# *LOTUS 800*
# CONTROLLER MANUAL

Revision 01

## NOTICE

Every attempt has been made to make this manual complete, accurate and up-to-date. However, all information herein is subject to change due to updates. All inquiries concerning this manual should be directed to POINT 4 Data Corporation.

POINT 4 Data Corporation
15442 Del Amo Avenue
Tustin, CA 92680
(714) 259-0777

# REVISION RECORD

PUBLICATION NUMBER: HM-128-0070

| Revision | Description | Date |
|---|---|---|
| 01 | Preliminary Release | 3/18/88 |

# LIST OF EFFECTIVE PAGES

Changes, additions, and deletions to information in this manual are indicated by vertical bars in the margins or by a dot near the page number if the entire page is affected. A vertical bar by the page number indicates pagination rather than content has changed. The effective revision for each page is shown below.

| Page | Rev | Page | Rev | Page | Rev |
|---|---|---|---|---|---|
| Cover | - | | | | |
| Title thru x | 01 | | | | |
| 1-1 thru 1-6 | 01 | | | | |
| 2-1 thru 2-28 | 01 | | | | |
| 3-1 thru 3-44 | 01 | | | | |
| 4-1 thru 4-9 | 01 | | | | |
| 5-1 | 01 | | | | |
| A-1 | 01 | | | | |
| Comment Sheet | 01 | | | | |
| Mailer | - | | | | |
| Back Cover | - | | | | |

# PREFACE

This manual describes the POINT 4 LOTUS 800 Controller, a multi-function controller that supports a maximum of four ESDI 5-1/4-inch hard disk drives and one 1/4-inch streaming tape drive.

Section 1, Overview, includes a general description, features, and specifications for the LOTUS 800.

Section 2, Interface Definitions, provides information about the internal and external interfaces of the LOTUS 800.

Section 3, Programming Instructions, provides information about the formats and instructions for programmed input/output and command descriptor blocks.

Section 4, Installation, provides the information and instructions required to install the LOTUS 800 in a POINT 4 MARK system.

Section 5, Operating Procedures - TBS

## Related Documentation

Related manuals include:

| Title | Document Order No. |
|---|---|
| MARK 6 System Installation/Maintenance Manual | HTP0059 |
| MARK 12 System Installation/Maintenance Manual | HTP0061 |
| MARK 6E/12E System Installation/Maintenance Manual | HTP0071 |
| IRIS R9 System Configuration Manual | ITP0029 |
| IRIS R9 System Manager Manual | ITP0030 |
| IRIS R9 User Reference Manual | ITP0034 |
| LOTUS DISCUTILITY Manual | ITP0026 |
| LOTUS 800 Disk/Tape Controller Diagnostics Document | |

# CONTENTS

**APPENDICES**

# FIGURES

## TABLES

# Section 1

# OVERVIEW

## 1.1 GENERAL DESCRIPTION

The POINT 4 LOTUS 800 Controller is a multi-function controller that supports one to four 5-1/4-inch hard disk ESDI (Enhanced Small Device Interface) drives and one 1/4-inch streaming tape drive.

The LOTUS 800 Controller requires IRIS R9.1.3 or later revision and is also compatible with Nova* systems. Figure 1-1 illustrates the interfaces between the LOTUS 800 and a MARK 6E/12E system.

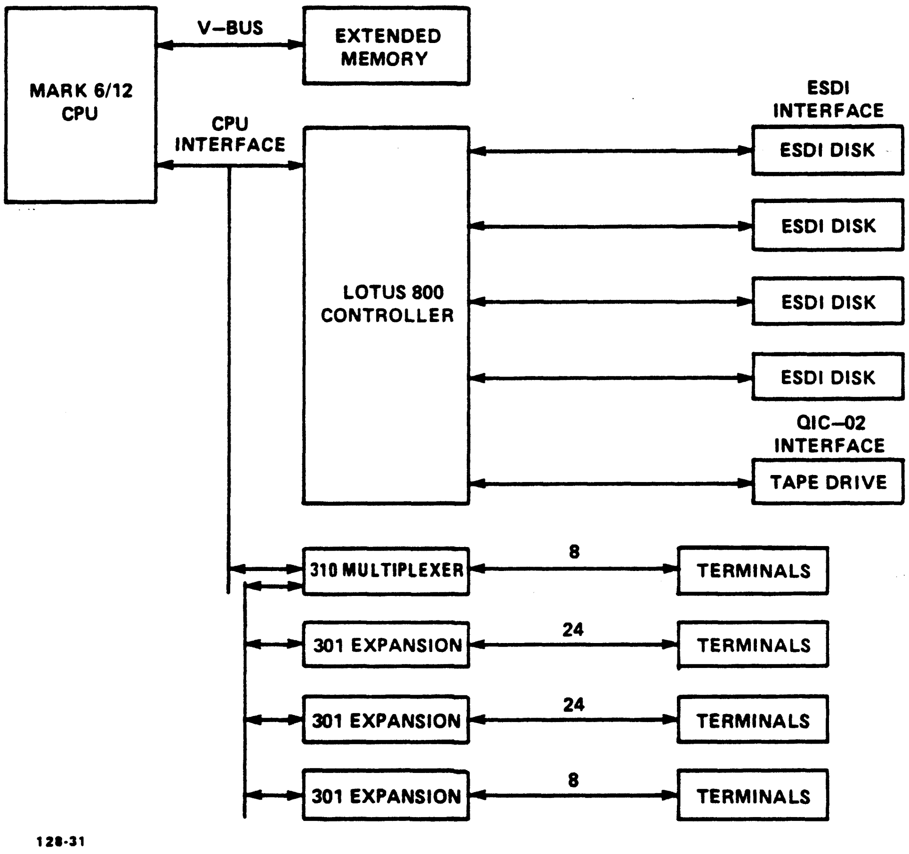


**Figure 1-1. LOTUS 800 Controller Interfacing with a MARK 6E/12E**

*Nova is a trademark of Data General Corporation.

## 1.2 FEATURES

Advanced design features ensure the compatibility, reliability, and flexibility of the LOTUS 800 Controller. Key features that relate to disk and tape operations are as follows:

**Disk Operations**

- Transfers data at a rate of up to 15 Mbits per second
- Uses linked Command Descriptor Blocks (CDBs) to enable multiple disk operations with minimum central processing unit (CPU) intervention
- Has full track buffering to enable maximum data throughput
- Includes an onboard microprocessor for intelligent CPU/disk interface, buffer level management, initialization and internal diagnostics
- Supports overlapped seeks when two or more ESDI drives are present
- Supports a sector size of 512 bytes
- Provides single- or double-burst Reed-Solomon error detection and correction

**Tape Operations**

- Provides a QIC-02 intelligent interface and QIC-24, QIC-120, and QIC-150 tape formats
- Supports concurrent file-by-file backup

## 1.3 SPECIFICATIONS

LOTUS 800 Controller specifications are provided in Tables 1-1 through 1-6. Interface specifications include information about central processing unit (CPU), disk, and tape interfaces; physical specifications include information about the dimensions, cables, and connectors; and power and environmental specifications state the requirements to ensure optimal operation.

**TABLE 1-1. LOTUS 800-CPU INTERFACE**

| Item | Parameter |
|---|---|
| I/O bus | POINT 4 and Nova-type computer I/O bus compatible |
| Backplane wiring | None required* |
| I/O bus loading | Single 74ALS00-type input load<br>Single 74F244-type output driver |
| Device code | Switch-programmable<br>Standard = 45 (disk), 42 (tape) |
| Priority mask bit | 8 (disk), 3 (tape) |
| DMA transfer rate | 1.209 megabytes/second<br>(1.65 microsecond/DMA cycle) |

*Modification of the backplane wiring is necessary in a non-POINT 4 chassis, i.e., jumper pins A94 and A96 (DCHPIN- and INTPIN-).

**TABLE 1-2. LOTUS 800-DISK DRIVE INTERFACE (1 of 2)**

| Item | Parameter |
|---|---|
| Disk drives supported | ESDI interface standard compatible |
| Drives per controller | 4 maximum |
| Drive size | Any mixture |
| Encoding method | Differential NRZ (non-return to zero) 2,7 or 1,7 RLL on disk |
| Cables | A cable - Control cable<br>B cable - Data cable |
| Cylinders | 65,535 maximum* |
| Number of heads per drive | 16 maximum* |
| Data transfer rate | 15Mbits/second maximum |
| Access time | 1/2 revolution average;<br>1 revolution maximum* |
| Header data | 512 bytes + 9-15 bytes of ECC |
| Sector size | 5 bytes + 2 bytes of CRC |
| Sectors per track | 33 with 2 spares or 33 with 1 spare at 10Mbits/second data transfer rate* |
| Number of possible consecutive sectors transferable in one operation | 255 maximum |

*Depends on the disk drive used.

## TABLE 1-2. LOTUS 800-DISK DRIVE INTERFACE (2 of 2)

| Item | Parameter |
|---|---|
| Sectoring | Hard |
| Overlap seek execution | Yes |
| Alternate tracking | Yes |
| ECC error detection and correction | Single-/double-burst Reed-Solomon; single is default |

## TABLE 1-3. LOTUS 800-TAPE DRIVE INTERFACE

| Item | Parameter |
|---|---|
| Tape drives supported | QIC-02 interface standard compatible; supports QIC-24, QIC-120, and QIC-150 tape formats |
| Block size | 512 bytes maximum |

## TABLE 1-4. PHYSICAL SPECIFICATIONS

| Item | Parameter |
|---|---|
| Dimensions | Single board 15 inch x 15 inch |
| Cables to disk drive(s) | Control cable A: 34-pin, daisy chained, terminated at the last drive<br>Data cable B: 20-pin, radial, one per drive |
| Cable to tape drive | Signal cable: 50-pin |
| Connectors | Cable connectors for drives are located on the board front<br>CPU interface connector: two 100-pin connectors plug directly into the backplane |

## TABLE 1-5. POWER REQUIREMENTS

| DC Output | Maximum Current | Tolerance |
|---|---|---|
| +5V DC | 8.0 amps | +/- 5% |

## TABLE 1-6. OPERATING ENVIRONMENT

| Item | Parameter |
|---|---|
| Operating temperature | 0° to 50°C<br>32° to 122°F |
| Relative humidity | 0 to 90% non-condensing |

# Section 2

# INTERFACE DEFINITIONS

This section provides information about the internal and external interfaces of the LOTUS 800 Controller. Information on the internal interface consists of an internal block diagram. Information on the external interface is divided into three subsections: central processing unit (CPU), ESDI hard disk drive(s), and tape drive. Each subsection includes a description of the interface as well as signal descriptions and pin assignments.

## 2.1 INTERNAL INTERFACE

A block diagram of the LOTUS 800 internal interface is shown in Figure 2-1.

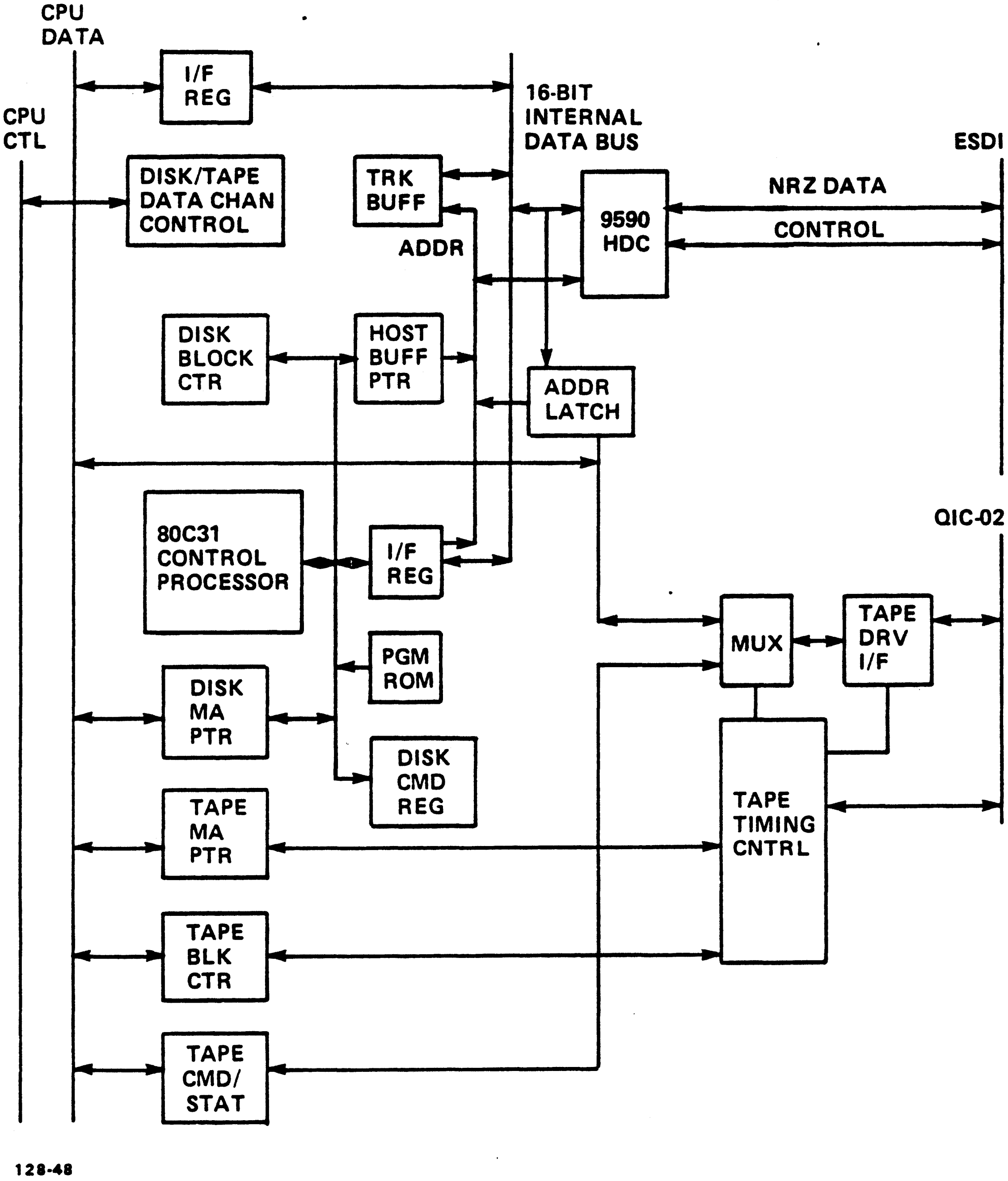


Figure 2-1. LOTUS 800 Internal Block Diagram

## 2.2 CENTRAL PROCESSING UNIT (CPU) INTERFACE

This section provides information about the interfaces of the CPU: the programmed input/output (I/O) interface, data channel interface, the interrupt interface, and I/O bus interface signals. It also provides information and illustrations about I/O timing.

### 2.2.1 Programmed I/O Interface

The CPU uses programmed input/output (PIO) instructions to set up disk and tape registers on the LOTUS 800. The CPU places the disk or tape device code (45 or 42 octal respectively) on the device select lines. For a PIO read instruction, the LOTUS 800 places the contents of the LOTUS 800 register onto the I/O data bus, and the CPU transfers the contents into the appropriate accumulator. For a PIO write instruction, the CPU places the contents of the selected accumulator onto the I/O data bus, and the LOTUS 800 loads the contents into its register. Once this has been accomplished, the LOTUS 800 initiates the selected operation. Figure 2-2 illustrates the data transfer between the CPU and the LOTUS 800.

