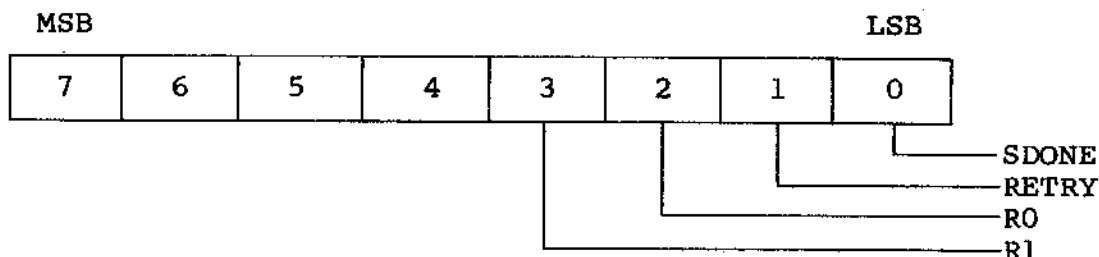
## Hardware Level Registers

### 8.5. Read-Only Registers - Register 1

Function: (READ) - Auxiliary Status Register  
Location 51H standard HDCA

This register is an Auxiliary Status Port for the controller and drive. It is found in location 51H in the standard system. It is four bits wide and contains the auxiliary status information regarding the drive and controller. The details of these bits are presented in Table 8-3.

Table 8-3: Auxiliary Status Register



#### 8.5.1. R0 and R1

These two bits are used by the controller to inform external software of the revision level of the board. The encoding scheme for R0 and R1 is given below:

Table 8-4: Encoding Scheme for R0 and R1

R1	R0	Rev Level
1	1	0
1	0	1
0	1	2
0	0	3 & 4

#### 8.5.2. RETRY

This bit is set to 0 whenever a command is issued to the controller. During transfers from the disk, the serial data stream is routed to two places: the shift register, where data is assembled into 8 bit bytes for storage in the data buffer, and to the CRC logic, where polynomial division is performed.

The last 16 bits of any transfer is the CRC error check word. These bits are not stored in the buffer. The last task of any command that transfers data from the disk to the controller is to compare the contents of the CRC register with the CRC error check word. If an error occurs, RETRY is set to 1. Thus, RETRY is the "read data" error flag. If RETRY is high after a read command and the TIMOUT bit of the main status register is 0, the calling

## Hardware Level Registers

program should read the data again. If RETRY remains high through 10 tries, a hard error is present on the sector.

When the controller is asked to search for a sector header image that does not exist on the current track, both TIMEOUT and RETRY will be high. If both TIMEOUT and RETRY are high, then

1. the track is not formatted,
2. there is a hard error in a sector header field, or
3. the controller has an improper sector header image in the header area of its data buffer (this is the most likely of the three possibilities).

### 8.5.3. SDONE

Whenever the COMPLETE bit of the main status register makes a transition from 0 to 1, SDONE is set to 1. SDONE is cleared by reading port base+2 (52H).

### 8.6. Read Only Registers - Register 2

Function: Clear Interrupt  
Location 52H standard HDCA

On HDCA controllers of revision level 4 or greater, the interrupt generated by SEEK COMPLETE or OPDONE remains latched until the CLEAR INTERRUPT port is accessed (I/O read from Port 52). Earlier revision boards may be configured this way, as described in ECN #1 for the HDCA-3, dated 4-8-82. Contact Morrow Designs Customer Service for details on this modification. Details on the use of interrupts are included in Section 6 of this manual.

### 8.7. Read/Write Registers - Register 3

Function: (READ/WRITE) - Header/Sector Data Port  
Location 53H standard HDCA

This is the controller data port at location 53H in the standard system. When the controller writes data to the disk, it is taken from an on board 1024 byte buffer. Conversely, when data is read from the disk, it is stored in this same buffer. Register 3 is

## Hardware Level Registers

the window between this buffer and the CPU. This register is used to fill the buffer before write commands are issued, and to empty it after read commands have completed.

Associated with the data port is a pointer which serves to address different locations of the on board data buffer. It is incremented after references to the data port. The pointer can be reset to either half of the buffer by commands to the controller (refer to the command port). Each half of the buffer is 512 bytes long. One half is for data, the other is for sector header information. The controller uses only the first six bytes of the header half of the buffer, the remainder is available to the system.

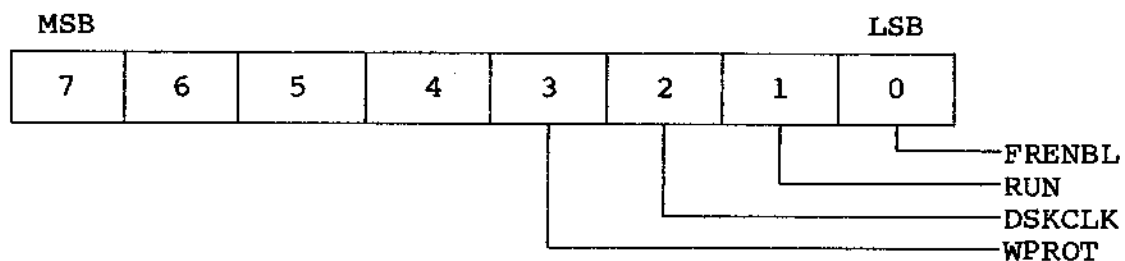
Data is transferred to the buffer by first resetting the pointer and then initiating successive I/O references to this register. The address pointer will automatically increment after each reference. After data is written in the buffer, it will remain stable unless a disk write command is issued or new data is written to the buffer by the CPU. Likewise, once a disk read operation has loaded the buffer, the CPU may retrieve data from it as often as desired.

### 8.8. Write-Only Registers - Register 4

Function: (WRITE) - HDCA Control Port  
Location 50H standard HDCA

Register 4 is the control port. Its location is 50H in the standard system. It is a four bit wide register which functions as the master control port for the board. The four high order data bits are ignored. The function of each low order bit is outlined in Table 8-5.

Table 8-5: Master Control Register



## Hardware Level Registers

### 8.8.1. WPROT

This bit serves two purposes, depending on the state of the NFAULT bit of the main status register. When WPROT is 0, the currently selected drive will be write protected as long as there are no write fault conditions present. If NFAULT=0, WPROT will reset a write fault condition when it is brought low and then high. The drive is write enabled when WPROT is 1 and NFAULT is 0. That is, the disk will accept write commands if there are no write faults present and the write protect bit, WPROT, is high.

### 8.8.2. DSKCLK

This bit determines how the master clock of the controller will be driven. If DSKCLK is 0, the master clock signal on the board will come from PHASE2 (pin-24) on the S-100 bus. If DSKCLK is 1, the currently selected drive will furnish the master clock to the controller. During data transfers, DSKCLK must be 1. However, if the selected drive is not ready or has encountered a write fault condition, it does not transmit any clock signals. In such a situation, the DSKCLK bit should be brought low so the controller will respond to commands.

### 8.8.3. RUN

When this bit is 0, the controller is reset and halted. It will not respond to commands - not even the buffer pointer reset commands. This bit is the master enable signal for the controller and should be set to 1 just after the first drive in the system is ready and does not have a write fault condition pending.

### 8.8.4. FRENBL

This bit enables the output of the drive select and drive function register. The function register outputs are TRI-STATE drivers that can be enabled or disabled. When FRENBL is 0, these drivers are disabled. When this bit is 1, the drivers are enabled.

Each bit in the control port is cleared to a zero whenever the S-100 bus signals POC (pin-99) or PRESET (pin-75) are active. Thus, the drive function register's outputs are disabled, the controller is in a reset state (with RUN false), the master clock is driven by PHASE2, and the disk is write enabled.

## Hardware Level Registers

### 8.9. Write-Only Registers - Register 5

Function: (WRITE) - Command Register  
Location 51H standard HDCA

This is the controller command port at location 51H in the standard system. This register is four bits wide and commands are transferred using the low order nibble of the data. The controller will execute six commands: reset the data buffer pointer to the first location of the data area, read a sector header, read a sector of data, write a sector of data, write a sector header, and reset the data buffer pointer to the first location of the header area. There are several more commands which the controller can accept, but these are for test purposes and should never be used in a normal environment. A loss of data on the disk could occur if the user should issue any unlisted command. A detailed explanation of the standard commands is presented below.

NOTE: The command numbers below are also the numeric values to output to Port 51H when issuing commands to the controller through this port.

Table 8-6: HDCA Command Summary

COMMAND	FUNCTION
0	Set RAM Pointer = Sector Data
1	Read Sector
3	Read Header
5	Write Sector Data
7	Write Sector Header
8	Set RAM Pointer = Sector Header

#### 8.9.1. COMMAND 0

As explained previously, the controller has a 1024 byte data buffer which is divided into two sections: sector data and sector header information. These two sections share a pointer which is used to address the buffer. Command 0 resets this pointer to the beginning of the data area of the buffer. All commands affect bits in the status registers. Command 0 resets OPDONE and TIMOUT. No other status bits are changed.

#### 8.9.2. COMMAND 1

This is the read sector command. As before, the first thing that occurs is OPDONE, TIMOUT and RESET are reset. The HALT bit is reset and access to the internal data bus is inhibited. As sector pulses arrive from the drive, the controller scans the sector header and compares it with the first four bytes of the header area of the internal buffer. If the sector header data matches the buffer and the CRC bytes match the contents of the

## Hardware Level Registers

CRC register, the data in the sector is read into the data area of the internal buffer.

The first byte of data is written into the third location of this area in the buffer. Successive bytes are written into successive locations. The next to last byte is written into the first location of the data area and the last byte from the disk is placed in the second location.

After the last byte is read, the controller compares the CRC checksum bytes with the contents of the CRC register. If there is a compare error, the RETRY bit of the auxiliary status register is set and the command terminates with OPDONE set. If the header does not match the pattern in the header area of the buffer, the controller continues to scan successive headers until a match is found (including CRC data) or until 16 index pulses have occurred. If no match is found by the 16th index, the TIMOUT bit of the main status register and the RETRY bit of the auxiliary status register are set and the command terminates with OPDONE and HALT being set.

### 8.9.3. COMMAND 3

This is the read a sector header command. After receipt of this command, the HALT bit of the main status register is brought false and no further access to the internal data bus is allowed until the command terminates. OPDONE, TIMOUT and RETRY are reset. When the next sector pulse from the drive arrives, the controller reads the four bytes of the header into the 3rd, 4th, 5th, and 6th locations of the sector header area of the buffer. The CRC checksum bytes are compared with the contents of the CRC register. If there was an error in the data transfer, the RETRY bit in the auxiliary status port is set. After the CRC bytes have been checked, the HALT bit is brought back to its true state and the command terminates with OPDONE set.

### 8.9.4. COMMAND 5

This is the write sector command. The sequence of events is exactly the same as with the read sector command, up to the point of the actual data transfer. If a match has occurred between the header area of the buffer and a sector header, the data area of the internal buffer is written to the sector where the header match took place. The data is transferred starting at the first location of the data area of the buffer. Successive bytes are taken from successive locations. The last byte written to the disk is fetched from the last location in the data area of the buffer.

If the header image in the buffer has no counterpart on the current track and head of the disk, the RETRY and TIMOUT bits are set in the status registers after 16 revolutions from the time

## Hardware Level Registers

the command was issued. As before, OPDONE and HALT are also set at the end of the command.

### 8.9.5. COMMAND 7

This command writes the first four bytes in the header area of the buffer on a sector header. The header in question is the very first that is encountered after receipt of the command. TIMEOUT or RETRY cannot be set by this command. As usual, however, OPDONE and HALT are set at the command's conclusion.

### 8.9.6. COMMAND 8

This command resets the internal pointer to the first location of the header area of the controller's data buffer. The other effects of this command are identical to COMMAND 0.

