MSC-9305 UNIVERSAL DISK CONTROLLER
OPERATION AND MAINTENANCE MANUAL

PRELIMINARY

TP74

DOCUMENT CONTROL RECORD

Document: MSC-9305 Universal Disk Controller

REV.   DATE   DESCRIPTION

Original Release

TABLE OF CONTENTS

SECTION 4 - MAINTENANCE

SECTION 5 - THEORY OF OPERATION

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

GENERAL DESCRIPTION

1.1   INTRODUCTION

The MSC-9305 Universal Disk Controller is a disk controller that interfaces a Shugart Technology ST-506 disk drive to a GPIB universal bus (also known as IEEE-488 and HPIB). Incorporating this GPIB interface standard as the host interface, the controller is universally applicable with a minimum of effort and cost using readily available LSI interface circuits or multi-sourced adaptors for popular computer systems.

At the heart of the MSC-9305 controller is the MSC-9000 module, which applies proven LSI techniques to achieve the task of bringing the cost of the Controller and the cost of the Disk Drive into proper balance. Yet it incorporates all the sophisticated features of much larger, much more complex controllers including automatic error correction and full-sector data buffering.

1.2   GENERAL DESCRIPTION

The MSC-9305 package consists of a controller and data separator PCB that fit within a compact case, and the MSC-9305 Operation and Maintenance Manual. Optionally disk cables, a power cable and disk subsystem diagnostics which run on an HP-85 computer are also available.

The controller is assigned a DIP-switch selectable device address in the GPIB system. During normal operation, the controller provides all signals needed to communicate commands and status information with the host and to perform data transfer with the disk.

The controller also interfaces to the disk drive. It issues disk commands and drives the control lines to the disk that selects the unit, head and cylinder. It monitors lines from the drive that carry disk status and controls the data lines carrying data to and from the drive.

Data Separator PCB
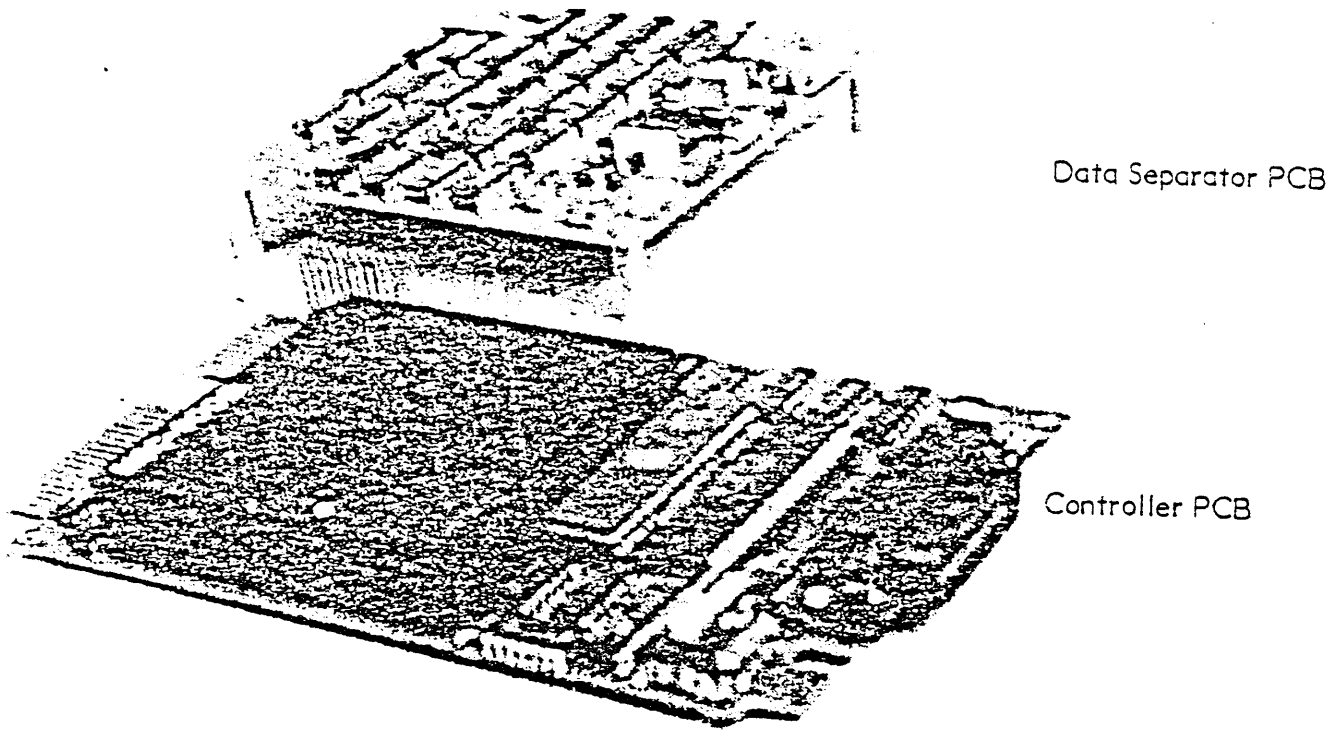
Controller PCB

Figure I-I. MSC-9305 Universal Disk Controller

1.3   CONTROLLER OPERATION

The MSC-9305   performs a set of commands for data transfer, status and diagnostic functions. See Table 1-1.

Table 1-1   Disk Commands

Mnemonic	Command Description
PARPOL	Parallel Poll
RDSJ	Read Device Specified Jump
RQSTS	Request Status
UCLR	Universal Clear
SLCLR	Selected Clear
INCLR	Interface Clear
SLFTST	Self Test
RSLTST	Read Self Test Results
ADRCD	Address Record
SEEK	Seek
RDADR	Read Address
CLRD	Cold Load Read
READ	Read
WRITE	Write
RLONG	Read Long
WLONG	Write Long
FORMAT	Format
VERIFY	Verify
WALT	Write Alternate Sector
STINT	Set Interleave
LPBK	Loopback

1.4   STANDARD AND OPTIONAL CAPABILITY

The standard capability of the MSC-9305 is summarized in Table 1-2. The options that can be specified with the MSC-9305 are summarized in Table 1-3.

Table 1-2 Controller Capability

Table 1-2 Controller Capability

Item	Description
Disk drive	Shugart Technology ST-506
Data transfer rate	Maximum: 200KB Minimum: host specified
Fault detection	Certain controller and drive fault conditions, are detectable and reported by the controller firmware in the normal course of operation.
Automatic head and cylinder switching	If a multiple-sector transfer occurs between disk and host memory and the end of a track is reached, the controller automatically advances the track. If the end of the track is reached and the track is the last one of the present cylinder, the controller automatically seeks to the next cylinder.
Error sensing, flagging and correction	A data stream being read from a disk is constantly being monitored for errors. If an error is detected, it is indicated to the host and burst errors of up to 11 bits long are automatically corrected by the controller.
Variables	The I/O device code of the controller is selectable with a dip-switch. The controller can be strapped to indicate if the disk is to be organized with 256 or 512 bytes per sector. Also, the selection of the parallel poll bit is switch-selectable.
Sector interleaving	This feature allows logically adjacent sectors on a given track to be mapped onto physical sectors on the track which are not adjacent. This feature might be used for either of two

reasons: (1) to smooth out the data rate in case the host CPU is slow or is overloaded with I/O activity and (2) to allow the host to read in sector I, perform some process on it, and then read in sector I+I without having to wait for disk latency.

Item	Description
Automatic position to alternate sector	If a sector that has been assigned an alternate sector is accessed, the controller automatically performs the data transfer at the alternate sector.
Data Buffer	The controller contains a full sector data buffer (512 bytes) to allow flexible data transfer rates with the host system without affecting data transfer integrity.
Position Verification	The controller automatically verifies that the heads are positioned (by reading identifier fields) before any data transfer is allowed.
Diagnostics	The controller command set includes a number of commands to thoroughly test the disk subsystem.
Disk Data Encoding	The controller contains a data separator and MFM encoder to receive and send the MFM data to the disk drive.
Sectors per Track	17 for 512 Bytes per Sector. 21 for 256 Bytes per Sector.

Table 1-3 Controller Options

Item	Description

Software Diagnostics   A diagnostic package can be provided to run
                              on an HP-85 computer to test the controller
                              and disk drive.

Disk Drive Cables   A cable set for the disk drive.

Power Connector   A power connector for the controller.

1.5   RELATED DOCUMENTS

Refer to the following documents for general description, operation, installation,
theory of operation and maintenance of related hardware and software.

1.   Shugart Technology ST-506 manual.
2.   IEEE-488 Interface Standard (1978).
3.   HP-85 Operating Manual.
4.   Texas Instruments TMS-9914 Manual.
5.   Intel 8048 Manual.
6.   MSC-9000 Series Product Specification.

SECTION 2

INSTALLATION

2.1   GENERAL

Section 2 contains information on inspecting the MSC-9305 for damage before and during unpacking.   It also describes the space, power, and environmental requirements of the controller and procedures for cable interconnection.

2.2   PRELIMINARY INSPECTION

All parts comprising the MSC-9305 are shipped in one container consisting of an inner envelope and an outer cardboard box. Before unpacking the unit from its shipping container, inspect the container for any obvious damage that might have occurred during shipping.

If in-transit damage to the container is obvious, contact the carrier and shipper immediately, and specify the nature and extent of damage. Do not open the container until the carrier's representative has inspected the damage. Inspect any accompanying disk drive containers in the same way.

2.3   UNPACKING AND INSPECTION

CAUTION

Use knives or other tools carefully during unpacking.

To unpack the controller, first open the outer cardboard box and remove the inner envelope. As each item is unpacked, inspect for damage and check against the shipping list. If any part or accessory is missing, notify the shipper of the shortage immediately.   (If a   claim for damages is filed, keep the original shipping containers.)

NOTE

> Refer to the appropriate manual from the list of
> related documents in section 1.5 for special
> instructions on the unpacking of any
> accompanying disk drives.

2.4   CONTROLLER REQUIREMENTS

For proper operation, operating environment and power supply must meet MSC-9305 requirements.

2.4.1   Operating Environment

Table 2-1 shows the minimum and maximum operating and storage limits for temperature, humidity and altitude.

Table 2-1 Environmental Limits

	Operating		Storage	
	Min	Max	Min	Max
Temperature range				
Fahrenheit	32	131	-40	167
Celsius	0	55	-40	75
Relative Humidity	10%	95%	10%	95%
at 40°F max				
wet bulb temperature, no				
condensation				
Altitude range				
Feet	Sea level	10,000	Sea level	15,000
Meters	Sea level	3,048	Sea level	4,572
Mag radiation			.5 gauss	

2.4.2 Space Requirements

Figure 2-1 shows the physical dimensions of the MSC-9305.

2.4.3   Power Requirements

The controller operates on +5 volts DC. The unit requires approximately 3 amps of +5 volts current.

2.4.4   Cooling Requirements

The controller must be mounted in an area with proper ventilation to dissipate 15 watts.
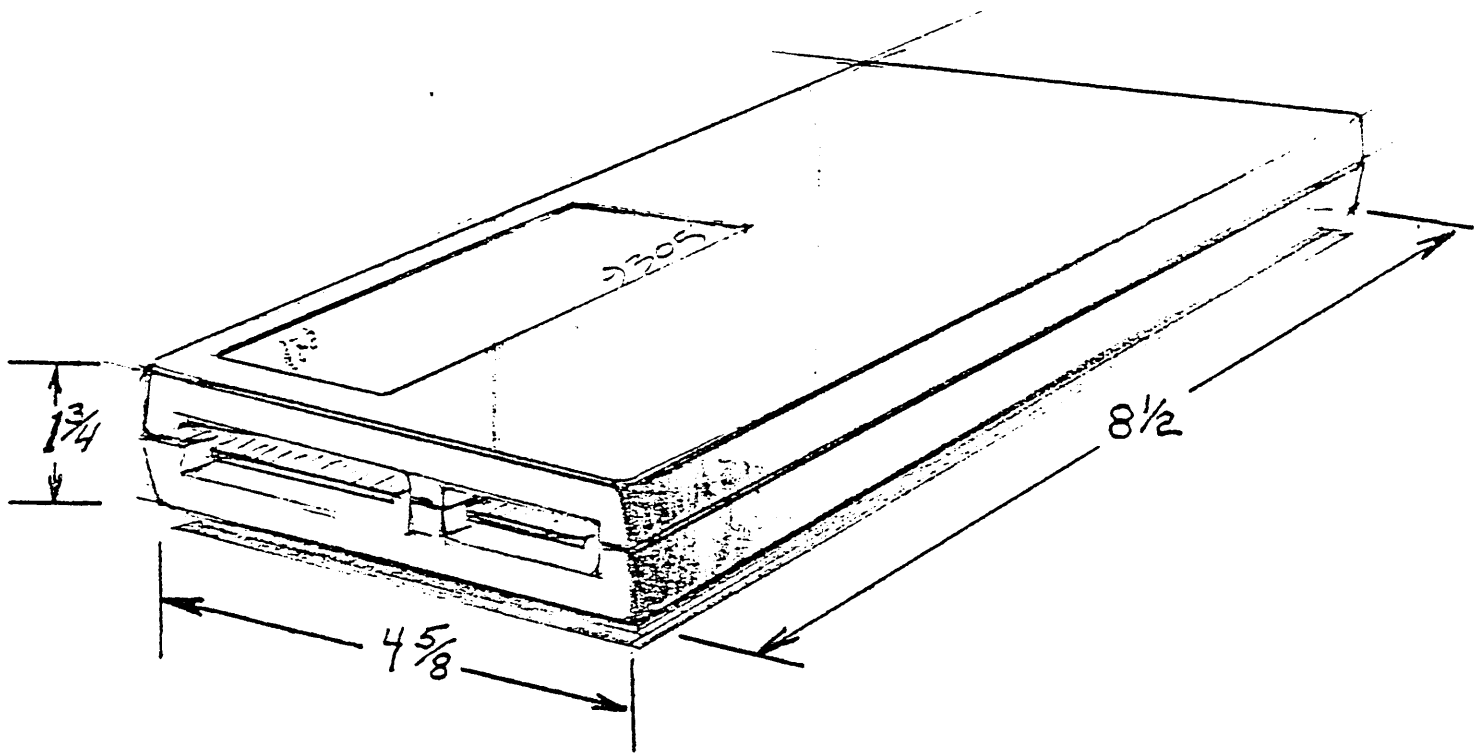
Figure 2-1. Disk Controller

2.5   INSTALLATION PROCEDURES

The various options must be switch selected or strapped on the controller card. The cables must be interconnected to the disk drive, power supply, and host IEEE-488 Bus.

