ADDENDUM 1:  (Prototype Boards ONLY)

Features not implemented on this board:

1. 512 Byte/Sector
2. Copy Command  Class 1, Opcode 00
3. RAM Diag.  Class 7, Opcode 00
4. Read ID  Class 7, Opcode 02
5. Drive Diag.  Class 7, Opcode 03
6. Seagate and Tandem Fast Step algorithm SEEK.

Discrepancy on the board layout:

1. LED designators on PC Board are backwards.

Additional codes for the LED display:

1. LED code 40 Hex = Controller is in the idle loop.
2. LED code C0 Hex = Controller is selected.

Discrepancy in the Firmware:

1. The Max Cylinder Adr Hi and Max Cyl Adr Lo in the Class 6 Opcode 2 command is in the wrong order on this board. For this board only, Byte 4 should contain Max Cylinder Adr Lo and Byte 5 should contain Max Cylinder Adr Hi as shown below:

   3  | Max Head Adr
   4  | Max Cylinder Adr Lo
   5  | Max Cylinder Adr Hi
   6  | Reduce Write Current Cyl

   Fixed: J. Tseng
ADDENDUM 2:

Changes concerning the STEP PULSE WIDTH parameter of the Class 6 Opcode 2 command.

Step Pulse Width for Winchester Drive.

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>Yield</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-6</td>
<td>6.8 us</td>
</tr>
<tr>
<td>7-13</td>
<td>13.6 us</td>
</tr>
<tr>
<td>14-20</td>
<td>20.4 us</td>
</tr>
<tr>
<td>21-27</td>
<td>27.2 us</td>
</tr>
</tbody>
</table>

Step Pulse Width for Floppy Drive

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Yield</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-17</td>
<td>17 us</td>
</tr>
<tr>
<td>18-25</td>
<td>25.5 us</td>
</tr>
<tr>
<td>26-34</td>
<td>34 us</td>
</tr>
<tr>
<td>35-42</td>
<td>42.5 us</td>
</tr>
<tr>
<td>43-51</td>
<td>51 us</td>
</tr>
<tr>
<td>52-59</td>
<td>59.5</td>
</tr>
<tr>
<td>60-68</td>
<td>68 us</td>
</tr>
<tr>
<td>Revision</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>1/14/82</td>
<td>Preliminary release</td>
</tr>
<tr>
<td>1/19/82</td>
<td>mod (C.T.)</td>
</tr>
<tr>
<td>2/1/82</td>
<td>Released as a &quot;PRELIMINARY SPECIFICATION&quot;</td>
</tr>
<tr>
<td>6/22/82</td>
<td>Updated Interleave Computation Section 4.3.1</td>
</tr>
<tr>
<td>7/30/82</td>
<td>Reorganized the document. JT</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

1.0 Introduction

2.0 DTC-520A Controller
   2.1 Features
   2.2 Electrical/Mechanical Requirements
   2.3 Physical Specification
   2.4 System Configuration
   2.5 Theory of Operation

3.0 DTC Host Bus
   3.1 Introduction
   3.2 Signal Definitions
   3.3 Theory of Operation
   3.4 Abnormal Terminations

4.0 Commands/Programming
   4.1 Description of Commands by Class
   4.2 Command Format
   4.3 Description of Commands by Opcode
   4.4 Completion Status Format
   4.5 Logical Address Computation
   4.6 Interleave Computation
   4.7 Error Code Descriptions
   4.8 LED Error Display

5.0 Diagnostic Philosophy
   5.1 Error Indicators

6.0 Sector Format
SECTION 1

INTRODUCTION

The DTC-520A Controller consists of a microprocessor-based controller with on-board data separator logic, and is able to control a maximum of 4 drives in any combination of up to 2 Seagate Technology ST506 fixed disk drives, or equivalent, and up to 4 5-1/4 inch floppy disk drives. The DTC-520A is a single printed circuit board, 5-3/4 inches by 10 inches.

Commands to the controller are issued over a bidirectional bus connected to the host computer via a host adapter. The data separator/"serdes" logic serializes bytes and converts to MFM data, and deserializes MFM data into 8-bit bytes.

Due to the microprogrammed approach utilized in the controller, extensive diagnostic capabilities are implemented. This methodology increases fault isolation efficiency and reduces system down time. Error detection and correction will tolerate media imperfections up to 4-bit burst errors.
## SECTION 2