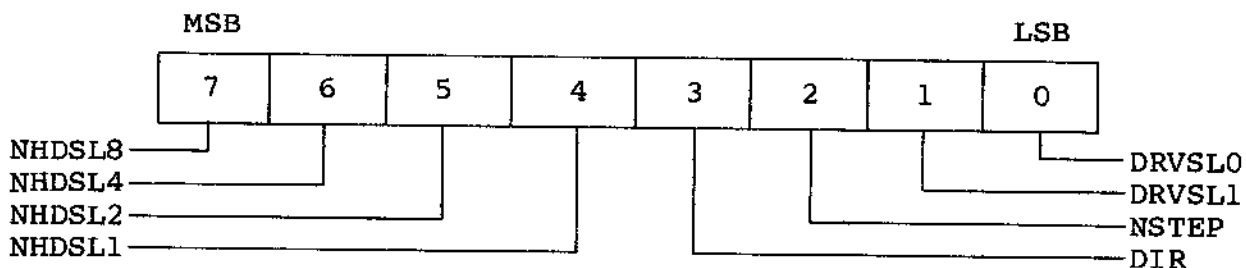
NOTE: Within the four bits allowed by commands, a total of sixteen are possible. However, the user is strongly cautioned not to use any except those listed above. If any other commands are issued to the controller, data on the disk could be lost if the WPROT bit in the control register is low. A good practice is to keep WPROT high at all times except during periods when write commands (5 & 7) are in progress.

## 8.10. Write-Only Registers - Register 5

Function: (WRITE) - Drive Control/Select Register  
Location 52H standard HDCA

Register 5 is the Drive Select and Drive Function port at location 52H in the standard system. It selects one of four drives, one of up to sixteen heads, and controls the two lines which step the heads. There are eight data bits in the register and their specific functions are presented in Table 8-7.

Table 8-7: Drive Function Register



## Hardware Level Registers

### 8.10.1. NHDSL

These are the head select bits. By using four, up to sixteen heads can be selected. On standard drives there are only eight heads. However, there is a model available which has eight extra fixed heads. This allows a user to have 131K of fast access memory on the disk which is independent of the position of the eight moving heads. This model must be specially ordered and customers may have a longer wait for such systems.

The relationship between the NHDSL bits and the head selected at the drive is detailed below. Heads numbered 0 through 7 move from track to track, and are present on all models of M26. The fixed heads are numbered 8 through 15, and are usually not present. Heads 8 through 15 should never be selected on a standard drive.

Table 8-8: NHDSL Bits and Head Select Relationship

NHDSL8	NHDSL4	NHDSL2	NHDSL1	HEAD NO.
0	0	0	0	15
0	0	0	1	14
0	0	1	0	13
0	0	1	1	12
0	1	0	0	11
0	1	0	1	10
0	1	1	0	9
0	1	1	1	8
1	0	0	0	7
1	0	0	1	6
1	0	1	0	5
1	0	1	1	4
1	1	0	0	3
1	1	0	1	2
1	1	1	0	1
1	1	1	1	0

### 8.10.2. DIR

This bit controls the direction the read/write heads move when the NSTEP bit is pulsed. DIR must never change when NSTEP makes a transition from 0 to 1. If DIR is 0, the heads will move toward the center of the disk when step pulses are issued. Conversely, when DIR is 1, the heads will move away from the center of the disk. Track 0 is the outermost track on the disk. Track numbers increase as the heads move closer to the center of the disk.



## Hardware Level Registers

### 8.10.3. NSTEP

This bit is used to issue step commands to the disk. Its idle state is 1. To issue a step command, NSTEP is brought low and then high. DIR must not change when NSTEP makes the transition from low to high. There are two modes of operation for stepping the heads: buffered and normal. In normal mode, the heads will move at the rate of the incoming NSTEP pulses and the minimum time between successive steps is 1 millisecond. In buffered mode, the step pulses are buffered into a counter at the drive. After the last pulse, the heads will begin stepping toward the appropriate track. The COMPLT bit (of the status register) will go true after the heads arrive at the proper track.

An example of this activity is provided in the HOME routine in the Driver Software, Appendix B.2.

### 8.10.4. DRVSL

These two bits select one of four drives. The relationship between these bits and the physical drive that is selected is given by the following table:

Table 8-9: DRVSL Bits and Drive Select Relationship

DRVSL1	DRVSLO	DRIVE NO.
0	0	0
0	1	1
1	0	2
1	1	3

When power is first applied to the system, the function register is in an unknown state. It is the user's responsibility to write a proper pattern to this port before its output drivers are enabled. In particular, NSTEP should be 1 when the outputs are first enabled.

A. TROUBLESHOOTING GUIDE

PROBLEM	PROBABLE CAUSE	REMEDY
UNABLE TO BOOT OR ADDRESS HARD DISK	Incorrect data cable connection - backwards or unplugged.	Check/correct internal/external data cable connections.
	Incorrect control cable connection - backwards or unplugged.	Check/correct internal/external data cable connections.
	HDCA SW8C set incorrectly.	Refer to Sect. 3.5 or Appendix C.
	Drive not on.	Check power connection and switch.
	Drive head locked.	Refer to Sect. 3.10.
	Drive not warmed up.	Refer to Sect. 3.10.
	Power supply connection loose.	Contact dealer or service center.
FORMATHD "HANGS" BEFORE BEGINNING TEST	Incorrect data cable connection - backwards or unplugged.	Check/correct internal/external data cable connections.
	Seek complete jumpers not installed.	Install jumpers on pins 7 & 8 of P3, P4 & P5.
FORMATHD "HANGS" AFTER 2 MINUTE WAITING PERIOD	Incorrect control cable connection - backwards or unplugged.	Check/correct internal/external data cable connections.
SYSTEM BOOTS UP IN MONITOR (:)	MPZ80 CPU SW16D set incorrectly.	Refer to CPU manual or installation guide.
UNABLE TO COPY DATA TO OR FROM HARD DISK.	Headers not formatted correctly	Format headers only.
	Header and data fields garbled.	Format headers and data field.
	Sector errors on hard disk.	Run diagnostics and reformat; contact dealer.

B. SOFTWARE LISTINGS

B.1. The Bootstrap Software

The following sample program will load the disk system software into memory. There is an option which will either load CP/M or only the utility low level disk drivers. The program starts at location 100H (Hex) but can be easily changed to start at the beginning of any page by altering the third byte of the jump (and conditional jump) instructions to the value of the desired page.

## Bootstrap Load Program

		1	*HARD DISK BOOTSTRAP SOFTWARE	
0100	C3 00 03	2	JMP	START
		3		
0103	01FD	4	DS	509 room for boot
		5		
0300	3E FC	6	START	MVI A,DRIVEA select
0302	D3 52	7		OUT FUNCTN -drive A
0304	3E 05	8		MVI A,DRENBL turn on drive
0306	D3 50	9		OUT CONTRL -command register
0308	DB 50	10	RLOOP	IN STATUS test for
030A	E6 20	11		ANI READY drive A ready
030C	C2 08 03	12		JNZ RLOOP
030F	3E 07	13		MVI A,DSKRUN enable the
0311	D3 50	14		OUT CONTRL -controller
0313	DB 50	15	WAITZ	IN STATUS test for heads
0315	1F	16		RAR . -at track zero
0316	D2 2B 03	17		JNC SDONE
0319	3E F8	18		MVI A,STEPO execute
031B	D3 52	19		OUT FUNCTN -the
031D	3E FC	20		MVI A,DRIVEA -step out
031F	D3 52	21		OUT FUNCTN -command
0321	DB 50	22	WAITC	IN STATUS wait for
0323	E6 04	23		ANI COMPLT -the seek
0325	CA 21 03	24		JZ WAITC -to complete
0328	C3 13 03	25		JMP WAITZ
032B	DB 50	26	SDONE	IN STATUS get an image
032D	4F	27		MOV C,A -of the status reg
032E	DB 50	28	IWAIT1	IN STATUS wait for
0330	91	29		SUB C -the index pulse
0331	CA 2E 03	30		JZ IWAIT1 -to arrive
0334	DB 50	31	IWAIT2	IN STATUS wait for the
0336	91	32		SUB C -next index pulse
0337	C2 34 03	33		JNZ IWAIT2 test for head settle
033A	3E 08	34		MVI A,HEADER reset the
033C	D3 51	35		OUT COMMD -buffer pointer
033E	AF	36		XRA A -to header area
033F	D3 53	37		OUT DATA head 0
0341	D3 53	38		OUT DATA track 0
0343	3E 01	39		MVI A,SECTOR 1 for CP/M
0345	D3 53	40		OUT DATA -or 30 for drivers
0347	3E 80	41		MVI A,SYSTEM system key
0349	D3 53	42		OUT DATA
034B	3E 01	43		MVI A,DREAD issue a
034D	D3 51	44		OUT COMMD -read command
034F	DB 50	45	WAITD	IN STATUS wait for command
0351	E6 02	46		ANI OPDONE -to complete
0353	CA 4F 03	47		JZ WAITD
0356	DB 53	48		IN DATA low order byte of
0358	6F	49		MOV L,A -bootstrap address
0359	5F	50		MOV E,A
035A	DB 53	51		IN DATA high order byte of
035C	67	52		MOV H,A -bootstrap address
035D	57	53		MOV D,A
035E	DB 53	54	LLOOP	IN DATA load
0360	12	55		STAX D -the
0361	1C	56		INR E -bootstrap
0362	C2 5E 03	57		JNZ LLOOP
0365	E9	58		PCHL . branch there

# Bootstrap Load Program

			1	*HARD DISK BOOTSTRAP SOFTWARE
001:000	303	000	2	JMP START
			3	
001:003	001:375		4	DS 509 room for boot
			5	
003:000	076	374	6	START MVI A,DRIVEA select
003:002	323	122	7	OUT FUNCTN -drive A
003:004	076	005	8	MVI A,DRENBL turn on drive
003:006	323	120	9	OUT CONTRL -command register
003:010	333	120	10	RLOOP IN STATUS test for
003:012	346	040	11	ANI READY drive A ready
003:014	302	010	12	JNZ RLOOP
003:017	076	007	13	MVI A,DSKRUN enable the
003:021	323	120	14	OUT CONTRL -controller
003:023	333	120	15	WAITZ IN STATUS test for heads
003:025	037		16	RAR . -at track zero
003:026	322	053	17	JNC SDONE
003:031	076	370	18	MVI A,STEPO execute
003:033	323	122	19	OUT FUNCTN -the
003:035	076	374	20	MVI A,DRIVEA -step out
003:037	323	122	21	OUT FUNCTN -command
003:041	333	120	22	WAITC IN STATUS wait for
003:043	346	004	23	ANI COMPLT -the seek
003:045	312	041	24	JZ WAITC -to complete
003:050	303	023	25	JMP WAITZ
003:053	333	120	26	SDONE IN STATUS get an image
003:055	117		27	MOV C,A -of the status reg
003:056	333	120	28	IWAIT1 IN STATUS wait for
003:060	221		29	SUB C -the index pulse
003:061	312	056	30	JZ IWAIT1 -to arrive
003:064	333	120	31	IWAIT2 IN STATUS wait for the
003:066	221		32	SUB C -next index pulse
003:067	302	064	33	JNZ IWAIT2 test for head settle
003:072	076	010	34	MVI A,HEADER reset the
003:074	323	121	35	OUT COMMD -buffer pointer
003:076	257		36	XRA A -to header area
003:077	323	123	37	OUT DATA head 0
003:101	323	123	38	OUT DATA track 0
003:103	076	001	39	MVI A,SECTOR 1 for CP/M
003:105	323	123	40	OUT DATA -or 30 for drivers
003:107	076	200	41	MVI A,SYSTEM system key
003:111	323	123	42	OUT DATA
003:113	076	001	43	MVI A,DREAD issue a
003:115	323	121	44	OUT COMMD -read command
003:117	333	120	45	WAITD IN STATUS wait for command
003:121	346	002	46	ANI OPDONE -to complete
003:123	312	117	47	JZ WAITD
003:126	333	123	48	IN DATA low order byte of
003:130	157		49	MOV L,A -bootstrap address
003:131	137		50	MOV E,A
003:132	333	123	51	IN DATA high order byte of
003:134	147		52	MOV H,A -bootstrap address
003:135	127		53	MOV D,A
003:136	333	123	54	LLOOP IN DATA load
003:140	022		55	STAX D -the
003:141	034		56	INR E -bcotstrap
003:142	302	136	57	JNZ LLOOP
003:145	351		58	PCHL . branch there

## Low Level Drivers

### B.2. Low Level Drivers

```

*****
*
*           * Low Level Hard Disk Drivers. The following routines are
the *       * lowest level drivers for the hard disk.
*
*           *
*           * Written By Bobby Dale Gifford.
*
*           * 12/8/80
*
*
*****