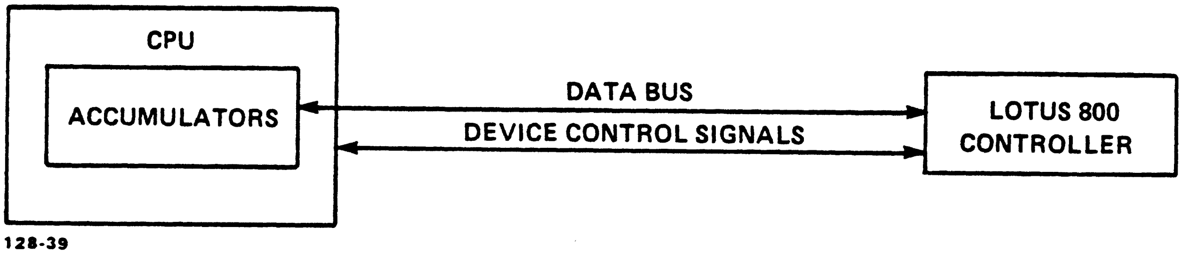


**Figure 2-2. Data Transfer Between Specific Accumulator and LOTUS 800 Device Buffer**

### 2.2.2 Data Channel Interface

To initiate a data channel transfer, the LOTUS 800 issues a data channel request to the CPU. The controller then transfers one or more data words to or from memory (see the data channel read and write timing diagrams, Figures 2-7 and 2-8). At the start of each instruction cycle, the CPU checks whether any device is requesting data channel service. If service is requested, the data channel transfer is performed before going on with the next instruction. Several data channel devices can be active at the same time. Those closest to the CPU have the highest data channel priority.

Transfers are written to or read from memory. The LOTUS 800 Controller enables and prioritizes data channel requests; then it sends them to the CPU. When the CPU is ready to process the request, the controller is signaled to place the direct memory address on the data bus. At the same time, the controller indicates the direction of the transfer. For a write transfer, the controller places the data word on the data bus where it is written to memory. For a read transfer, the CPU fetches the data word from memory and places it on the data bus where it is read from memory.

### 2.2.3 Interrupt Interface

When a device needs service, it sets its interrupt request flag. The CPU begins servicing when all the following conditions exist:

- o The CPU has just completed an instruction or a data channel transfer
- o At least one device has a pending interrupt request
- o Interrupts are enabled
- o No device is waiting for a data channel transfer

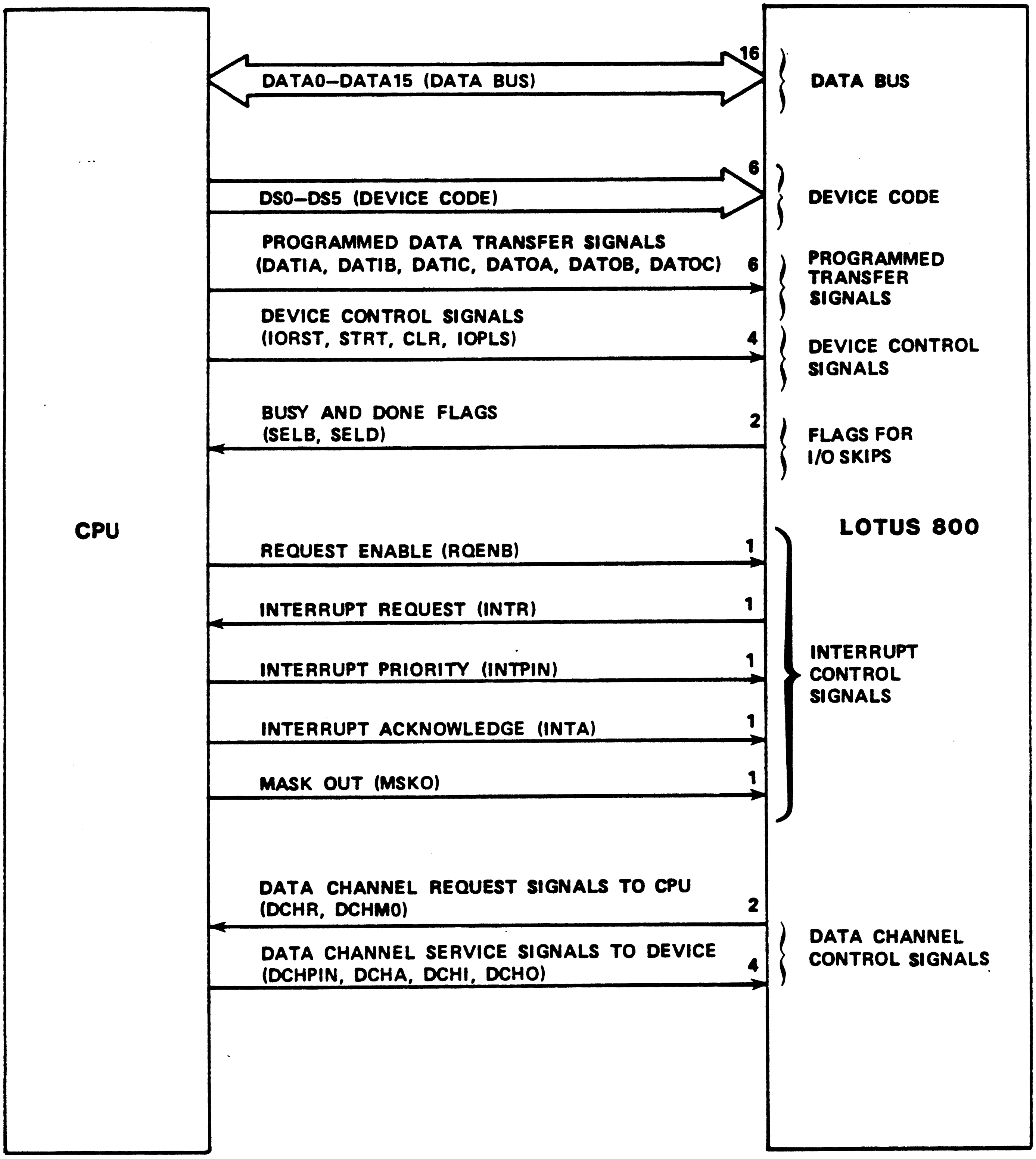
Devices closest to the CPU have the highest interrupt priority.

### 2.2.4 Input/Output Bus Interface Signals

The interface between the LOTUS 800 Controller and the CPU main data bus is defined by Tables 2-1 and 2-2, and Figure 2-3. Table 2-1 shows the signals on the I/O bus grouped by signal classifications. Figure 2-3 illustrates the I/O signals across the I/O bus. Table 2-2 lists the signals by classification group and signal name, indicating direction and describing each signal function.

## TABLE 2-1. INPUT/OUTPUT BUS SIGNAL CLASSIFICATIONS

| Signal Classification | Number of Lines | Description |
|---|---|---|
| Bidirectional data bus | 16 | Used for transfer of all data and address words between the CPU and a peripheral device, for both programmed I/O and data channel transfers. |
| Device code | 6 | Code generated by CPU to specify the address of a peripheral device used for an input/output instruction. |
| Programmed data transfer signals | 6 | These signals, generated by the CPU in response to input/output instructions for data transfers, are used to control data transfers for programmed I/O devices. |
| Device control signals | 4 | These signals are generated by the CPU in response to input/output instructions, and are used to initialize and control I/O devices. The signals affect only the device whose device code is in the instruction, except in the case of the IORST instruction which resets all devices. |
| Skip testing flags | 2 | Flags supplied to the CPU when skip-testing is required. |
| Interrupt control signals | 5 | Signals used to initialize and control the interrupt sequence. |
| Data channel control signals | 6 | Signals used to control data channel transfers between CPU memory and a peripheral device. |


CPU
LOTUS 800
DATA0–DATA15 (DATA BUS)
16
DATA BUS
DS0–DS5 (DEVICE CODE)
6
DEVICE CODE
PROGRAMMED DATA TRANSFER SIGNALS
(DATIA, DATIB, DATIC, DATOA, DATOB, DATOC)
6
PROGRAMMED TRANSFER SIGNALS
DEVICE CONTROL SIGNALS
(IORST, STRT, CLR, IOPLS)
4
DEVICE CONTROL SIGNALS
BUSY AND DONE FLAGS
(SELB, SELD)
2
FLAGS FOR I/O SKIPS
REQUEST ENABLE (RQENB)
1
INTERRUPT REQUEST (INTR)
1
INTERRUPT PRIORITY (INTPIN)
1
INTERRUPT ACKNOWLEDGE (INTA)
1
MASK OUT (MSKO)
1
INTERRUPT CONTROL SIGNALS
DATA CHANNEL REQUEST SIGNALS TO CPU
(DCHR, DCHMO)
2
DATA CHANNEL SERVICE SIGNALS TO DEVICE
(DCHPIN, DCHA, DCHI, DCHO)
4
DATA CHANNEL CONTROL SIGNALS

128-38

Figure 2-3. Input/Output Signal Diagram

### TABLE 2-2. INPUT/OUTPUT SIGNALS (1 of 5)

| Signal Group | Signal Name* | Origin | Description |
|---|---|---|---|
| Data bus | DATA0- to DATA15- | Bidirectional | All data and addresses are supplied to and from the device via these lines. DATA0- is the MSB. |
| Device code | DS0- to DS5- | CPU | The CPU places the device code (bits 10-15 of the instruction word) on these lines during the execution of an input/output instruction. DS0- is the MSB. |
| Programmed data transfer signals | DATIA+ | CPU | Data in A - generated by a DIA instruction. Causes buffer "A" of the selected device to drive the data bus for entry into the accumulator specified by the instruction. |
| | DATOA+ | CPU | Data out A - generated by a DOA instruction. Causes the contents of the accumulator specified in the DOA instruction to drive the data bus for entry into the "A" buffer of the selected device. |
| | DATIB+ | CPU | Data in B - generated by a DIB instruction. Functions like data in A, but uses buffer B. |
| | DATOB+ | CPU | Data out B - generated by a DOB instruction. Functions like data out A, but uses buffer B. |
| | DATIC+ | CPU | Data in C - generated by a DIC instruction. Functions like data in A, but uses buffer C. |
| | DATOC+ | CPU | Data out C - generated by a DOC instruction. Functions like data out A, but uses buffer C. |

*Signal names ending with + are active high; those ending with - are active low.

**TABLE 2-2. INPUT/OUTPUT SIGNALS (2 of 5)**

| Signal Group | Signal Name | Origin | Description |
|---|---|---|---|
| Device control signals | IORST+ | CPU | Input/output reset - generated when APL is pressed on the Mini-Panel, when RESET is pressed on the Operator Control Unit, when an IORST instruction is being executed, or during power turn-on. |
| | STRT+ | CPU | Start - generated when the CTRL field of an input/output transfer instruction contains code 01. It clears the Done flag and Interrupt request, and sets the Busy flag in the selected device. |
| | CLR+ | CPU | Clear - generated when the CTRL field of an input/output transfer instruction contains code 10. It usually clears the Busy and Done flags and the Interrupt request in the device whose device code is on the lines. |
| | IOPLS+ | CPU | I/O pulse - generated when the CTRL field of an input/output transfer instruction contains code 11. The effect, if any, depends on the device. |
| Skip testing flags | SELB- | Disk drive | Selected device busy - supplied to CPU by the selected device when its internal Busy flag is set. Indicates that the device is busy. |
| | SELD- | Disk drive | Selected device done - supplied to the CPU by the selected device when its internal Done flag is set. Indicates that the device is done. |

## TABLE 2-2. INPUT/OUTPUT SIGNALS (3 of 5)

| Signal Group | Signal Name | Origin | Description |
|---|---|---|---|
| Interrupt control signal | RQENB- | CPU | Request enable - generated during each memory read/write cycle to synchronize INTR- and DCHR-. In any device, changes in INTR- or DCHR- may only occur following the leading edge (high-to-low transition) of RQENB-. |
| | INTR- | Disk drive | Interrupt request - this signal goes low (following the leading edge of RQENB-) if the device requires CPU service. |
| | INTPIN- | From jumper saver | Interrupt priority input - on POINT 4 chassis, this signal is produced by jumper-saver logic on the backplane for highest priority device requesting an interrupt. |
| | | Jumpered to ground or from higher priority device | On non-POINT 4 chassis, this signal is produced by a jumper to ground on the board closest to the CPU (highest in priority). If the highest priority I/O board in the non-POINT 4 chassis has not requested an interrupt, this signal is output as INTPOUT- to the device next in priority. |
| | INTA+ | CPU | Interrupt acknowledge - generated by an INTA instruction. Causes the device whose INTPIN- line is low to place its device code in bits 10-15 of the data bus for entry into accumulator specified in the instruction. |
| | MSKO- | CPU | Mask out - generated by a MSKO instruction. Commands all I/O devices to set their interrupt disable flags according to the state of the associated mask bit in the word on the data bus. |

**TABLE 2-2. INPUT/OUTPUT SIGNALS (4 of 5)**

| Signal Group | Signal Name* | Origin | Description |
|---|---|---|---|
| Data channel control signals | DCHR- | Disk drive | Data channel request - this signal goes low (following the leading edge of RQENB-) if the device wants to request a data channel transfer. |
| | DCHMO- | Disk drive | Data channel mode - generated by a device connected to the data channel while DCHA- is low. Indicates the type of data channel cycle being requested as follows:<br>DCHMO- Type of Cycle<br>high Data out (from CPU)<br>low Data in (to CPU) |
| | DCHPIN- | From jumper saver | Data channel priority input - on POINT 4 chassis, this signal is produced by jumper-saver logic on the backplane for highest priority device requesting a data channel transfer. |
| | | Jumpered to ground or from higher priority device | On non-POINT 4 chassis, this signal is produced by a jumper to ground the board closest to the CPU (highest in priority). If the highest priority I/O board in the non-POINT 4 chassis has not requested a data channel transfer, this signal is output as DCHPOUT- to the device next in priority. |
| | DCHA- | CPU | Data channel acknowledge - generated by CPU in response to a data channel request. Initiates a data channel cycle in the device whose DCHPIN- is low. The devices places the memory address for data channel access on the data bus. |

TABLE 2-2. INPUT/OUTPUT SIGNALS (5 of 5)

| Signal Group | Signal Name | Origin | Description |
|---|---|---|---|
| | DCHI+ | CPU | Data· channel in - DCHI+ is generated whenever the device is placing a data word on the data bus. |
| | DCHO+ | CPU | Data channel out - DCHO+ is generated whenever the word accessed from memory is on the data bus. |

### 2.2.5 Backplane Pin Signal Connectors

All signal connections between the LOTUS 800 Controller and the CPU take place via two 100-pin connectors. Figure 2-4 shows the connector pin layout for all backplane I/O signals (not all are used by the LOTUS 800). The labeled pins refer to the I/O control signals, data transfer signals, and the DC voltages used by the LOTUS 800.