2.5.1   Strapping Procedure

Three controller variables can be strapped to meet a variety of applications. These consist of the I/O device code address, the selection of the parallel poll bit, and the selection of 256 or 512 bytes per sector. Refer to Figure 2-2 for the locations of the various switches and straps.

2.5.1.1 Device Address Selection

The device address is an octal address selectable for the GPIB bus in the range of 00_8 through 37_8. To select an address refer to Figure 2-3.

The selection switch shown in the figure illustrates the switches and the corresponding address bit associated with each switch. Set switches where binary ones would appear. The equivalent binary value for 26_8 for example, is $010\text{-}110_2$. To select this address, switches 2, 3 and 5 are set. The OPEN position is equivalent to a 0.

The device address is typically set at the factory to 12_8.

2.5.1.2   Parallel Poll Bit Selection

Any of the 8 bits on the GPIB data bus can be defined as the parallel poll bit. The switch positions 6, 7, and 8 are encoded to represent which parallel poll bit is desired. Switch 6 is the least significant position and an open switch represents a zero. See figure 2-3.

5.5.1.3   Sector Size Strapping

The number of bytes per sector can be strapped for 256 or 512. The strapping block shown in Figure 2-3 illustrates the strapping pins and the corresponding capacity specified when a strap is inserted. For the strapping block location on the controller, see Figure 2-2.
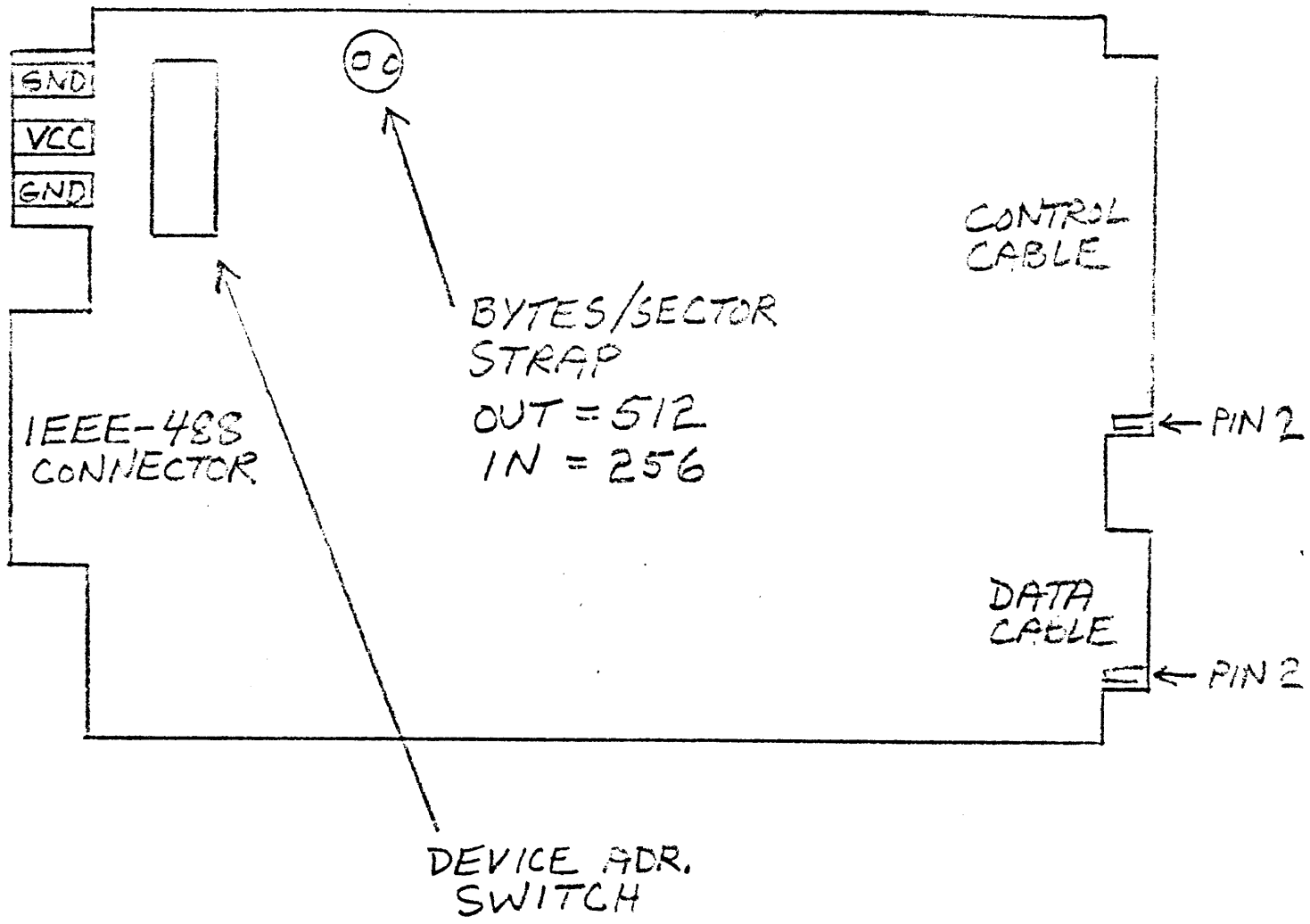
Figure 2-2. Controller Straps

OPEN

1 A_0

2 A_1

3 A_2

4 A_3

5 A_4

6
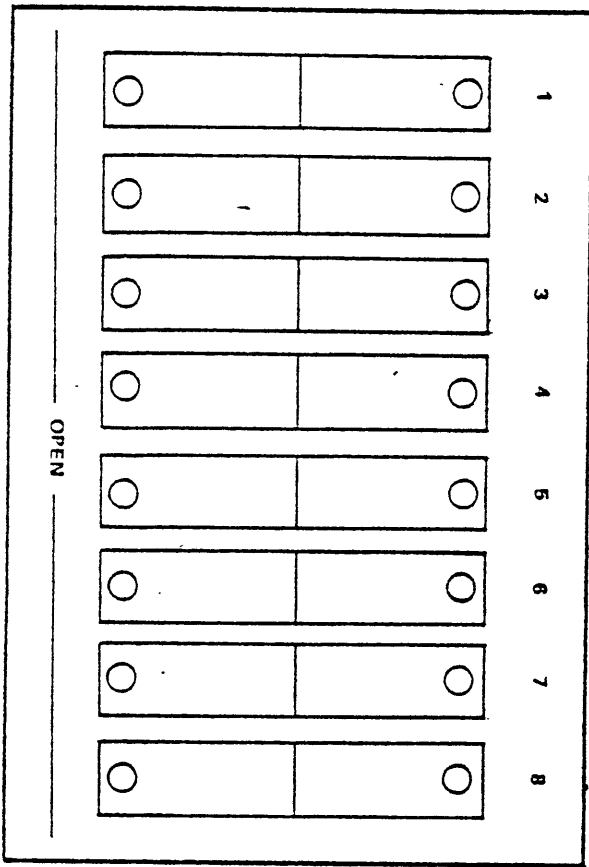
7 } USED FOR PARALLEL POLL BIT SELECTION (PPB)

8

⇓

PPB BIT	SW 6	SW 7	SW 8
0	O	O	O
1	C	O	O
2	O	C	O
3	C	C	O
4	O	O	C
5	C	O	C
6	O	C	C
7	C	C	C

O = OPEN
C = CLOSED

Figure 2-3. Device Address Selection

2.5.2 Installation Procedure

Before installation of the controller or disk drive the power supplies should be checked for proper voltages. all power, including host processor power should then be turned off. All the cables should then be interconnected as shown in Figure 2-6. After checking for proper interconnections and controller strapping (see section 2.5.1) power can then be applied.

Four cables are required to interconnect the Host, controller and disk drive. See Table 2-2. Figure 2-4 shows the interconnections as well as the required connectors.

Table 2-2 Interconnection Cables

Cable	Description
Disk Control Cable	A 34-line cable containing Control signals that connects the disk drive to the controller. Also called the daisy-chain interface cable, it carries control, select and status signals.
Disk Data Cable	A 20-line cable containing data signals that connects controller and disk drive. Also called the radial interface cable, it carries serial read and serial write data as well as the unit selected signal.
Power Cable	A cable from the +5 volt power supply to the controller.
Host Cable	A standard IEEE-488 cable.

IEEE-488
(1978)

AMP 88373-6

HOST

MSC-9305

DATA
CABLE

ST-506

AMP 88373-3

CONTROL
CABLE

AMP 640136-3
WITH CONTACT 350011-1

+5VDC
POWER SUPPLY
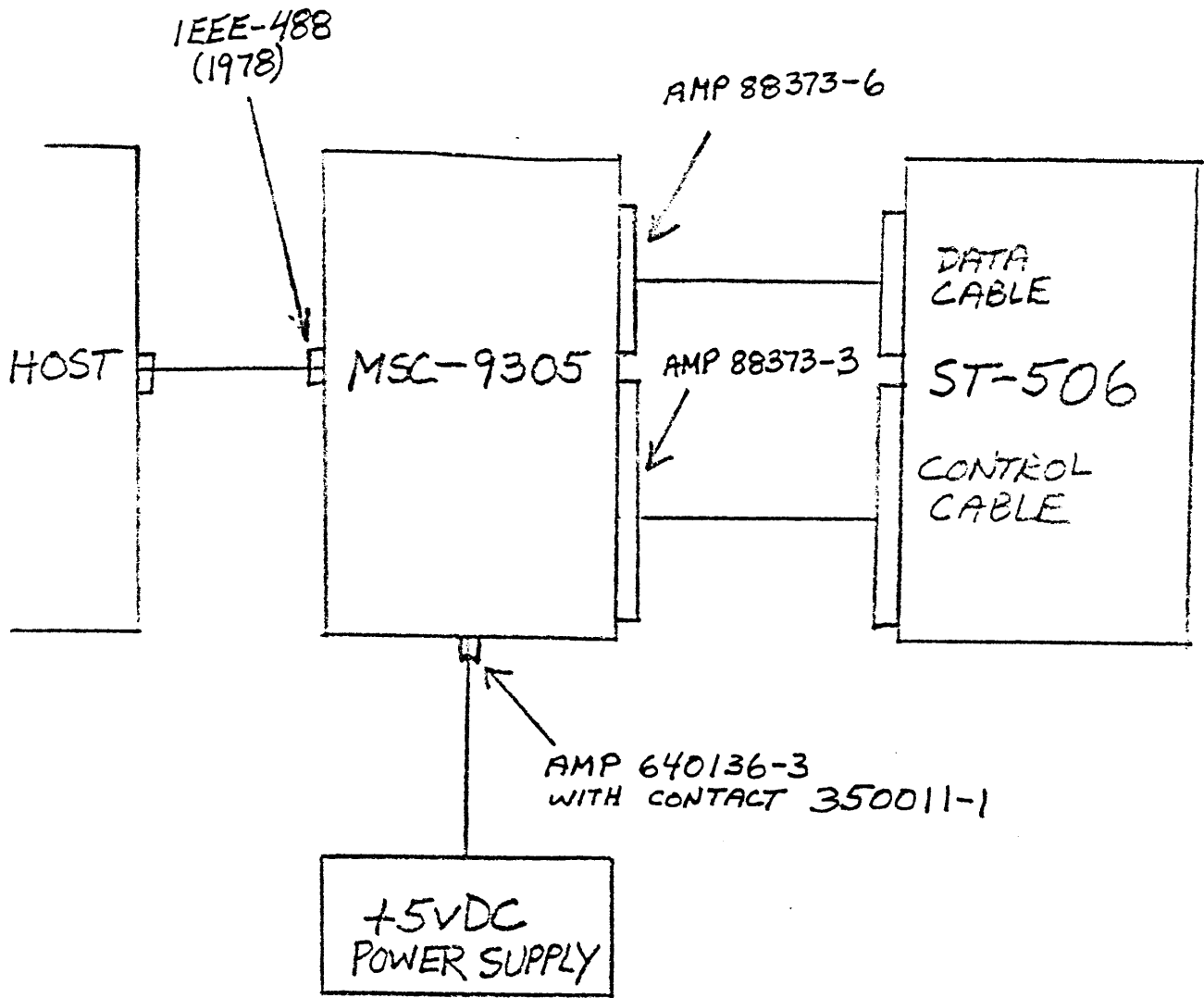
Figure 2-4. Interconnection Diagram

COMMUNICATION PROTOCOL

3.1   General

The MSC-9305 communicates with a host system using the GPIB bus.   The communication consists of a control byte from the host which has the disk controllers talk or listen address, followed by one or more bytes to describe the task, followed by a number of bytes to transfer the data and finally an unlisten or untalk command which completes the sequence.   The actual communication sequences are described in each command description. Before issuing any disk data transfer commands an address record or seek command must be issued.

Note that the controller has the same talk and listen address.

3.2   Parallel Poll

When at idle, the controller will respond to a parallel poll by asserting the bit specified by switches 6-8 of the controller address DIP switch. Upon receiving a command, the controller stops asserting the bit when polled and re-asserts it upon command completion.   The Host must have issued the termination untalk or unlisten for a command before the controller will return to idle and assert parallel poll.

3.3   Read - Device - Specified - Jump - Byte

The DSJ byte provides a quick means of checking the status of the controllers last operation.   The sequence is as follows:

```
ATN (X 1 0 T T T T T)   -   Controller talk address
    (D D D D D D D D)   -   DSJ byte from controller
ATN (X 1 0 1 1 1 1 1)   -   Untalk
```

This is the only command which begins with a talk address. All others begin with this controller addressed to listen. The DSJ bytes is encoded as follows:

0	-	The last operation completed without error.
1	-	The last operation completed with error and it is necessary to issue the READ-STATUS command to further define the error.
2	-	The last operation was a successful interface-clear, universal-clear, selected-clear or self-test command.