### DTC520A CONTROLLER

#### 2.1 FEATURES

The capabilities supplied as standard with the DTC-520A are listed below:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AUTOMATIC SEEK AND VERIFY</strong></td>
<td>A seek command is implied in every data transfer command (READ, WRITE, CHECK, etc.). If the heads are not positioned over the correct cylinder, a seek is initiated, and a cylinder verification is performed after the seek completes.</td>
</tr>
<tr>
<td><strong>FAULT DETECTION</strong></td>
<td>Two classes of faults are flagged to improve error handling:</td>
</tr>
<tr>
<td></td>
<td>* Controller faults</td>
</tr>
<tr>
<td></td>
<td>* Disk faults</td>
</tr>
<tr>
<td><strong>AUTOMATIC HEAD AND CYLINDER SWITCHING</strong></td>
<td>If during a multi-block data transfer the end of a track is reached, the controller automatically switches to the next track. If the end of a cylinder is reached, the controller issues a seek and resumes the transfer.</td>
</tr>
<tr>
<td><strong>DATA ERROR SENSING AND CORRECTION</strong></td>
<td>If a data error is detected during a disk data transfer, the controller indicates whether or not it is correctable. If correctable the error is automatically corrected.</td>
</tr>
<tr>
<td><strong>LOGICAL TO PHYSICAL DRIVE CORRELATION</strong></td>
<td>Logical unit numbers (LUN's) are independent of physical port numbers. All accesses specify LUN's.</td>
</tr>
<tr>
<td><strong>ON BOARD SECTOR BUFFER</strong></td>
<td>A sector buffer is provided on the controller to eliminate the possibility of data overruns during a data transfer.</td>
</tr>
<tr>
<td>EFFICIENT HOST INTERFACE PROTOCOL</td>
<td>A bidirectional bus between the controller and host provides a simple yet efficient communication path. In addition, a high level command set permits effective command initiation.</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BYTE TRANSFER</td>
<td>The byte to byte data mode transfer is less than 1.5us.</td>
</tr>
<tr>
<td>SECTOR INTERLEAVE</td>
<td>Sector interleaving is programmable with up to 16 way interleave.</td>
</tr>
<tr>
<td>ODD PARITY</td>
<td>The 8 data bits on the interface bus can have odd parity. Depending on user preference, parity can be disabled.</td>
</tr>
<tr>
<td>256/512 BYTES PER SECTOR ON HARD DISKS</td>
<td>Switch selectable sector size.  256 bytes/sector – 33 sectors/track  512 bytes/sector – 18 sectors/track</td>
</tr>
<tr>
<td>NUMBER OF DRIVES</td>
<td>The controller will connect to a maximum of four (4) drives. The drives can be a combination of up to two (2) ST506 interface compatible Winchester drives and up to four (4) industry standard mini-floppy disk drives that are single or double density, single or double-sided.</td>
</tr>
<tr>
<td>PROGRAMMABLE DISK PARAMETERS</td>
<td>The disk parameters for both hard and floppy can be passed to the controller to define the drive characteristics.</td>
</tr>
<tr>
<td>PROGRAMMABLE FLOPPY TRACK FORMAT</td>
<td>The type of track format on the floppy media that is going to be used can be passed to the controller through software.</td>
</tr>
<tr>
<td>ALTERNATE TRACK ASSIGNMENT</td>
<td>The host can assign an alternate track for a defective track. Subsequent accesses to the defective track will cause the controller to transfer data from the new track automatically.</td>
</tr>
</tbody>
</table>
2.2 ELECTRICAL/MECHANICAL SPECIFICATIONS

Physical Parameters

Width 5-3/4 inches (tentative)
Length 10.0 inches
Height 0.49 inches
Weight 2.0 lbs.
Mounting holes are the same as that of the disk drive.

Environmental Parameters

Temperature
(degrees F/C) Operating: 32/0 to 131/55 Storage: -40/-10 to 167/75
Relative Humidity
(@ 40 degrees F, wet bulb temp., no condensation) 10% to 95% 10% to 95%
Altitude sea level to 10K feet sea level to 15K feet

Power Requirement

Voltage @ current +5 VDC + 5% @ 2.6A (max) 50 MVOLD P/P Ripple (max)

Controller Power Connections

1) No Connection
2) GND
3) GND
4) +5V @ 2.6A

power connector diagram
The electrical interface to the 5 1/4 in. drives will conform to the requirements described in the disk drive interface specification.
2.3 PHYSICAL SPECIFICATION

DTC520A Board Outline Diagram is shown in Fig. 2.0
<table>
<thead>
<tr>
<th>SWITCH</th>
<th>DRIVE TYPE SELECT</th>
<th>ON</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>LUN 0</td>
<td>HARD DISK</td>
<td>FLOPPY DISK</td>
</tr>
<tr>
<td>7</td>
<td>LUN 1</td>
<td>HARD DISK</td>
<td>FLOPPY DISK</td>
</tr>
<tr>
<td>6</td>
<td>LUN 2</td>
<td>ILLEGAL</td>
<td>FLOPPY DISK</td>
</tr>
<tr>
<td>5</td>
<td>LUN 3</td>
<td>ILLEGAL</td>
<td>FLOPPY DISK</td>
</tr>
<tr>
<td>4</td>
<td>NOT USED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>NOT USED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NOT USED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>HARD DISK</td>
<td>256 BYTES/SECTOR</td>
<td>512 BYTES/SECTOR</td>
</tr>
<tr>
<td></td>
<td>SECTOR SIZE</td>
<td>33 SECTORS/TRACK</td>
<td>18 SECTORS/TRACK</td>
</tr>
</tbody>
</table>

NOTE: LUN 2 & 3 CAN ONLY BE ASSIGNED AS FLOPPY DRIVES.

DEFAULT DRIVE TYPES

HARD DISK...SEagate TECHNOLOGY ST506 WITH BUFFERED STEP
FLOPPY DISK...SHUGART ASSOCIATES SA460, 16 SEC/TRACK

THESE ARE THE DEFAULT DRIVE TYPES ASSIGNED TO EACH DEVICE.
USE THE CLASS 6 COMMAND TO CHANGE TO THE ACTUAL DRIVE TYPE.
Parity Jumper Setting

If terminal A is jumpered to B then parity is enabled.
FIG. 2.0

J1....WINCHESTER CONTROL CABLE
J2, J3....WINCHESTER READ/WRITE CABLES
J4....HOST BUS CABLE
J5....UNUSED
J9....FLOPPY CONTROL CABLE
J10....POWER CONNECTOR
2.4 SYSTEM CONFIGURATION

The controller and data separator comprise a single PCB. A maximum of four (4) drives may be connected as shown in Fig. 2.1.

diagram goes here
### DTC 520A System Configuration

**Host Adapter**

- J4
- J1
- J2
- J3
- J9

**Winchester Drives**

**Floppy Drives**

**FIG. 2.1**

---

**Data Technology Corporation**

**Contract No.**

**Material**

- Drawn: L. FOK
- Checked: 8-5-82

**Finish**

**Issue**

**Size**

**FSCM No.**

**DWG. No.**

**Rev.**

**Scale**

**Sheet**
2.5 THEORY OF OPERATION

Disk commands are issued to the DTC-520A via the host bus following a defined protocol. The host initiates a command sequence by selecting the controller on the bus. If the controller is not busy, it requests command bytes from the host for task execution. (Command structure is described in Section 4.0). Depending on the type of command, the controller will request up to 10 bytes. Upon reception of the last command byte, the controller begins execution of the command.

For the data transfer commands, a check is performed on the disk address and status flagged if it exceeds the drive limits. The data is stored in a sector buffer before transfer to the host or disk drive. This buffer eliminates any possibility of data overruns between the host and the disk.

Upon completion of the command, the controller will send the completion status to the host and if error is indicated further delineation of the completion status may be requested by issuing the appropriate sense commands.