```

```

000C =      HDREV  EQU    12      ;Revision number
0004 =      MAXHD  EQU     4      ;Maximum # of Hard Disk
Drives

0050 =      HDORG  EQU    50H     ;Hard Disk Controller origin
0050 =      HDSTAT EQU    HDORG   ;Hard Disk Status
0050 =      HDCNTL EQU    HDORG   ;Hard Disk Control
0053 =      HDDATA EQU    HDORG+3 ;Hard Disk Data
0052 =      HDFUNC EQU    HDORG+2 ;Hard Disk Function
0051 =      HDCMND EQU    HDORG+1 ;Hard Disk Command
0051 =      HDRESLT EQU    HDORG+1 ;Hard Disk Result
0002 =      RETRY  EQU     2      ;Retry bit of result
0001 =      TKZERO EQU     1      ;Track zero bit of status
0002 =      OPDONE EQU     2      ;Operaction done bit of
status
0004 =      COMPLT EQU     4      ;Complete bit of status
0008 =      TMOUT  EQU     8      ;Time out bit of status
0010 =      WFAULT EQU    10H    ;Write fault bit of status
0020 =      DRVRDY EQU    20H    ;Drive ready bit of status
0040 =      INDEX  EQU    40H    ;Index bit of status
0004 =      PSTEP  EQU     4      ;Step bit of function
00FB =      NSTEP  EQU    0FBH   ;Step bit mask of function
0004 =      HDRLEN EQU     4      ;Sector header length
0200 =      SECLEN EQU    512    ;Sector data length
000F =      WENABL EQU    0FH    ;Write enable
000B =      WRESET EQU    0BH    ;Write reset of function
0005 =      SCENBL EQU     5      ;Controller control
0007 =      DSKCLK EQU     7      ;Disk clock for control
00F7 =      MDIR  EQU    0F7H   ;Direction mask for function
00FC =      NULL  EQU    0FCH   ;Null command
0000 =      IDBUFF EQU     0      ;Initialize data command
0008 =      ISBUFF EQU     8      ;Initialize header command
0001 =      RSECT  EQU     1      ;Read sector command
0005 =      WSECT  EQU     5      ;Write sector command
0015 =      SECCNT EQU     21     ;Sectors per track
                                ;32 for M26
                                ;21 for M10 & M20

0100          ORG    100H

```

```

0100 317603          LXI    SP,STACK
0103 0E00           MVI    C,0
0105 CD7603         CALL   SETDRV
0108 CD7903         CALL   HOME
010B 014401         LXI    B,BXX
010E CD8203         CALL   SETDMA
0111 0E00           MVI    C,0
0113 CD7C03         CALL   SETTRK
0116 0E00           MVI    C,0
0118 CD8503         CALL   SETHEAD
011B 0E01           MVI    C,1
011D CD7F03         CALL   SETSEC
0120 0E80           MVI    C,80H
0122 CD8803         CALL   SETKEY
0125 CD7B04         CALL   HDREAD
0128 00            NOP
0129 0EC8           MVI    C,200
012B CD7C03         CALL   SETTRK
012E 0E00           MVI    C,0
0130 CD8503         CALL   SETHEAD
0133 0E01           MVI    C,1
0135 CD7F03         CALL   SETSEC
0138 0E00           MVI    C,0
013A CD8803         CALL   SETKEY
013D CD7B04         CALL   HDREAD
0140 00            NOP
0141 C31101         JMP    LOOP

```

```

0144          BXX    DS    512
0344          DS    50
0376 =        STACK EQU  $

```

```

0376 C39703     SETDRV  JMP    HDDRV      ;Select disk
0379 C3E203     HOME    JMP    HDHOME     ;Recalibrate
037C C31804     SETTRK  JMP    HDTRK      ;Seek to specified track
037F C36704     SETSEC  JMP    HDSEC      ;Prep for sector #
0382 C34A04     SETDMA  JMP    HDDMA      ;Prep for DMA address
0385 C37404     SETHEAD JMP    HDHEAD     ;Set head #
0388 C3DD03     SETKEY  JMP    HDKEY      ;Set the key in next
transfer
038B C37B04     READ    JMP    HDREAD     ;Read one sector
038E C3B004     WRITE   JMP    HDWRITE    ;Write one sector
0391 C34B05     DMASTAT JMP    DMAHD      ;Return DMA address
0394 C35305     GETSTAT JMP    STATHD     ;Get drive status

0397 3E03       HDDRV  MVI    A,3
0399 A1         ANA    C
039A 324705     STA    HDDISK

```



```

039D F6FC'      ORI      NULL      ;Select drive
039F D352      OUT      HDFUNC
03A1 3E05      MVI      A,SCENBL  ;Enable the controller
03A3 D350      OUT      HDCNTL
03A5 0EEF      MVI      C,239      ;Wait for Disk to ready
                                ; 2 minutes for M26
                                ; 30 seconds for M10 & M20

03A7 210000    TDELAY  LXI      H,0
03AA 2B        DCX      H
03AB 7C        MOV      A,H
03AC B5        ORA      L
03AD CCDB03    CZ        DCRC
03B0 37        STC
03B1 C8        RZ
03B2 DB50     IN        HDSTAT      ;Test if ready yet
03B4 E620     ANI      DRVRDY
03B6 C2AA03   JNZ      TDELAY
03B9 2AFC03   LHL      SETTLE
03BC 7C        MOV      A,H
03BD B5        ORA      L
03BE C8        RZ
03BF 210000    LXI      H,0      ;Time one revolution of the
drive
03C2 0E40     MVI      C,INDEX
03C4 DB50     IN        HDSTAT
03C6 A1        ANA      C
03C7 47        MOV      B,A      ;Save current index level in
B
03C8 DB50     INDX1  IN        HDSTAT
03CA A1        ANA      C
03CB B8        CMP      B      ;Loop util index level
changes
03CC CAC803   JZ        INDX1
03CF 23        INDX2  INX      H
03D0 DB50     INDX2  IN        HDSTAT      ;Start counting until index
returns to
03D2 A1        ANA      C      ; previous state
03D3 B8        CMP      B
03D4 C2CF03   JNZ      INDX2
03D7 22FC03   SHLD    SETTLE      ;Save the Count for timeout
delay
03DA C9        RET
03DB 0D        DCRC    DCR      C      ;Conditional decrement C
routine
03DC C9        RET
03DD 79        HDKEY  MOV      A,C
03DE 321A05   STA     NKEY
03E1 C9        RET
03E2 CD3405   HDHOME CALL    DRVPTN
03E5 3600     MVI     M,0      ;Set track to zero
03E7 23        INX     H      ;Point to seek flag
03E8 3601     MVI     M,1      ;Set not seeking, but must
delay
03EA DB50     STEPO  IN        HDSTAT      ;Test status

```

```

03EC E601      ANI      TKZERO      ;At track zero ?
03EE C8        RZ
03EF 3E01      MVI      A,1
03F1 37        STC
03F2 CD3904    CALL     ACCOK      ;Take one step out
03F5 CD5004    CALL     WSDONE     ;Wait for previous seek to
finish
03F8 C3EA03    JMP      STEPO
03FB 210000    DELAY   LXI      H,0      ;Get delay
03FC =         SETTLE EQU      $-2
03FE 2B        DELOOP DCX      H
03FF 7C        MOV      A,H      ;Wait 20ms
0400 B5        ORA      L
0401 23        INX      H
0402 2B        DCX      H
0403 C2FE03    JNZ     DELOOP
0406 215C05    LXI     H,DRIVES-1
0409 0605      MVI     B,MAXHD+1
040B 23        DELUP  INX      H
040C 23        INX      H
040D 05        DCR      B
040E C8        RZ
040F 7E        MOV     A,M
0410 3D        DCR     A
0411 C20B04    JNZ     DELUP
0414 77        MOV     M,A
0415 C30B04    JMP     DELUP
0418 CD3405    HDTRK  CALL     DRVPTR   ;Get pointer to current
track
041B 5E        MOV     E,M      ;Get current track
041C E5        PUSH    H        ;Save pointer to current
track
041D 1C        INR     E        ;Ever homed this drive ?
041E CCE203    CZ      HDHOME
0421 E1        POP     H        ;Restore track pointer
0422 5E        MOV     E,M      ;Get current track
0423 71        MOV     M,C      ;Update the track
0424 7B        MOV     A,E      ;Need to seek at all ?
0425 91        SUB     C
0426 C8        RZ
0427 F5        PUSH    PSW      ;Save # of steps
0428 23        INX     H        ;Point to the seek complete
flag
0429 7E        MOV     A,M      ;Get current seek progress
flag
042A 3C        INR     A        ;Currently seeking ?
042B E5        PUSH    H        ;Save seek flag pointer
042C CC5004    CZ      WSDONE   ;Wait if currently seeking
042F E1        POP     H
0430 36FF      MVI     M,OFFH   ;Set seek in progress flag
0432 F1        POP     PSW
0433 3F        CMC     ;Get carry into direction
0434 DA3904    JC      ACCOK
0437 2F        CMA
0438 3C        INR     A

```

```

0439 47      ACCOK  MOV     B,A           ;Prep for build
043A CD4005  CALL   BUILD
043D E6FB    SLOOP  ANI     NSTEP      ;Get step pulse low
043F D352    OUT    HDFUNC     ;Output low step line
0441 F604    ORI     PSTEP      ;Set step line high
0443 D352    OUT    HDFUNC     ;Output high step line
0445 05      DCR    B           ;Update repeat count
0446 C23D04  JNZ    SLOOP      ;Keep going the required #
of tracks
0449 C9      RET

044A 60      HDDMA  MOV     H,B           ;Save the DMA address
044B 69      MOV    L,C
044C 229504  SHLD  HDADD
044F C9      RET

0450 DB50    WSDONE IN     HDSTAT      ;Wait for seek complete to
finish
0452 E604    ANI    COMPLT
0454 CA5004  JZ     WSDONE
0457 215C05  LXI    H,DRIVES-1 ;Update all seek in progress
flags
045A 0605    MVI    B,MAXHD+1
045C 23      WSUP  INX    H
045D 23      INX    H
045E 05      DCR    B
045F C8      RZ
0460 7E      MOV    A,M
0461 E601    ANI    1
0463 77      MOV    M,A
0464 C35C04  JMP    WSUP

0467 AF      HDSEC  XRA    A
0468 B1      ORA    C
0469 37      STC
046A C8      RZ
046B 3E15    MVI    A,SECCNT
046D 91      SUB    C
046E D8      RC
046F 79      MOV    A,C
0470 321605  STA    HDSECTR
0473 C9      RET

0474 79      HDHEAD MOV    A,C
0475 E607    ANI    7           ; 7 for M26 & M20, 3 for M10
0477 324105  STA    HEAD
047A C9      RET

047B CDF904  HDREAD CALL   HDPREP
047E D8      RC
047F AF      XRA    A
0480 D351    OUT    HDCMND
0482 2F      CMA
0483 D353    OUT    HDDATA
0485 D353    OUT    HDDATA
0487 3E01    MVI    A,RSECT
0489 D351    OUT    HDCMND      ;Read sector command

```

```

048B CDDF04          CALL    PROCESS
048E D8             RC
048F AF             XRA     A
0490 D351           OUT     HDCMND
0492 0680           MVI     B,SECLEN/4
0494 210000         LXI     H,0
0495 =              HDADD   EQU     $-2
0497 DB53           IN      HDDATA
0499 DB53           IN      HDDATA
049B DB53           RTLOOP  IN      HDDATA      ;Move four bytes
049D 77             MOV     M,A
049E 23             INX    H
049F DB53           IN      HDDATA
04A1 77             MOV     M,A
04A2 23             INX    H
04A3 DB53           IN      HDDATA
04A5 77             MOV     M,A
04A6 23             INX    H
04A7 DB53           IN      HDDATA
04A9 77             MOV     M,A
04AA 23             INX    H
04AB 05             DCR     B
04AC C29B04         JNZ    RTLOOP
04AF C9             RET

04B0 CDF904         HDWRITE CALL   HDPREP      ;Prepare header
04B3 D8             RC
04B4 AF             XRA     A
04B5 D351           OUT     HDCMND
04B7 2A9504         LHLD   HDADD
04BA 0680           MVI     B,SECLEN/4
04BC 7E             WTLOOP  MOV     A,M      ;Move 4 bytes
04BD D353           OUT     HDDATA
04BF 23             INX    H
04C0 7E             MOV     A,M
04C1 D353           OUT     HDDATA
04C3 23             INX    H
04C4 7E             MOV     A,M
04C5 D353           OUT     HDDATA
04C7 23             INX    H
04C8 7E             MOV     A,M
04C9 D353           OUT     HDDATA
04CB 23             INX    H
04CC 05             DCR     B
04CD C2BC04         JNZ    WTLOOP
04D0 3E05           MVI     A,WSECT      ;Issue write sector command
04D2 D351           OUT     HDCMND
04D4 CDDF04         CALL   PROCESS
04D7 D8             RC
04D8 3E10           MVI     A,WFAULT
04DA A0             ANA     B
04DB 37             STC
04DC C8             RZ
04DD AF             XRA     A
04DE C9             RET

04DF DB50           PROCESS IN   HDSTAT      ;Wait for command to finish