3.4   Request-Status

Returns controller status. The sequence is as follows:

```
ATN (X 0 1 A A A A A)  -  Controllers listen address
    (X X X 0 0 0 1 1)  -  Request status command
    (X X X X X X X X)  -  Unused
ATN (X 0 1 1 1 1 1 1)  -  Controllers talk address
ATN (X 1 0 A A A A A)  -  Unlisten
    (X X X C C C C C)  -  Controller termination status
    (0 0 0 0 0 0 S S)  -  Zero byte
    (S S S S S S S S)  -  Drive status high
    (S S S S S S S S)  -  Drive status low
ATN (X 1 0 1 1 1 1 1)  -  Untalk
```

Controller termination status is encoded as follows in hexadecimal:

00	-	Normal Termination
01	-	Illegal command byte
13	-	Drive Error. The error is further defined in the drive status bytes.

The drive status bytes are encoded in hexadecimal as follows:

0000	-	Normal Termination
8002	-	Drive Not Ready
8003	-	1 Second Seek Timeout
8004	-	Invalid track 00 indication from drive
8005	-	all ID fields bad on track
8006	-	Sector not found

8007	-	Sector not found and ID ECC error
8008	-	Position error
8009	-	Data transfer start error
800A	-	Write fault error
800B	-	Timeout waiting for index or address mark
800C	-	Invalid disk address
800D	-	Uncorrectable ECC error
801X	-	Correctable ECC Error
8020	-	Write alternate error
8021	-	Alternate sector is defective
8022	-	Alternate already assigned
8023	-	Direct access to alternate sector
8024	-	Defective MSC-9056 Module
8025	-	Defective buffer memory inside module
8026	-	Defective ECC circuitry inside module
8027	-	Defective controller program memory inside module
8028	-	Illegal address mark pulse during diagnostic
8029	-	Illegal interleave table
8080	-	8048 program storage failure
8081	-	8048 data storage failure

X = length of the burst error which can be 1 to B to note if correction span was 1 to 11 bits.

3.5   Universal-Clear

Resets the controller. The sequence is as follows:

ATN (X 0 0 1 0 1 0 0)

The controllers address registers are set to cylinder, head and sector zero and a recalibrate is issued to the drive. On successful completion of the recalibrate, the DSJ byte is set to 2.

3.6   Selected-Clear

Resets the controller. The sequence is as follows:

ATN (X 0 1 A A A A A) - Controllers listen address
ATN (X 0 0 0 0 1 0 0) - Selected Clear
ATN (X 0 1 1 1 1 1 1) - Unlisten

The controllers address registers are set to cylinder, head and sector zero and a recalibrate is issued to the drive. On successful completion of the recalibrate, the DSJ byte is set to 2.

3.7   Interface-Clear

Asserting the interface clear line resets the controller. Its address registers are set to cylinder, head and sector zero. A recalibrate is not issued to the drive as it is with universal and selected clear.

3.8   Initiate-Self-Test

Causes the controller to check itself for proper operation. The sequence is as follows:

ATN (X 0 1 A A A A A) - Controllers listen address
    (0 0 0 1 1 0 1 1) - Initiate self test command
    (0 0 0 0 0 0 0 0) - Zero byte
ATN (X 0 1 1 1 1 1 1) - Unlisten

If the test completes without error, the DSJ byte is set to 0. If an error is detected the DSJ byte will be set to 1 and the result may be obtained by issuing the Read-Self-Test-Result command sequence.

3.9   Read-Self-Test-Result

Returns the result of the last initiate-self-test command sequence. The sequence is as follows.

ATN (X 0 1 A A A A A) - Controller's listen address

```
     (0  0  0  1  1  1  0  0)   -   Read self test result command
     (0  0  0  0  0  0  0  0)   -   Zero byte
ATN (X  0  1  1  1  1  1  1)   -   Unlisten
ATN (X  1  0  A  A  A  A  A)   -   Controller's talk address
     (R  R  R  R  R  R  R  R)   -   Self test result byte
ATN (X  1  0  1  1  1  1  1)   -   Untalk
```

The self test result byte is encoded the same as the drive status byte.

3.10   Address-Record

Loads the controllers disk address registers.   This command (or the SEEK command) must be issued before any data transfer command with the disk.   The sequence is as follows:

```
ATN (X  0  1  A  A  A  A  A)   -   Controller's listen address
     (X  X  X  0  1  1  0  0)   -   Address record command
     (0  0  0  0  0  0  0  0)   -   Zero byte
     (C  C  C  C  C  C  C  C)   -   Cylinder high
     (C  C  C  C  C  C  C  C)   -   Cylinder low
     (H  H  H  H  H  H  H  H)   -   Head
     (S  S  S  S  S  S  S  S)   -   Sector
ATN (X  0  1  1  1  1  1  1)   -   Unlisten
```


3.11   Seek

Loads the controllers disk address registers, seeks to the specified cylinder, selects the specified head and reads an ID from the disk to validate the position.   This command (or the address-record command) must be issued before any data transfer command with the disk.   The sequence is as follows:

```
ATN (X  0  1  A  A  A  A  A)   -   Controllers listen address
     (X  X  X  0  0  0  1  0)   -   Seek command
     (0  0  0  0  0  0  0  0)   -   Zero byte
     (C  C  C  C  C  C  C  C)   -   Cylinder high
     (C  C  C  C  C  C  C  C)   -   Cylinder low
```

```
        (H  H H H H H H H)     -    Head
        (S  S S S S S S S)     -    Sector
ATN (X  0  1  1  1  1  1  1)    -    Unlisten
```

Since all data tranfer commands imply a seek to the address in the disk address registers, the seek command is provided only for diagnostic purposes.

3.12   Read-Address

Returns the contents of the controllers disk address registers and the residual count from the most recent read, write, cold-load-read, format or verify command. If the most recent of these commands is completed without error the residual count will not be meaningful. If it is completed with error the count will be the number of sectors not transferred. The sequence is as follows:

```
ATN (X  0  1  A A A A A)     -    Controller's listen address
        (X X X 1  0  1  0  0)     -    Read-Address command
        (0  0  0  0  0  0  0  0)     -    Zero byte
ATN (X  0  1  1  1  1  1  1)     -    Unlisten
ATN (X  1  0  A A A A A)     -    Controller's talk address
        (C C C C C C C C)     -    Cylinder high
        (C C C C C C C C)     -    Cylinder low
        (H H H H H H H H)     -    Head                Returned by MSC-9305
        (S S S S S S S S)     -    Sector
        (K K K K K K K K)     -    Residual count high
        (K K K K K K K K)     -    Residual count low
ATN (X  1  0  1  1  1  1  1)     -    Untalk
```

3.13   Cold-Load-Read

Seeks to cylinder zero and transfers the specified number of sectors starting with the specified head and sector. The sequence is as follows:

```
ATN (X  0  1  A A A A A)     -    Controllers listen address
        (X X X 1  1  1  1  1)     -    Cold load read command
```

```
        (H H S S S S S S)   -   Head and sector
        (K K K K K K K K)   -   Sector count high
        (K K K K K K K K)   -   Sector count low
ATN (X 0 1 1 1 1 1 1)       -   Unlisten
ATN (X 1 0 A A A A A)       -   Controllers talk address
        (D D D D D D D D)

                .
                .           -   Data
                .

        (D D D D D D D D)
ATN (X 1 0 1 1 1 1 1)       -   Untalk
```

If no errors or only ECC correctable errors are encountered, the disk address registers are left pointing at the sector following the last sector transferred. If any other error is encountered, the disk address registers are left pointing at the sector in error and the count returned by the read-address commands indicates the number of sectors not transferred. Upon detection of a hard error, the controller returns zero bytes until the expected number of bytes (initial sector count times sector size) have been transferred to the Host. The data transfer may be suspended by untalking the controller and resumed by addressing it to talk again.

3.14 Read

Seeks to the cylinder and selects the head specified by the contents of the disk address registers and reads the specified number of sectors starting with the sector specified in the disk address registers. The sequence is as follows:

```
ATN (X 0 1 A A A A A)       -   Controller's listen address
        (X X X 0 0 1 0 1)   -   Read command
        (0 0 0 0 0 0 0 0)   -   Zero byte
        (K K K K K K K K)   -   Sector count high
        (K K K K K K K K)   -   Sector count low
ATN (X 0 1 1 1 1 1 1)       -   Unlisten
ATN (X 1 0 A A A A A)       -   Controllers talk address
        (D D D D D D D D)

                .
```

```
                    .     -  Data
                    .
     (D  D D D D D D D)
ATN (X  I  0  I  I  I  I  I)  -  Untalk
```

If no errors or only ECC correctable errors are encountered, the disk address registers are left pointing at the sector following the last sector tranferred. If any other error is encountered, the disk address registers are left pointing at the sector in error and the count returned by the read-address commands indicates the number of sectors not tranferred. Upon detection of a hard error, the controller returns zero bytes until the expected number of bytes (initial sector count times sector size) have been transferred to the Host. The data transfer may be suspended by untalking the controller and resumed by addressing it to talk again.

3.15   Write

Seeks to the cylinder and selects the head specified by the contents of the disk address registers and writes the specified number of sectors starting with the sector specified in the disk address register. The sequence is as follows:

```
ATN (X  0  I  A A A A A)  -  Controllers listen address
     (X  X X 0  I  0  0  0)  -  Write command
     (0  0  0 0 0 0 0 0)  -  Zero byte
     (K  K K K K K K K)  -  Sector count high
     (K  K K K K K K K)  -  Sector count low
     (D  D D D D D D D)
                    .
                    .     -  Data
                    .
     (D  D D D D D D D)
ATN (X  0  I  I  I  I  I  I)  -  Unlisten
```

If no errors are encountered, the disk address registers are left pointing at the sector following the last one written. If an error is encountered, the disk address registers are left pointing at the sector where the error was encountered and the residual count returned by the read-address command is the number of sectors not

written. Upon detection of an error, the controller will continue to accept data from the Host until the expected number of bytes (initial sector count times sector size) have been transferred but the data will not be written to the disk. The data transfer may be suspended by unlistening the controller and resumed by addressing it to listen again.

3.16 Read-Long

Seeks to the cylinder and selects the head specified by the contents of the disk address registers and reads the sector specified in the disk address register. The data is returned followed by the 4 byte ECC field. The sequence is as follows:

```
ATN (X 0 1 A A A A A)  -  Controllers listen address
    (0 0 0 0 0 1 1 0)  -  Read long command
    (0 0 0 0 0 0 0 0)  -  Zero byte
ATN (X 0 1 1 1 1 1 1)  -  Unlisten
ATN (X 1 0 A A A A A)  -  Controllers talk address
    (D D D D D D D D)
            .
            .          -  256 or 512 data bytes
    (D D D D D D D D)
    (E E E E E E E E)  -  ECC byte 0
    (E E E E E E E E)  -  ECC byte 1
    (E E E E E E E E)  -  ECC byte 2
    (E E E E E E E E)  -  ECC byte 3
ATN (X 1 0 1 1 1 1 1)  -  Untalk
```

On completion of a read-long, the disk address registers are left unchanged.

The ECC bytes is a polynomial derived from the data consisting of $(X^{32} + X^{23} + X^{21} + X^{11} + X^2 + 1)$.

3.17 Write-Long

Seeks to the cylinder and selects the head specified by the contents of the disk address registers and writes the sector specified in the disk address registers. The

data is written to the disk followed by the 4 byte ECC field. The sequence is as follows:

```
ATN (X 0 1 A A A A A)  -  Controllers listen address
    (0 0 0 0 1 0 0 1)  -  Write long command
    (0 0 0 0 0 0 0 0)  -  Zero byte
    (D D D D D D D D)
                 .
                 .     -  256 or 512 data bytes
    (D D D D D D D D)
    (E E E E E E E E)  -  ECC byte 0
    (E E E E E E E E)  -  ECC byte 1
    (E E E E E E E E)  -  ECC byte 2
    (E E E E E E E E)  -  ECC byte 3
ATN (X 0 1 1 1 1 1 1)  -  Unlisten
```

On completion of a write-long, the disk address registers are left unchanged.

The ECC bytes appended to the data should be a polynomial calculation of $(X^{32} + X^{23} + X^{21} + X^{11} + X^2 + 1)$.

3.18   Format

Recalibrates the drive, then seeks to the cylinder and addresses the head specified in the disk address registers and formats the specified number of tracks. A repeating data pattern of B6DB6D is written in the data field of each sector on the track. The sequence is as follows:

```
ATN (X 0 1 A A A A A)  -  Controllers listen address
    (X X X 1 1 0 0 0)  -  Format command
    (0 0 0 0 0 0 0 0)  -  Zero byte
    (K K K K K K K K)  -  Track count high
    (K K K K K K K K)  -  Track count low
ATN (X 0 1 1 1 1 1 1)  -  Unlisten
```

If an error occurs during the format the disk address registers are left pointing at

the track which was being formatted when the error occured. If no errors occur, the cylinder and head portions of the disk address register are left pointing at the track one past the last one formatted. The sector portion of the disk address registers is unchanged by the format command.

3.19   Verify

Seeks to the cylinder and selects the head specified in the disk address registers and reads the specified number of sectors starting with the sector specified in the disk address registers. the data is checked for valid ECC but is not transferred to the Host. The sequence is as follows:

```
ATN (X 0 1 A A A A A)  -   Controllers listen address
    (X X X 0 0 1 1 1)  -   Verify command
    (0 0 0 0 0 0 0 0)  -   Zero byte
    (K K K K K K K K)  -   Sector count high
    (K K K K K K K K)  -   Sector count low
ATN (X 0 1 1 1 1 1 1)  -   Unlisten
```

If no errors are encountered, the disk address registers will be left pointing at the sector one beyond the last one verified and the residual count will equal zero. If an ECC correctable error is encountered, the command will terminate with the address registers one sector beyond the sector which was correctable and with the residual count equal to the number of sector not verified. If any other error occurs, the command terminates with the address registers pointing at the sector in error and with the corresponding residual count.