| BOTTOM | A | | TOP |
|---|---|---|---|
| GND | 2 | 1 | GND |
| +5V | 4 | 3 | +5V |
| –5V | 6 | 5 | +5BU |
| | 8 | 7 | PWRGON– |
| +15V | 10 | 9 | –5BU |
| | 12 | 11 | PWRF |
| | 14 | 13 | |
| | 16 | 15 | |
| | 18 | 17 | |
| | 20 | 19 | |
| | 22 | 21 | |
| | 24 | 23 | |
| | 26 | 25 | |
| | 28 | 27 | |
| | 30 | 29 | |
| | 32 | 31 | |
| GND | 34 | 33 | GND |
| | 36 | 35 | |
| MSKO– | 38 | 37 | |
| INTA+ | 40 | 39 | |
| DATIB+ | 42 | 41 | |
| DATIA+ | 44 | 43 | |
| DS3– | 46 | 45 | |
| DATOC+ | 48 | 47 | |
| CLR+ | 50 | 49 | |
| STRT+ | 52 | 51 | |
| DATIC+ | 54 | 53 | |
| DATOB+ | 56 | 55 | |
| DATOA+ | 58 | 57 | |
| DCHA– | 60 | 59 | |
| DS4– | 62 | 61 | |
| DS5– | 64 | 63 | |
| DS2– | 66 | 65 | |
| DS1– | 68 | 67 | |
| IORST+ | 70 | 69 | |
| DSO– | 72 | 71 | |
| IOPLS+ | 74 | 73 | |
| | 76 | 75 | |
| | 78 | 77 | |
| SELD– | 80 | 79 | |
| SELB– | 82 | 81 | |
| | 84 | 83 | |
| | 86 | 85 | |
| | 88 | 87 | |
| | 90 | 89 | |
| | 92 | 91 | |
| DCHPIN– | 94 | 93 | DCHPOUT– |
| INTPIN– | 96 | 95 | INTPOUT– |
| +5V | 98 | 97 | +5V |
| GND | 100 | 99 | GND |

| BOTTOM | B | | TOP |
|---|---|---|---|
| GND | 2 | 1 | GND |
| +5V | 4 | 3 | +5V |
| | 6 | 5 | |
| | 8 | 7 | |
| | 10 | 9 | |
| | 12 | 11 | |
| | 14 | 13 | |
| | 16 | 15 | |
| | 18 | 17 | DCHMO– |
| | 20 | 19 | |
| | 22 | 21 | |
| | 24 | 23 | |
| | 26 | 25 | |
| | 28 | 27 | |
| | 30 | 29 | INTR– |
| | 32 | 31 | |
| | 34 | 33 | DCHO+ |
| | 36 | 35 | DCHR– |
| | 38 | 37 | DCHI+ |
| | 40 | 39 | |
| | 42 | 41 | RQENB– |
| | 44 | 43 | |
| +15V | 46 | 45 | |
| | 48 | 47 | |
| GND | 50 | 49 | |
| | 52 | 51 | |
| | 54 | 53 | |
| DATA14– | 56 | 55 | DATA7– |
| DATA11– | 58 | 57 | DATA5– |
| DATA8– | 60 | 59 | DATA12– |
| DATA0– | 62 | 61 | DATA4– |
| DATA13– | 64 | 63 | DATA9– |
| DATA15– | 66 | 65 | DATA1– |
| | 68 | 67 | |
| | 70 | 69 | |
| | 72 | 71 | |
| | 74 | 73 | DATA3– |
| | 76 | 75 | DATA10– |
| | 78 | 77 | |
| | 80 | 79 | |
| DATA2– | 82 | 81 | –5V |
| +15V | 84 | 83 | |
| | 86 | 85 | |
| | 88 | 87 | |
| | 90 | 89 | |
| GND | 92 | 91 | –15V |
| +12BU | 94 | 93 | –15V |
| | 96 | 95 | DATA6– |
| +5V | 98 | 97 | +15V |
| GND | 100 | 99 | GND |

128-37

**Figure 2-4. Backplane I/O Signals**

### 2.2.6 Input/Output Timing

Three types of operation take place on the I/O bus: those associated with programmed I/O instructions, interrupt handling, and data channel transfers. In the timing diagrams in this section, each signal or group of signals is represented by a horizontal line with a raised section to show the active state. Control signals, which are generated at a specific time to control a particular function, show a raised line to indicate the time that the signal is active. For signals carrying binary information, the raised line indicates the amount of time that information remains on the bus. Raised lines may represent either high or low voltage levels, depending on whether the signal is active when low or high. Times on the diagrams are given in nanoseconds. Figures 2-5 and 2-6 show programmed I/O instruction timing, Figures 2-7 and 2-8 show data channel timing, and Figure 2-9 shows interrupt timing.

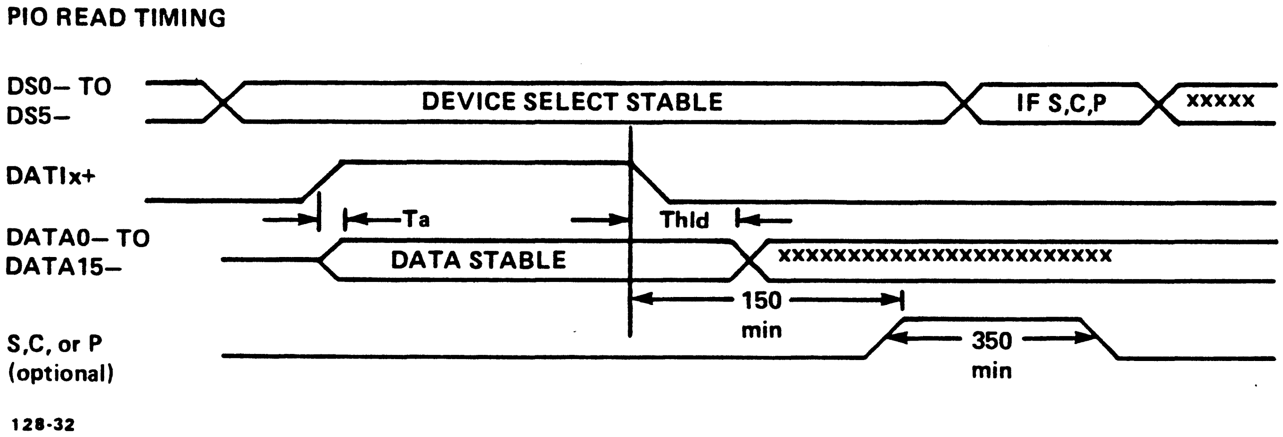


**Figure 2-5. PIO Read Timing Diagram**

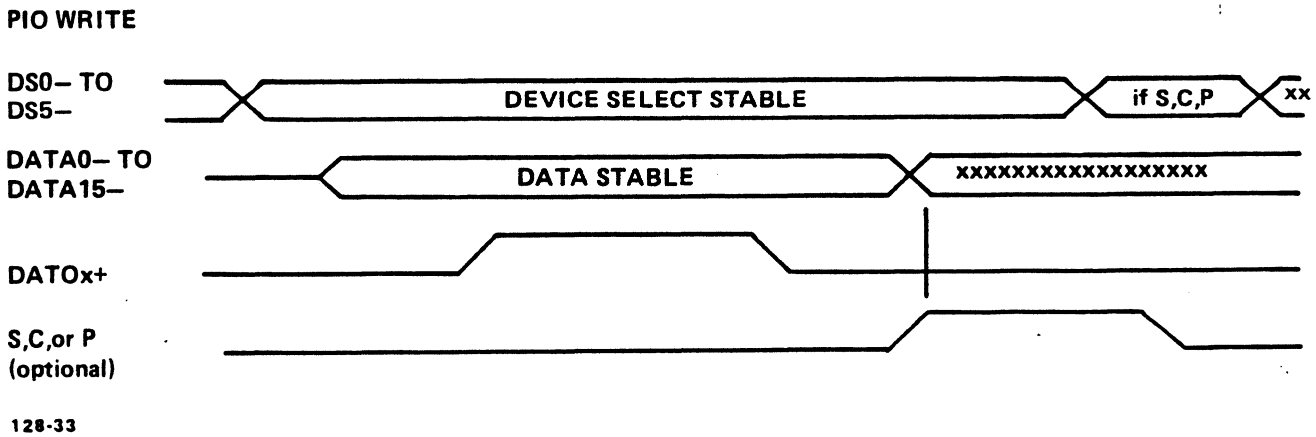


Figure 2-6. PIO Write Timing Diagram

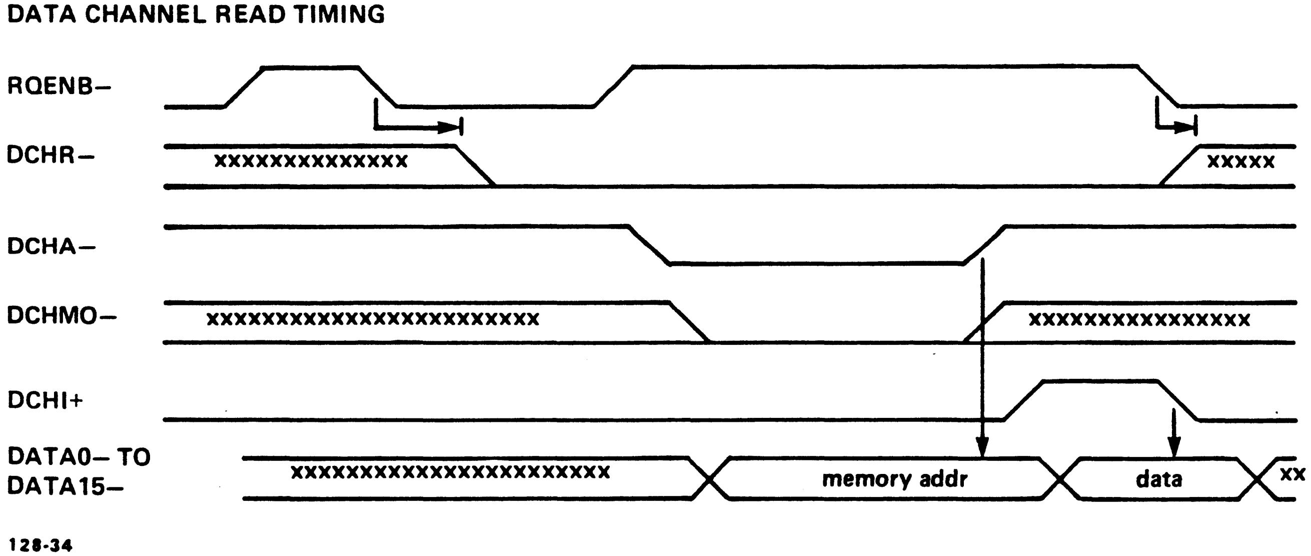


**Figure 2-7. Data Channel Read Timing Diagram**

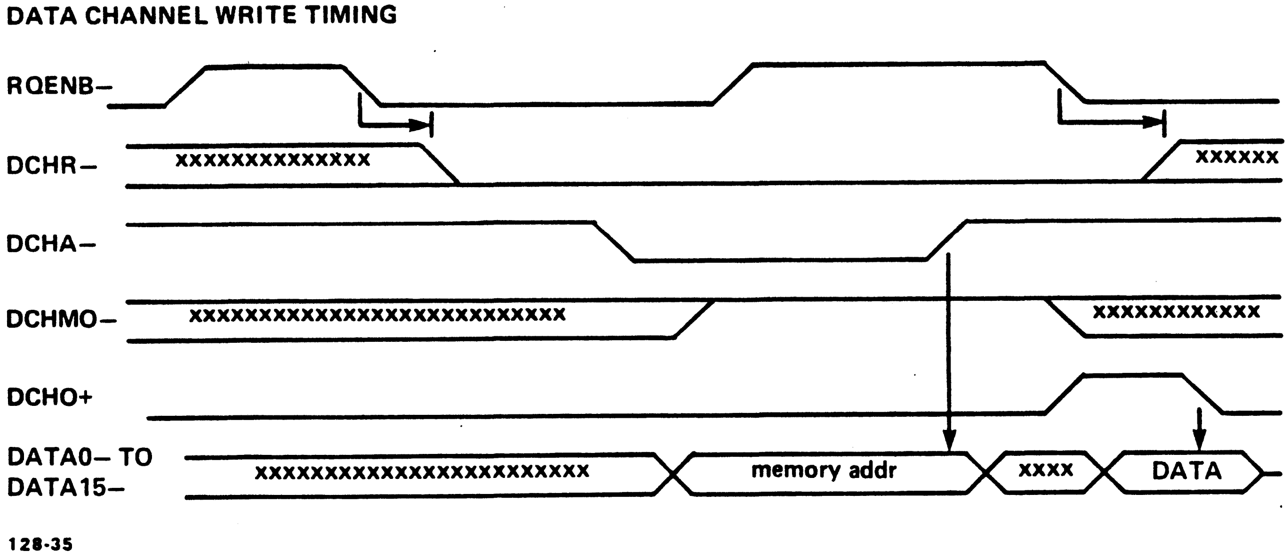


**Figure 2-8. Data Channel Write Timing Diagram**

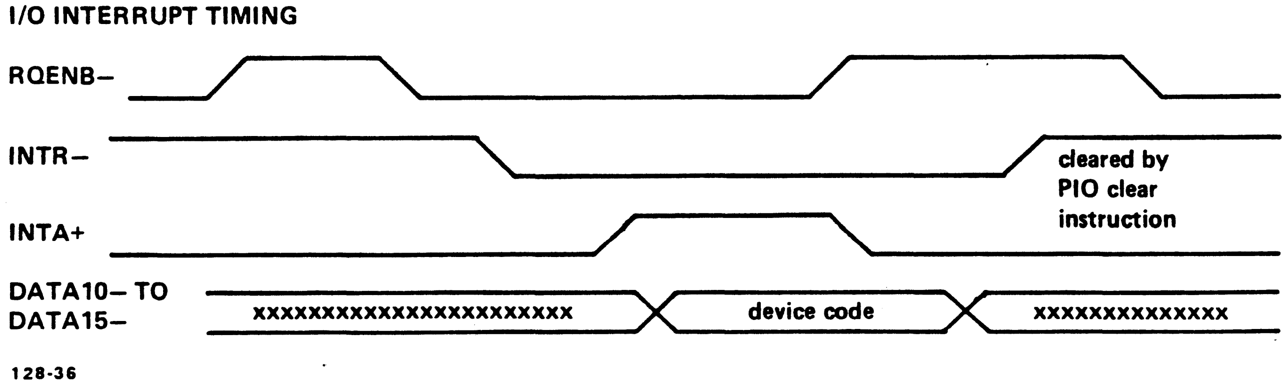


Figure 2-9. Interrupt Timing Diagram

## 2.3 DISK DRIVE INTERFACE

The LOTUS 800 disk interface is designed in accordance with the ESDI interface specification. The interface consists of a control cable and a data cable. The control cable can be daisy-chained to a maximum of four ESDI drives with only the last physical disk drive being terminated. Figures 2-10 and 2-11 illustrate the signals and the pin connectors respectively; Table 2-3 provides descriptions of the LOTUS 800-disk drive interface signals. Section 2.3.2 provides a description of the initialization procedures.

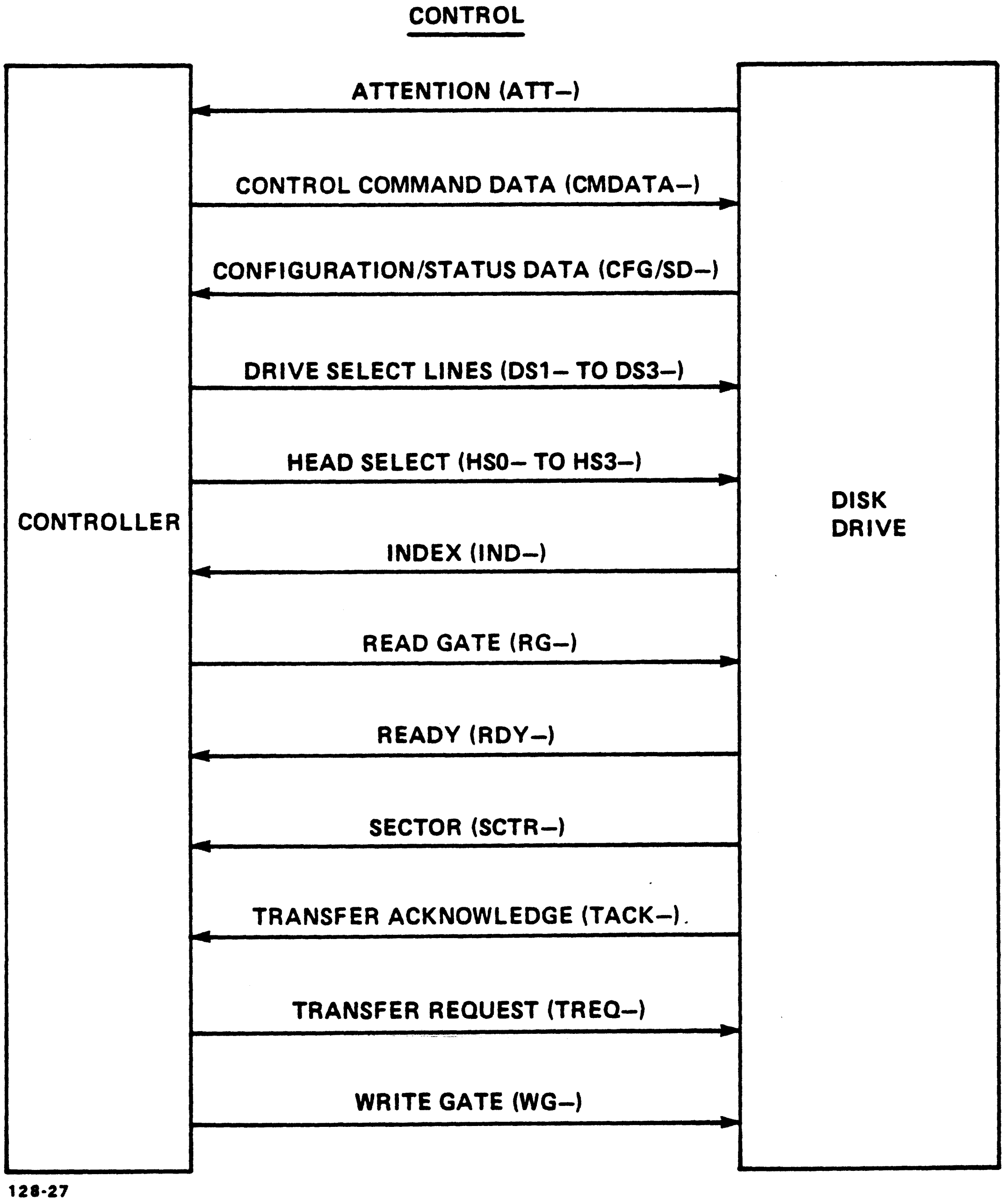

CONTROL
ATTENTION (ATT–)
CONTROL COMMAND DATA (CMDATA–)
CONFIGURATION/STATUS DATA (CFG/SD–)
DRIVE SELECT LINES (DS1– TO DS3–)
HEAD SELECT (HS0– TO HS3–)
CONTROLLER
DISK DRIVE
INDEX (IND–)
READ GATE (RG–)
READY (RDY–)
SECTOR (SCTR–)
TRANSFER ACKNOWLEDGE (TACK–)
TRANSFER REQUEST (TREQ–)
WRITE GATE (WG–)
128-27