Odd parity is generated by the DTC-520A for all information that it puts on the I/O bus. If enabled, the DTC-520A flags all information that it receives with bad parity.
SECTION 3

DTC HOST BUS

3.1 INTRODUCTION

The DTC Host Bus is a negative-logic, bidirectional 8-bit data bus utilizing odd parity. The electrical interface consists of an open collector bus terminated on each end by a 220/330 ohm resistor network. The controller regulates transfers across the bus in such a way that permits connection host systems that utilize direct memory transfer capability as well as those that only support program input/output implementations.

3.2 ELECTRICAL REQUIREMENTS ON THE HOST INTERFACE

The Host Bus is based on an open-collector philosophy terminated on each end with 220 ohms to Vcc and 330 ohms to ground. The typical cable used is a 50-pin, mass terminated, cable with a characteristic impedance of approximately 100 ohms. The recommended drivers and receivers for the Host Bus is 7438 and 74LS240, respectively. The maximum length of the Host Bus is 20 feet.

3.3 SIGNAL DEFINITION

The signals that comprise the Host Bus are separated into those that are driven by the controller, those that are driven by the host adaptor (H/A), and those that are bidirectional. The term "asserted" means that the signal on the Host Bus is between 0V and 0.8V. The term "deasserted" means that the signal on the host bus is between 2.5V and 3.5V (negative or low true logic).

I. Unidirectional Signals Driven By Controller

<table>
<thead>
<tr>
<th>I/O</th>
<th>Input/Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When asserted, the data on the bus is driven by the controller. When deasserted, the data on the bus is driven by the host adapter. The host adapter must use this line to enable its drivers onto the data bus.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C/D</th>
<th>Command/Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When asserted, the bytes transmitted across the bus are interpreted as command bytes. When deasserted, the bytes are data bytes.</td>
</tr>
</tbody>
</table>

| BUSY  | This bit is asserted as a response to the SEL line from the host adapter and to indicate that the host bus is currently in use. |
MSG  Message
When asserted, indicates that the
command is completed. This bit is
always followed with the assertion of
I/O, and the assertion of REQ.

REQ  Request
This bit operates in conjunction with
I/O, C/D, & MSG. When asserted and I/O
is asserted, REQ will mean that the
data on the host bus is driven by the
controller. When asserted and I/O is
deasserted, REQ will mean that the data
is driven by the host adapter (H/A),
The following table lists the legal
states of the control bits on the
bus; the states are valid only when
REQ is asserted.

<table>
<thead>
<tr>
<th>I/O</th>
<th>C/D</th>
<th>MSG</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>a</td>
<td>d</td>
<td>Get a command byte from H/A</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td>d</td>
<td>Get a data byte from H/A</td>
</tr>
<tr>
<td>a</td>
<td>d</td>
<td>d</td>
<td>Send a data byte to H/A</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>d</td>
<td>Send the status byte to H/A</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
<td>Send the message byte to H/A</td>
</tr>
</tbody>
</table>

(Command is completed.)

a = asserted, d = deasserted

II. Unidirectional Signals Driven By Host Adapter

ACK  Acknowledge
This bit is asserted as a response to REQ from the
ccontroller. ACK must be returned for each REQ
assertion. The controller will wait for the
assertion of ACK before REQ is deasserted. The H/A
must not deassert ACK until after REQ has been
deasserted. If the H/A keeps ACK asserted, the
controller will not reassert REQ until after ACK is
deasserted. This provides the H/A with a means of
regulating the transfer of bytes across the bus.
Byte transfer regulation can occur for either
command or data bytes.

RST  Reset
When asserted, this bit forces the controller to the
beginning of its microcode. Following a Reset, the
controller will monitor the bus and wait for the H/A
to assert Select. Reset will immediately terminate
any pending command without the transmission of the
status or the message bytes. All signals to the
drives are deasserted. RST must be asserted for a
minimum of 250ns and a maximum of 1 second.
SEL
Select
When asserted, indicates the beginning of the command transaction. The H/A asserts SEL to gain the attention of the controller. Data Bit 0 on the bus must also be asserted during SEL time to select the controller. The controller will return BUSY as acknowledgement for SEL. After the assertion of BUSY, the H/A will deassert SEL. The controller will wait until SEL is deasserted before it asserts REQ. This provides a way for the H/A to hold off the command byte fetch until it is ready for the controller. SEL can be asserted immediately following a Reset.

III. Bidirectional Data

DB(7-0,P) - Data lines 7 thru 0 represent the eight data bits (DB0 = lsb). Parity is represented by P. The controller utilizes odd parity (the number of asserted bits on the host bus is always odd).
3.4 THEORY OF OPERATION

Every command to the controller is performed in 4 phases as follows:

1) Select Phase
2) Command Fetch Phase
3) Data Transfer Phase (if required)
4) Status & Message Phase

I. Select Phase

Following a Reset, or the completion of a command, the controller will monitor the Host Bus for the assertion of SEL. The host adapter asserts SEL and DB0 (controller address bit) on the Host Bus to indicate that a command is ready for the controller. It then waits for the controller to respond with BUSY. Upon reception of BUSY, the H/A deasserts SEL. The controller now has control of the host bus. Note: The H/A may keep SEL/DB0 asserted until it is ready to enter Phase 2.

II. Command Fetch Phase

After the H/A deasserts Select and DB0, the controller asserts C/D (to indicate command mode transfer), and deasserts I/O (to indicate output from the host adapter) to fetch the command bytes from the H/A. The command bytes are transferred over the host bus with the REQ/ACK handshake protocol until all command bytes are transferred to the controller. (The command byte fetch mode ends after the last REQ pulse from the controller is deasserted.) The H/A must not assert ACK until after REQ is asserted; H/A must not deassert ACK until after REQ is deasserted. The controller will wait until ACK from the previous byte transfer is deasserted before it reasserts REQ for to transfer the next byte. Note: This provides a means for the H/A to regulate the byte transfer across the Host Bus.