3.20   Write-Alternate-Sector

The last sector on each track is reserved as an alternate in case one other sector on the track has a hard defect. The spare may be assigned using this command to replace the defective. The command seeks to the cylinder and selects the head specified in the disk address registers. It then assigns the alternate to replace the sector specified in the disk address register and writes it with the data sent with the command. The sequence is as follows:

```
ATN (X 0 1 A A A A A)  -  Controllers listen address
    (X X X 1 1 0 1 0)  -  Write alternate command
    (0 0 0 0 0 0 0 0)  -  Zero byte
    (D D D D D D D D)
             .
             .             -  256 or 512 data bytes
    (D D D D D D D D)
ATN (X 0 1 1 1 1 1 1)  -  Unlisten
```

3.21   Set-Interleave

The controller will perform a logical to physical sector mapping as specified by this command.   At power on or following an interface-clear, universal-clear or selected-clear, the mapping is reset to logical equals physical.   The number of bytes sent with the command must equal the number of sectors per track.   (17 with 512 byte sectors and 31 with 256 byte sectors.)   Each byte must have a unique value from 16 to 10 or 0 to 30.   The contents of byte 0 tells physically where logical sector 0 is located.   The contents of byte 1 tells physically where logical sector 1 is located, etc.   The sequence is as follows:

```
ATN (X 0 1 A A A A A)  -  Controllers listen address
    (X X X 1 1 0 0 1)  -  Set interleave command
    (0 0 0 0 0 0 0 0)  -  Zero byte
    (D D D D D D D D)  -  Physical location of logical sector 0
             .
             .
             .
    (D D D D D D D D)  -  Physical location of logical sector 16 or 30.
ATN (X 0 1 1 1 1 1 1)  -  Unlisten
```

3.22   Loopback

This command is a diagnostic command to test proper communication between the 8048 in the controller and the Host system.   A data byte is sent to the controller where it is complemented and returned to the Host.   The sequence is as follows:

ATN (X 0 1 A A A A A) - Controllers listen address

(X X X 1 1 1 0 1) - Loopback command

(D D D D D D D D) - Data Byte

ATN (X 0 1 1 1 1 1 1) - Unlisten

ATN (X 1 0 A A A A A) - Controllers talk address

(D D D D D D D D) - Complemented data byte

ATN (X 1 0 1 1 1 1 1) - Untalk

SECTION 4
MAINTENANCE

4.1   General

The MSC-9305 requires no periodic maintenance.   Numberous commands are included within the command set to aid in verifying the operation of the major subassemblies of an MSC-9305 based disk subsystem and the major components within the MSC-9305 itself.

When a fault is suspected within the disk subsystem maintenance should proceed in an organized fashion to ensure rapid fault isolation and repair.   The major faulty subassembly should be identified first, then the faulty component within the subassembly should be identified.

4.2   Troubleshooting Guide

Shown in Figure 4-1 is a flow chart to assist troubleshooting a system containing the MSC-9305.

Shown in the flow chart is a "Host Level Disk Diagnostic" - this is optionally provided by Microcomputer Systems Corporation to be used with HP-85 Host computers.   If the MSC-9305 is used with another Host computer the diagnostic should be written to exercize and test the disk subsystem by envoking all controller commands and disk functions and testing for proper results.

Suspect Failure in Disk Subsystem

Test Other Devices on GPIB Bus

GPIB BUS FAILURE — Y — Failure — N

Run Host Level Disk Diagnostic

DISK SYSTEM OK — N — Failure — Y

All combinations of 8 Bits

Run Loopback Test

Problem in MSC-9305 3048 9914 or interface circuits — Y — Failure — N

Run Self Test

N — Failure — Y

STATUS = 8XXX — Y — BAD 3048
N

BAD 9056 or interface to 3048

Run Address Record Test (All valid disk address and Read address must be same)

Bad 3048 — Y — Failure — N

Run Recalibrate Test (Universal Clear)

A

A

Failure — Y — any Head Movement — N — STATUS 2, 3, or 4 — Y — Defective: Control Cable Drive Interface Disk Drive
N   Y   N
Run Seek Test   Bad Disk Drive   Defective 9056

Failure — Y — all cylinders — Y — Status 5 or 3 — Y — Defective: Data Cable Disk Interface Disk Drive Data Separator
N   N   N
Write Data Test   Defective: Disk Format Disk Drive   Defective 9056

Failure — Y — all Heads and Cylinders — Y — STATUS 5, 6, 7, of A — Y — Defective: Data Cable Data Separator Disk Interface Disk Drive
N   N   N
Read Data Test (same as previously written)   Bad Format Bad Media Bad Disk Drive   Defective 9056

STATUS FAILURE — Y — all areas written — Y — STATUS 5, 6, 7, C, or D — Y — Defective: Data Cable Data Separator Disk Interface Disk Drive
N   N   N
                 Defective: Media Disk Drive   Defective 9056

Data Compare Failure — N
Y
Run Verify Test

Failure — N — Defective: 9056 8048 Interface circuits between 9056 and Host
Y
Defect is either: Intermittant, in Formatting or Host software

Figure 4-1

-31-

SECTION 5
THEORY OF OPERATION

5.1   General

At the center of the MSC-9305 is the 8048 microprocessor unit (MPU) which controls the TMS-9914 to provide all host communication and activates the MSC-9056 Disk I/O processor to perform disk functions.

5.2   GPIB Protocol Introduction

The GPIB is designed to allow up to 15 devices within a localized area to communicate with each other over a common bus. Each device has a unique address, read from external switches at power-on, to which it responds. Information is transmitted in byte serial bit parallel format and may consist of either device data or interface control information.

Device data may be sent by any one device (the TALKER) and received by a number of other devices (LISTENERS). Instructions such as a select range, select function, or measurement data for processing or printout may be sent in this way.

One of the devices on the bus, designated the Controller in charge (Controller), may send interface control messages. Devices can be assigned to the bus as listeners or talkers by sending their unique talk or listen address, and may be switched between remote and local control.

The bus itself consists of a 24 wire shielded cable. 8 lines carry data; 8 are control lines; 8 are signal and system grounds. A diagram showing the IEEE bus configuration is given in Figure 5-1.

Three of the bus management lines operate as a three line handshake between talker (or controller) and listeners. No new data is sent until each device addressed to listen has received the last byte and is ready for the next. This method of asynchronous communication ensures that the data rate is suited to the slowest active listener, as well as ensuring compatibility over a wide range of devices. A

TP74   -32-

SHIELD SRQ NDAC DAV DIO4 DIO2

LOGIC GND ATN IFC NRFD EOI DIO3 DIO1

Signature   Description

GND GND GND REN DIO7 DIO5
11   9   7
LOGIC GND GND GND DIO8 DIO6
GND 10   8   6
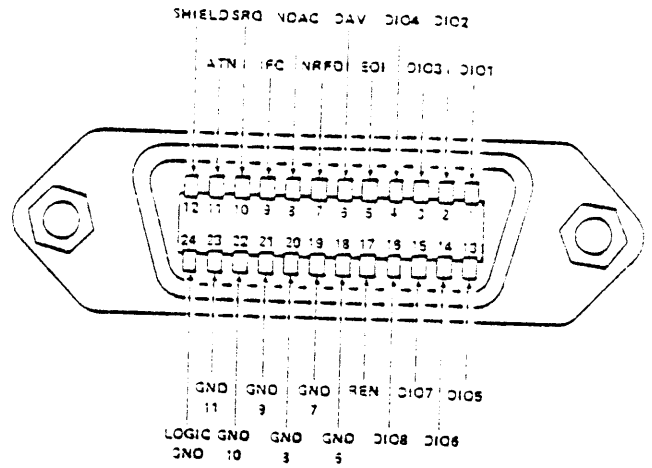
DIO8 (MSB)   DIO8 through DIO1 are the data input/output line on the GPIB side.
DIO7   These pins connect to the IEEE-488 bus via non-inverting
DIO6   transceivers.
DIO5
DIO4
DIO3
DIO2
DIO1 (LSB

DAV   DATA VALID: handshake line controlled by source to show
                 acceptors when valid data is present on the bus.

NDAC   NOT DATA ACCEPTED: handshake line. Acceptor sets this false
                 (high) when it has latched the data from the I/O lines.

NRFD   NOT READY FOR DATA: handshake line. Sent by acceptor to
                 indicate readiness for the next byte.

ATN   ATTENTION: sent by controller in charge.   When true (low)
                 interface commands are being sent over the DIO lines.   When
                 false (high) these line carry data.

REN   REMOTE ENABLE: sent by system controller to select control
                 either from the front panel or from the IEEE bus.

IFC   INTERFACE CLEAR: sent by the system controller to set the
                 interface system into a known quiescent state.   The system
                 controller becomes the controller in charge.

SRQ   SERVICE REQUEST: set true (low) by a device to indicate a need
                 for service.

EOI   END OR IDENTIFY: if ATN is false this indicates the end of a
                 message block. If ATN is true the controller is requesting a
                 parallel poll.

FIGURE 5-1 GPIB Bus Configuration
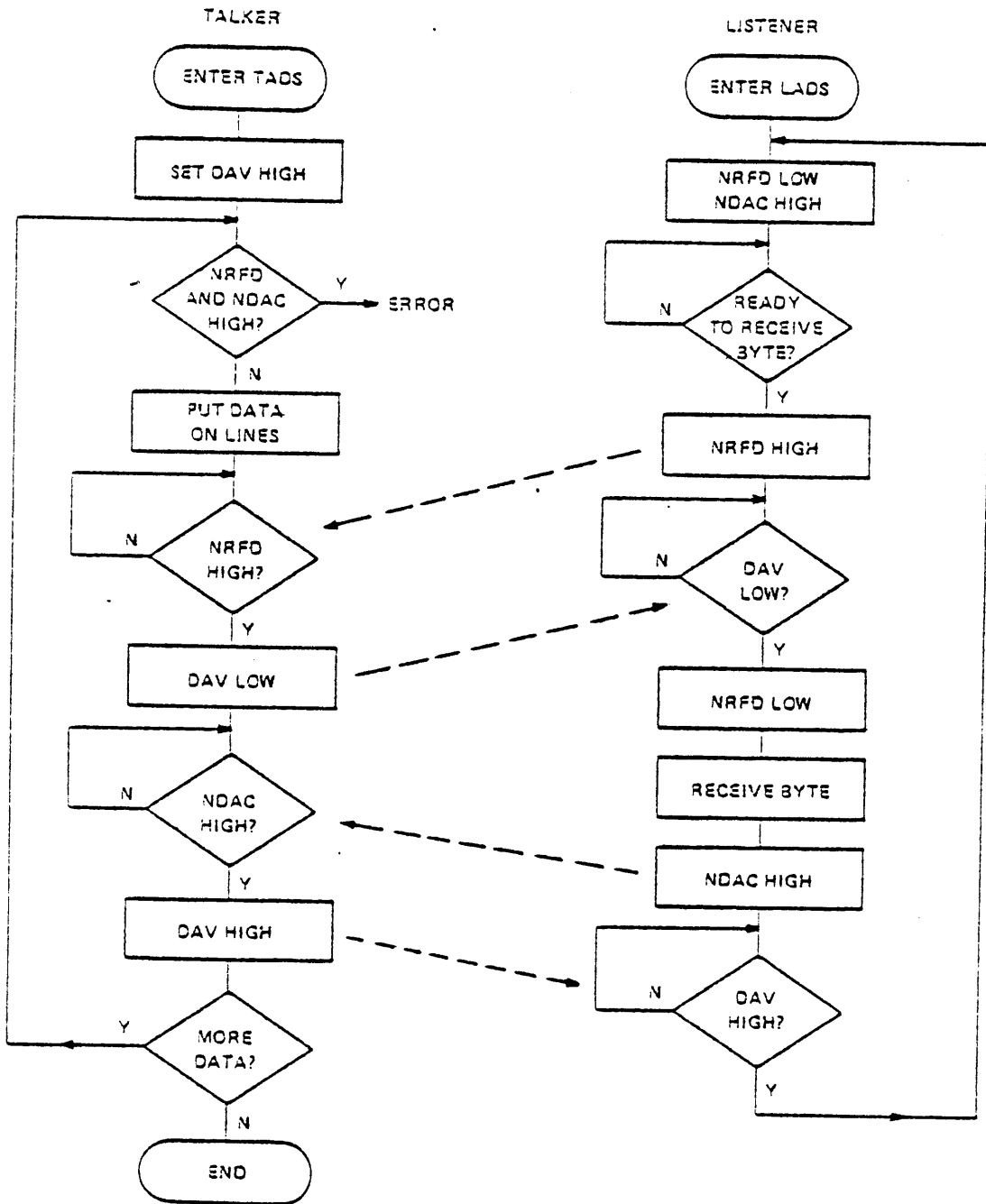
Figure 5-2 GPIB 3-Wire Handshake Flow Chart

5.3   Block Diagram Description

Figure 5-3 illustrates the block diagram of the MSC-9305. The following sections will describe each major block in greater detail.

5.3.1   GPIB Interface

The GPIB input/output pins are connected to the IEEE-488 bus via bus transceivers. The direction of data flow is controlled by the TE and $\overline{\text{CONTROLLER}}$ outputs generated on the TMS 9914.   The SN 75160, 75161 and 75162 ar designed specifically for use with a GPIB interface. The TE and $\overline{\text{CONTROLLER}}$ signals are routed within the devices so that the buffers on particular lines are controlled as required by the TMS 9914.

5.3.2   GPIB Adapter

The TMS 9914 is used to enable the 8048 MPU to communicate with an IEEE-488 General Purpose Interface Bus (GPIB). It performs the interface function between the microprocessor and bus and relieves the processor of the task of maintaining the IEEE protocol. By utilizing the interrupt capabilities of the device the bus does not have to be continually polled, and fast responses to changes in the interface configuration are achieved.