```

```

04E1 47          MOV      B,A
04E2 E602       ANI      OPDONE
04E4 CADF04     JZ       PROCESS
04E7 3E07       MVI      A,DSKCLK
04E9 D350       OUT      HDCNTL
04EB DB50       IN       HDSTAT
04ED E608       ANI      TMOUT          ;Timed out ?
04EF 37        STC
04F0 C0        RNZ
04F1 DB51       IN       HDRESLT
04F3 E602       ANI      RETRY          ;Any retries ?
04F5 37        STC
04F6 C0        RNZ
04F7 AF        XRA      A
04F8 C9        RET

04F9 DB50       HDPREP  IN       HDSTAT
04FB E620       ANI      DRVRDY
04FD 37        STC
04FE C0        RNZ
04FF 3E08       MVI      A,ISBUFF      ;Initialize pointer
0501 D351       OUT      HDCMND
0503 CD4005     CALL     BUILD
0506 F60C       ORI      OCH
0508 D352       OUT      HDFUNC
050A 3A4105     LDA      HEAD
050D D353       OUT      HDDATA        ;Form head byte
050F CD3405     CALL     DRVPTR
0512 7E        MOV      A,M          ;Form track byte
0513 D353       OUT      HDDATA
0515 3E00       MVI      A,0          ;Form sector byte
0516 =         HDSECTR EQU    $-1
0517 D353       OUT      HDDATA
0519 3E00       MVI      A,0
051A =         NKEY   EQU    $-1
051B D353       OUT      HDDATA
051D 23        INX      H          ;Bump to seek flag
051E 7E        MOV      A,M
051F 3C        INR      A          ;Update the seek in progress
flag
0520 E5        PUSH     H
0521 CC5004     CZ       WSDONE
0524 E1        POP      H
0525 7E        MOV      A,M
0526 3D        DCR      A          ;Test for delay also
0527 CCFB03     CZ       DELAY
052A 3E07       MVI      A,DSKCLK
052C D350       OUT      HDCNTL
052E 3E0F       MVI      A,WENABL
0530 D350       OUT      HDCNTL
0532 AF        XRA      A
0533 C9        RET

0534 2A4705     DRVPTR  LHLD     HDDISK
0537 EB        XCHG
0538 1600       MVI      D,0
053A 215D05     LXI      H,DRIVES

```

053D	19		DAD	D
053E	19		DAD	D
053F	C9		RET	
0540	3E00	BUILD	MVI	A,0
0541	=	HEAD	EQU	\$-1
0542	17		RAL	
0543	17		RAL	
0544	17		RAL	
0545	17		RAL	
0546	F600		ORI	0
0547	=	HDDISK	EQU	\$-1
0548	EEF0		XRI	OFOH
054A	C9		RET	
054B	E5	DMAHD	PUSH	H
054C	2A9504		LHLD	HDADD
054F	44		MOV	B,H
0550	4D		MOV	C,L
0551	E1		POP	H
0552	C9		RET	
0553	DB51	STATHD	IN	HDRESLT
0555	E603		ANI	3
0557	47		MOV	B,A
0558	DB50		IN	HDSTAT
055A	EE31		XRI	31H
055C	C9		RET	
055D	FFFF	DRIVES	DW	OFFFFH
055F	FFFF		DW	OFFFFH
0561	FFFF		DW	OFFFFH
0563	FFFF		DW	OFFFFH
0565			END	

# Optional Port Addresses

## C. OPTIONAL PORT ADDRESSES

HDCA controller switch settings:

S1	S2	S3	S4	S5	S6	PORT ADDRESSES
ON	ON	ON	ON	ON	ON	00 - 03
ON	ON	ON	ON	ON	OFF	04 - 07
ON	ON	ON	ON	OFF	ON	08 - 0BH
ON	ON	ON	ON	OFF	OFF	0CH - 0Fh
ON	ON	ON	OFF	ON	ON	10H - 13h
ON	ON	ON	OFF	ON	OFF	14H - 17h
ON	ON	ON	OFF	OFF	ON	18H - 1Bh
ON	ON	ON	OFF	OFF	OFF	1CH - 1Fh
ON	ON	OFF	ON	ON	ON	20H - 23H
ON	ON	OFF	ON	ON	OFF	24H - 27H
ON	ON	OFF	ON	OFF	ON	28H - 2BH
ON	ON	OFF	ON	OFF	OFF	2CH - 2FH
ON	ON	OFF	OFF	ON	ON	30H - 33H
ON	ON	OFF	OFF	ON	OFF	34H - 37H
ON	ON	OFF	OFF	OFF	ON	38H - 3BH
ON	ON	OFF	OFF	OFF	OFF	3CH - 3FH
ON	OFF	ON	ON	ON	ON	40H - 43H
ON	OFF	ON	ON	ON	OFF	44H - 47H
ON	OFF	ON	ON	OFF	ON	48H - 4BH
ON	OFF	ON	ON	OFF	OFF	4CH - 4FH
ON	OFF	ON	OFF	ON	ON	50H - 53H
ON	OFF	ON	OFF	ON	OFF	54H - 57H
ON	OFF	ON	OFF	OFF	ON	58H - 5BH
ON	OFF	ON	OFF	OFF	OFF	5CH - 5FH
ON	OFF	OFF	ON	ON	ON	60H - 63H
ON	OFF	OFF	ON	ON	OFF	64H - 67H
ON	OFF	OFF	ON	OFF	ON	68H - 6BH
ON	OFF	OFF	ON	OFF	OFF	6CH - 6FH
ON	OFF	OFF	OFF	ON	ON	70H - 73H
ON	OFF	OFF	OFF	ON	OFF	74H - 77H
ON	OFF	OFF	OFF	OFF	ON	78H - 7BH
ON	OFF	OFF	OFF	OFF	OFF	7CH - 7FH

## Optional Port Addresses

S1	S2	S3	S4	S5	S6	PORT ADDRESSES
OFF	ON	ON	ON	ON	ON	80H - 83H
OFF	ON	ON	ON	ON	OFF	84H - 87H
OFF	ON	ON	ON	OFF	ON	88H - 8BH
OFF	ON	ON	ON	OFF	OFF	8CH - 8FH
OFF	ON	ON	OFF	ON	ON	90H - 93H
OFF	ON	ON	OFF	ON	OFF	94H - 97H
OFF	ON	ON	OFF	OFF	ON	98H - 9BH
OFF	ON	ON	OFF	OFF	OFF	9CH - 9FH
OFF	ON	OFF	ON	ON	ON	A0H - A3H
OFF	ON	OFF	ON	ON	OFF	A4H - A7H
OFF	ON	OFF	ON	OFF	ON	A8H - ABH
OFF	ON	OFF	ON	OFF	OFF	ACH - AFH
OFF	ON	OFF	OFF	ON	ON	B0H - B3H
OFF	ON	OFF	OFF	ON	OFF	B4H - B7H
OFF	ON	OFF	OFF	OFF	ON	B8H - BBH
OFF	ON	OFF	OFF	OFF	OFF	BCH - BFH
OFF	OFF	ON	ON	ON	ON	C0H - C3H
OFF	OFF	ON	ON	ON	OFF	C4H - C7H
OFF	OFF	ON	ON	OFF	ON	C8H - CBH
OFF	OFF	ON	ON	OFF	OFF	CCH - CFH
OFF	OFF	ON	OFF	ON	ON	D0H - D3H
OFF	OFF	ON	OFF	ON	OFF	D4H - D7H
OFF	OFF	ON	OFF	OFF	ON	D8H - DBH
OFF	OFF	ON	OFF	OFF	OFF	DCH - DFH
OFF	OFF	OFF	ON	ON	ON	E0H - E3H
OFF	OFF	OFF	ON	ON	OFF	E4H - E7H
OFF	OFF	OFF	ON	OFF	ON	E8H - EBH
OFF	OFF	OFF	ON	OFF	OFF	ECH - EFH
OFF	OFF	OFF	OFF	ON	ON	F0H - F3H
OFF	OFF	OFF	OFF	ON	OFF	F4H - F7H
OFF	OFF	OFF	OFF	OFF	ON	F8H - FBH
OFF	OFF	OFF	OFF	OFF	OFF	FCH - FFH



# Performance Specifications

## D. PERFORMANCE SPECIFICATIONS

CLOCK SPEED: 2Mhz - 5Mhz

SYSTEM BUS: IEEE-696 (S-100)

DEVICE TYPE: SLAVE

PHYSICAL CHARACTERISTICS: 10.0" X 5.425" X .062"  
0.6 lbs. (S-100 standard size)

LAYOUT: 2 layer, plated, silkscreened

### DC POWER REQUIREMENTS

Nominal Voltage	Tolerance	Peak	Average
+8 Volts	Unregulated: filtered must be > +7.0 volts < +25.0 volts	2.5 Amps	1.75 Amps

POWER DISSIPATION: 20 Watts (Absolute maximum)  
15 Watts (Typical)

### ENVIRONMENTAL CONSIDERATIONS:

#### Operating:

Temperature	10 C to 40 C
Relative Humidity	10% to 90%
Elevation	Sea Level to 12,000 Ft.
Air	Filtered, positive

#### Non-Operating:

Temperature	-40 C to + 60 C
Humidity	10% to 90%
Elevation	Sea Level to 12,000 Ft.
Air	Unfiltered

## Performance Specifications

### ELECTRICAL INTERFACE:

Power Input:                    +8 V - pins 1 and 51  
                                  0 V - pins 50 and 100

### SIGNAL REQUIREMENTS

Signal interface to and from the HDCA module is accomplished through the 100 pin edge connector. These signals are defined in accordance with the IEEE-696 standards for interfacing to the S-100 bus.

### DRIVERS

All drivers on the board have the following specification:

High Level output:                2.4V min.  
                                     3.4V typ.  
                                     2.6 ma typ.