III. Data Transfer Phase

If the command does not require a transfer of data, this phase is skipped. If a data transfer is required for the command, such as read or write, the following occurs. The controller deasserts the C/D line to indicate data mode. Depending on the command type (read/write disk), the I/O bit on the host bus is asserted or deasserted by the controller, and the data is transferred (one byte at a time) with the same REQ/ACK handshake protocol used during the Command. After all bytes are transferred the controller exits this phase.

IV. Status & Message Phase

After all the data bytes have been transferred, a completion status byte is placed on the data bus by the controller - C/D and I/O are asserted, MSG deasserted. REQ is asserted and the controller waits for ACK from the host adapter. After the status byte transfer, the controller places zeros on the data bus and asserts C/D, I/O and MSG along with REQ to indicate to the host that the command is complete (this action can be used to generate an interrupt on the host system). After the H/A responds with ACK, the controller deasserts REQ, BUSY and
the command execution and the controller is now ready to be selected for the next command.

3.5 ABNORMAL COMMAND TERMINATION

If no errors occur, the command will proceed as previously described. However, if errors do occur the controller will terminate the command and immediately enter the status & message phase. Error conditions can be classified as follows:

1) Bus Parity errors - Upon the detection of a parity error in the byte (command or data) that is received from the H/A, the controller will complete the REQ/ACK handshake and enter the status and message phase. The status byte will indicate that a parity error has occurred.

2) Drive Interface or Controller related errors - After the six command bytes have been accepted, errors of this type can be detected. Upon the detection of the error condition (Drive Fault, Drive Not Ready, Illegal Command) the controller will enter the status & message phase. The status byte will have the error bit set.

3) Read/Write Channel errors - Upon detection of these errors (Read data error, Record not found, Drive Fault during a write) the controller may transfer a sector, or more, of data before it enters the status & message phase. The error bit is set in the status byte.

In cases 2 & 3 a Request Sense command may be issued to retrieve error information. In either case, any another command may be issued if the host system does not care about the details of the errors.
Timing Requirements for Controller Selection

I. Selection Phase

SEL

BUSY
| no time
| limit-->
| typ

DB0

100ns max--> | <--

REQ

ACK

Note: SEL must be deasserted before the controller will assert REQ. Busy will remain asserted during the entire command. The three assertion times for REQ and ACK are shown merely to indicate that the handshake occurs during the assertion time of Busy. It does not imply that there are only three REQ/ACK transactions during a command.
Timing Requirements for Data Transfer

(To host adapter, typical byte)

I/O

C/D

DB

REQ

ACK

Data driven by controller

|<->|100ns ->|<-- 0ns

min

min

500ns to no time

1st REQ limit

500ns typ

no time limit

*see note 3

*see note 3

Note 1. For Status Byte Transfer (I/O, C/D asserted & MSG deasserted); or Interrupt Byte Transfer (MSG, I/O, C/D asserted), REQ is asserted 500ns (typical) after the assertion of any of the above bits.

2. Data driven by the controller is stable 100ns min at the host adapter end before REQ is asserted and 0ns min after REQ is deasserted.

3. Although there is no time limit on the REQ-ACK handshake the entire transfer must be done within 52.43ms otherwise a time-out error will occur.

4. The Status & Message bytes are transferred in the same manner described in this subsection. Of course MSG is also asserted during this time.
Timing Requirements for Data Transfer  
(From host adapter, one byte)

I/O

C/D |

Data driven by host adapter

DB | | 100ns max

REQ

ACK

Note 1. Data driven by the host adapter is stable 100ns max (at the host adapter end) after ACK is asserted and 0ns min after REQ is deasserted.

2. For command mode transfers, SEL must be deasserted before ACK is asserted. This sequence follows the selection protocol.

3. Although no time limit on the REQ-ACK handshake, the entire transfer must be done within 52.43ms otherwise a time-out error will occur.

4. Command bytes are transferred using the same protocol. C/D, REQ are asserted and I/O, MSG are deasserted.
Host I/O Connector Pin Assignment.

The Host I/O Bus uses a 50-pin connector (AMP P/N 2-87227-5 or equivalent). The unused signal pins are considered to be spares for future use. The pin assignments are as follows:

<table>
<thead>
<tr>
<th>Signal</th>
<th>Pin Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA0</td>
<td>2</td>
</tr>
<tr>
<td>DATA1</td>
<td>4</td>
</tr>
<tr>
<td>DATA2</td>
<td>6</td>
</tr>
<tr>
<td>DATA3</td>
<td>8</td>
</tr>
<tr>
<td>DATA4</td>
<td>10</td>
</tr>
<tr>
<td>DATA5</td>
<td>12</td>
</tr>
<tr>
<td>DATA6</td>
<td>14</td>
</tr>
<tr>
<td>DATA7</td>
<td>16</td>
</tr>
<tr>
<td>PARITY</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>34</td>
</tr>
<tr>
<td>BUSY</td>
<td>36</td>
</tr>
<tr>
<td>ACK</td>
<td>38</td>
</tr>
<tr>
<td>RST</td>
<td>40</td>
</tr>
<tr>
<td>MSG</td>
<td>42</td>
</tr>
<tr>
<td>SEL</td>
<td>44</td>
</tr>
<tr>
<td>C/D</td>
<td>46</td>
</tr>
<tr>
<td>REQ</td>
<td>48</td>
</tr>
<tr>
<td>I/O</td>
<td>50</td>
</tr>
</tbody>
</table>

---

Future

Usage

Note: All signals are negative true and all odd pins are connected to ground. The signal lines are terminated with 220 ohms to 5V and 330 ohms to ground.
4.1 DESCRIPTION OF COMMANDS BY CLASS

Commands to the controller are categorized into various classes as indicated below:

Class 0  - Non-data Transfer, Data Transfer and Status Commands
Class 1  - Disk Copy Commands
Class 2-5 - Reserved
Class 6  - Disk Parameter Assign Commands
Class 7  - Diagnostic Commands
### 4.2 COMMAND FORMATS

#### Class 0 Command Format

The Class 0 commands follow the general format shown:

<table>
<thead>
<tr>
<th>Byte #</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OPCODE</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>LUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SLAD 2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>LAD 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>LAD 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td># of BLOCKS / INTERLEAVE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>CONTROL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Class 1 Command Format