Communication between the microprocessor and TMS 9914 is carried out via memory mapped registers. There are 13 registers within the TMS 9914, 6 of which are read and 7 write. They are used both to pass control data to, and get status information from, the device.

The 3 least significant address lines from the MPU are connected to the register selected line RSO, RS1, and RS2 and determine the particular register selected. The high order address lines are decoded by external logic to cause the $\overline{\text{CE}}$ input to the TMS 9914 to be pulled low when any one of 8 consecutive addresses are selected.   Thus the internal registers appear to be situated at 8 consecutive locations within the MPU address space. Reading or writing to these locations transfers information between the TMS 9914 and the microprocessor. Note that reading and writing to the same location will not access the same register within

the TMS 9914 since they are either read ony or write only registers. For example, a read operations with RS2-RS0 = 011 gives the current status of the GPIB interface control lines, whereas a write to this location loads the auxiliary command register.

Each device on the bus interface is given a 5-bit address enabling it to be addressed as a talker or listener. This address is set on a DIP switch before power-on and is both read by the microprocessor and written into the address register as part of the initialization procedure. The TMS 9914 responds by causing a MA (My Address) interrupt and entering the required addressed state when this address is detected on the GPIB data lines.
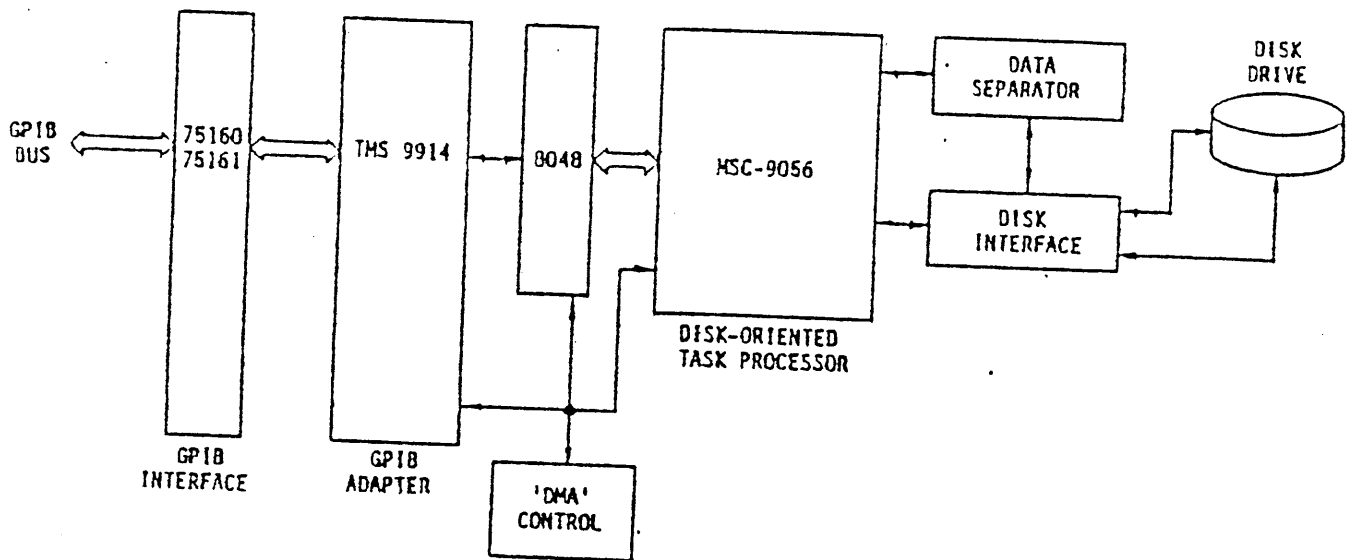
Figure 5-3 MSC-9305 Block Diagram

5.3.3 8048 Microprocessor

The microprocessor provides the central intelligence of the MSC-9305 controller to translate commands from the host into a series of control functions for the MSC-9056 Disk I/O processor. The 8048 contains a 1KX8 program memory, a 64X8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter.

5.3.4 Disk I/O Processor

The MSC-9056 is a Module which incorporates most of the functions required to interface to the Shugart Technology ST-506 disk drive. The functions incorporated within the MSC-9056 allow high level tasks to be communicated with it, achieving sophisticated control of the disk drive with minimum additional circuitry.

There are twelve separate commands which the Module will execute. Each of these commands requires multiple 8 bit bytes to fully specify the task.

Seek	Read Sector	Write Long	Set Interleave
Recalibrate	Write Sector	Status Request	Write Alt. Sector
Diagnostic	Read Long	Format Track	Write Check

5.3.5 DMA Control

This is a group of circuits which allow the Disk Data to be communicated directly between the MSC-9056 and the TMS 9914, thus achieving maximum data transfer rate without being restricted by the speed of the MPU. The MPU handles all command, control, and status functions and enables the DMA circuits at the proper time for the data transfer.

5.3.6 Data Separtor

This is an individual printed circuit-board to provide conversion between the MFM data format of the disk and the NRZ format of the MSC-9056 module. This block also provides a means for generating and detecting address marks which have a unique encoding to divide a track into a number of fixed sectors.

5.3.7 Disk Interface

This is a group of circuits to provide the electrical interface between the MSC-9056 module and the disk drive.

5.4   Controller Schematic Description

A schematic of the MSC-9305 controller is included in Appendix A. The following sections will detail the function of the controller. The controller function can best be described by the functions of the I/O signals of the MPU and 9056 module.

5.4.1 MPU I/O Ports

The following list describes the functions of each 8048 MPU I/O port.

P10   -   Clear signal to the 9914 used to intilialize the GPIB adaptor.

P11   -   DBIN signal to the 9914 to define the direction of data transfer. During DMA this line is high for data transfer from the MSC-9056 to the host. During transfers between the MPU and the 9914 this line has the opposite interpretation. When the tranfer is from the 8048 to the 9914 this signal is low.

P12   -   This line enables the LS243 to transfer data from the 9056 to the MPU data bus. The MPU data bus is also routed to the 9914, so this signal will also be high when data is sent from the 9056 to the 9914.

P13   -   This signal is connected to the LS257 multiplexer to define when 9056 communication is to be performed with the MPU or the 9914. When P13 is high communication is between the 8048 and the 9056. When P13 is low "DMA" is enabled and the 9056 communicates (Data) directly with the 9914.

P14   -   This is an enable signal to activate 9056 and 9914 communication.

P15   -   This is the Command (CMD) signal to the 9056. Whenever a task is to be performed by the 9056 this signal gets set (active low) to activate the module to accept a command.

P16   -   This is a Clear signal for the 9056 module whenever the MPU needs to reset the module.

P17   -   This signal is routed through the LS257 multiplexer to activate a

Strobe (STB) to the module as a handshake signal to note that the MPU has accepted or has available a byte of information. The Strobe is a reponse to Load Data In (LDI) or Data Out (DOUT) from the module. See Figure 5-4.

P24   —   This signal gets input to the MPU as the READY (RDY) signal from the 9056. The Ready signal is active whenever the 9056 is transferring data on its data bus with the 9914 or the 8048. See Figure 5-4.

DISK READ   DISK WRITE

READY

DOUT

EDI

STROBE

DATA BUS   VALID   VALID
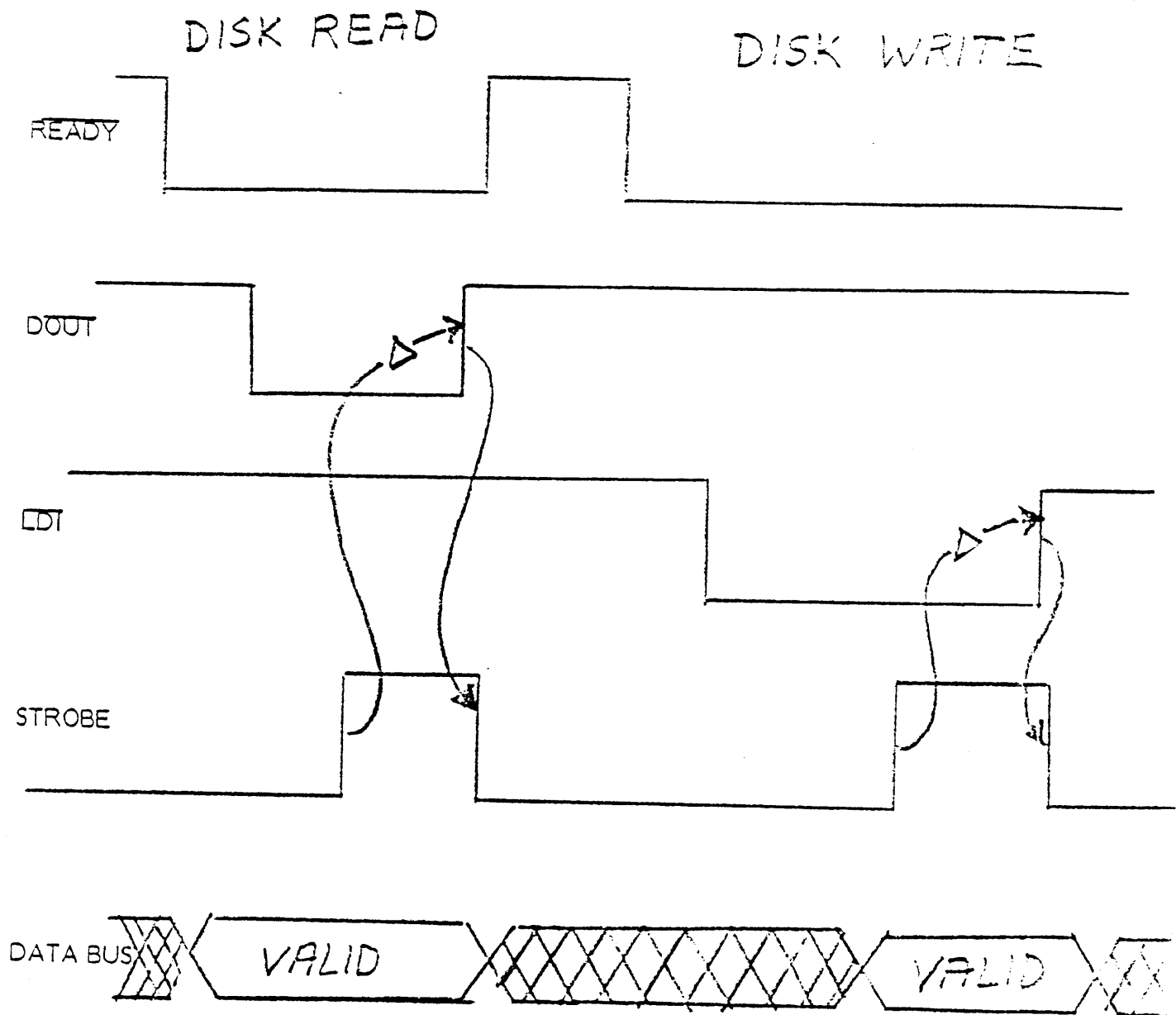
Figure 5-4 Module Control Timing

P25 — This is the TRIGGER (TR) signal out of the 9914 to note when the GET command (Group Execute Trigger) is received over the GPIB interface or the GET command is given by the MPU.

P26 — This is an input to the MPU to note if the controller is set for 256 or 512 bytes per sector. The signal is high for 512, at which time the strap is out.

P27 — This is a Clear signal to the LS174 address register. It is low whenever the MPU is using memory map data transfers to communicate with the 9056, thus the CE to the 9914 and the LS240 switch receiver will be disabled from driving the data bus.

INT — This the Interrupt signal from the 9914 to the MPU to note whenever the GPIB adapter requires attention.

XTAL — These two lines are the clock input to the MPU. The clock frequency is 4MHZ which is derived from dividing 16MHZ by 4 by the two 74S74 Flip-Flops.

ALE — This is the Address Latch enable signal out of the MPU which is active every memory I/O cycle to latch the address off the MPU data bus.

RESET — This is the Reset signal for the MPU derived from the RC power up detect circuit or the Interface Clear (IFC) signal from the GPIB bus.

TI — This is an input signal to the MPU which is active (low) whenever the 9056 is transferring a byte of information. See Figure 5-4.

WR — This is the Write signal out of the MPU whenever data is output on the MPU data bus. The WR signal enables the LS139 address decoder to activate chip enable (CE) to the 9914, and the WR signal directly enables the Write enable signal to the 9914.

D0-D7 — This is the MPU data bus connected to the address register, 9914, LS243 transceiver to the 9056, and the LS240 switch receiver.

RD — This is the Read signal out of the MPU whenever data is input on the MPU data bus. It is connected to enable the LS139 address decoder to enable the 9914 chip enable or the LS240 switch receiver.

TØ — This is an input to the MPU from the 9056 BUSY signal to denote whenever the 9056 is processing a command.

5.4.2. MSC-9056 I/O Lines

TP74

The following list describes the functions of each Module I/O Line which was not previously described.

D0-D7   –   This is the bidirectional data bus of the Module which is routed to the MPU bus via the LS243 transceiver and the disk interface via the LS245 transceiver.

Ready   –   This is a Ready signal out of the module to note whenever the module can transfer data on the data bus with other than the disk interface. See Figure 5-4.

LDI   –   This is Load Data In signal out of the module to note when the module can input a byte on it's data bus from other than the disk interface. The LDI signal is combined with the RDY signal to enable the LS243 to drive the module data bus. The combined signal is used to generate the $\overline{XFER}$ signal which is routed to the MPU, the Strobe latch, and the DMA logic via the LS257 multiplexer. The Strobe latch gets reset whenever the $\overline{LDI}$ signal returns inactive (high), see Figure 5-4. The $\overline{XFER}$ signal, after being routed through the LS257 is used in conjunction with the Access Request $\overline{(ACCRQ)}$ of the 9914 to enable a sequencer to generate the Access Granted $\overline{(ACCGR)}$ function to the 9914. See Figure 5-5.