Low level output:                 0.5V max.  
                                     0.4V typ.  
                                     24 ma typ

High Impedance state:            20 ua typical leakage

### RECEIVERS

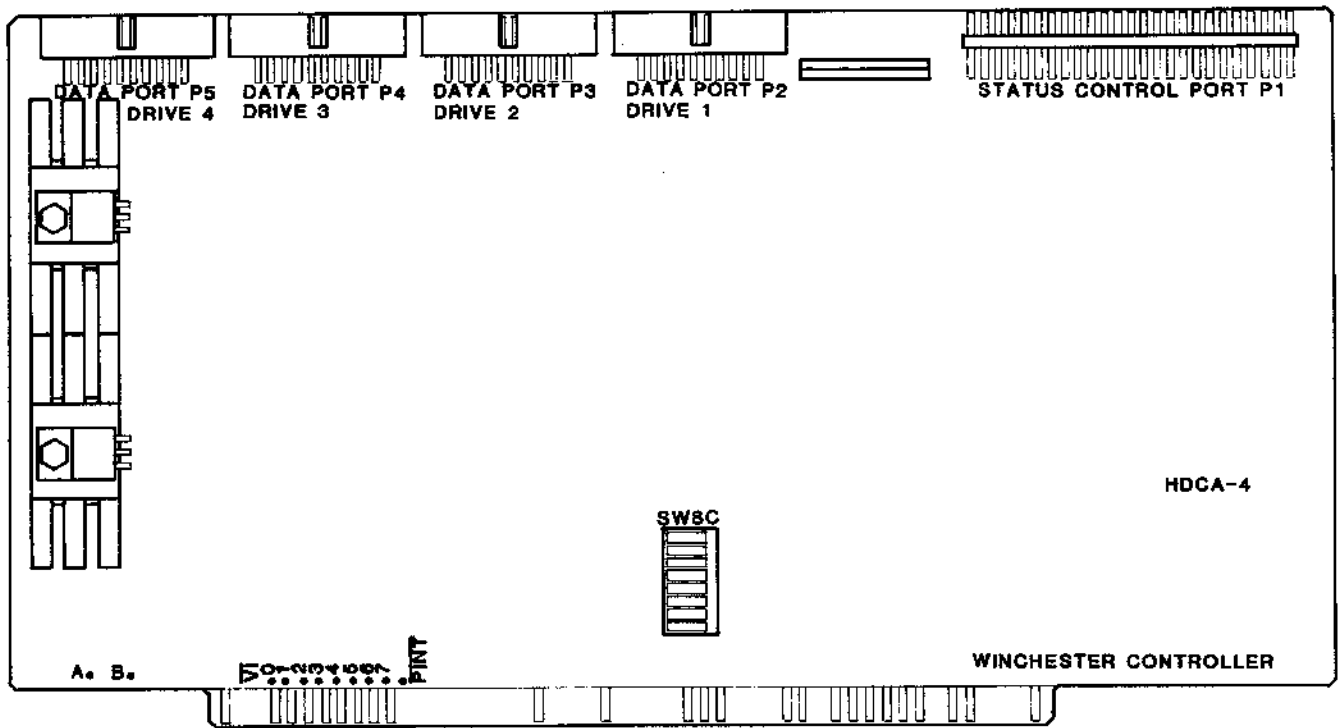
All receivers on the module have the following characteristics:

High level input:                 2.0 V min.  
                                     20 ua

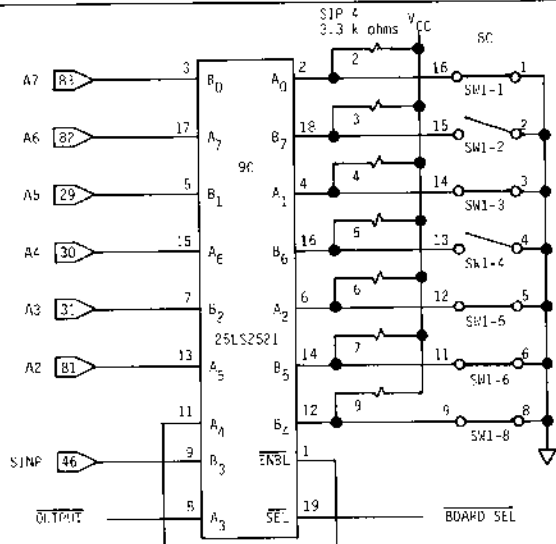
Low level input:                 0.7 V max  
                                     .4 ma

S-100 Signal lines A0 and A1 present 2 LS TTL loads (.8 ma) to the system bus.

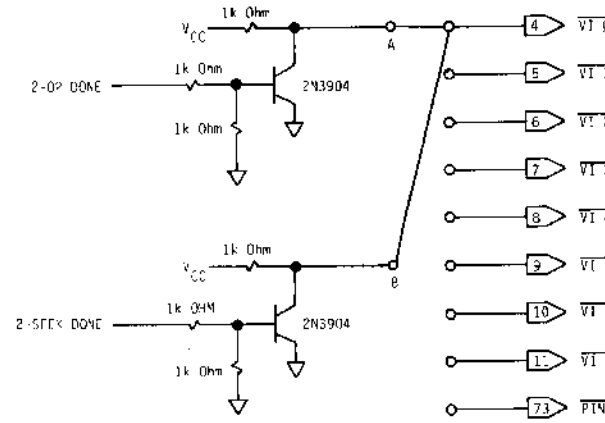
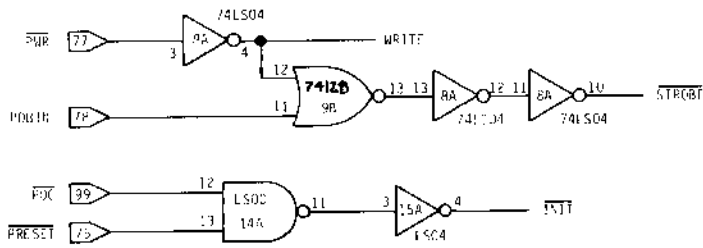
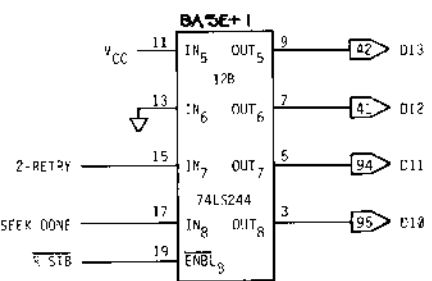
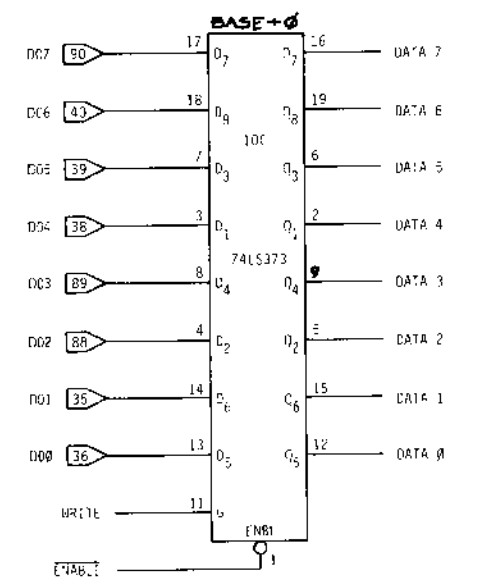
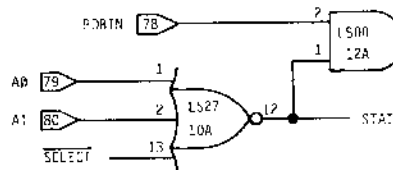
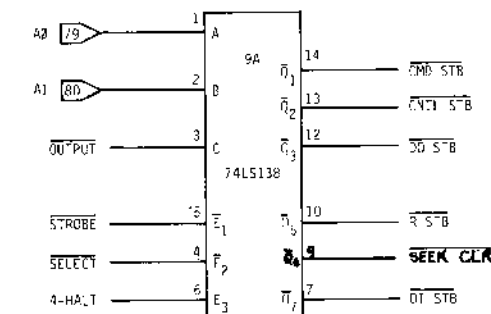
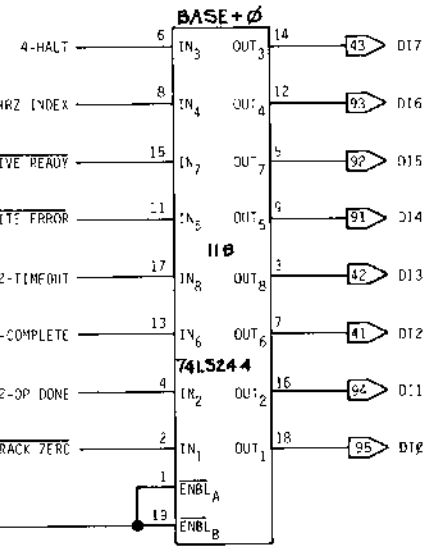
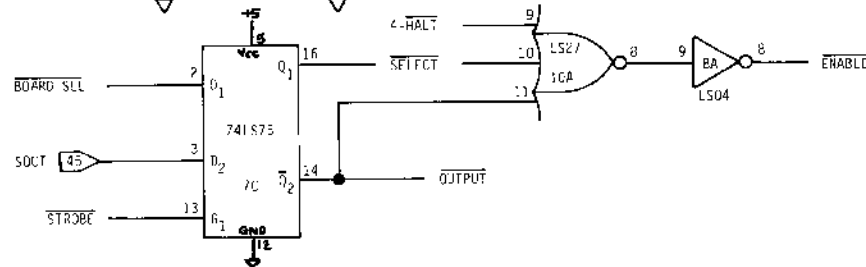
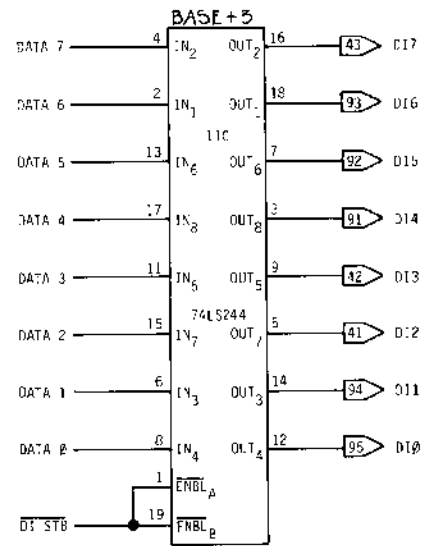
COMPONENT LAYOUT/SCHEMATICS