The Class 1 commands follow the general format shown:

<table>
<thead>
<tr>
<th>Byte #</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OPCODE</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>SLUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SLAD 2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SLAD 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SLAD 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td># of BLOCKS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>DLUN</td>
<td></td>
<td></td>
<td></td>
<td>DLAD 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DLAD 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DLAD 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>set to 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CONTROL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Class 6 Command Format

The Class 6 commands follow the general format as shown:

<table>
<thead>
<tr>
<th>Byte #</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OPCODE

LUN set to 0

set to 0

set to 0

set to 0

set to 0 except Opcode 0
Class 7 Command Format

The Class 7 commands follow the general format shown:

<table>
<thead>
<tr>
<th>Byte</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OPCODE</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>LUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LAD 2</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>LAD 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>LAD 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INTERLEAVE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CONTROL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Description of the fields used in the Command Descriptor Blocks

LUN

This field specify the logical unit number of the drive we are attempting to communicate with. It corresponds to Drive Select jumper on the drives as follows:

- Drive Select 1 = LUN 0
- Drive Select 2 = LUN 1
- Drive Select 3 = LUN 2
- Drive Select 4 = LUN 3

SLUN = Source drive LUN
DLUN = Destination drive LUN

LAD (0-2)

This is a 21 bit logical sector address of the beginning sector of the group of sectors we wish to access. LAD 0 is the LSB. The LAD field can be from 0 up to the total number of sectors available on a particular drive.

# OF BLOCKS

This field holds the number of sectors we wish to transfer per command. A value of 0 will result in a transfer of 256 sectors.

INTERLEAVE

This field is used to tell the controller what interleave to use during the format commands, check track command or the read id command.

Interleave is a factor used while formatting a drive so the user can optimize the throughput of the data transfer of the system. The throughput of the system depends on the controller's turnaround time for next sector and the data transfer rate on the host bus.
CONTROL

This field contains the control bits that tells the controller how to react if an error condition is encountered during the command execution phase.

This byte is defined as follows:

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disable Data Error Correction</td>
<td>Disable Retry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Disable Retry

If this bit is set, the controller will not attempt to retry the command upon certain retryable error conditions. If it is not set, a total of eight retries will be performed before an error is reported. If a retry is successful, the controller will not report the error to the host. The following errors can result in a retry attempt:

a) Record not found during Read or Write commands
b) Seek errors during Read or Write commands
c) Uncorrectable data error during Read commands

For error conditions (a) & (b), a recalibrate, reseek, and reread are performed. For error condition (c), only a reread is performed.

Disable Data Error Correction

If this bit is set, the controller will not correct the data that is read from the disk if an ECC error occurred during a read. If it is not set, data errors will always be corrected if correctable, before being transferred to the host.

The information returned by the Request Sense command will indicate whether or not the data error is correctable.

Regardless of the error condition, the data is transferred to the host.
4.3 DESCRIPTION OF COMMANDS BY OPCODE

Class 0 Commands

Opcodes 00 Test Drive Ready

This command selects the specified drive and verifies that the drive is ready for access.

The required fields for this command are: OPCODE and LUN.

Opcodes 01 Recalibrate

This command positions the R/W arm of the drive to Track00 position and clears any possible error status in the drive.

The required fields for this command are: OPCODE and LUN.

Opcodes 02 Not Used

Opcodes 03 Request Sense

This command returns 4 bytes of drive and controller sense as Data (C/D deasserted) for the specified LUN. It must be issued immediately after an error to get the valid sense of the error.

The required fields for this command are: OPCODE and LUN.

The sense bytes are returned in the following format.

<table>
<thead>
<tr>
<th>Byte</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SENSE BYTE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>LUN</td>
<td>LAD 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LAD 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>LAD 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Definitions of the fields used in the Sense Block

SENSE BYTE

This is the byte that describe the details or the nature of the error itself. The bits are defined as follows:

<table>
<thead>
<tr>
<th>Byte 0</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
</table>
|        |   |   |   |   |   |   |   |<--Error code-->
|        |   |   |   |   |   |   |   | Error type
|        |   |   |   |   |   |   |   | Spare (set to zero)
|--------|---|---|---|---|---|---|---| Block address valid

Block Address Valid (Bit 7)
This bit indicates that the Logical Sector Address, LAD in bytes 1 thru 3 contain the valid logical address of the block at which the error occurred.

Error Type (Bit 5,4)
These two bits describes the general type of error. It can contain either one of the following:

00 for controller related errors
01 for drive related errors
10 for command related errors
11 for miscellaneous errors

Error Code (Bit 3-0)
This is a 4 bit field that describes the actual error interpretation under each general type of error.

LUN

This byte indicates the logical unit number of the drive where the error had occurred.

LAD 2,1,0

This is a 21 bit logical address of the sector at which the error occurred. This field is valid only if the Block Address Valid bit is set in the SENSE byte.
Opcodes

04 Format Drive

This command formats all the tracks on the specified drive with the selected track format. The sectors will be placed on the tracks according to the interleave code specified in the command block and the data fields will be filled with data pattern E5 hex.

The required fields for this command are: OPCODE, LUN and INTERLEAVE.

05 Check Track

Check track command checks the track format on the specified track for the correctness of the contents of the ID fields and the interleave of the sectors. It does not read the data field.

The required fields for this command are: OPCODE, LUN, LAD 2,1,0 and INTERLEAVE.

06 Format Track

This command formats the specified track with no flags set in the ID fields of all sectors on the track. It also fills the data field with data pattern E5 hex.

The required fields for this command are: OPCODE, LUN, LAD 2,1,0 and INTERLEAVE.

07 Format Bad Track

Format bad track command formats the specified track with the bad block flag set in all ID fields on the track. Data pattern of E5 hex is filled in the data field.

The required fields for this command are: OPCODE, LUN, LAD 2,1,0, and INTERLEAVE.

08 Read

This command reads the specified number of sectors starting from the initial block address given in the LAD field and transfers them to the host.

The required fields for this command are: OPCODE, LUN, LAD 2,1,0, # of blocks and CONTROL.