**Figure 2-10. LOTUS 800-Disk Drive Interface Signals (1 of 2)**

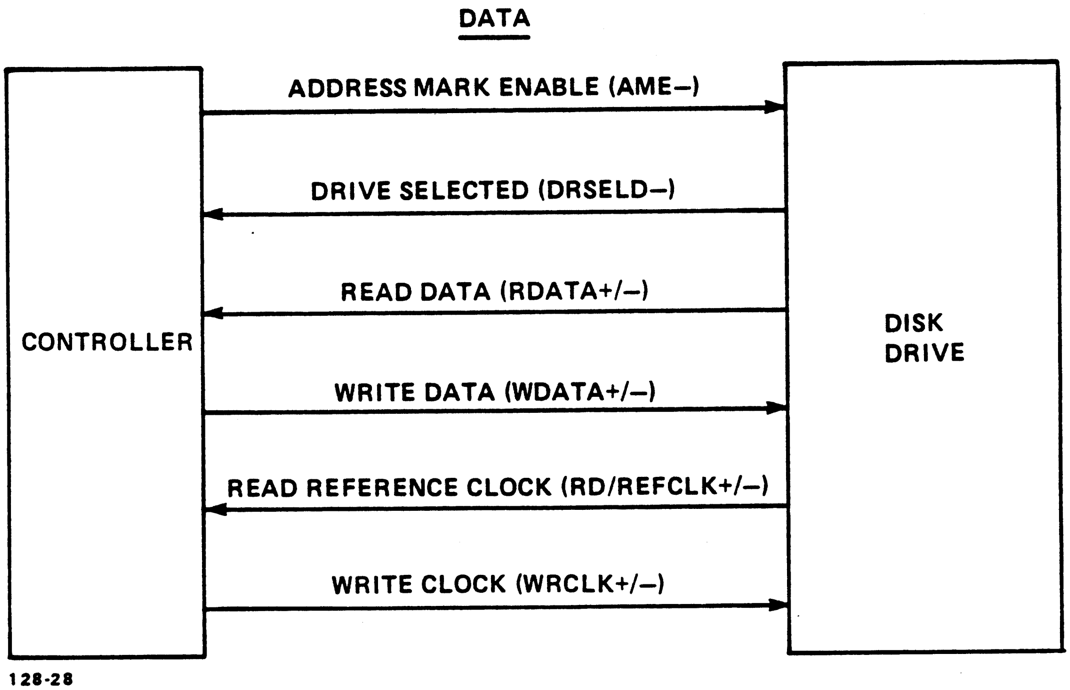


**Figure 2-10. LOTUS 800-Disk Drive Interface Signals (2 of 2)**

## TABLE 2-3. LOTUS 800-DISK DRIVE INTERFACE SIGNAL DESCRIPTIONS (1 of 3)

| Signal Name* | Origin | Description |
|---|---|---|
| **ESDI Control** | | |
| ATT- | Disk drive | Attention - Inhibits writing to disk. Alerts hard disk controller to read standard status. Usually results from a fault or status change. |
| CMDATA- | Ctrlr | Command Data - Upon receipt of serial command, drive performs function specified by the 17-bit configuration (16 command plus 1 odd parity bit). Data transfers MSB first. CMDCMPLT should be activated before CMDATA is transferred. |
| CFG/SD- | Disk drive | Configuration/Status Data - In response to hard disk controller command, 17 bits of serial data (16 data plus 1 odd parity bit) transmits MSB first. CMDCMPLT deactivates during command phase through data portion. If a command e.g., SEEK requires no response CMDCMPLT terminates after command phase. |
| DS1- to DS3- | Ctrlr | Drive select lines 0 to 3. Encodes drives select lines as follows: |

| Drive Selected | DS3- | DS2- | DS1- |
|---|---|---|---|
| none | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 |

| Signal Name* | Origin | Description |
|---|---|---|
| HS0- to HS3 | Ctrlr | Head Select 0 top 3. Binary encoded lines select 1 of 16 drive heads. All lines high = head 0. If more heads are addressed than drives contain, ATT and ILLEGAL HEAD ADDRESS activate. Subsequent writing results in write fault. |

*Signal names ending with + are active high; those ending with - are active low.

## TABLE 2-3. LOTUS 800-DISK DRIVE INTERFACE SIGNAL DESCRIPTIONS (2 of 3)

| Signal Name | Origin | Description |
|---|---|---|
| **ESDI Control** | | |
| IND- | Disk drive | Index - The leading edge occurs once/ 16.67 millisecond to indicate the beginning of a track (physical sector 1 for the selected drive). |
| RG- | Ctrlr | Read Gate - Enables data to be read from disk. Should be activated only during phase lock oscillator sync field and prior to ID or data sync bytes. Must be deactivated when passing over a write splice area. |
| RDY- | Disk drive | Ready - Indicates disk spindle is up to speed. When activated with CMDCMPLT, drive is ready to read, write, or seek. Otherwise, all writing/seeking is inhibited. |
| SCTR- | Disk drive | Sector - The leading edge indicates start of a hard sector but not physical sector 1 (IND- indicates physical sector 1). |
| TACK- | Disk drive | Transfer Acknowledge - during CFG/SD- transfers in response to TREQ- activating/deactivating for each command, parity, status, or configuration bit transferred. |
| TREQ- | Ctrlr | Transfer Request - Issued during CFG/SD- transfers in response to TACK- activating/deactivating before proceeding to transfer the next bit. |
| WG- | Ctrlr | Write Gate - Enables write data to be written on the disk. |

**TABLE 2-3. LOTUS 800-DISK DRIVE INTERFACE SIGNAL DESCRIPTIONS (3 of 3)**

| Signal Name* | Origin | Description |
|---|---|---|
| **ESDI Data** | | |
| AME- | Ctrlr | Address Mark Enable - When WG- is active, enables writing of address mark. IF WG- or RG- is not active, AME- triggers search for address mark. |
| DRSELD- | Disk drive | Drive Selected - Indicates drive selection by DS0- through DS3-. |
| RDATA+/- | Disk drive | Read Data - Differential disk read data. This data is clocked by READ CLOCK. NRZ (non return to zero) encoded. |
| WDATA+/- | Ctrlr | Write Data - Differential disk write. This data is clocked by WRCLK. NRZ (non return to zero) encoded. |
| RD/ REFCLK+/- | Disk drive | Read Reference Clock - REFCLK is valid if RG- is inactive; RD is valid if RG- is active and PLO sync is established. |
| WRCLK+/- | Ctrlr | Write Clock - Generated by hard disk controller and is same frequency as RD/REFLCK. |

### 2.3.1 Connector Pin Assignments

Signal connections between the LOTUS 800 Controller and the disk drive take place via a 34-pin control cable connector and a 20-pin data cable connector. Figure 2-11 shows the pin layout for the disk I/O signals.

| SIGNAL GROUND | SIGNAL PIN | I/O | SIGNAL NAME |
|---|---|---|---|
| CONTROL | | | |
| 1 | 2 | O | HS3– |
| 3 | 4 | O | HS2– |
| 5 | 6 | O | WG– |
| 7 | 8 | I | CFG/SD– |
| 9 | 10 | I | TACK– |
| 11 | 12 | I | ATT– |
| 13 | 14 | O | HS0– |
| 15 | 16 | I | SCTR– |
| 17 | 18 | O | HS1– |
| 19 | 20 | I | IND– |
| 21 | 22 | I | RDY– |
| 23 | 24 | O | TREQ– |
| 25 | 26 | O | DS1– |
| 27 | 28 | O | DS2– |
| 29 | 30 | O | DS3– |
| 31 | 32 | O | RG– |
| 33 | 34 | O | CMDATA– |
| DATA | | | |
| – | 1 | I | DRSELD– |
| – | 2 | I | SCTR– |
| – | 3 | I | CMDCMPLT– |
| – | 4 | O | AME– |
| 6 | 5 | – | RESERVED |
| – | 7 | O | WRCLK+ |
| – | 8 | O | WRCLK– |
| – | 9 | – | |
| – | 10 | I | RD/REFCLK+ |
| 12 | 11 | I | RD/REFCLK– |
| – | 13 | O | WDATA+ |
| 15 | 14 | O | WDATA– |
| 16 | 17 | I | RDATA+ |
| 19 | 18 | I | RDATA– |
| – | 20 | I | IND– |

128-29

**Figure 2-11. Disk I/O Signals**

### 2.3.2 Initialization Procedures

On power-up, buffer and hard disk controller (HDC) self-test diagnostics are initiated, and the LOTUS 800 LED lights. If there are no self-test errors, the LOTUS issues four dummy Load Drive Parameter Block Command Descriptor Blocks (CDBs) to initialize the HDC. Once the dummy CDBs are executed, LOTUS 800 firmware spins up drives 0, 1, 2, and 3.* Any non-existent drive causes an HDC timeout, and a corresponding Drive Present Map is established in firmware. This map, along with the ESDI channel CDB, is used by the firmware to determine the correct drive parameter for each existing drive. The Drive Parameter Block for each existing drive is then updated. If no further errors are encountered, the LOTUS 800 LED goes out.

If a self-test failure occurs, further self-tests are aborted and the LED remains on until further CPU intervention.

*Drive spin-ups and self-test diagnostics occur only during power-up or by pressing the LOTUS 800 reset button.

## 2.4 TAPE DRIVE INTERFACE

The LOTUS 800 supports the QIC-02 interface and the QIC-24, QIC-120, and QIC-150 tape formats. Data and commands are transferred to and from the tape drive on an eight-bit bidirectional data bus. One tape drive is supported by the LOTUS 800. The interface is a 50-pin flat-ribbon signal cable. Figure 2-12 illustrates the signals and Table 2-4 provides descriptions of the LOTUS 800-tape drive interface. Section 2.4.2 provides a description of the initialization procedures.

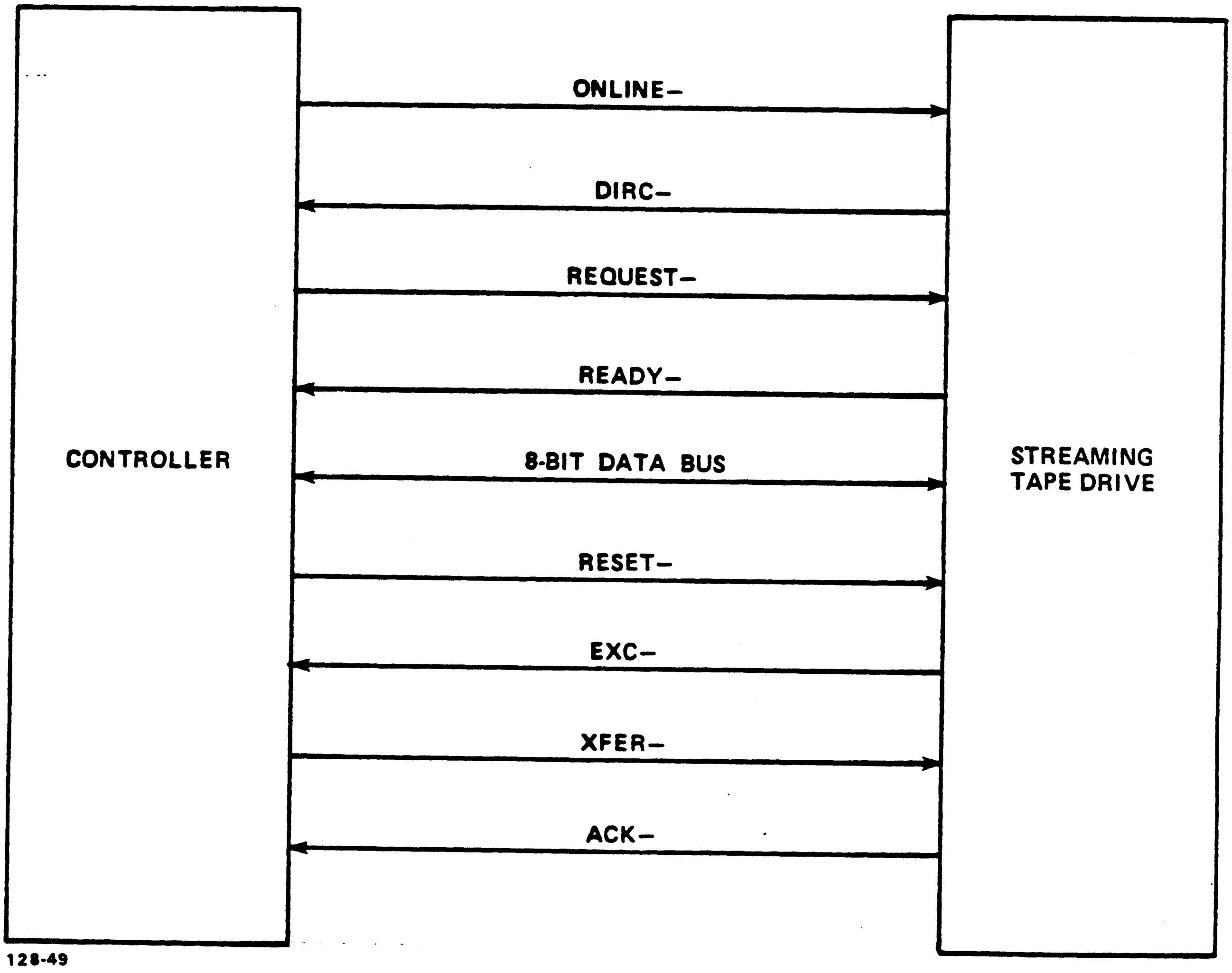


**Figure 2-12. LOTUS 800-Tape Drive Interface Signals**

## TABLE 2-4. LOTUS 800-TAPE DRIVE INTERFACE SIGNAL DESCRIPTIONS (1 of 2)

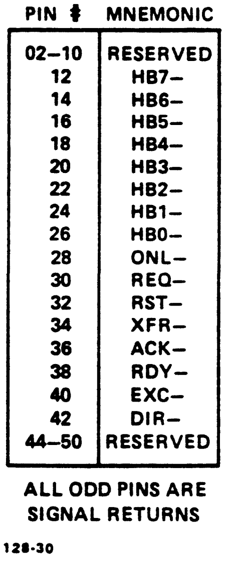
| Signal Name | Origin | Description |
|---|---|---|
| ONLINE- | Ctrlr | Activates when drive writes, reads, or searches. Deactivating terminates a write/read operation and rewinds tape to the beginning of the tape. |
| DIRC- | Tape drive | Indicates data direction on the drive data bus. When activated, transfers are from drive to controller. |
| REQUEST- | Ctrlr | Indicates that command data has been placed on the data bus or that status data has been taken from the data bus. Activated only when READY- or EXC- are activated. |
| READY- | Tape drive | Indicates one of the following:<br>o Drive is available to receive and execute a command<br>o A new block is ready for transfer during a read/write operation<br>o Drive is ready to receive a new block during a write operation<br>o Drive is ready to transfer status information to controller when requested |
| HB0- thru HB7- | Ctrlr or tape drive | 8-bit bidirectional data bus. HB0- is the least significant bit; HB7- is most significant. |
| RESET- | Ctrlr | Initializes tape drive. Drive select defaults to 0. |

## TABLE 2-4. LOTUS 800-TAPE DRIVE INTERFACE SIGNAL DESCRIPTIONS (2 of 2)

| Signal Name | Origin | Description |
|---|---|---|
| EXC- | Tape drive | Indicates that controller should issue a command to read the 6-byte standard status. After reset, EXC- is activiated by tape drive 0. |
| XFR- | Ctrlr | Indicates that data has been placed on or taken from the data bus. Used in conjunction with ACK- to move data between the CPU and the tape drive. |
| ACK- | Tape drive | Indicates that data has been taken from or placed on data bus. Used in conjunction with XFR- to move data between the CPU and tape drive. |

### 2.4.1 Connector Pin Assignments

Signal connections between the LOTUS 800 Controller and the tape drive take place via a 50-pin cable connector. Figure 2-13 shows the pin layout for the tape I/O signals.



| PIN # | MNEMONIC |
|---|---|
| 02–10 | RESERVED |
| 12 | HB7– |
| 14 | HB6– |
| 16 | HB5– |
| 18 | HB4– |
| 20 | HB3– |
| 22 | HB2– |
| 24 | HB1– |
| 26 | HB0– |
| 28 | ONL– |
| 30 | REQ– |
| 32 | RST– |
| 34 | XFR– |
| 36 | ACK– |
| 38 | RDY– |
| 40 | EXC– |
| 42 | DIR– |
| 44–50 | RESERVED |

ALL ODD PINS ARE SIGNAL RETURNS

128-30

**Figure 2-13. Tape I/O Signals**

### 2.4.2 Initialization Procedures

A power-up reset or an IORST instruction issued by the CPU causes the tape drive to initialize and default to drive 0. Following an EXC-, the CPU issues the Read Status command to the tape controller. A programmed input/output instruction then causes the LOTUS 800 to transfer block 0 from tape and send it to memory address 0 through 377 (octal). This initiates the tape boot program.