DOUT   –   This is the signal out of the Module to denote whenever the module is outputting a byte on its data bus. The DOUT signal is connected to the disk interface circuits via LS32 gates to generate control functions for the disk interface circuits. The DOUT signal is also combined with the RDY signal to generate the XFER signal which is routed to the MPU, the Strobe latch reset, and the DMA circuits. The XFER signal, during a DOUT function, is used in a sequencer to generate Access Granted and Write enable functions to the 9914. See Figure 5-5.
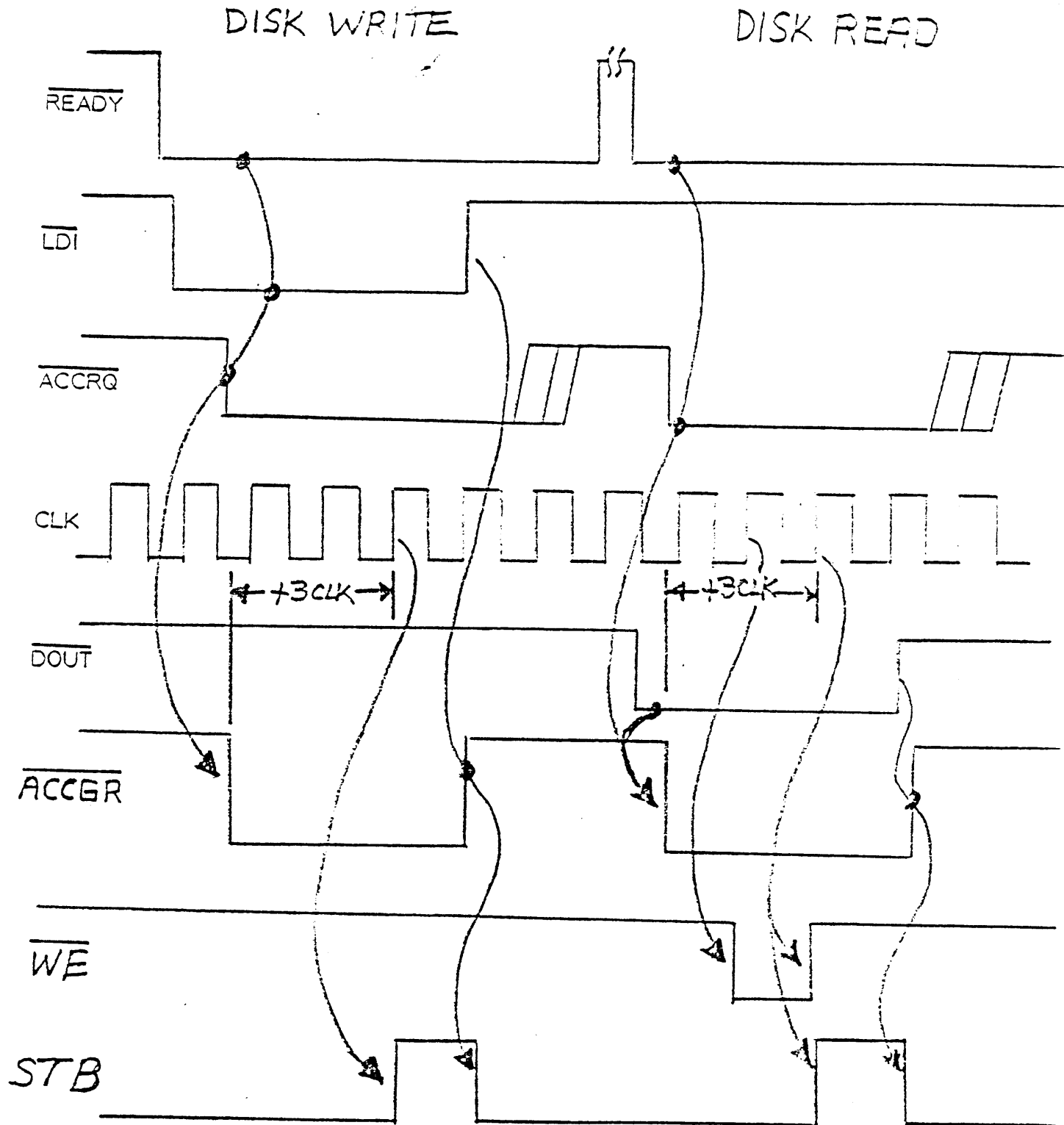
Figure 5-5 DMA Timing

DC0,   –   These two signals, out of the module, are encoded to define what
DC1   information is on the module data bus and when a write address mark
pulse is to be generated. The signals are decoded by an LS139
decoder, for the following functions:

State   0 –   LS174 enable to store disk control functions.

Bit 0 – Head 0 select

1 – Head 1 select

4 – Step control

5 – Direction control

7 – Reduce current control

State   1 –   Latch Enable to store the drive select and
AM Search functions. Note that in this
application of the 9056 the drive select is
always enabled. And bit 4 is the only bit
latched for the AM Search enable function.

State   2 –   Generates a Control signal to enable the
input of disk status into the module.

Bit 0 – Track 00

1 – Write Fault

2 – $\overline{\text{Seek Complete}}$

3 – $\overline{\text{Ready}}$

4 – $\overline{\text{Selected}}$

State   3 –   Generates a pulse which is routed to the
data separator to generate an address mark.

$\overline{\text{DCV}}$ –   This is the DC Valid signal out of the module to denote whenever
there is a valid state on the DC0 and DC1 control signals. The $\overline{\text{DCV}}$
signal is used to enable the LS139 Decoder.

Index   –   This signal is routed into the module from the disk drive via a timing
circuit to denote whenever the disk has reached the index point of it
rotation. The timing circuit is required to insure an accurate
reference point from the leading edge of index. See Figure 5-6.

AMD   –   This is the Address Mark detect signal into the module from the data
separator PCB. The disk is formatted into fixed sectors by writing
address marks to delineate the beginning of each sector.

PLO   –   This is the disk data clock input to the module coming from the data
Clock   Separator, it is used to synchronize the Read or Write data.

TP74

Read   -   This is the NRZ Disk Read data input to the module coming from the
Data   data separator.

Disk Index

CLK

Module Index

< 250nS
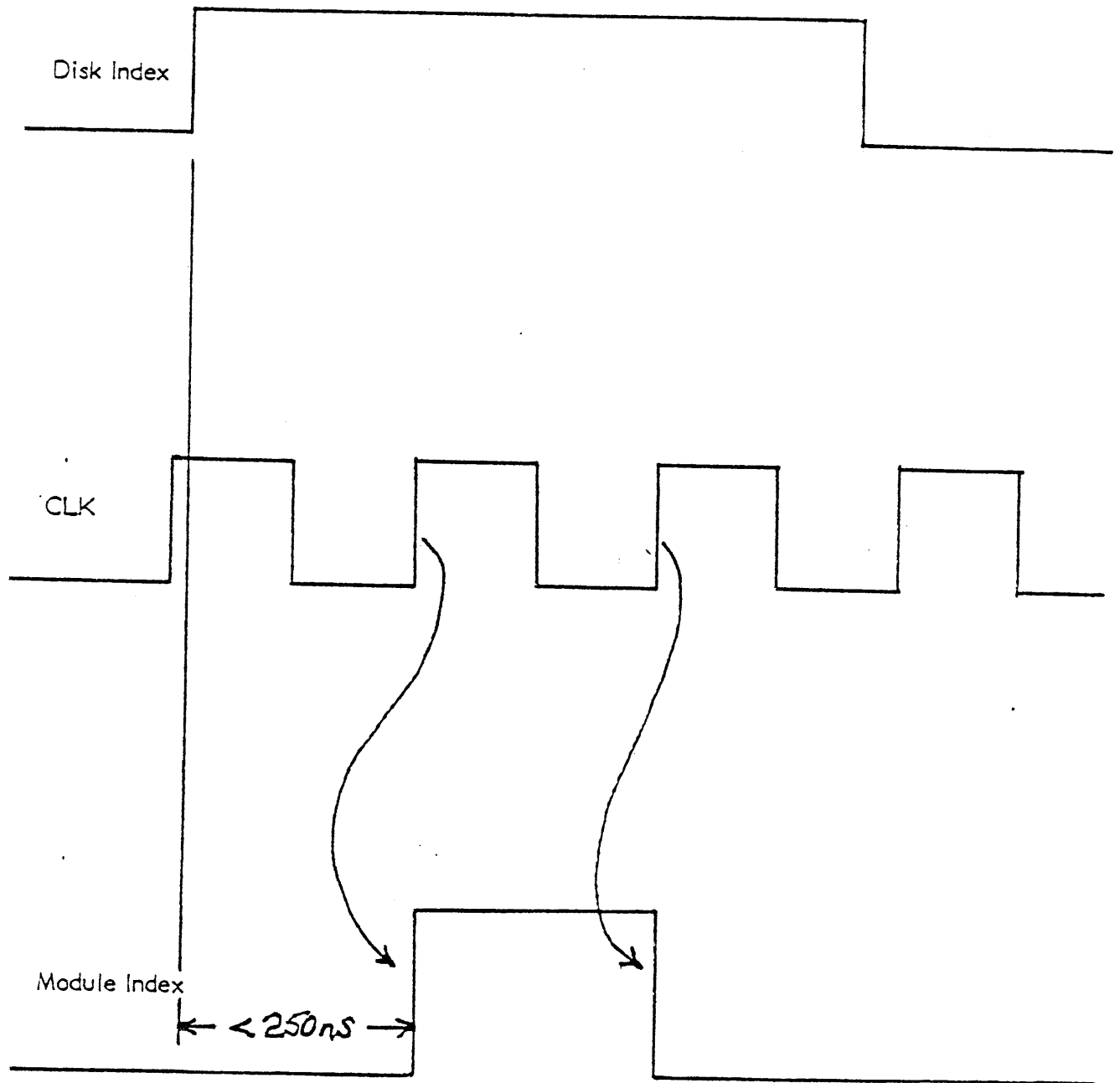
Figure 5-6 Index Timing

Write Data	-	This the NRZ Disk Write data out of the module going to the data separator.
Write Gate	-	This the Disk Write enable signal out of the module going to the data separator.
Read Gate	-	This is the Disk Read Enable signal out of the module going to the data separator.

5.5   MPU Firmware Description

The firmware in the 8048 MPU is structured into 2 major routines, a number of command execute routines and a number of common subroutines. A listing of the Firmware is provided in Appendix B.

5.5.1   Major Routines

Power on Initialization - This routine initializes the 9914, reads the controllers talk and listen address (both the same), clears the MPU RAM, determines the parallel poll bit (also obtained from the switches) and initializes various control signals.

Idle Routine - This routine waits for activity to be initialized from the host.

5.5.2   Command Execute Routines

These routines perform the unique functions required for each command.

5.5.3   Major Subroutines

Read GPIB Subroutine - This subroutine reads data bytes from the GPIB bus via the 9914.

Write GPIB Subroutines - This subroutine sends bytes to the GPIB bus via the 9914.

Bump Address Subroutine - This subroutine provides the mechanism to enable the 9305 to transfer multiple sectors, automatically changing the head and cylinder address.

Write 9914 Register Subroutine - This subroutine writes the control registers within the 9914.

Read 9914 Registers Subroutine - This subroutine reads the status registers
out of the 9914.