0A Write

The write command gets the data from the host and writes the specified number of sectors starting from the initial block address given in the LAD field.

The required fields for this command are: OPCODE, LUN, LAD 2,1,0, NUMBER OF BLOCKS and CONTROL.
Opcode 0B Seek

This command initiates a seek to the cylinder where the block specified in the LAD field is located. For winchester drives capable of doing overlap seeks, depending on how the drive parameter is set up, it could immediately return completion status before the seek complete is found from the drive. Normally it returns the completion status only after the seek is all done.

The required fields for this command are: OPCODE, LUN, and LAD 2,1,0.

Opcode 0C Not Used

Opcode 0D Not Used

Opcode 0E Assign Alternate Track

This command formats the track specified in the LAD field with the alternated bad track flag set in the ID fields and with the track address of the alternate track written in the data fields. Also it formats the alternate track itself with the alternate track identifier flag set in its ID fields and data pattern E5 hex filled in the data field.

Future write/read access to that specified track will cause the drive to seek to the alternate track, transparent to the host software, and do the write/read operations there.

The required fields for this command are: OPCODE, LUN, LAD 2,1,0 and INTERLEAVE.

After the controller received the command, it will collect the alternate track address as data (C/D deasserted) from the host in the following format:

<table>
<thead>
<tr>
<th>Byte #</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LAD 2</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LAD 1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LAD 0</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>set to 0</td>
</tr>
</tbody>
</table>

NOTE: Alternate tracks can be assigned one level only. That is, an alternate track cannot have another alternate track assigned to it.
Class 1 Commands

Opcodes 00: Copy Blocks

<<TBD>>
Class 6 Commands

Opcode 00 Define Floppy Disk Track Format

This command is used for defining the track format of the floppy media to the controller, i.e. sides, density and bytes per sector.

The required fields for this command are: OPCODE, LUN and Byte 5.

The following track format codes are supported by the controller and the code byte is passed to the controller through Byte 5 of the Class 6 command block in the following format.

<table>
<thead>
<tr>
<th>Track Format Code</th>
<th>Track Format Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 hex</td>
<td>Single density, single-sided; FM recording, 128 bytes/sector, 16 sectors/track.</td>
</tr>
<tr>
<td>01 hex</td>
<td>Single density, double-sided; FM recording, 128 bytes/sector, 16 sectors/track.</td>
</tr>
<tr>
<td>06 hex</td>
<td>Double density, single-sided; Side 0, cylinder 0 - FM recording, 128 bytes/sector, 16 sectors/track. All other tracks - MFM recording, 256 bytes/sector, 16 sectors/track.</td>
</tr>
<tr>
<td>07 hex</td>
<td>Double density, double-sided; Side 0, cylinder 0 - FM recording, 128 bytes/sector, 16 sectors/track. All other tracks - MFM recording, 256 bytes/sector, 16 sectors/track.</td>
</tr>
<tr>
<td>86 hex</td>
<td>Double density, single-sided; MFM recording, 256 bytes/sector, 16 sectors/track.</td>
</tr>
<tr>
<td>87 hex</td>
<td>Double density, double-sided; MFM recording, 256 bytes/sector, 16 sectors/track.</td>
</tr>
<tr>
<td>8A hex</td>
<td>Double density, single-sided; MFM recording, 512 bytes/sector, 8 sectors/track.</td>
</tr>
<tr>
<td>8B hex</td>
<td>Double density, double-sided; MFM recording, 512 bytes/sector, 8 sectors/track.</td>
</tr>
</tbody>
</table>

Note: If track format information for floppy is not specified after each reset or power-on, the default code will be as follows:

Track format code 06 (hex).
Double density, single-sided;
512 bytes/sector, 8 sectors/track.
Opcodes 02 Assign Drive Parameters

This command allows the host to setup the disk step pulse width, step period, step mode, max head address, max cylinder address, timing/delays and cylinder address to apply reduced write current for the specified LUN.

The required fields for this command are: OPCODE, and LUN.

Depending on how the switches are set up on the controller, 10 bytes of parameters will be collected from the host as Data (C/D deasserted) using 2 different formats for floppy drives or Winchester drives.

The parameters for a Winchester drive are passed as data to the controller using the following format.

<table>
<thead>
<tr>
<th>Byte #</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Step Pulse Width</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Step Period</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Step Mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Max Head Adr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Max Cylinder Adr Hi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Max Cylinder Adr Lo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Reduce Write Current Cyl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Drive Type Identifier</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>set to 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>set to 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Description of the parameters for Winchester drives

Step Pulse Width

This is the time the STEP signal is asserted in lusec increments.
Range- 1.0usec to 256.0usec

Step Period

This is the time between two step pulses. For the purpose of simplifying logics, this is the time the step signal is deasserted between two step pulses in 50usec increments.
Range- 50usec to 128msec.

Step Mode

This defines the type of stepping the drive is set up to do. The following describes the modes supported by the controller.

Mode 0    Normal or Buffered Step
This mode generates step pulses using the pulse width & rate specified in Byte 0,1)

Mode 1    Seagate fast step algorithm
This mode steps the drive according to the Seagate fast step algorithm and it ignores the pulse width and rate specified in Byte 0,1.

Mode 2    Tandon fast step algorithm
This mode steps the drive as described in the Tandon fast step algorithm and it ignores the pulse width and rate specified in Byte 0,1.

Note: For Mode 1 & 2, refer to the drive specification on how to set up the drive to get the fast step mode.

Max head address

This bytes specifies the maximum head address on the drive.
Range- 0 to 7 (i.e. 1 to 8 heads)

Max cylinder address

These two bytes specify the maximum cylinder address on the drive.
Range- 0 to 1023 (i.e. up to 1024 cylinders)

E.g. Byte 4 = 0, Byte 5 = 179 => 180 cylinders.

Reduced Write Current Cyl

This byte specifies the cylinder address where the reduced write current is to be first applied. If the value is '0', reduced write current is never asserted.
Range- 1 to 255

E.g. 77 = the reduced write current is applied for all cylinders greater than or equal to 77 (during write operation)
Drive type identifier

This byte is defined as follows.