# Section 3

# PROGRAMMING INSTRUCTIONS

Programmed input/output (PIO) instructions enable the central processing unit (CPU) to communicate directly with the LOTUS 800 Controller. This section contains the following information about programmed input/output:

o PIO instruction word format and bit descriptions for disk and tape

o PIO interface instructions for disk and tape operations

In addition to the programmed input/output (PIO) instructions, all disk operations are performed using Command Descriptor Blocks (CDBs). This section contains the following information about Command Descriptor Blocks:

o CDB format and bit descriptions

o Summary and illustration of each CDB

## 3.1 PROGRAMMED INPUT/OUTPUT INSTRUCTION WORD FORMAT AND BIT DESCRIPTIONS

Although the instructions for disk and tape operations differ, the programmed input/output (PIO) instruction word format is the same for both devices. The word format is described below; specific instructions for disk and tape operations are given in Sections 3.2 and 3.3, respectively.

An input/output instruction is designated by 011 (bits 0-2). The operation code (bits 5-7) and control (bits 8 and 9) fields define the input/output operation to be performed. When a register data transfer is involved, the accumulator field (bits 3 and 4) specifies the accumulator involved in the data transfer. Bits 10-15 select the device to respond to the instruction. Figure 3-1 illustrates the input/output instruction word format.

In order to program register data transfers, the programmer must properly code the central processing unit (CPU) instruction and also load the specified CPU accumulator (if required) with the information to be transferred to the LOTUS 800 controllers. Data transfers to the controllers require both instruction coding and loading of an accumulator with information for the controllers. Data transfers to the CPU require only instruction coding since information is read from the disk or tape controller into the specified accumulator.

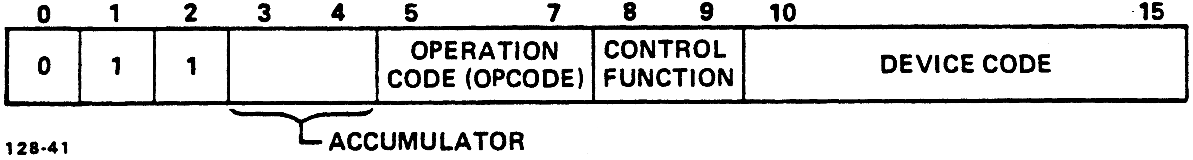


Figure 3-1. Input/Output Instruction Word Format

The programmed input/output (PIO) instructions that support the disk and tape operations are as follows:

| Disk Operations | Tape Operations |
|---|---|
| DOA | DOA |
| DOAS | DOB |
| DIA | DOC |
| NIOC | DIA |
| NIOS | DIB |
| IORST | DIC |
| | IORST |

Instructions are coded in the following format:

**MNEMONIC** [optional mnemonics] **OPERAND STRING**

The mnemonic shown in bold print must be coded exactly as shown in the instruction description. For example, the mnemonic for the Data Out to Buffer A instructions is:

**DOA**

Operands shown in bold print must also be coded exactly as shown.

Some instructions have optional mnemonics that are appended to the main mnemonic if the option is desired. These mnemonics are enclosed in brackets [ ]. Optional mnemonics may require substitution of a specific control character in order to be properly decoded. Instructions for controlling data transfer between the processor and the disk or tape controller use two optional fields.

ac = accumulator

f = control function

The accumulator field (ac) designates one of four accumulators (0, 1, 2, or 3). The accumulator sends information to the controller or receives information from the controller.

The control function field controls the disk controller's busy and done flags as follows:

f = S    Function field = start. Sets the busy flag to one; clears the done flag to zero; clears out all controller error flags; starts read/write timeout.

f = C    Function field = clear. Clears disk interrupts.

The control function controls the tape controller's busy and done flags as follows:

f = S Function field = start. Sets the busy flag, clears the done flag. Initiates request for command transfer to streaming tape drive.

f = C Function field = clear. Clears the busy, done, and interrupt flags; stops data transfer.

f = P Function field = pulse. Clears the interrupt flag and requests the next byte from the streaming tape drive.

## 3.2 DISK INSTRUCTIONS

The following subsections describe the programmed input/output instructions used in performing disk operations.

### 3.2.1 Write Memory Address Register

Instruction Mnemonic: **DOA ac,45**

Instruction Function: Writes the contents of the specified accumulator into the controller's Memory Address Register. Figure 3-2 illustrates the contents of the register.

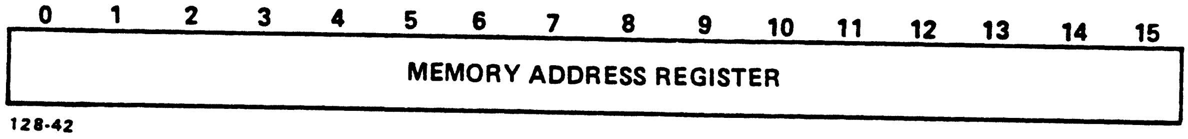


**Figure 3-2. Write Memory Address Register**

### 3.2.2 Read Memory Address Register

Instruction Mnemonic: **DIA ac,45**

Instruction Function: Reads the Memory Address Register from the controller. Figure 3-3 illustrates the contents of the register.

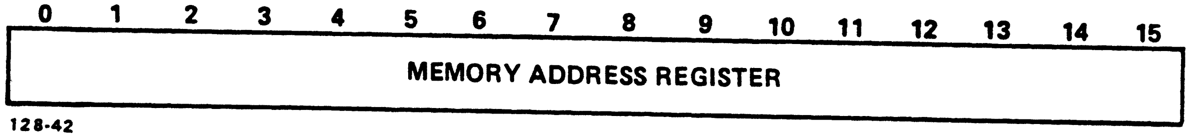


**Figure 3-3. Read Memory Address Register**

### 3.2.3 Write a CDB Starting Address to the Memory Address Register

Instruction Mnemonic: **DOAS ac,45**

Instruction Function: Writes a starting address to the Memory Address Register. This address points to the first Command Descriptor Block (CDB) in memory and initiates a disk operation. Section 3.4 provides a summary for each CDB. Figure 3-4 illustrates the contents of the CDB starting address for the next data channel transfer.

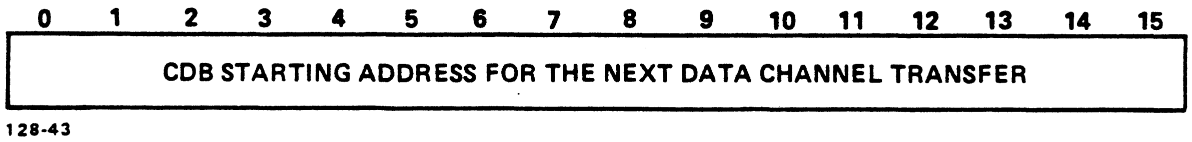


**Figure 3-4. Write a Starting Address to the Memory Address Register**

### 3.2.4 Initiate Initial Program Load

Instruction Mnemonic: **NIOS 45**

Instruction Function: Initiates an Initial Program Load (IPL) that causes the controller to read Cylinder 0, Track 0, and Sector 0 from Drive 0; and transfers data to main memory locations 0 through 377 (octal).

### 3.2.5 Clear Disk Interrupt

Instruction Mnemonic: **NIOC 45**

Instruction Function: Clears disk interrupt.

### 3.2.6 Input/Output Reset (Disk)

Instruction Mnemonic: **IORST**

Instruction Function: Sets the busy flag and clears the done flag, terminates the current disk operation, resets the hard disk controller (HDC), reinitializes all HDC internal registers, waits for a programmed input/output instruction.

## 3.3 TAPE INSTRUCTIONS

The following subsections describe the programmed input/output instructions used in performing tape operations.

### 3.3.1 Specify Command

Instruction Mnemonic: **DOA[f] ac,42**

Instruction Function: Loads the contents of the specified accumulator into the tape controller's Command Register in the input/output interface logic. Bits 0 and 1 of the Command Register are connected to the tape drive through dedicated lines. Figure 3-5 illustrates the accumulator format for the tape controller Specify Command instruction. Table 3-1 defines bit functions.

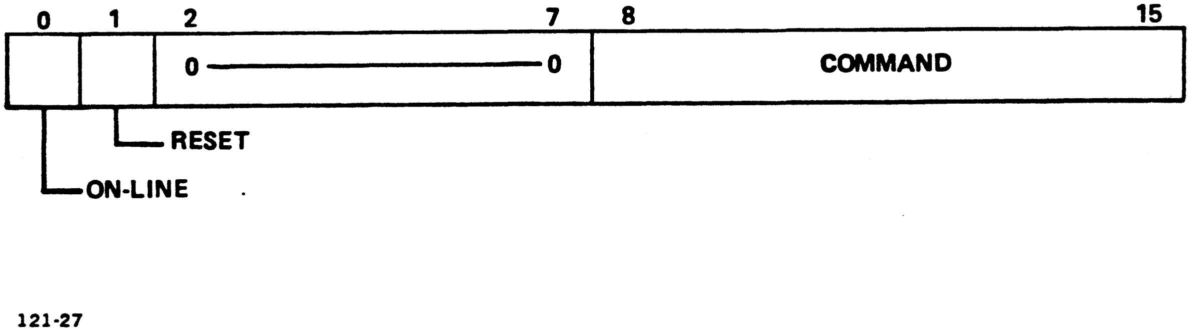


**Figure 3-5. Specify Command Instruction**

## TABLE 3-1. SPECIFY COMMAND BIT FUNCTIONS

| Bits | Function |
|---|---|
| 0 | On-line - Must be set to one for read, read-file-mark, write, and write-file-mark operations. |
| 1 | Reset - Issues master reset to the streaming tape drive. Must be set for 13 microseconds minimum. |
| 2-7 | Unused. |
| 8-15 | Command: |

| | |
|---|---|
| 00000001 | Select drive 0 |
| 00000010 | Select drive 1 |
| 00000100 | Select drive 2 |
| 00001000 | Select drive 3 |
| 00010000 | Reserved |
| 00100001 | Rewind to beginning of tape |
| 00100010 | Erase tape |
| 00100100 | Retension tape |
| 00100110 | Select Archive mode |
| 00101000 | Reserved |
| 00110000 | Reserved |
| 01001001 | Select read-Cipher-quarterback mode |
| 01000000 | Write data |
| 01100000 | Write file mark |
| 10000000 | Read data |
| 10100000 | Read file mark |
| 11000000 | Read status |
| 111xxxxx | Reserved |

All other commands are illegal.

### 3.3.2 Specify Memory Address Register

**Instruction Mnemonic: DOB[f] ac,42**

Instruction Function: Loads the contents of the specified accumulator into the controller's Memory Address Register. Figure 3-6 illustrates the accumulator format for the Memory Address Register instruction. Table 3-2 defines the accumulator bit functions.

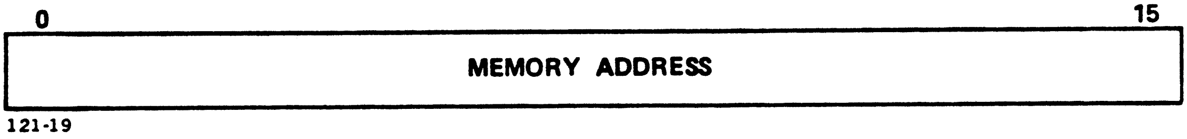


**Figure 3-6. Specify Memory Address Register Instruction**

**TABLE 3-2. SPECIFY MEMORY ADDRESS BIT FUNCTIONS**

| Bits | Function |
|---|---|
| 0-15 | Memory address for the next data channel transfer. |

### 3.3.3 Specify Block Count Instruction

Instruction Mnemonic: **DOC**[f] ac,**42**

Instruction Function: Loads bits 8-15 of the specified accumulator into the controller's Block Count Register. Figure 3-7 illustrates the accumulator format for the Specify Block Count instruction. Table 3-3 defines accumulator bit functions.

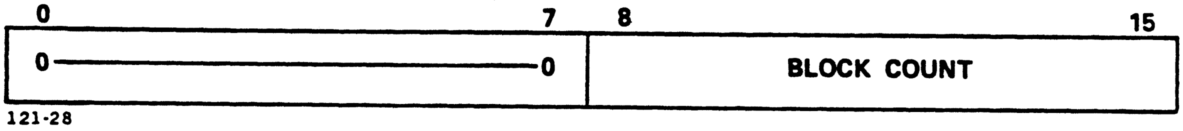


**Figure 3-7. Specify Block Count Instruction**

**TABLE 3-3. SPECIFY BLOCK COUNT BIT FUNCTIONS**

| Bits | Function |
|---|---|
| 0-7 | Unused. Set to zero. |
| 8-15 | Two's complement of the number of data blocks to be transferred in a read or write operation. One block is 256 words (512 bytes). |

### 3.3.4 Read Tape Controller Status Instruction

Instruction Mnemonic: **DIA**[f] ac,**42**

Instruction Function: Loads the tape controller's status flags and the data on the bidirectional data bus into the specified accumulator. Figure 3-8 illustrates the accumulator format for the Read Tape Controller Status instruction. Table 3-4 defines accumulator bit functions.

Do not use this command while the controller busy flag is set.