Command to Module Subroutine - This subroutine performs the transmission
of the command task bytes to the module.

```
                  m.aclib z80
         ;####################################################
         ;
         ;              GPIB CONTROLLER ROUTINES
         ;
         ;####################################################
         ;
                  public  send,recv,initgpib
         ; 8291 registers
F700 =            gptldi  equ     0f700h          ; data in
F700 =            gptldo  equ     0f700h          ; data out
F701 =            gptlis1 equ     0f701h          ; int status 1
F701 =            gptlm1  equ     0f701h          ; int mask 1
F702 =            gptlis2 equ     0f702h          ; int status 2
F702 =            gptlm2  equ     0f702h          ; int mask 2
F703 =            gptlsps equ     0f703h          ; serial poll status
F703 =            gptlspm equ     0f703h          ; serial poll mode
F704 =            gptlas  equ     0f704h          ; address status
F704 =            gptlam  equ     0f704h          ; address mode
F705 =            gptlcpt equ     0f705h          ; command pass thru
F705 =            gptlaxm equ     0f705h          ; aux mode
F706 =            gptla0  equ     0f706h          ; address 0
F706 =            gptla01 equ     0f706h          ; address 0/1
F707 =            gptla1  equ     0f707h          ; address 1
F707 =            gptleos equ     0f707h          ; eos
         ;
         ;
         ; 8292 registers
         ;
F708 =            gpcdata equ     0f708h          ; data
F709 =            gpcsc   equ     0f709h          ; control/status
         ;
         ; option switches
         ;
F710 =            gpsw1   equ     0f710h          ; switch 1
F711 =            gpsw2   equ     0f711h          ; switch 2
F712 =            gpsw3   equ     0f712h          ; switch 3
F713 =            gpio    equ     0f713h          ; ioport
         ;
         ; "dma" address
         ;
00F6 =            gpdma   equ     0f6h            ; one byte dma address
         ;
         ; ioport bit significance
         ;
0001 =            iointb  equ     01h             ; 8291 int
0002 =            iotrigb equ     02h             ; trigger
0004 =            iotcib  equ     04h             ; 8292 tci
0008 =            iospib  equ     08h             ; 8292 spi
0010 =            ioobfib equ     10h             ; 8292 obfi
0020 =            ioibfib equ     20h             ; 8292 ibfi
0040 =            iodreqb equ     40h             ; 8291 dreq
0080 =            iowtto  equ     80h             ; wait time out
         ;
         ;8291 int 1 constants
         ;
```

```
0002 =          bom      equ      02h       ; bo
0001 =          bim      equ      01h       ; bi
0010 =          endmk    equ      10h       ; end intrp
0080 =          cpt      equ      80h       ; cpt
                ;
                ; gpib commands
                ;
005F =          unt      equ      5fh                   ; untalk
003F =          unl      equ      3fh                   ; unlisten
0008 =          get      equ      08h                   ; group execute trigger
0004 =          sdc      equ      04h                   ; device clear
0018 =          spe      equ      18h                   ; serial poll enable
0019 =          spd      equ      19h                   ; serial poll disable
0009 =          tct      equ      09h                   ; take control
                ;
                ; 8292 contrl values
                ;
00A0 =          intmb    equ      0a0h      ; tci
                ;
                ;
                ; 8292 commands
                ;
00F0 =          spcni    equ      0f0h                  ; stop counter interrupts
00F1 =          gidl     equ      0f1h                  ; go to idle
00F2 =          rset     equ      0f2h                  ; reset
00F3 =          rsti     equ      0f3h                  ; reset ints
00F4 =          gsec     equ      0f4h                  ; goto standby,count
00F5 =          expp     equ      0f5h                  ; execute pp
00F6 =          gtsb     equ      0f6h                  ; goto standby
00F7 =          sloc     equ      0f7h                  ; set local mode
00F8 =          srem     equ      0f8h                  ; set remote
00F9 =          abort    equ      0f9h                  ; abort
00FA =          tcntr    equ      0fah                  ; take control (recieve control)
00FC =          tcasy    equ      0fch                  ; take control async
00FD =          tcsy     equ      0fdh                  ; take control sync
00FE =          stcni    equ      0feh                  ; start counter ints
                ;
                ; 8292 utility commands
                ;
                ;
00E1 =          wout     equ      0e1h                  ; write to time out register
00E2 =          wevc     equ      0e2h                  ; write to event counter
00E3 =          revc     equ      0e3h                  ; read event counter status
00E4 =          rerf     equ      0e4h                  ; read error flag register
00E5 =          rinm     equ      0e5h                  ; read int flag register
00E6 =          rcst     equ      0e6h                  ; read controller status reg
00E7 =          rbst     equ      0e7h                  ; read gpib bus status reg
00E9 =          rtout    equ      0e9h                  ; read timeout status register
00EA =          rerm     equ      0eah                  ; read error mask reg
000B =          iack     equ      00bh                  ; int ack
                ;
                ;
                ;
                ; 8291 commands
0000 =          ipon91   equ      0                     ; immediate pon
0002 =          cr91     equ      2                     ; chip reset
```

```
0003 =            fh91     equ      3               ; finish hand shake
0004 =            get91    equ      4               ; get
0006 =            seoi91   equ      6               ; send eoi next byte
                  ;
0040 =            lon      equ      40h             ; listen only mode
0080 =            ton      equ      80h             ; talk only mode
                  ;
                  ;
                  ;
                  ;
                  ;
                  ;##############################################################
                  ;
                  ; gpib intialise routine
                  ;
                  ;##############################################################
                  initgpib:
0000 3EA0                  mvi      a,intmb         ; enable TCI on 8292
0002 0109F7                lxi      b,gpcsc         ; 8292 command register
                           outp     a
0005+ED79                  DB       0EDH,A*8+41H
0007 3E60                  mvi      a,060h
0009 0E06                  mvi      c,low(gptla01)
                           outp     a               ; disable major talker/listener
000B+ED79                  DB       0EDH,A*8+41H
000D 3EE0                  mvi      a,0e0h
                           outp     a               ; disable minor talker/listener
000F+ED79                  DB       0EDH,A*8+41H
0011 3E80                  mvi      a,ton           ; talk only mode
0013 0E04                  mvi      c,low(gptlam)
                           outp     a               ; set talk only mode
0015+ED79                  DB       0EDH,A*8+41H
0017 3E28                  mvi      a,28h           ; 8Mhz clock
0019 0E05                  mvi      c,low(gptlaxm)
                           outp     a               ; set clock rate
001B+ED79                  DB       0EDH,A*8+41H
001D AF                    xra      a
001E 0E01                  mvi      c,low(gptlm1)
                           outp     a               ; clr all ints mask 1
0020+ED79                  DB       0EDH,A*8+41H
0022 0C                    inr      c
                           outp     a               ; and mask 2
0023+ED79                  DB       0EDH,A*8+41H
0025 0E05                  mvi      c,low(gptlaxm)
                           outp     a               ; immediate execute pon
0027+ED79                  DB       0EDH,A*8+41H
0029 3A0C03                lda      mda             ; calc mta,mla
002C C620                  adi      20h
002E 320D03                sta      mla'
0031 C620                  adi      20h
0033 320E03                sta      mta
0036 0E09                  mvi      c,low(gpcsc)
                           inp      a
0038+ED78                  DB       0EDH,A*8+40H
003A B7                    ora      a
003B F2AD01                jp       initasnc        ; initialise as not controller
```

```
003E 3EFF              mvi        a,0ffh
0040 321A03            sta        ctrlflg          ; set c in c flag
0043 C9               ret
                 ;
                 ;
                 ;
                 ;
                 ;##############################################################
                 ;
                 ;    send routine
                 ;
                 ;##############################################################
                 send:
                 ;
                 ; send a data block addressed by HL of (B) byte long to listener
                 ; address (D), if E=0 use slow data rate else fast
                 ;
0044 C5               push       b                ; save BC
0045 3A0E03            lda        mta              ; my talk address
0048 CDDF01            call       gpibcom          ; send gpib command
004B 7A               mov        a,d              ; listen address
004C CDDF01            call       gpibcom
004F 3EF6             mvi        a,gtsb           ; 8292 goto standby
0051 CDEC01            call       com92            ; send command to 8292
0054 C1               pop        b                ; resore count
0055 05               dcr        b                ; last byte for eoi
0056 7B               mov        a,e
0057 B7               ora        a
                      jrz        send1
0058+280D             DB         28H,SEND1-$-1
                 ; use fast data rate
                 send2:
005A CDFA01            call       ioportr          ; read ioport(and reset to)
005D E640             ani        iodreqb          ; test dreq
                      jrz        send2            ; wait for dreq
005F+28F9             DB         28H,SEND2-$-1
                 ;
0061 0EF6             mvi        c,gpdma          ; "dma" address
                      outir                       ; output data
0063+EDB3             DB         0EDH,0B3H
                      jr         send3            ; output eoi
0065+180D             DB         18H,SEND3-$-1
                 send1:
                 ; use slow data rate
0067 CDFA01            call       ioportr          ; read io port
006A E640             ani        iodreqb          ; test dreq
                      jrz        send1
006C+28F9             DB         28H,SEND1-$-1
                 ;
006E 0EF6             mvi        c,gpdma
                      outi                        ; output byte
0070+EDA3             DB         0EDH,0A3H
                      jrnz       send1
0072+20F3             DB         20H,SEND1-$-1
                 ;
                 send3:
```

```
                        ; check io status and send eoi
0074  CDFA01                    call      ioportr
0077  E680                      ani       iowtto
0079  5F                        mov       e,a                   ; save time out status
                        send4:
007A  CDFA01                    call      ioportr
007D  E640                      ani       iodreqb
                                jr        send4                 ; wait till last byte sent
007F+18F9                       DB        18H,SEND4-$-1
0081  3E06                      mvi       a,seoi91              ; send eoi next byte
0083  C5                        push      b
0084  0105F7                    lxi       b,gptlaxm
                                outp      a                     ; ouput command
0087+ED79                       DB        0EDH,A*8+41H
0089  C1                        pop       b
008A  0EF6                      mvi       c,gpdma
                                outi                            ; send last byte
008C+EDA3                       DB        0EDH,0A3H
008E  3EFD                      mvi       a,tcsy                ; take control syncronously
0090  CDEC01                    call      com92
0093  CD0602                    call      waitt                 ; wait for task complete
0096  3E3F                      mvi       a,unl                 ; unlisten command
0098  CDDF01                    call      gpibcom
009B  7B                        mov       a,e
009C  C9                        ret
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;*************************************************************
                        ;
                        ;       recieve routine
                        ;
                        ;*************************************************************
                        ;
                        recv:
                        ;
                        ; recieve data block from GPIB, Address of block is in HL
                        ; Length is in B and address of talker is in D.
                        ; If E=0 then use slow data rate else fast.
009D  C5                        push      b                     ; save count
009E  7A                        mov       a,d
009F  CDDF01                    call      gpibcom               ; talker address
00A2  3A0D03                    lda       mla                   ; my listen address
00A5  CDDF01                    call      gpibcom
00A8  3E40                      mvi       a,lon                 ; listen only
00AA  0104F7                    lxi       b,gptlam              ;
                                outp      a                     ; set listen only
00AD+ED79                       DB        0EDH,A*8+41H
00AF  AF                        xra       a
00B0  0E05                      mvi       c,low(gptlaxm)
                                outp      a                     ; immediate pon
00B2+ED79                       DB        0EDH,A*8+41H
00B4  3EF6                      mvi       a,gtsb                ; goto standby
```

```
00B6 CDEC01            call    com92
00B9 CD0602            call    waitt              ; wait task complete
00BC C1                pop     b                  ; restore count
00BD 7B                mov     a,e
00BE B7                ora     a
                       jrz     recv1
00BF+280D              DB      28H,RECV1-$-1
           ; use fast data rate
recv2:
00C1 CDFA01            call    ioportr            ; read io port
00C4 E640              ani     iodreqb            ; test dreq
                       jrz     recv2
00C6+28F9              DB      28H,RECV2-$-1
           ;
00C8 0EF6              mvi     c,gpdma            ; "dma" address
                       inir                       ; input block
00CA+EDB2              DB      0EDH,0B2H
                       jr      recv3
00CC+180D              DB      18H,RECV3-$-1
           ;
recv1:
           ; use slow data rate
00CE CDFA01            call    ioportr            ; read i/o port
00D1 E640              ani     iodreqb            ; test dreq
                       jrz     recv1
00D3+28F9              DB      28H,RECV1-$-1
           ;
00D5 0EF6              mvi     c,gpdma
                       ini                        ; input one byte
00D7+EDA2              DB      0EDH,0A2H
                       jrnz    recv1
00D9+20F3              DB      20H,RECV1-$-1
           ;
recv3:
00DB CDFA01            call    ioportr
00DE E680              ani     iowtto             ; check to status
00E0 5F                mov     e,a                ; save it
           ;
00E1 3EFD              mvi     a,tcsy             ; take control sync
00E3 CDEC01            call    com92
00E6 CD0602            call    waitt              ; wait task complete
00E9 3E80              mvi     a,ton
00EB 0104F7            lxi     b,gptlam
                       outp    a                  ; set talk only mode
00EE+ED79              DB      0EDH,A*8+41H
00F0 3E5F              mvi     a,unt              ; untalk command
00F2 CDDF01            call    gpibcom
00F5 7B                mov     a,e                ; get io status
00F6 C9                ret
           ;
           ;
           ;################################################################
           ;
           ;          get control of GPIB
           ;
           ;################################################################
```

```
                    GETCTRL:
                    ; this routine aquires control of GPIB if needed
00F7 AF                     xra     a
00F8 321203                 sta     freeflg         ; set busy
00FB 3A1A03                 lda     ctrlflg         ; check if already c in c
00FE B7                     ora     a
00FF C0                     rnz                     ; already in control
                    ; enable pp
0100 3E09                   mvi     a,9h            ; parallel poll enable
0102 0105F7                 lxi     b,gptlaxm        ; aux mode register
                            outp    a
0105+ED79                   DB      0EDH,A*8+41H
                    ; now assert SRQ
0107 0103F7                 lxi     b,gptlspm        ; serial poll mode
010A 3E40                   mvi     a,40h           ; rsv bit
                            outp    a
010C+ED79                   DB      0EDH,A*8+41H
                    ; now wait for take control message
                    getc2:
010E 0E01                   mvi     c,low(gptlis1)  ; int 1
                    getc1:
                            inp     a
0110+ED78                   DB      0EDH,A*8+40H
0112 E630                   ani     cpt             ; wait for cpt
                            jrz     getc1
0114+28FA                   DB      28H,GETC1-$-1
                    ;
0116 0E05                   mvi     c,low(gptlcpt)
                            inp     a               ; get command
0118+ED78                   DB      0EDH,A*8+40H
011A FE09                   cpi     tct             ; check if take control command
                            jrnz    getc3           ; acknowledge and try again
011C+2036                   DB      20H,GETC3-$-1
011E 0E04                   mvi     c,low(gptlas)   ; address status
                            inp     a
0120+ED78                   DB      0EDH,A*8+40H
0122 E602                   ani     02h             ; check if addressed
                            jrz     getc3           ; not my address
0124+282E                   DB      28H,GETC3-$-1
                    ;
0126 0E03                   mvi     c,low(gptlspm)
0128 AF                     xra     a
                            outp    a               ; reset SRQ
0129+ED79                   DB      0EDH,A*8+41H
                    ;
012B 3E60                   mvi     a,60h
012D 0E06                   mvi     c,low(gptla01)  ; disable talker listener
                            outp    a
012F+ED79                   DB      0EDH,A*8+41H
0131 3E80                   mvi     a,ton
0133 0E04                   mvi     c,low(gptlam)
                            outp    a               ; set ton mode
0135+ED79                   DB      0EDH,A*8+41H
0137 AF                     xra     a
0138 0E01                   mvi     c,low(gptlis1)
                            outp    a
```

```
013A+ED79                    DB          0EDH,A*8+41H
013C  0C                     inr         c
                             outp        a                       ; clear both int masks
013D+ED79                    DB          0EDH,A*8+41H
013F  0E05                   mvi         c,low(gptlaxm)
                             outp        a
0141+ED79                    DB          0EDH,A*8+41H