Bit 7
Must be set to 0 to indicate parameters are for hard disk.

Bit 6
This bit tells the controller whether to wait for the seek complete after a seek command (Class 0, Opcode 0B) to the Winchester drive. If the drive is capable of overlap seek, this bit can be set to utilize the function.

   e.g. Bit 6 = 0 ==> wait for seek complete after the last step pulse.
       Bit 6 = 1 ==> do not wait for seek complete after the last step.

NOTE: The seek incomplete timeout is set to approximately 700 msec. That is, any seek must complete within 700 msec or the controller will report an error.

Default Parameters for Winchester drive type.

Upon reset, the controller will default to the parameters of Seagate Technology ST506 disk drive as shown below for the LUN's that are preassigned as Winchester drives on the dipswitch. The user should set the parameters for the type of Winchester drive being used after every reset or power-on.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>11 Usec Step pulse</td>
</tr>
<tr>
<td>1</td>
<td>60</td>
<td>3.0 Msec Step period</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>Buffered step mode</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4 heads</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>High cylinder byte is zero</td>
</tr>
<tr>
<td>5</td>
<td>152</td>
<td>153 cylinders</td>
</tr>
<tr>
<td>6</td>
<td>77</td>
<td>Reduce Wr Current at cyl. 77 and above</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
The parameters for a mini floppy drive is passed to the controller using the following format.

<table>
<thead>
<tr>
<th>Byte #</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Step Pulse Width</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Step Interval</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max Cylinder Address</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Head Settling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hd Sel --&gt; Valid Rd/WR</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dr Sel --&gt; Valid Rd/WR</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Delay after Wr Gate off</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description of the parameters for floppy drives

Step Pulse Width

This is the period of time when the STEP signal is asserted in nusec increments.
Range- 1.0nusec to 256nusec

Step Interval

This byte specifies the period of time when the STEP signal is deasserted in msec increments
Range- 1.0msec to 256.0msec

Max Cylinder Address

This byte specifies whether the mini floppy drive is a single or double TPI drive. For compatibility sake, it should contain either the value 34 (decimal) for a single TPI drive or 79 (decimal) for a double TPI drive.

Head Settling

This specifies the delay required from last STEP pulse to valid Rd/WR in msec increments.
Range- 1.0msec to 256.0msec
Hd Sel --> Valid Rd/Wr

This byte indicates the delay required from head selection to valid Rd/WR in lusec increments.
Range- 1.0usec to 256.0usec

Dr Sel --> Valid Rd/WR

This byte contains the delay required from drive selection to valid Rd/WR in 1 msec increments.
Range- 1.0msec to 255.0msec

A value of 0 in this field means that this delay will not be applied as in the case of some mini floppy drives.

Delay after Wr Gate off

The delay required from the time Write Gate becomes inactive to Side Selection/ Drive Deselection/ Next Step Sequence in 1msec increments.
Range- 0.1msec to 25.5msec.

A value of 0 in this field means that this delay will not be applied as in the case of some mini floppy drives.

Default Parameters for a mini floppy drive.

Upon reset, the controller will default to the parameters of Shugart SA460 drive as shown below for the drives that are preassigned as floppy drives on the dipswitch. The user should set the parameters for the type of floppy drive being used after every reset or power-on.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>2.0usec Step Pulse Width</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>7.0msec Step Interval</td>
</tr>
<tr>
<td>2</td>
<td>79</td>
<td>80 cylinders</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>22.0msec Head Settling</td>
</tr>
<tr>
<td>4</td>
<td>205</td>
<td>205.0usec Hd Sel to Valid Rd/Wr</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.0msec Dr Sel to Valid Rd/Wr</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>1.1 msec Delay after Wr Gate off.</td>
</tr>
<tr>
<td>7</td>
<td>80 (Hex)</td>
<td>Mini floppy parameter indicator</td>
</tr>
<tr>
<td>8,9</td>
<td>0</td>
<td>Set spare bytes to zero</td>
</tr>
</tbody>
</table>
Class 7 Commands

Opcodes 00 RAM Diagnostic

This command performs a data pattern test on the RAM on the controller.

The required field for this command is: OPCODE.

Opcodes 01 Write ECC

This command writes a block of data to the disk without generating ECC for the data. It displaces the data on the disk by three bytes so that the ECC bytes for the data can be written from the host in order to verify the ECC logic of the controller.

The required fields for this command are: OPCODE, LUN, LAD 2,1,0, and CONTROL.

Opcodes 02 Read ID

This command reads only the 3 ID bytes and 3 ECC bytes of the specified sector and transfers them to the host as data (C/D deasserted).

The required fields for this command are: OPCODE, LUN, LAD 2,1,0, and INTERLEAVE.

Opcodes 03 Drive Diagnostic 0

This command performs a diagnostic on the specified LUN. It reads sector 0 on all tracks sequentially and then reads sector 0 on 256 random tracks.

The required fields for this command are: OPCODE and LUN.

Opcodes 06 Request Logout

This command is used to retrieve four bytes of error log for the specified LUN. Each device has its own error log area which is accumulated every time certain errors occurred and it is cleared every time after transferring the error log.

The required fields for this command are: OPCODE and LUN.
The log is transferred to the host as Data (C/D deasserted) using the following format.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Retry count high</td>
</tr>
<tr>
<td>1</td>
<td>Retry count low</td>
</tr>
<tr>
<td>2</td>
<td>Permanent error high</td>
</tr>
<tr>
<td>3</td>
<td>Permanent error low</td>
</tr>
</tbody>
</table>

The following errors get accumulated in the log:
1. Uncorrectable error in ID field.
2. Uncorrectable error in data field.
3. No ID address mark
4. No data address mark.
5. Seek error.
6. Record not found.

4.4 COMPLETION STATUS FORMAT

The following describes the Completion Status Byte returned to the host after every command.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUN</td>
<td>Spare</td>
<td>Error</td>
<td>Parity error</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Definition of the bits

Parity error
A 1 in this bit indicates that a parity error occurred during transfer from host to controller.

Error bit
A 1 in this bit indicates that an error occurred during the execution of a command.

Spare bits are always zero.