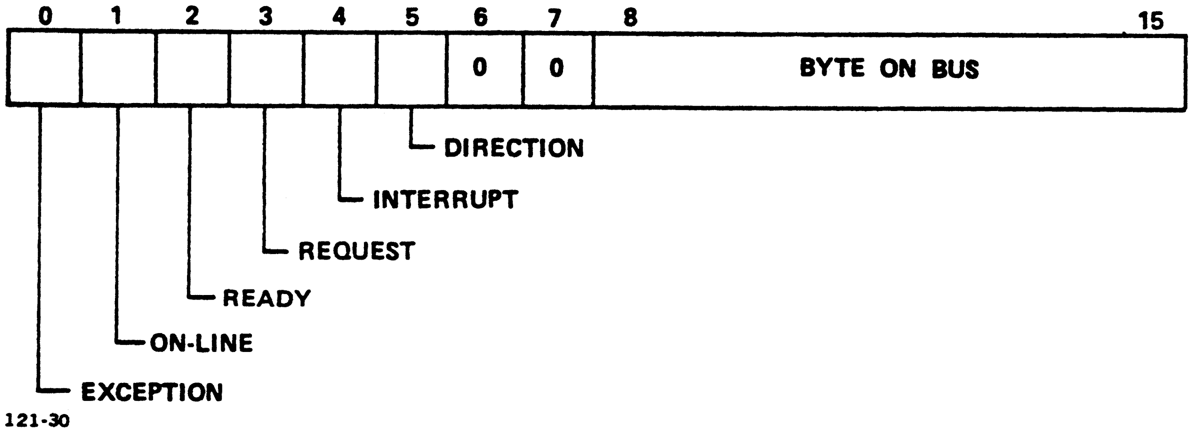


**Figure 3-8. Read Tape Controller Status Instruction**

### TABLE 3-4. READ TAPE CONTROLLER STATUS BIT FUNCTIONS

| Bits | Function |
|---|---|
| 0 | Exception (EXC) - The streaming tape drive detected an error condition, found a file mark, or received a reset signal. |
| 1 | On-line - Status of Bit 0 of the command register. |
| 2 | Ready - Current state of the ready signal from the streaming tape drive. |
| 3 | Request - Current state of the request signal. |
| 4 | Interrupt - Next byte is on the bidirectional data bus in a read status operation. |
| 5 | Direction (DIRC) - When set, byte on bidirectional data bus is transmitted by streaming tape drive. |
| 6-7 | Unused. Set to zero. |
| 8-15 | Byte on Bus - If busy flag is not set and DIRC is not set, this byte is the content of the controller's command register. |

### 3.3.5 Read Memory Address Instruction

Instruction Mnemonic: **DIB[f] ac,42**

Instruction Function: Loads the contents of the controller's Memory Address Register into the specified accumulator. Figure 3-9 illustrates the accumulator format for the Read Memory Address instruction. Table 3-5 defines accumulator bit functions.

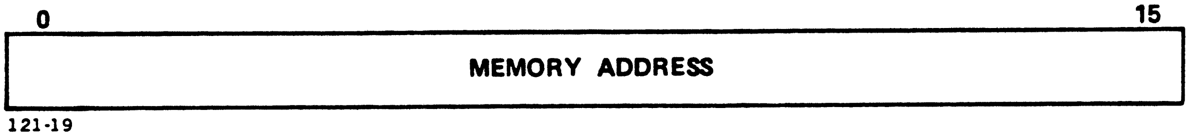


**Figure 3-9. Read Memory Address Instruction**

**TABLE 3-5. READ MEMORY ADDRESS BIT FUNCTION**

| Bits | Function |
|---|---|
| 0-15 | Memory address of next data channel transfer. |

### 3.3.6 Read Block Count Instruction

Instruction Mnemonic: **DIC[f] ac,42**

Instruction Function: Loads the contents of the controller's Block Counter Register into the specified accumulator. Figure 3-10 illustrates the accumulator format for the Read Block Count instruction. Table 3-6 defines accumulator bit functions.

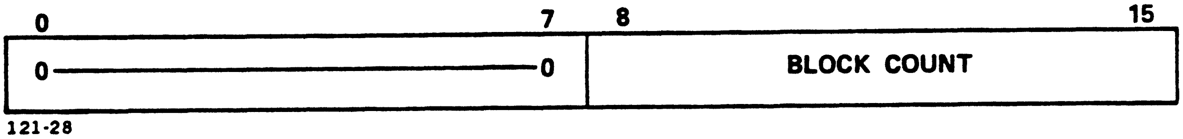


**Figure 3-10. Read Block Count Instruction**

**TABLE 3-6. READ BLOCK COUNT BIT FUNCTIONS**

| Bits | Function |
|---|---|
| 0-7 | Unused. Set to zero. |
| 8-15 | Two's complement of the number of data blocks to be transferred in a read or write operation. One block is 256 words (512 bytes). |

### 3.3.7 Input/Output Reset (Tape)

Instruction Mnemonic: **IORST**

Instruction Format: Performs all operations listed at f = C, resets controller Memory Address Register, resets Block Count Register, and starts bootstrap sequence.

## 3.4 DISK COMMAND DESCRIPTOR BLOCKS

Disk instructions are performed via direct memory access (DMA) using Command Descriptor Blocks (CDBs). A CDB consists of ten 16-bit words. It is set up by software before a disk operation begins, and is located in main memory. The actual execution of a CDB occurs according to the following sequence:

- o The central processing unit (CPU) executes a DOAS instruction, which writes the CDB starting address into the LOTUS 800 disk Memory Address Register.
- o The controller uses the address in the Memory Address Register to transfer all CDBs from main memory to controller buffer memory.
- o The controller executes the CDBs.
- o After a command is completed, the controller writes the completion status of each disk operation into Words 8 and 9 of the CDBs in main memory.

Figure 3-11 illustrates the Disk CDB. Figures 3-12 and 3-13 show the bit format of Words 8 and 9, respectively. Table 3-7 defines the bits of Words 0 through 9 of the CDB. Note that Words 0 through 9 are written in ascending order in main memory.

|  | BIT |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | MEMORY ADDRESS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 0 ——— 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | DISK COMMAND |  |  |  |  |  |  |  | 0 | SE | 0 | 0 | TV | 0 | 0 | 0 |
| 4 | 0 ——— 0 |  |  |  |  |  |  |  |  |  |  |  |  |  | DRV # |  |
| 5 | CYLINDER |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | 0 ——— 0 |  |  |  | HEAD |  |  |  | SECTOR |  |  |  |  |  |  |  |
| 7 | 0 ——— 0 |  |  |  |  |  |  |  | SECTOR COUNT |  |  |  |  |  |  |  |
| 8 | DISK CONTROLLER STATUS |  |  |  |  |  |  |  | DISK DRIVE STATUS |  |  |  |  |  |  |  |
| 9 | DISK DRIVE ERROR STATUS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

128-01

**Figure 3-11. Disk Command Descriptor Block Format**

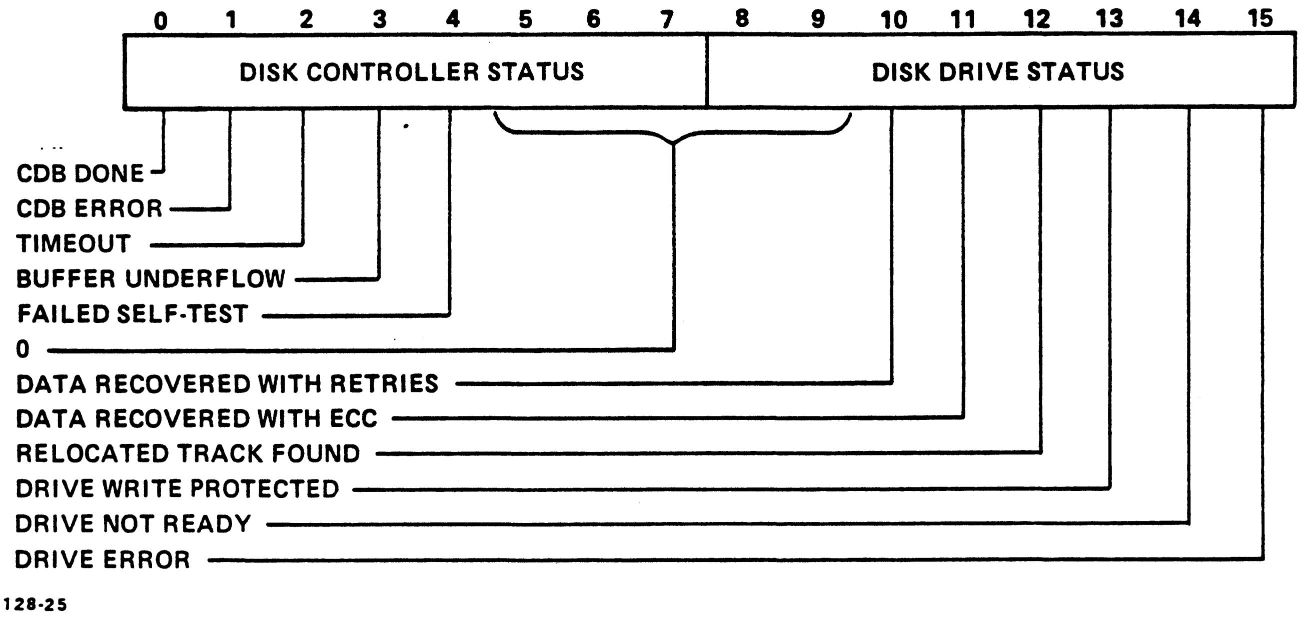


Figure 3-12. Disk Controller and Disk Drive Status Format

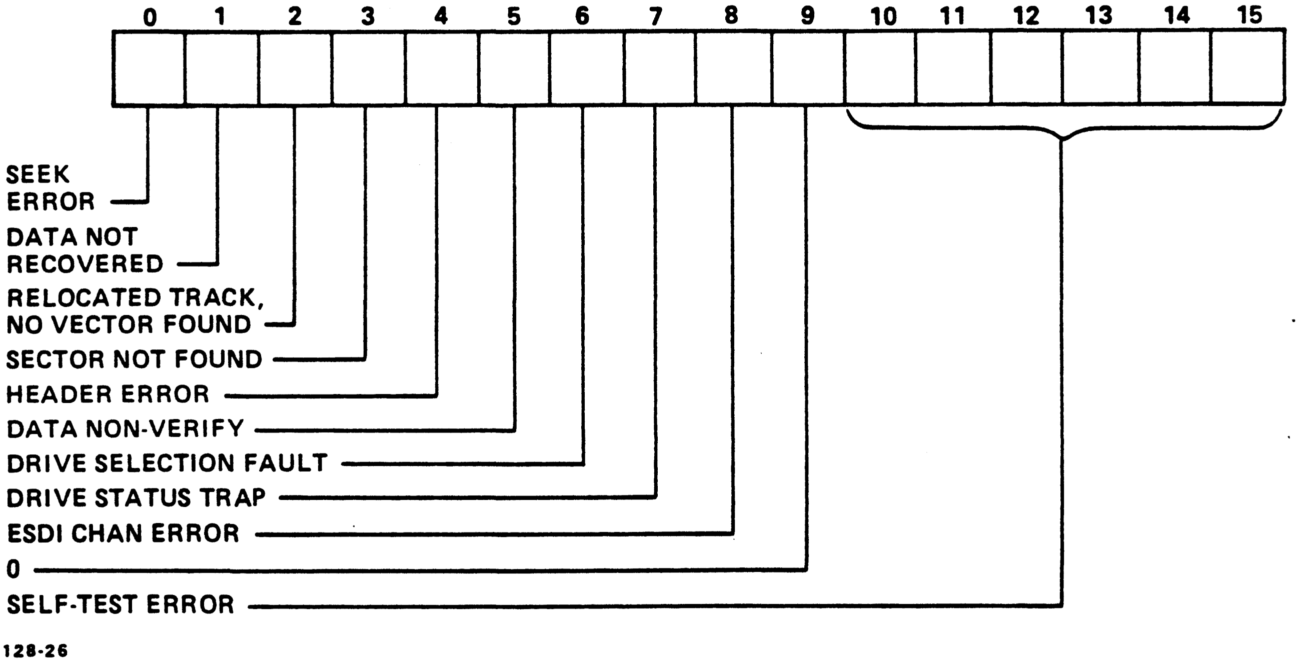


Figure 3-13. Disk Drive Error Status Format

## TABLE 3-7. DISK COMMAND DESCRIPTOR BLOCK BIT DEFINITIONS (1 of 5)

| Word | Bits | Definition |
|---|---|---|
| 0 | 0-15 | Next CDB Pointer (set to 0 for last CDB) - Specifies the link address to the next CDB. If no chaining is desired, or the CDB is the last one in the chain, each bit of the Next CDB Pointer is set to 0. |
| 1 | 0-15 | Memory Address - Specifies starting main memory address where disk data is to be read or written. If the data destination or source is the controller buffer, each bit of the memory address pointer is set to 1. |
| 2 | 0-15 | All set to zeros. This field is used by the controller firmware to assign an ID number to the CDB. |
| 3 | 0-7 | Disk Command - Specifies particular disk operation. |
|  | 8 | Set to zero. |
|  | 9 | Stop on Error (SE) - If the current CDB terminates with a fatal error, no further CDBs are executed. Otherwise, if the CDBs are linked, the controller continues to execute them. |
|  | 10-11 | Set to zero. |
|  | 12 | Track Verify (TV) - If set, a sector ID field is read to verify correct position of heads. If position is incorrect, a Seek Error is generated in the Disk Drive Status. Track Verify applies only to Seek, Rezero, or Read Physical Sector commanas. |
|  | 13-15 | Set to zero. |

## TABLE 3-7. DISK COMMAND DESCRIPTOR BLOCK BIT DEFINITIONS (2 of 5)

| Word | Bits | Definition |
|---|---|---|
| 4 | 0-13 | Set to zero. |
| | 14-15 | Drive number - Specifies drive 0, 1, 2, or 3. |
| 5 | 0-15 | Cylinder - Specifies starting cylinder number. |
| 6 | 0-3 | Set to zero. |
| | 4-7 | Head - Specifies starting head number. |
| | 8-15 | Sector - Specifies starting sector number. |
| 7 | 0-7 | Set to zero. |
| | 8-15 | Sector Count - Specifies number of sectors to be transferred. |
| 8 | | Disk Controller and Disk Drive Status. |
| | 0 | Command Descriptor Block (CDB) Done - Disk command has completed. |
| | 1 | CDB Error - CDB has undefined and/or conflicting parameters. Command execution will not begin. |
| | 2 | Timeout - Data transfer operation initiated by the last start pulse did not complete, and operation is terminated. The Head/ Sector address counter points to the sector where error occurred. |

## TABLE 3-7. DISK COMMAND DESCRIPTOR BLOCK BIT DEFINITIONS (3 of 5)

| Word | Bits | Definition |
|---|---|---|
| 8 | | (Continued) |
| | 3 | Buffer Overflow - Because the controller did not halt data channel operation in time, controller buffer underflowed during a read operation. If this bit is set, data transferred to main memory is not valid. |
| | 4 | Failed Self-Test - If the controller fails self-test diagnostics after power-up, this bit is set. A 6-bit error word describes the failure. |
| | 5-9 | Set to zero. |
| | 10 | Data Recovered with Retries - Specified sector data was recovered using the number of retries specified in the drive parameter block. |
| | 11 | Data Recovered with ECC - Specified sector data was recovered using Reed-Solomon error correction. |
| | 12 | Relocated Track Found - A normal read, write, or verify operation has continued on a relocated track. |
| | 13 | Drive Write-Protected - Selected drive is write-protected. Any write operation will abort. |
| | 14 | Drive Not Ready - Selected drive is not ready to read, write, or seek. |
| | 15 | Drive Error - A disk drive error has occurred. To determine specific error, read Word 9 of CDB Format. |

## TABLE 3-7. DISK COMMAND DESCRIPTOR BLOCK BIT DEFINITIONS (4 of 5)

| Word | Bits | Definition |
|---|---|---|
| 9 | | Disk Drive Error Status |
| | 0 | Seek Error - Selected drive was not able to seek the specified cylinder/head or to rezero properly. |
| | 1 | Data Not Recovered - Despite retries and/or error correction, data on specified sector was not recovered. |
| | 2 | Relocated Track, No Vector Found - Hard disk controller (HDC) was not able to seek to the relocated track because the relocation vector could not be read from the defective track fields. |
| | 3 | Sector Not Found - Specified sector was not found after at least two disk revolutions. No sector ID errors were encountered. |
| | 4 | Header Error - Header compare or header Cyclic Redundancy Check (CRC) errors have occurred. |
| | 5 | Data Non-Verify - Specified sector does not match the data block in memory to which it was compared. This bit is set only by the Verify command. |
| | 6 | Drive Selection Fault - A fault occurred when selecting or deselecting the drive. |
| | 7 | Drive Status Trap - A fault occurred during a normal I/O operation when read gate or write gate was active. |
| | 8 | ESDI Channel Error - A fault occurred during an ESDI serial communication. |

## TABLE 3-7. DISK COMMAND DESCRIPTOR BLOCK BIT DEFINITIONS (5 of 5)

| Word | Bits | Definition |
|---|---|---|
| 9 | | (Continued) |
| | 9 | Set to zero. |
| | 10-15 | Self-Test Error Code - If the failed self-test bit is set in the Disk Controller Status word, bits 10 through 15 contain a code for that error. See Appendix A for a listing of Self-Test error codes. |

### 3.4.1 Read Command

The Read command performs multi-sector data transfers from the disk to main memory via direct memory access (DMA). If the disk memory address is set to 177777 (octal), data remains in the controller buffer. The sector count should not exceed the size of the controller data buffer, which is 64 sectors (32KB). Figure 3-14 illustrates the Read command.

| | BIT | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | MEMORY DESTINATION ADDRESS | | | | | | | | | | | | | | | |
| 2 | 0 ——— 0 | | | | | | | | | | | | | | | |
| 3 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | SE | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 ——— 0 | | | | | | | | | | | | | | DRV # | |
| 5 | CYLINDER | | | | | | | | | | | | | | | |
| 6 | 0 ——— 0 | | | | HEAD | | | | SECTOR | | | | | | | |
| 7 | 0 ——— 0 | | | | | | | | SECTOR COUNT | | | | | | | |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

128-40

**Figure 3-14. Read Command**

### 3.4.2 Write Command

The Write command performs a multi-sector data transfer via direct memory access (DMA) from main memory to disk. If the disk memory address is set to 177777 (octal), the controller buffer is the data source. The sector count should not exceed 64 sectors (32KB). Figure 3-15 illustrates the Write command.