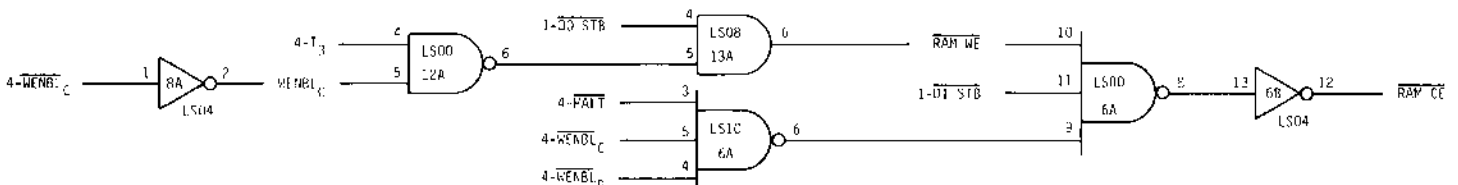
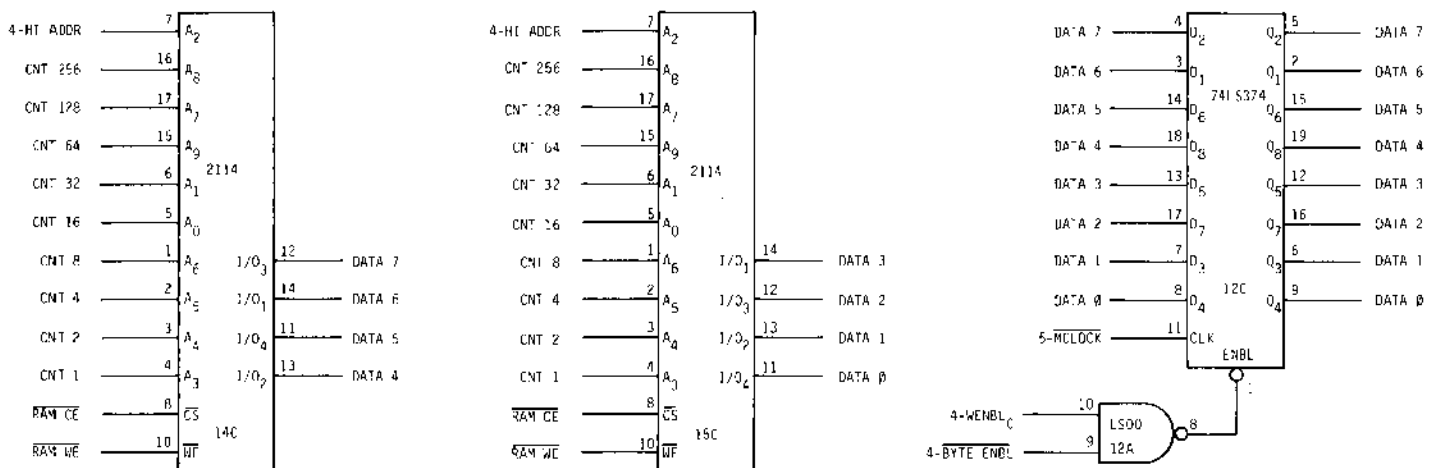
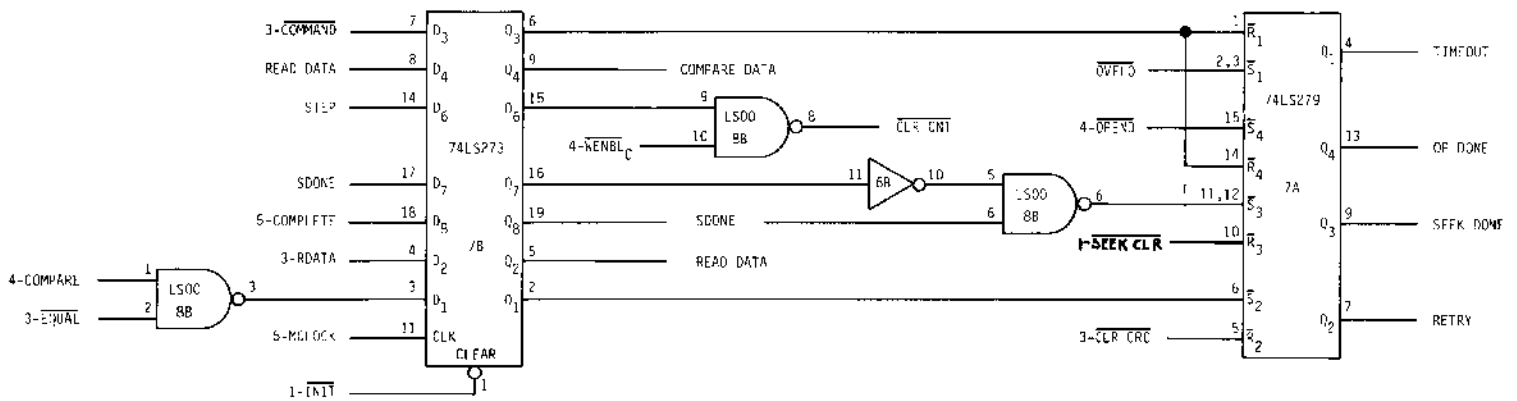
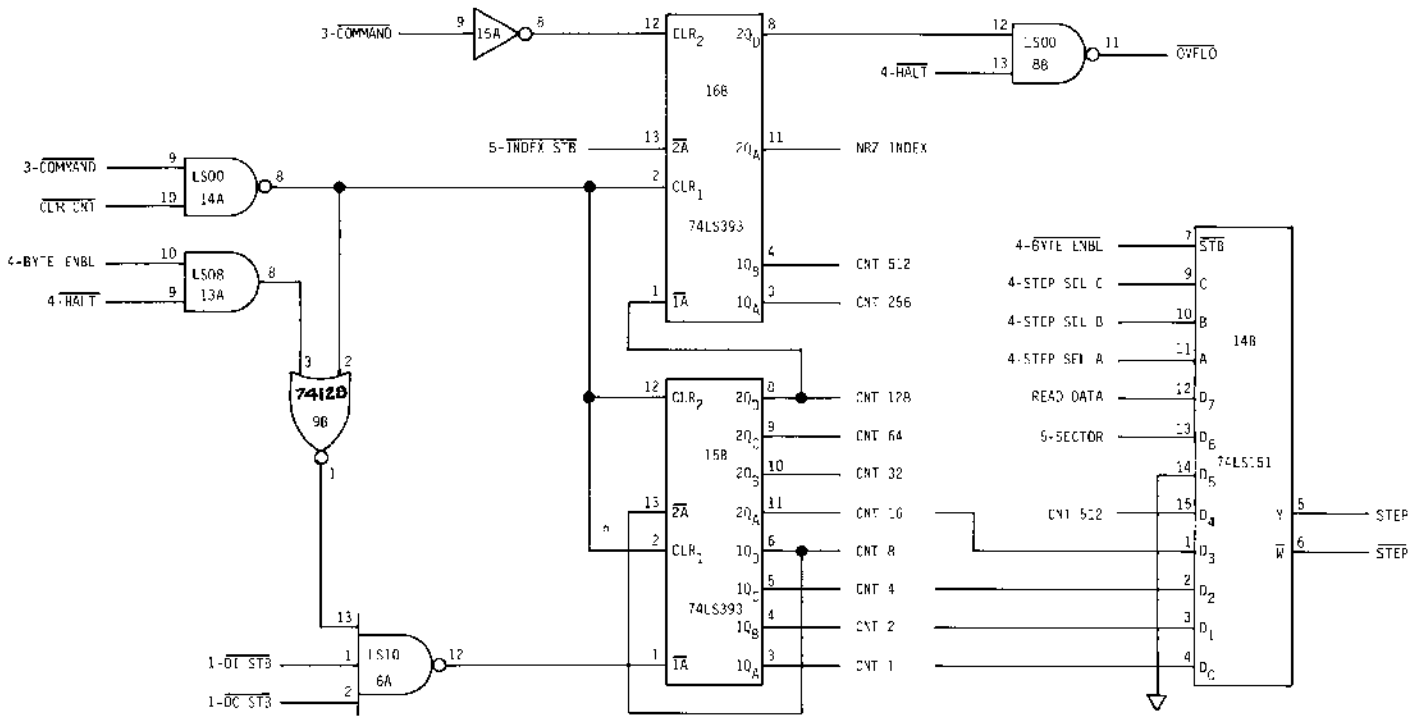


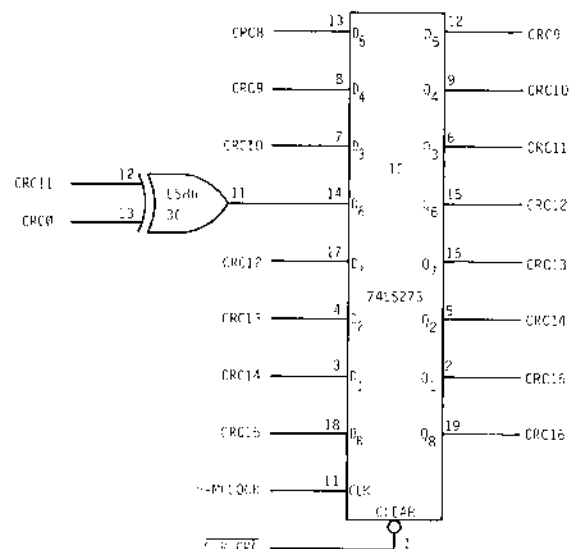
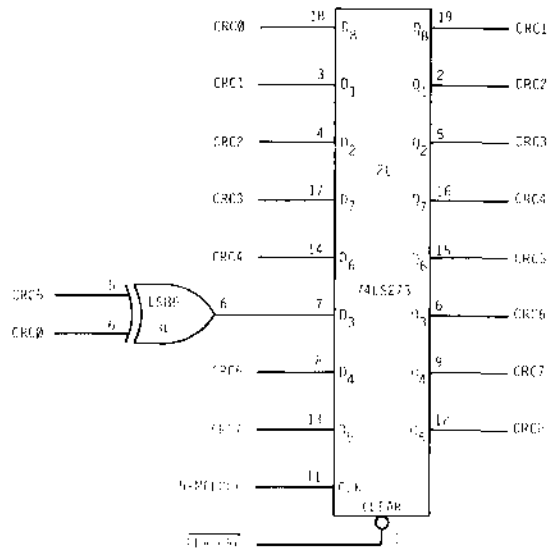
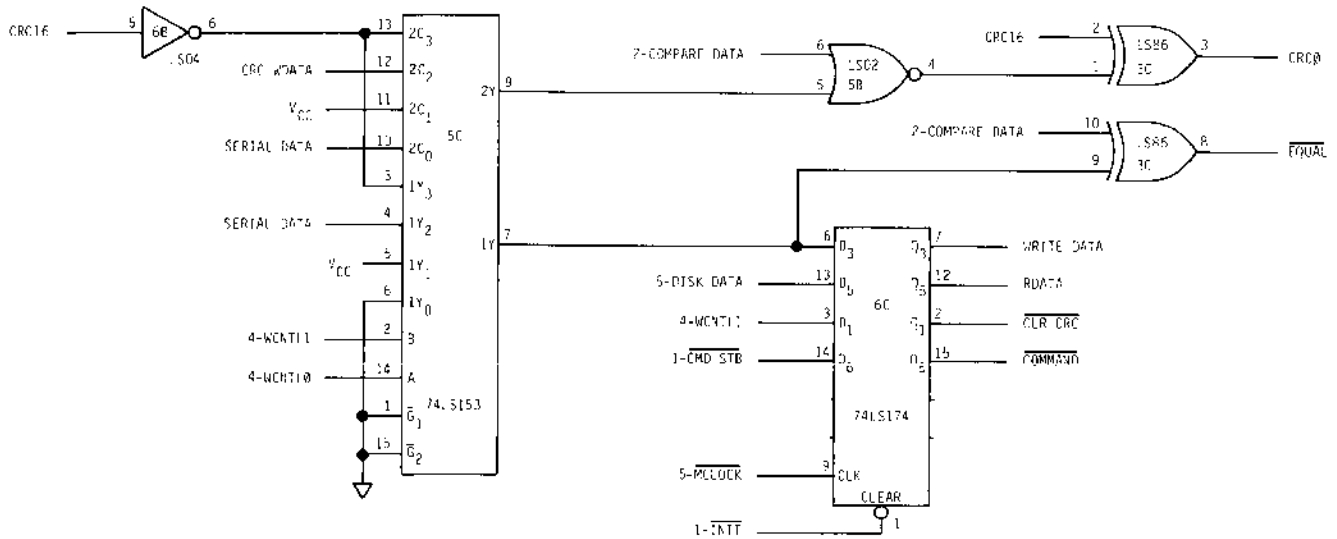
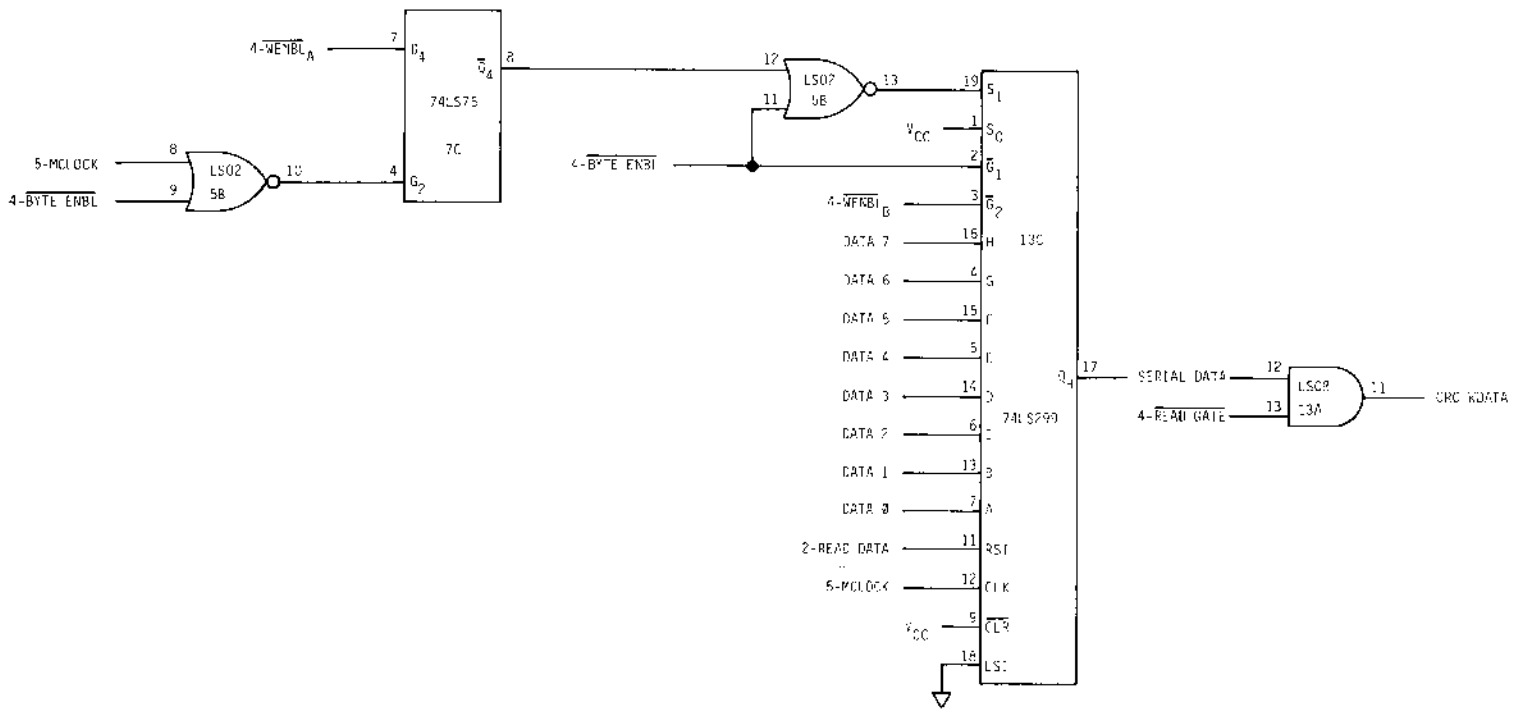
**HDCA-4 Hard Disk Controller Component Layout**