LUN
These bits indicate the logical unit number of the drive where the error occurred.
4.5 LOGICAL ADDRESS COMPUTATION

The logical address is computed using the following formula:

Logical addr = (CYA * #HD per CYL + HDA) * #SE per TRK + SEA

Where: CYA = cylinder address
       HDA = head address
       SEA = sector address
       #HD per CYL = number of heads per cylinder
       #SE per TRK = number of sectors per track

Note: Bit Ø of LAD Ø = the least significant bit.
      Bit 4 of LAD 2 = the most significant bit.

4.6 INTERLEAVE COMPUTATION

The interleave mechanism on a track can be represented as follows.

Example: Interleave Code = 10 (decimal)
         Maximum Sector Number = 33

The physical sectors are assigned the following logical sector numbers.

| Phy. Sec. | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 |
| Log. Sec. | 0,10,20,30, 1,11,21,31, 2,12,22,32, 3,13,23,33, 4,14,24, 5 |

| Phy. Sec. | 20 21 22 23 24 25 26 27 28 29 30 31 32 33 |
| Log. Sec. | 15,25, 6,16,26, 7,17,27, 8,18,28, 9,19 29 |
4.7 ERROR CODE DESCRIPTIONS

These error code descriptions are related to Byte 0 of the Sense Block after the Request Sense command (Class 0, Opcode 03).

Type 0 (Drive) Error Codes

0  No status
1  No Index signal.
2  No Seek Complete.
3  Write fault
4  Drive not ready
5  Drive not selected.
6  No Track00
7  Multiple winchester drives selected.
D  Seek in progress

Type 1 (Controller) Error Codes

0  ID read error. ECC error in the ID field.
1  Uncorrectable data error during a read.
2  ID Address Mark not found.
3  Data Address Mark not found.
4  Record not found. Found correct cylinder and head but not sector.
5  Seek error. R/W head positioned on a wrong cylinder and/or selected a wrong head.
6  Unused.
7  Write protected.
8  Correctable data field error.
9  Bad block found.
A  Format Error. The controller detected that during the Check Track command, the format on the drive was not expected.
C  Unable to read the Alternate Track address.
E  Attempted to directly access an Alternate Track.
F  Sequencer time out during disk or host transfer.

Type 2 (Command) Error codes

0  Invalid Command received from the host.
1  Illegal disk address. Address is beyond the maximum address.
2  Illegal function for the current drive type.
3  Volume overflow.

Type 3 (Misc) Error codes

0  RAM error. Data error detected during Sector buffer RAM diagnostic.
4.8 LED ERROR DISPLAY

The following error codes are displayed on the LEDs on the controller in conjunction with any error condition on the controller indicated by the Error bit in the Completion Status Byte.

<table>
<thead>
<tr>
<th>0 0 0 0 0 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS7</td>
</tr>
<tr>
<td>DS0</td>
</tr>
</tbody>
</table>

Error Code (HEX) Interpretation

00 No Error
01 No Index from drive
02 No Track 00 from drive
03 Sector Address Out of Bounds
04 Winchester disk not selected
05 No Seek Complete from Winchester disk
06 No ID Address Mark
07 No Data Address Mark
08 Seek Error (Cylinder or Head not correct)
09 Sector not found
0A ID ECC error
0B Not used
0C Invalid Command
0D Incorrect DATA MARK
0E Incorrect ID MARK
0F Incorrect cylinder address from drive
10 Incorrect sector address from drive
11 Incorrect head address from drive
12 Uncorrectable Data Error
13 Correctable Data Error
14 Drive not READY
15 Write fault
16 not used
17 Drive write protected
18 RAM diagnostic error
19 - 1E not used
1F Unable to read the Alternate Track Address
20 Parity Error from host adaptor. If this error occurs, the host adaptor has a fault in the parity generation circuitry.
21 Bad Block detected from drive
22 Invalid function for this type
31 Attempted to directly access an alternate Track
32 Seek in progress
33 Volume overflow
81 Multiple Winchester disks selected. Fatal error.
82 Sequencer time-out during disk or Host transfer.
SECTION 5

DIAGNOSTIC PHILOSOPHY

5.1 ERROR INDICATORS

The controller contains 8 diagnostic LED error indicators. Each time an error occurs the controller deposits a value in the LED's and returns a failure status to the host adapter. The LED value can be decoded, but the error it indicates will always be available to the host software. The errors that are returned by the controller are very detailed. As a result, preliminary fault isolation is made fairly easily, narrowing the failure to the particular interface portion of the controller.
SECTION 6

SECTOR FORMATS

Winchester Drive Sector Formats

The track layout for the 256 bytes/sector 33 sectors/track is shown below.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>a</td>
<td>F</td>
<td>I</td>
<td>I</td>
<td>e</td>
<td>0</td>
</tr>
<tr>
<td>bytes</td>
<td>m</td>
<td>E</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>c</td>
</tr>
<tr>
<td>00's</td>
<td>012</td>
<td>00's</td>
<td></td>
<td>data</td>
<td>c</td>
<td>4E's</td>
</tr>
</tbody>
</table>

am, FE, ID0, ID1, ID2, 00, F8 = 1 byte
ecc = 3 bytes

Track Capacity = 10416

16 = Index Gap (4E)
10197 = 33 sectors @ 309 bytes/sector
203 = Speed Tolerance Gap (4E)
--------
10416

309 bytes/sector including ID and overhead

This track format provides (+ or -) 1.77% speed tolerance.
The track layout for the 512 bytes/sector, 18 sectors/track is shown below.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>a</td>
<td>F</td>
<td>I</td>
</tr>
<tr>
<td>bytes</td>
<td>m</td>
<td>E</td>
<td>D</td>
</tr>
<tr>
<td>00's</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

am, FE, ID0, ID1, ID2, 00, F8 = 1 byte
ecc = 3 bytes

Track Capacity = 10416

16 = Index Gap (4E)
10242 = 18 sectors @ 569 bytes/sector
158 = Speed Tolerance Gap (4E)

10416

569 bytes/sector including ID and overhead

This track format provides (+ or -) 1.29% speed tolerance.