BIT

| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | MEMORY SOURCE ADDRESS | | | | | | | | | | | | | | | |
| 2 | 0 | | | | | | | | | | | | | | | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | SE | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | | | | | | | | | | | | | 0 | DRV # | |
| 5 | CYLINDER | | | | | | | | | | | | | | | |
| 6 | 0 | | | 0 | HEAD | | | | SECTOR | | | | | | | |
| 7 | 0 | | | | | | | 0 | SECTOR COUNT | | | | | | | |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

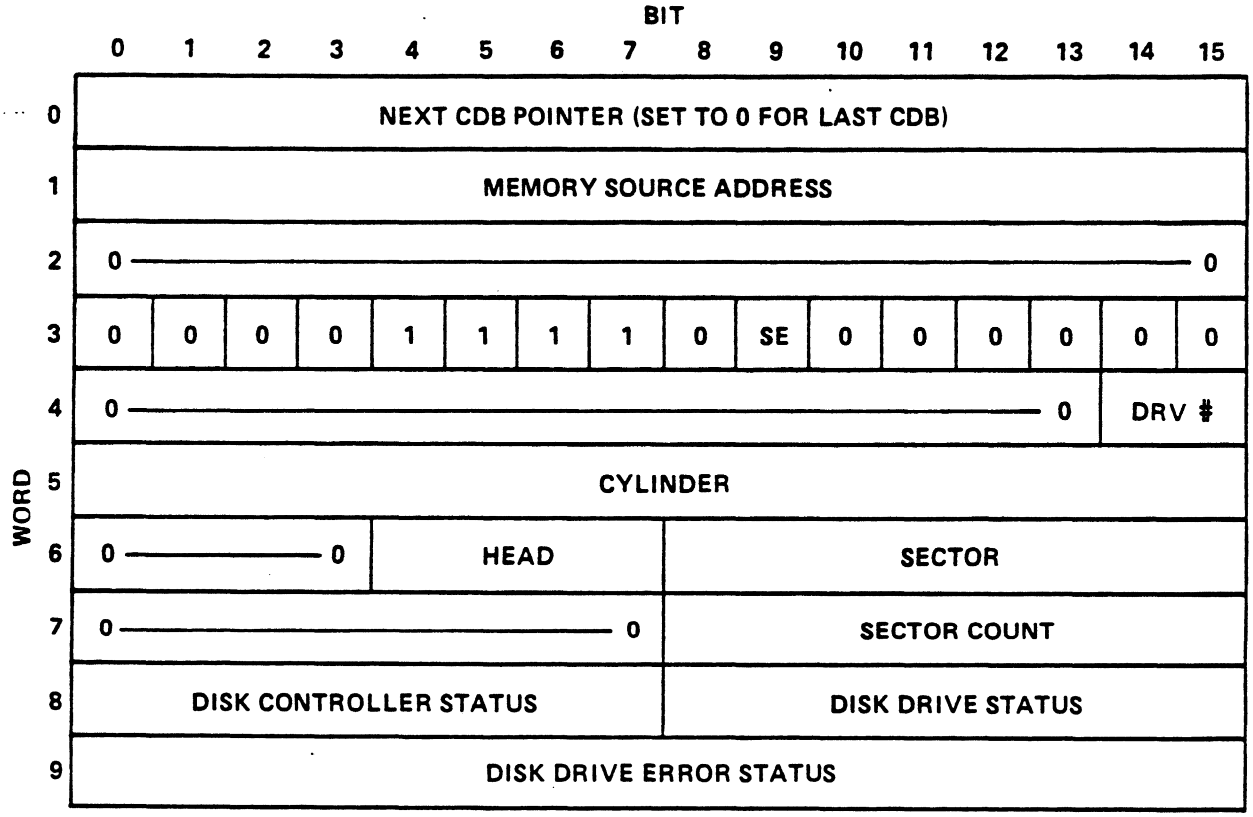
128-02

**Figure 3-15. Write Command**

### 3.4.3 Verify Command

The Verify command compares main memory data with disk data track-by-track. The sector count should not exceed the number of sectors per track (spare sectors excluded). Any sector miscompare causes the Data Non-Verify bit in the Disk Drive Error Status to be set. In addition, the Sector and Sector Count fields are updated. The Sector field specifies the first miscompared sector encountered when verifying, and the Sector Count field reflects the sector count at that point (the number of unprocessed sectors).

The Verify command must be repeated for each bad sector of a track, because only one bad sector can be reported per command descriptor block. Figure 3-16 illustrates the Verify command.



| | BIT | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | MEMORY SOURCE ADDRESS | | | | | | | | | | | | | | | |
| 2 | 0 | | | | | | | | | | | | | | | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | SE | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | | | | | | | | | | | | | 0 | DRV # | |
| 5 | CYLINDER | | | | | | | | | | | | | | | |
| 6 | 0 | | | 0 | HEAD | | | | SECTOR | | | | | | | |
| 7 | 0 | | | | | | | 0 | SECTOR COUNT | | | | | | | |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

128-03

**Figure 3-16. Verify Command**

### 3.4.4 Format Command

The Format command formats the number of tracks specified by Track Count starting at Head and Cylinder. The head and track numbers are incremented according to the general select byte in the Drive Parameter Block. Sector parameters such as PLO Sync, Write Splice, and End Gap are also specified in the Drive Parameter Block. Pattern specifies the data pattern to be written to the data field of each formatted sector. Header and data sync bytes are 376 (octal). Figure 3-17 illustrates the Format command.

At the initial power-up, the Drive Parameter Block specifies the number of physical sectors per track. The sectors are numbered sequentially from zero to N-1 (N = the number of physical sectors per track). For example, a 35-sector track is numbered consecutively from 0 through 34. Spare or defective sector numbers must be greater than N-1. Figure 3-18 illustrates how a sector is organized on the disk.

Two examples of sector maps are provided below. In both of the examples, 35 zero bytes follow the last sector number. The first zero byte is the most significant byte of the word that contains the last sector number. These bytes account for the flag byte in each header.

Sector map (octal) for a 35-sector track (0 through 34)

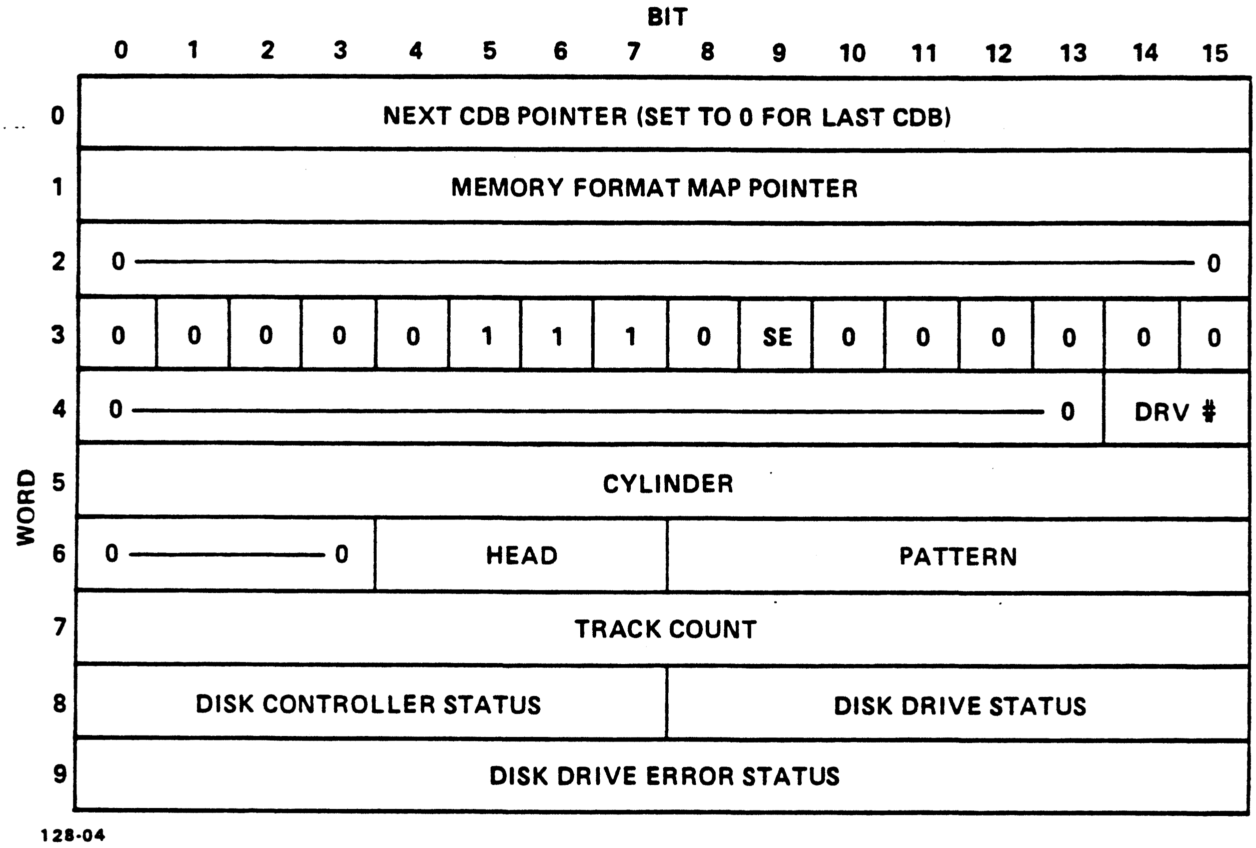
```
  400   1402   2404   3406   4410   5412   6414   7416
10420  11422  12424  13426  14430  15432  16434  17436
20440     42      0      0      0      0      0      0
    0      0      0      0      0      0      0      0
    0      0      0
```

Sector map (hex) for a 35-sector track (0 through 34)

```
  100    302    504    706    908    BOA    DOC    FOE
 1110   1312   1514   1716   1918   1B1A   1D1C   1F1E
 2120     22      0      0      0      0      0      0
    0      0      0
```

Once the Format command is given, the Map Pointer points to the sector map in main memory. The sector map then supplies the logical sector number sequence to the selected track as follows: the first byte (LS) of the sector map is written into the sector number field of the first physical sector; the second byte (MS) is written into the second physical sector. This sequence continues until the required number of sectors has been formatted.

The same sector map can be repeated for multiple track format commands. Spare sectors can be accessed using a single-sector Read or Write command descriptor block.



| WORD | BIT 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | MEMORY FORMAT MAP POINTER | | | | | | | | | | | | | | | |
| 2 | 0 ——— 0 | | | | | | | | | | | | | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | SE | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 ——— 0 | | | | | | | | | | | | | | DRV # | |
| 5 | CYLINDER | | | | | | | | | | | | | | | |
| 6 | 0 ——— 0 | | | | HEAD | | | | PATTERN | | | | | | | |
| 7 | TRACK COUNT | | | | | | | | | | | | | | | |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

128-04

Figure 3-17. Format Command

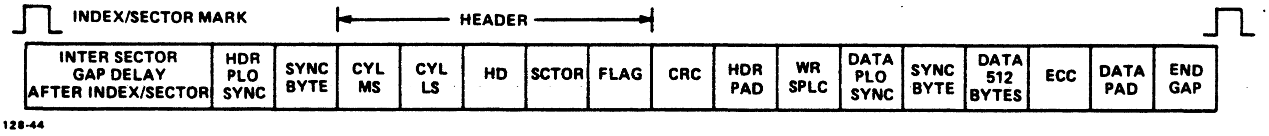
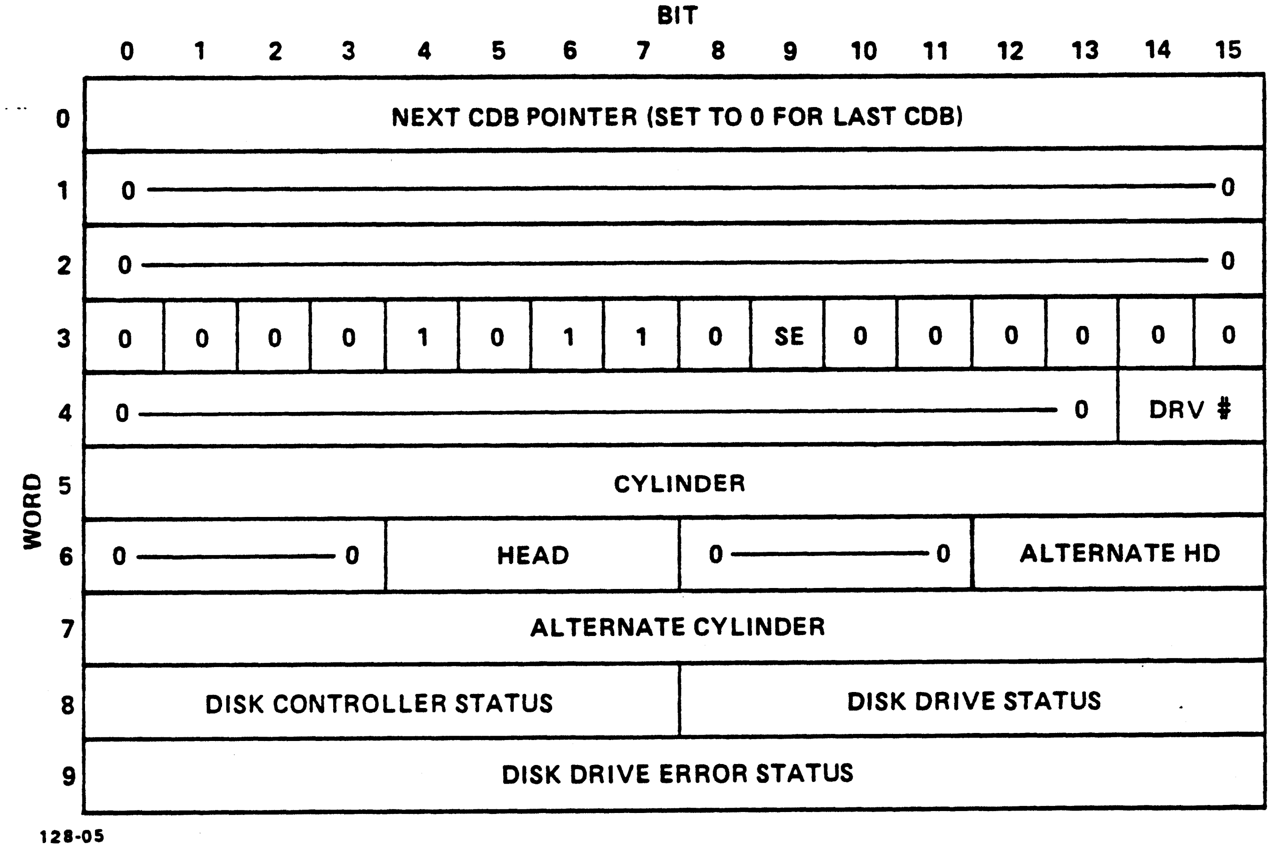


Figure 3-18. Sector Organization on a Disk

## 3.4.5 Relocate Track Command

The Relocate Track command is normally used with the Verify and Format commands to relocate bad tracks. The Verify command determines which sectors on a track are bad. Once the bad tracks are identified, the Format command uses spare sectors to reformat the bad sectors as long as the number of bad sectors does not exceed the number of spare sectors. If the number of bad sectors exceeds the number of spare sectors, the track is considered bad; the Relocate Track command causes the hard disk controller to write a relocation vector to all data fields of the bad track while changing the sync byte for the header field from 376 to 375 (octal). If a bad track is encountered during normal disk operations, the hard disk controller automatically vectors to the alternate track. The Relocate Track command does not reformat or write to the alternate track. Figure 3-19 illustrates the Relocate Track command.