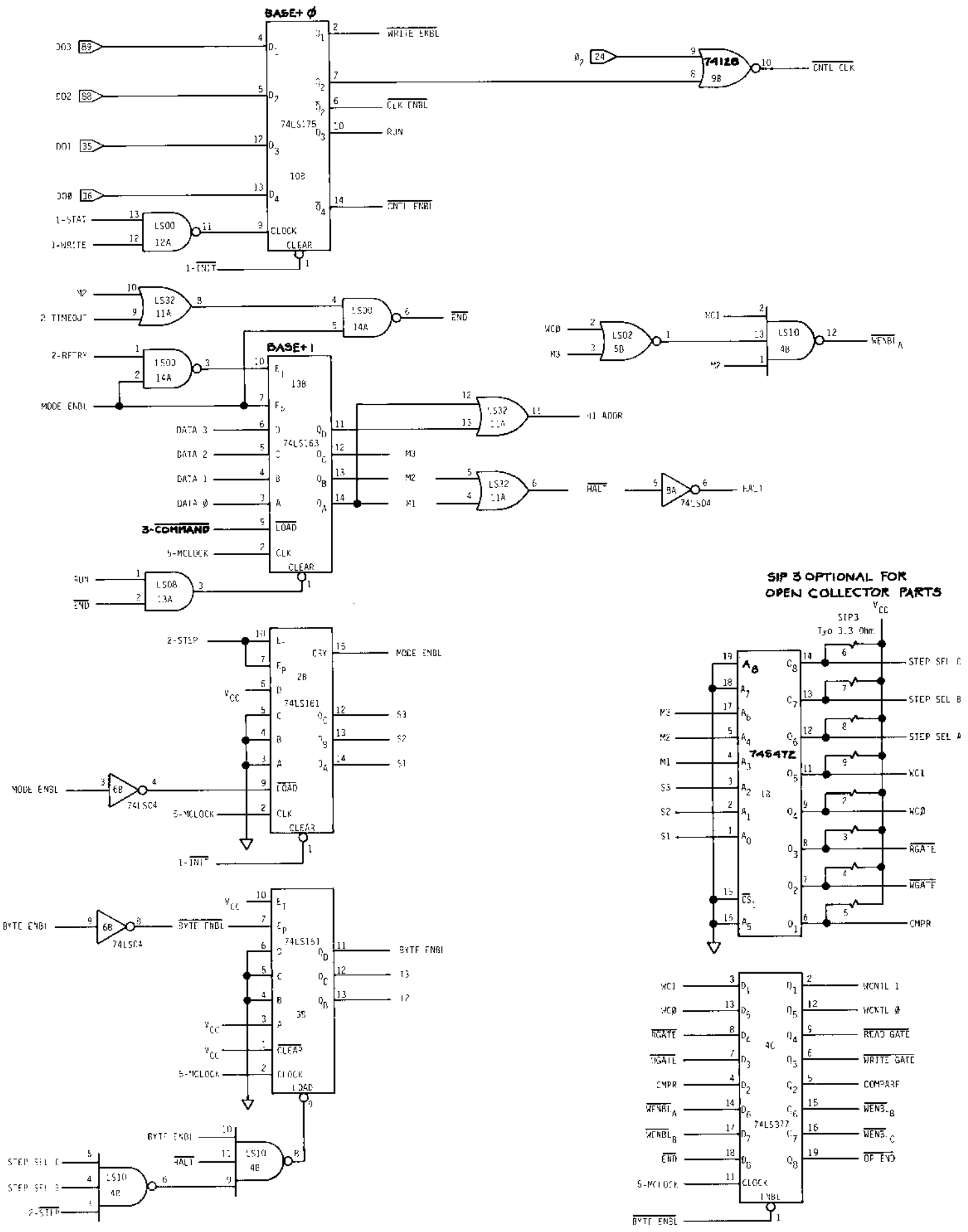


SWITCHES SET FOR  
BASE ADDRESS 50H

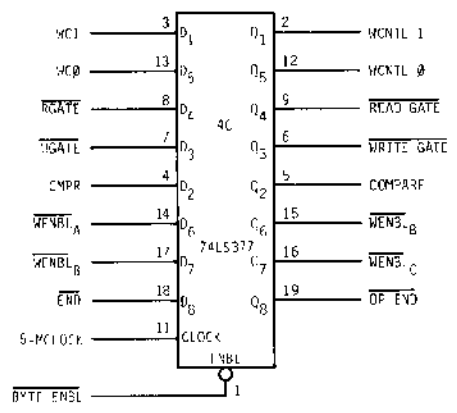
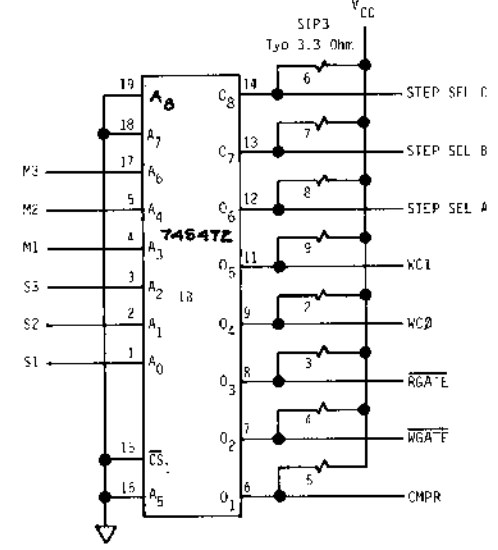




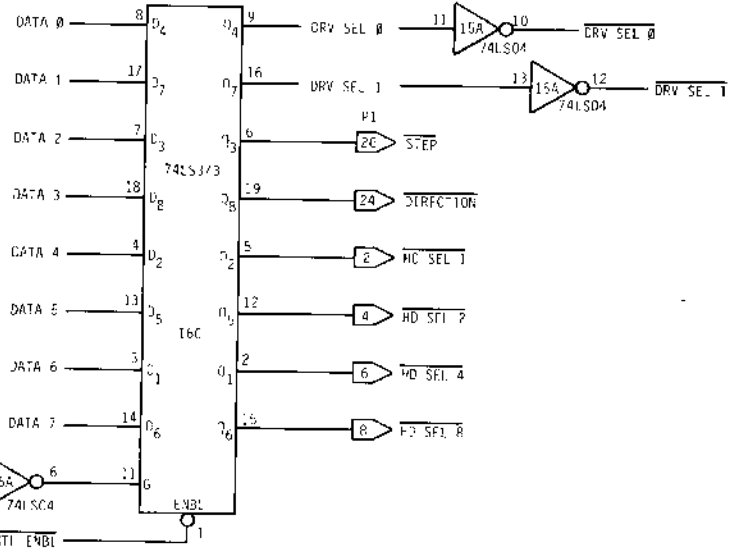
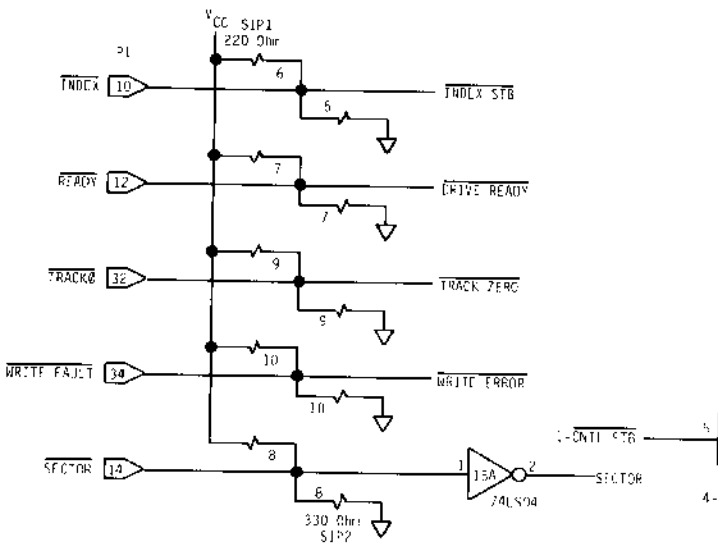
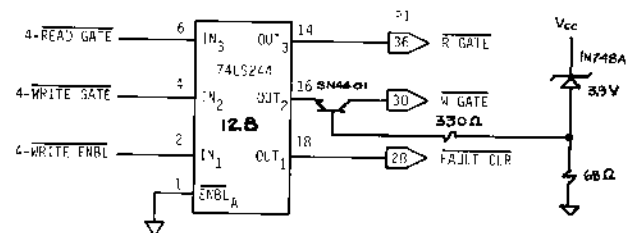
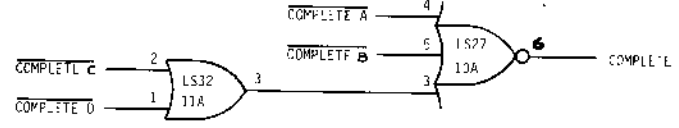
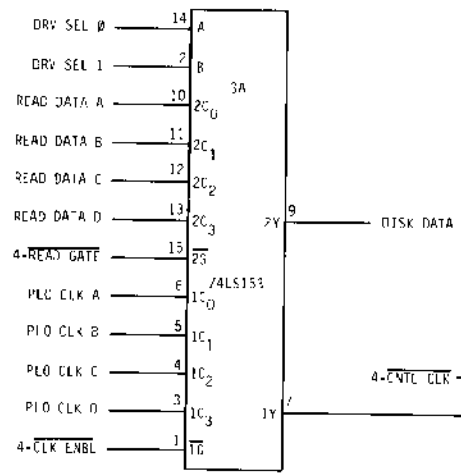
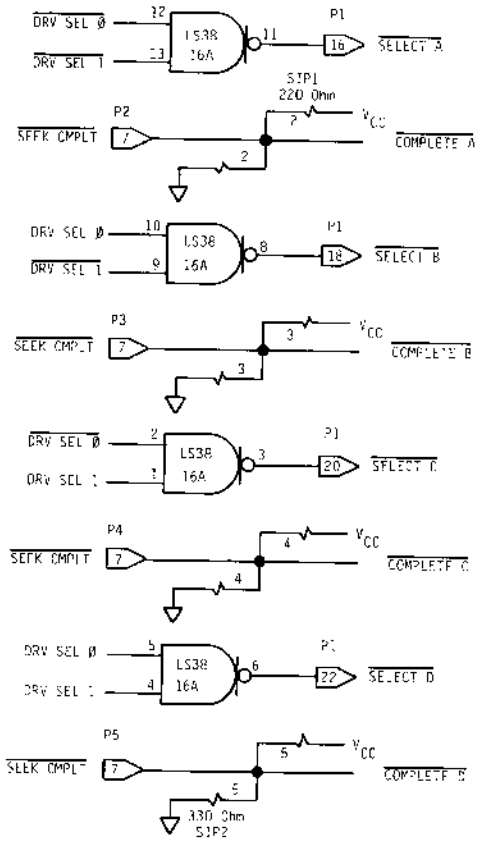
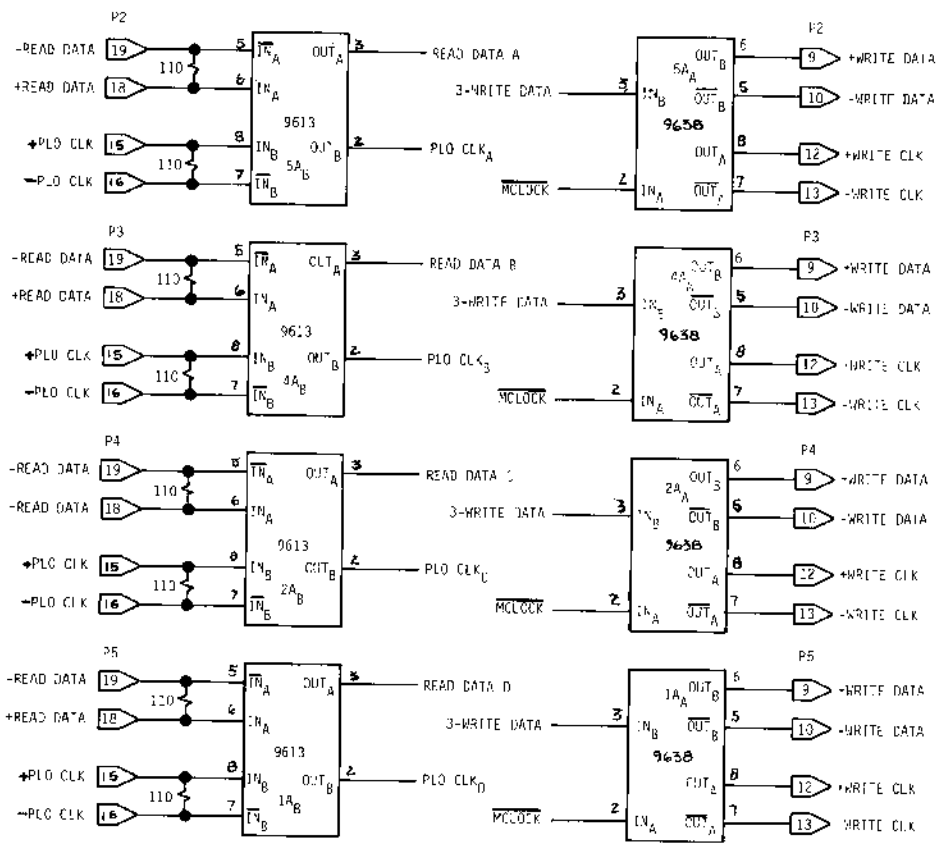