0143  3EFA                   mvi         a,tcntr
0145  CDEC01                 call        com92                   ; take control
0148  3E0F                   mvi         a,0fh                   ; reset holdoff on cpt command
014A  0E05                   mvi         c,low(gptlaxm)
                             outp        a
014C+ED79                    DB          0EDH,A*8+41H
014E  3EFF                   mvi         a,0ffh
0150  321A03                 sta         ctrlflg                 ; set c in c flag
0153  C9                     ret
                      ;
                getc3:
0154  0E05                   mvi         c,low(gptlaxm)
0156  3E0F                   mvi         a,0fh                   ; reset holdoff
                             outp        a
0158+ED79                    DB          0EDH,A*8+41H
                             jr          getc2                   ; try again
015A+18B2                    DB          18H,GETC2-$-1
                      ;
                      ;
                      ;
                      ;
                      ;
                      ;
                      ;
                      ;#########################################################################
                      ;
                      ; pass control to another terminal if requested
                      ;
                      ;#########################################################################
                passctrl:
                      ; this routine is called from the real time clock interrupt
                      ; EI and RETI have already been executed
                      ;
015C  3A1A03                 lda         ctrlflg
015F  B7                     ora         a
0160  C8                     rz                                  ; i dont have control anyway
0161  3A1903                 lda         freeflg
0164  B7                     ora         a
0165  C0                     rnz                                 ; im not free to do this
                      ;
                      ; check SRQ
                      ;
0166  C5                     push        b
0167  0109F7                 lxi         b,gpcsc                 ; 8292 status
                             inp         a
016A+ED78                    DB          0EDH,A*8+40H
016C  E620                   ani         20h                     ; check SRQ bit
016E  C1                     pop         b
016F  C8                     rz                                  ; SRQ not set - return
                      ; SRQ set - who was it?
```

```
                         ; first clear SRQ int
0170 C5                  push     b
0171 F60B                ori      0bh
0173 CDEC01              call     com92              ; IACK1
                    ;
                    ;initiate paralel poll
0176 3E40                mvi      a,lon
0178 0104F7              lxi      b,gptlsm
                         outp     a                  ; set listen only
017B+ED79                DB       0EDH,A*8+41H
017D AF                  xra      a
017E 0E05                mvi      c,low(gptlsxm)
                         outp     a                  ; reset lon
0180+ED79                DB       0EDH,A*8+41H
0182 3EF5                mvi      a,expp
0184 CDEC01              call     com92              ; execute pp
0187 3E80                mvi      a,ton
0189 0E04                mvi      c,low(gptlsm)
                         outp     a                  ; set ton
018B+ED79                DB       0EDH,A*8+41H
018D AF                  xra      a
018E 0E05                mvi      c,low(gptlsxm)
                         outp     a                  ; reset lon
0190+ED79                DB       0EDH,A*8+41H
0192 0E00                mvi      c,low(gptldi)
                         inp      c                  ; input pp response byte
0194+ED48                DB       0EDH,C*8+40H
0196 AF                  xra      a
            ppf2:
                         srar     c
0197+CB29                DB       0CBH, 28H+C
                         jrc      ppf1
0199+3803                DB       38H,PPF1-$-1
019B 3C                  inr      a                  ; this bit calculate device addres
                         jr       ppf2               ; of responding terminal
019C+18F9                DB       18H,PPF2-$-1
                    ;
            ppf1:
019E C640                adi      40h                ; calc talker address
01A0 CDDF01              call     gpibcom            ; send on GPIB
01A3 3E09                mvi      a,tct              ; take control message
01A5 CDDF01              call     gpibcom            ; send on GPIB
                    ;
                    ;
01A8 CDAD01              call     initasnc           ; initialise as not controller
01AB C1                  pop      b
01AC C9                  ret
                    ;
                    ;
                    ;
            initasnc:
            ; initialise as not controller routine
01AD 3E01                mvi      a,1
01AF 0104F7              lxi      b,gptlsm
                         outp     a
01B2+ED79                DB       0EDH,A*8+41H
```

```
                        ; not talker only or nor listener only
01B4 AF                 xra      a
01B5 0E05               mvi      c,low(gptla×m)
                        outp     a                         ; immediate pon
01B7+ED79               DB       0EDH,A*8+41H
01B9 321A03             sta      ctrlflg                   ; c in flag clear
01BC 3A0C03             lda      mda                       ; my device address
01BF 0E06               mvi      c,low(gptla01)
                        outp     a                         ; my address enabled
01C1+ED79               DB       0EDH,A*8+41H
01C3 3EA1               mvi      a,0a1h                    ; cpt enabled
01C5 0E05               mvi      c,low(gptla×m)
                        outp     a
01C7+ED79               DB       0EDH,A*8+41H
01C9 3A0C03             lda      mda
01CC F661               ori      61h                       ; form pp enable command
                        outp     a
01CE+ED79               DB       0EDH,A*8+41H
                        ; pp enabled
01D0 3EF1               mvi      a,gidl                    ; 92 go to idle
01D2 CDEC01             call     com92
01D5 CD0602             call     waitt                     ; wait for tct true
01D8 C9                 ret
                        ;
                        freegpib:
                        ; simply set free flag
01D9 3EFF               mvi      a,0ffh
01DB 321903             sta      freeflg
01DE C9                 ret
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;############################################################
                        ;
                        ;         utility routines
                        ;
                        ;############################################################
                        ;
                        gpibcom:
                        ; send command over gpib
                        ;
01DF 0100F7             lxi      b,gptldo
                        outp     a                         ; output command
01E2+ED79               DB       0EDH,A*8+41H
01E4 0C                 inr      c
                        gcom1:
                        inp      a                         ; get int 1 status
01E5+ED78               DB       0EDH,A*8+40H
01E7 E602               ani      bom
                        jrz      gcom1                     ; wait for completion
01E9+28FA               DB       28H,GCOM1-$-1
01EB C9                 ret
                        ;
                        ;
```

```
                          ;
                          com92:
                          ; send command to 8292
   01EC  0109F7                   lxi       b,gpcsc              ; 8292 control register
                                  outp      a
   01EF+ED79                      DB        0EDH,A*8+41H
   01F1  0E13                     mvi       c,low(gpio)
                          com921:
                                  inp       a
   01F3+ED78                      DB        0EDH,A*8+40H
   01F5  E604                     ani       iotcib               ; tci bit
                                  jrnz      com921               ; wait for tci false
   01F7+20FA                      DB        20H,COM921-$-1
   01F9  C9                       ret
                          ;
                          ;
                          ;
                          ;
                          ioportr:
                          ; read ioport and reset time out status
                          ;
   01FA  C5                       push      b
   01FB  0113F7                   lxi       b,gpio
                                  inp       a
   01FE+ED78                      DB        0EDH,A*8+40H
   0200  0E10                     mvi       c,low(gpswl)
                                  inp       c                    ; reset time out status
   0202+ED48                      DB        0EDH,C*8+40H
   0204  C1                       pop       b
   0205  C9                       ret
                          ;
                          ;
                          ;
                          ;
                          waitt:
                          ; wait for 8292 task complete
   0206  0113F7                   lxi       b,gpio
                          waitt1:
                                  inp       a
   0209+ED78                      DB        0EDH,A*8+40H
   020B  E604                     ani       iotcib
                                  jrz       waitt1
   020D+28FA                      DB        20H,WAITT1-$-1
   020F  C9                       ret
                          ;
                          ;
                          ;
                          ;
                          ;########################################################
                          ; hard disk controller routines
                          ;########################################################
                          ;
                          RDSJB:
                          ; read device specified jump byte from drive (A).
   0210  C640                     adi       40h                  ; form talker address
   0212  57                       mov       d,a
```

```
                    rdsj1:
0213  0601          mvi    b,1              ; one byte message
0215  210F03        lxi    h,statblk
0218  1E00          mvi    e,0              ; slow data rate
021A  CD9D00        call   recv
021D  3A0F03        lda    statblk
0220  C9            ret
                    ;
                    ;
                    COMSEN:
                    ; send command to drive (A) - command in C.
0221  F5            push   psw
0222  C620          adi    20h              ; form listen address
0224  57            mov    d,a
0225  79            mov    a,c              ; command
0226  320F03        sta    statblk
0229  0602          mvi    b,2              ; two byte message
022B  210F03        lxi    h,statblk
022E  1E00          mvi    e,0              ; use slow data rate
0230  CD4400        call   send
0233  F1            pop    psw
0234  C640          adi    40h
0236  57            mov    d,a              ; leave talker address in D
0237  DE40          sbi    40h              ; leave device address in A
0239  C9            ret
                    ;
                    ;
                    ;
                    REQSTAT:
                    ; request status from device (A)
                    ;
023A  0E03          mvi    c,3              ; request status command
023C  CD2102        call   comsen           ; send command
023F  0604          mvi    b,4              ; 4 byte reply
0241  210F03        lxi    h,statblk
0244  1E00          mvi    e,0              ; slow data rate
0246  CD9D00        call   recv             ; recieve reply
0249  210F03        lxi    h,statblk        ; point to reply block
024C  C9            ret
                    ;
                    ;
                    ;
                    INITST:
                    ; initiate self test on disk (A)
024D  0E1B          mvi    c,1bh            ; selftest command
024F  CD2102        call   comsen
0252  0E1C          mvi    c,1ch            ; read result byte command
0254  CD2102        call   comsen
0257  210F03        lxi    h,statblk
025A  1E00          mvi    e,0              ; slow data rate
025C  0601          mvi    b,1              ; one byte reply
025E  CD9D00        call   recv
0261  3A0F03        lda    statblk
0264  C9            ret
                    ;
                    ;
```

```
                        ;
                        ADDREC:
                        ;   address - record on drive (A) - address info in (HL)
                        ;
0265  E5                        push     h
0266  0E0C                      mvi      c,0ch              ; address record command
0268  CD2102                    call     comsen
026B  E1                        pop      h
026C  C620                      adi      20h
026E  57                        mov      d,a
026F  0604                      mvi      b,4                ; 4 more bytes to send
0271  1E00                      mvi      e,0                ; slow data rate
0273  CD4400                    call     send
0276  C9                        ret
                        ;
                        ;
                        ;
                        RDADD:
                        ;   read address record in (HL) from drive (A)
                        ;
0277  E5                        push     h
0278  0E14                      mvi      c,14h              ; read address command
027A  CD2102                    call     comsen
027D  E1                        pop      h
027E  0606                      mvi      b,6                ; 6 bytes to recieve
0280  1E00                      mvi      e,0                ; use slow data rate
0282  CD9D00                    call     recv
0285  C9                        ret
                        ;
                        ;
                        ;
                        ;
                        READ:
                        ;   read 1 physical sector into (HL) from drive (A)
                        ;   zero flag set on return if no time out error
                        ;
0286  E5                        push     h
0287  0E05                      mvi      c,5                ; read command
0289  CD2102                    call     comsen
028C  CDA202                    call     sensec             ; set one sector read
                        ;  talk address left in D
028F  E1                        pop      h
0290  1EFF                      mvi      e,0ffh             ; use fast data rate
0292  D5                        push     d
0293  0600                      mvi      b,0
0295  CD9D00                    call     recv               ; get first 256 bytes
0298  D1                        pop      d
0299  B7                        ora      a
029A  C0                        rnz
029B  0600                      mvi      b,0
029D  CD9D00                    call     recv               ; next 256 bytes
02A0  B7                        ora      a
02A1  C9                        ret
                        ;
                        ;
                        ;
```

```
                    SENSEC:
                    ; send sector count
02A2  210100                lxi     h,1
02A5  220F03                shld    statblk
02A8  0602                  mvi     b,2                 ; two byte sector count
02AA  1E00                  mvi     e,0
02AC  C620                  adi     20h
02AE  57                    mov     d,a
02AF  CD4400                call    send
02B2  C9                    ret
                    ;
                    ;
                    ;
                    ;
                    WRITE:
                    ; write 1 physical sector from (HL) to drive (A)
                    ; zero flag set if no time out error
02B3  E5                    push    h
02B4  0E08                  mvi     c,8h                ; write command
02B6  CD2102                call    comsen
02B9  CDA202                call    sensec              ; send byte count of 1
                    write1:
02BC  C620                  adi     20h                 ; calc listen address
02BE  57                    mov     d,a
02BF  E1                    pop     h
02C0  1EFF                  mvi     e,0ffh              ; use fast data rate
02C2  D5                    push    d
02C3  0600                  mvi     b,0
02C5  CD4400                call    send                ; send first 256 bytes
02C8  D1                    pop     d
02C9  B7                    ora     a
02CA  C0                    rnz
02CB  0600                  mvi     b,0
02CD  CD4400                call    send                ; send last 256 bytes
02D0  B7                    ora     a
02D1  C9                    ret
                    ;
                    ;
                    ;
                    ;
                    FORMAT:
                    ; format one track
02D2  0E18                  mvi     c,18h               ; format command
02D4  CD2102                call    comsen
02D7  CDA202                call    sensec              ; send track count of 1
02DA  C9                    ret
                    ;
                    ;
                    ;
                    VERIFY:
                    ; verify one sector
02DB  0E07                  mvi     c,7                 ; write command
02DD  CD2102                call    comsen
02E0  CDA202                call    sensec              ; send sector count of 1
02E3  C9                    ret
                    ;
```

```
                      ;
                      ;
                      ;
             WRTALT:
             ;write alternate sector - same parameters as write
  02E4 E5                   push      h
  02E5 0E1A                 mvi       c,1ah               ; write alternate command
  02E7 CD2102               call      comsen
                            jr        write1
  02EA+18D0                 DB        18H,WRITE1-$-1
                      ;
                      ;
                      ;
                      ;
             SETINT:
             ; set interleave ,(HL) points to interleave pattern.
                      ;
  02EC E5                   push      h
  02ED 0E19                 mvi       c,19h               ; set interleave command
  02EF CD2102               call      comsen
  02F2 C620                 adi       20h
  02F4 57                   mov       d,a
  02F5 E1                   pop       h
  02F6 0610                 mvi       b,16                ; sixteen bytes to send
  02F8 1E00                 mvi       e,0
  02FA CD4400               call      send
  02FD C9                   ret
                      ;
                      ;
                      ;
                      ;
             LOOPBACK:
             ; send byte D to drive (A)
             ; complement and return in A
  02FE 0E1D                 mvi       c,1dh               ; loopback command
  0300 F5                   push      psw
  0301 7A                   mov       a,d
  0302 321003               sta       statblk+1
  0305 F1                   pop       psw
  0306 CD2102               call      comsen
  0309 C31302               jmp       rdsj1
                      ;
                      ;##########################################################
                      ;
                      ;       data section
                      ;
                      ;##########################################################
                      ;
  030C         mda:      ds        1
  030D         mla:      ds        1
  030E         mta:      ds        1
  030F         statblk:  ds        10
  0319         freeflg:  ds        1
  031A         ctrlflg:  ds        1
  031B                   end
```