SIP 3 OPTIONAL FOR OPEN COLLECTOR PARTS







## Parts List

### PARTS LIST

[ ]	1	5" x 10" S-100 circuit board w/solder mask & legend	
[ ]	8	110 Ohm 1/4 watt 5% resistors	brown-brown-brown
[ ]	6	1K Ohm 1/4 watt 5% resistors	brown-black-red
[ ]	1	220 Ohm 10 pin 9 position SIP resistor array	SIP3
[ ]	1	330 Ohm 10 pin 9 position SIP resistor array	SIP4
[ ]	1	1K Ohm 10 pin 9 position SIP resistor array	SIP1
[ ]	1	3.3K Ohm 10 pin 9 position SIP resistor array	SIP2
[ ]	14	.01 - .1 microfarad Disk ceramic by-pass capacitors	
[ ]	2	1.0 - 2.0 microfarad tantalum capacitors	
[ ]	2	39 microfarad tantalum capacitors	
[ ]	1	8 position DIP switch array	
[ ]	2	6-32 x 5/16 machine screws	
[ ]	2	6-32 x 1/4 hex machine nuts	
[ ]	2	Heat sinks	
[ ]	3	Slide-on jumpers	
[ ]	3	20 position dual-in-line right angle header connectors	
[ ]	1	50 position dual-in-line right angle header connectors	
[ ]	17	14 pin low profile IC sockets	
[ ]	15	16 pin low profile IC sockets	
[ ]	2	18 pin low profile IC sockets	
[ ]	13	20 pin low profile IC sockets	
[ ]	3	74LS00 quad 2-input NAND gate	2G, 4C, 6B
[ ]	2	74LS02 quad 2-input NOR gate	3G, 5C
[ ]	3	74LS04 hex inverter	1G, 4C, 6A
[ ]	1	74LS08 quad 2-input AND gate	5A
[ ]	2	74LS10 triple 3-input NAND gate	2C, 5B

## Parts List

[ ]	1	74LS27 triple 3-input NOR gate	1F
[ ]	1	74LS32 quad 2-input OR gate	2F
[ ]	1	74LS37/74LS38/7437/7438 quad 2-input NAND buffer	1H
[ ]	1	74LS75 quad follower-latch	6C
[ ]	1	74LS86 quad 2-input XOR gate	4H
[ ]	1	74LS138 1 of 8 decoder	2H
[ ]	1	74LS151/74151 8 to 1 multiplexor	3C
[ ]	2	74LS153/74153 dual 4 to 1 multiplexor	1C,4G
[ ]	2	74LS161 hexadecimal counter	2A,2D
[ ]	1	74LS163 hexadecimal counter	2E
[ ]	2	74LS175 quad dual rail latch	3F,3H
[ ]	3	74LS244 octal TRI-STATE* buffer	6F,6G,6H
[ ]	4	74LS273 octal latch	3A,3B,5G,5H
[ ]	1	74LS279/74279 quad R-S flip-flop	4A
[ ]	1	74LS299 octal shift register with TRI-STATE outputs	4F
[ ]	3	74LS373 octal latch with TRI-STATE outputs	5E,5F,6E
[ ]	2	74LS393/74393 dual hexadecimal counter	3D,4D
[ ]	1	25LS2521 octal comparator	6D
[ ]	1	MMI6309 8 x 256 PROM	2B
[ ]	2	2114-3L low power 4 x 1024 RAM	3E,4E
[ ]	1	9612 dual balanced line transmitter	1Eb
[ ]	1	9613 dual balanced line receiver	1Ea
[ ]	2	7805 Monolithic 5 volt 1 amp regulators	

\*TRI-STATE is a registered trade mark of National Semiconductor

# Subject Index

## SUBJECT INDEX

### A

ASCII characters, 31  
Access keys, 43  
Application programs, 31  
Automatic Hard Disk Boot, 22

### B

Bad sector file, 42  
Bad sectors, 39  
Blocks, 38, 40  
Bootstrap EPROMs, 18  
Bootstrap Software, 19  
Bootstrap code, 18  
Bootstrap load software, 59  
Bootstrap operation, 18  
Bootstrap starting address, 18

### C

CALL instruction, 36  
COMMAND 0, 53  
COMMAND 1, 53  
COMMAND 3, 54  
COMMAND 5, 54  
COMMAND 7, 55  
COMMAND 8, 55  
COMPLETE (COMPLT), 42  
CONIN, 31  
CONOUT, 31  
CONST, 31  
CP/M jump instructions, 30  
CP/M memory locations, 31  
CP/M utilities  
  COMPLT, 42  
  GETSTAT, 41  
  HALT, 41  
  HOME, 37  
  INDEX, 41  
  OP DONE, 42  
  READ, 38  
  READY, 41  
  RETRY, 43  
  SDONE, 43  
  SELDRV, 40  
  SETDMA, 38  
  SETHEAD, 43  
  SETKEY, 43  
  SETSEC, 38  
  SETTRK, 37  
  TIMEOUT, 42

## Subject Index

### C, CONT.

WRITE, 39  
    WRITE FAULT, 42  
CRC Errors, 39  
Cable connections, 1, 42  
Carry flag, 36, 37, 38, 40  
Character I/O operations, 31  
Clock signals, 52  
Cold Booting the Hard Disk (BOOTH.D.COM), 22  
Command Console Processor (CCP), 31  
Communication with I/O devices, 29  
Control characters, 31  
Control-Z, 32

### D

DESPOOL, 31  
DIP switch, 5, 6  
DIR, 56  
DRVSL, 57  
DSKCLK, 52  
Data Input (DI) bus, 47  
Data buffer, 53  
Data field, 37  
Data transfer, 37, 44, 51  
Data transfer command, 47  
Disk drive addresses, 18  
Disk drive sensors, 37  
Drive I/O configurations, 31  
Drive select, 57  
Driver installation instructions, 33

### E

End-of-file condition, 32  
Error bit map, 40  
Error bits, 40  
    0, 39  
    3, 39  
Error conditions, 35, 36, 37, 38, 40  
Error flag  
    RETRY, 50  
Error handling routine, 36  
Error message, 32  
Errors  
    compare, 54  
    CRC, 39, 40, 43, 44, 50, 54  
    data transfer, 54  
    hard, 39, 42, 50  
    read, 39  
    RECORD NOT FOUND, 40  
    RECORD NOT FOUND, 39  
Exceptional signal, 42

## Subject Index

### F

FRENBL, 52

### G

GETSTAT, 41

### H

HALT, 41, 46

HDCA Modifications and Connections, 16

HDCA controller logic, 18

HDFIRM.ASM code, 35

HOME, 37

Head selection, 43, 56

Header field, 37

### I

I/O devices, 29

I/O drivers, 29

ILEVEL, 47

INDEX, 41

Implementing the Bootstrap, 18

Index pulse, 47

Installing a System on the Hard Disk, 21

Internal pointer reset command, 55

Interrupt lines, 33

### J

Jump instructions, 36

Jump table, 36

Jump-to-self instruction, 31, 33

Jumpers, 33

options, 5

### K

Key field, 42, 44

### L

LIST, 31

Logical devices, 31

### M

M10 and M20, 14

M10/M20 Daisy Chaining, 16

M26 Daisy Chaining, 16

M26 Drive Connection, 13

Master clock signal, 52

Memory sections, 33

Micronix operating system, 1, 3

## Subject Index

### M, CONT.

Modems, 29  
Modes  
    buffered, 57  
    normal, 57  
Morrow Designs System, 20  
Multiple drive systems, 38

### N

$\overline{\text{NFAULT}}$ , 47  
NHDSL, 56  
NOP command, 42  
NREADY, 47  
NSTEP, 57  
NTRCK0, 48

### O

$\overline{\text{OP DONE}}$  (OPDONE), 42  
OPDONE, 48  
Origin statement, 35  
Overlapped seeks and commands, 34

### P

$\overline{\text{PHASE2}}$ , 52  
PUNCH, 31  
Peripheral Interchange Program (PIP), 32  
Platters, 25  
Port addressing, 7  
Port  
    drive function/select, 55  
Ports  
    50H, 7  
    auxiliary status, 49, 54  
    base+2, 33, 43, 50  
    clear interrupt, 50  
    command, 51  
    control, 46, 51, 52  
    controller command, 53  
    controller data, 50  
    controller data, 46  
    data, 51  
    drive function, 46  
    function register, 52  
    main status, 46  
    status, 47  
Power down requirements, 24  
Power up requirements, 18  
Priority levels  
    OPDONE, 33  
    SDONE, 33

## Subject Index

### R

$\overline{R0}$  and  $R1$ , 49  
READ, 38  
READ GATE, 42, 47  
READER, 31  
READY, 41  
RET instructions, 31, 33, 36  
RETRY, 43, 49  
RUN, 52  
Read sector command, 54  
Read sector header command, 54  
Read/Write commands, 37  
Read/Write heads, 25, 37, 43  
Recording of data, 37  
Register locations, 45  
Registers  
  0, 45  
  1, 49  
  3, 50, 51  
  4, 51  
  5, 53, 55  
  A, 32, 36, 37, 39, 40, 41  
  B, 38, 41, 42  
  B-C, 38  
  C, 32, 37, 38, 40, 43, 44  
  CRC, 49, 54  
  Clear Interrupt, 50  
  I/O device, 45  
  Read Only, 50  
  auxiliary status, 33, 49, 54  
  command, 46  
  control, 55  
  controller, 45  
  controller status, 46  
  drive function, 52, 55  
  hardware, 35  
  main status, 33, 47, 50, 52, 54  
  master control, 51  
  shift, 49  
  status, 41, 53, 55, 57  
Requirements for I/O drivers, 29  
Reset pointer, 51  
Reset pointer command, 53

### S

$\overline{SDONE}$ , 50  
SEEK COMPLETE, 48  
SEEK DONE ( $\overline{SDONE}$ ), 43  
SET KEY special routine, 42  
SETDMA, 38  
SETDRV, 40  
SETHEAD, 43  
SETKEY, 43  
SETSEC, 38



## Subject Index

### S, CONT.

SETTRK, 37

Sector

description of, 26

Sectors, 37

Seek completes, 42

Shipping damage, 5

Stabilizing drive, 40

Step commands, 38, 42, 57

Step in commands, 25

Step out commands, 25

Steps for transferring data, 35

System console, 29

### T

TIME OUT (TIMEOUT), 42, 47

Track 0, 25, 44

Track numbering, 25

Tracks per disk, 25

### W

WPROT, 52

WRITE, 39

WRITE FAULT, 42, 47

WRITE FAULT bit, 40

WRITE GATE, 47

WRITE GATE signal, 40, 42

Winchester disk drives, 23

Write fault, 52

Write sector header command, 54