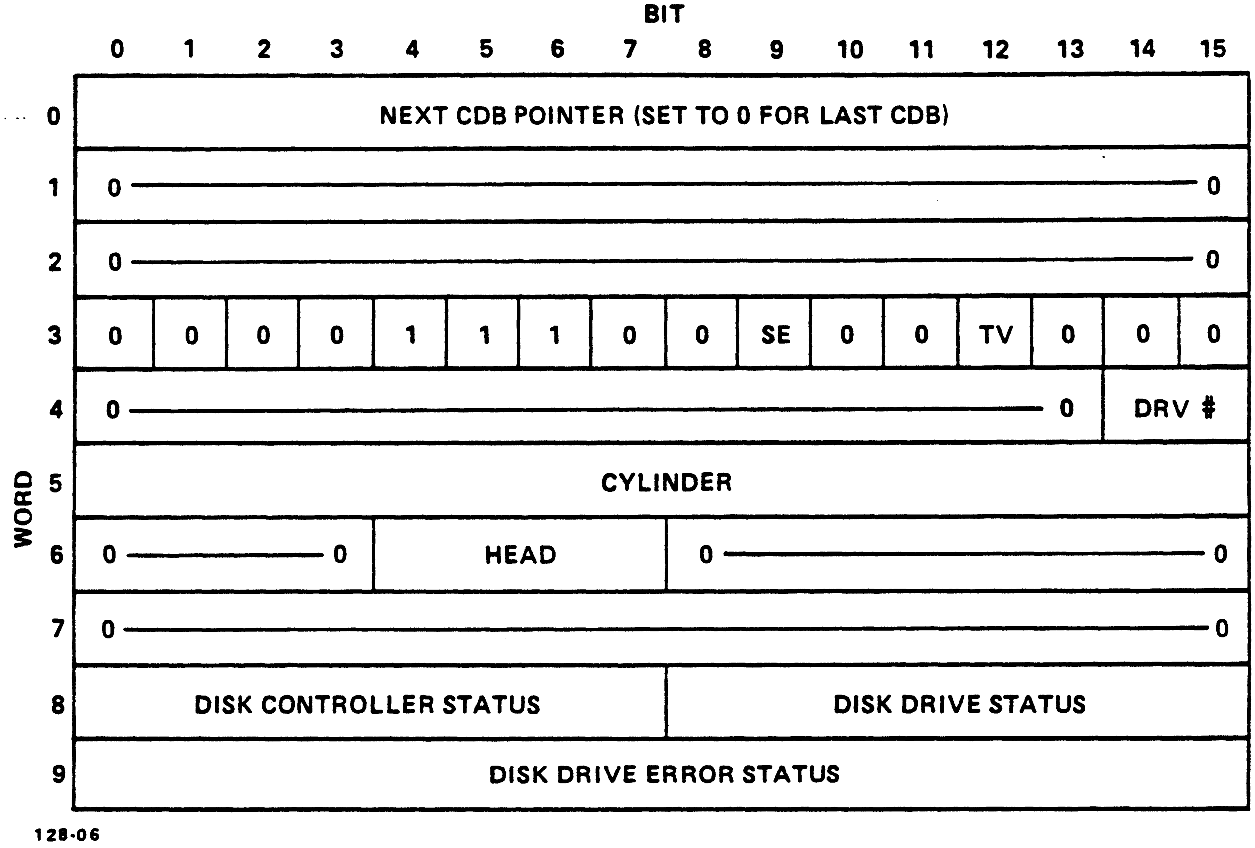


| WORD | BIT 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | 0 |
| 2 | 0 | | | | | | | | | | | | | | | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | SE | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | | | | | | | | | | | | | 0 | DRV # | |
| 5 | CYLINDER | | | | | | | | | | | | | | | |
| 6 | 0 | | | 0 | HEAD | | | | 0 | | | 0 | ALTERNATE HD | | | |
| 7 | ALTERNATE CYLINDER | | | | | | | | | | | | | | | |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

**Figure 3-19. Relocate Track Command**

### 3.4.6 Seek Command

The Seek command moves the head of the selected drive to the specified cylinder/head. If Track Verify (TV) is selected, the hard disk controller reads the first header it encounters on the specified track to verify that it is on the correct track. If not, a Seek Error status is generated. Figure 3-20 illustrates the Seek command.



| WORD | BIT 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | 0 ———— 0 | | | | | | | | | | | | | | | |
| 2 | 0 ———— 0 | | | | | | | | | | | | | | | |
| 3 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | SE | 0 | 0 | TV | 0 | 0 | 0 |
| 4 | 0 ———— 0 | | | | | | | | | | | | | | DRV # | |
| 5 | CYLINDER | | | | | | | | | | | | | | | |
| 6 | 0 ———— 0 | | | | HEAD | | | | 0 ———— 0 | | | | | | | |
| 7 | 0 ———— 0 | | | | | | | | | | | | | | | |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

**Figure 3-20. Seek Command**

### 3.4.7 Rezero Command

The Rezero command moves the heads of the selected drive back to cylinder zero, head zero. If Track Verify (TV) is selected, the hard disk controller verifies a successful Rezero by reading one header field. Figure 3-21 illustrates the Rezero command.

| WORD \ BIT | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | 0 |
| 2 | 0 | | | | | | | | | | | | | | | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | SE | 0 | 0 | TV | 0 | 0 | 0 |
| 4 | 0 | | | | | | | | | | | | | 0 | DRV # | |
| 5 | 0 | | | | | | | | | | | | | | | 0 |
| 6 | 0 | | | | | | | | | | | | | | | 0 |
| 7 | 0 | | | | | | | | | | | | | | | 0 |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

128-07

**Figure 3-21. Rezero Command**

### 3.4.8 Write Buffer Command

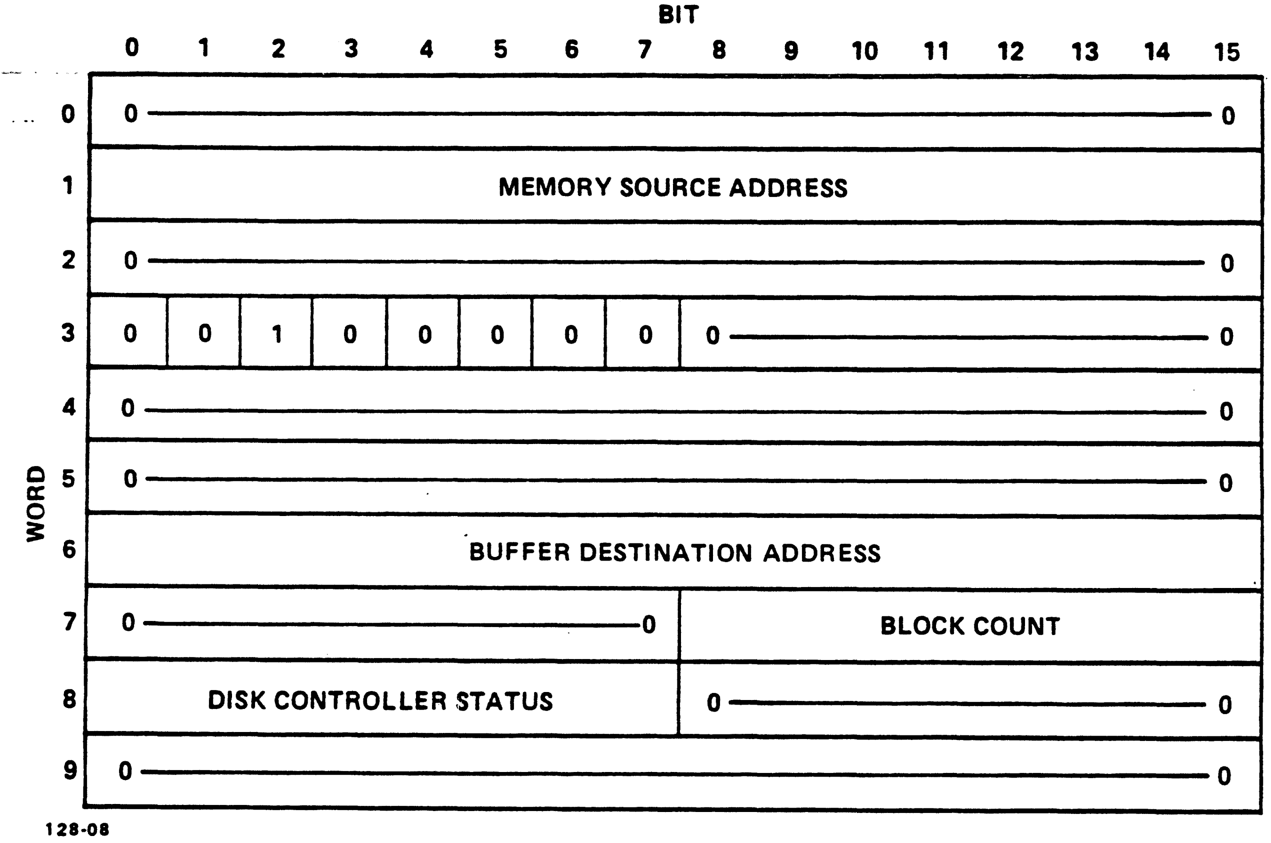
The Write Buffer command is a diagnostic command that writes to the controller track buffer without invoking any disk operations. The controller track buffer is 32,768 words (128 blocks). The Write Buffer command descriptor block (CDB) should not be used in a linked CDB list.

Controller buffer memory allocations are defined in octal as follows:

| | |
|---|---|
| 0 through 37777 | Data buffer |
| 40000 through 40237 | Command Descriptor Block buffer |
| 40240 through 77653 | Status Result Block buffer |
| 77654 through 77717 | Sector map |
| 77720 through 177777 | Drive parameter buffer (4 drives) |

Word 6 is a 16-bit buffer destination address that specifies the starting controller buffer address.

Word 7, bits 8 through 15, is the block count that indicates the number of 256 word blocks to be transferred from main memory to controller buffer memory. Figure 3-22 illustrates the Write Buffer command.



| WORD \ BIT | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | | | | | | | | | | | | | | | 0 |
| 1 | MEMORY SOURCE ADDRESS | | | | | | | | | | | | | | | |
| 2 | 0 | | | | | | | | | | | | | | | 0 |
| 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | 0 |
| 4 | 0 | | | | | | | | | | | | | | | 0 |
| 5 | 0 | | | | | | | | | | | | | | | 0 |
| 6 | BUFFER DESTINATION ADDRESS | | | | | | | | | | | | | | | |
| 7 | 0 | | | | | | | 0 | BLOCK COUNT | | | | | | | |
| 8 | DISK CONTROLLER STATUS | | | | | | | | 0 | | | | | | | 0 |
| 9 | 0 | | | | | | | | | | | | | | | 0 |

**Figure 3-22. Write Buffer Command**

### 3.4.9 Read Buffer Command

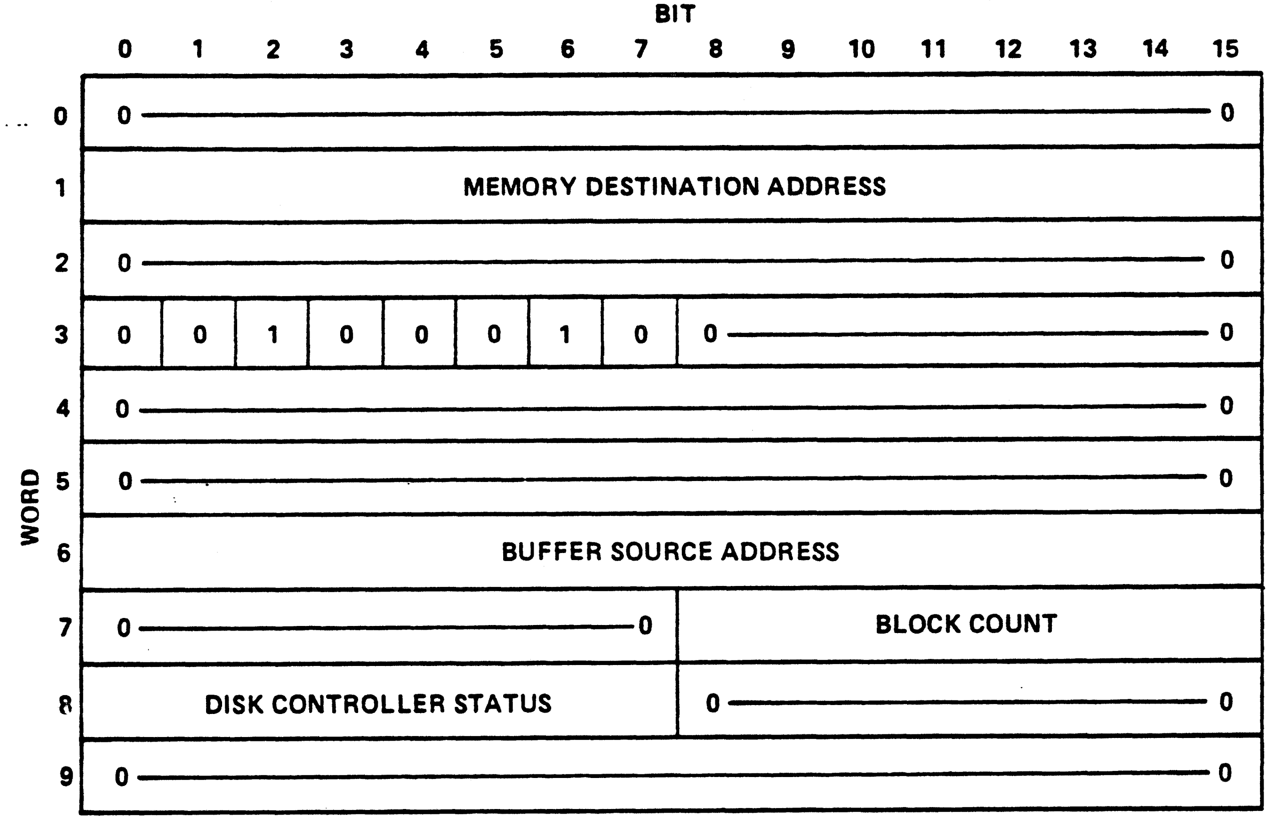
The Read Buffer command is a diagnostic command to read the contents of the controller track buffer without invoking disk operations. The controller buffer is 32,768 word (128 blocks). The Read Buffer command descriptor block (CDB) should not be used in a linked CDB list.

Controller buffer memory allocations are defined in octal as follows:

| | |
|---|---|
| 0 through 37777 | Data buffer |
| 40000 through 40237 | Command Descriptor Block buffer |
| 40240 through 77653 | Status Result Block buffer |
| 77654 through 77717 | Sector map |
| 77720 through 177777 | Drive parameter buffer (4 drives) |

Word 6 is a 16-bit buffer source address that specifies the starting controller buffer address.

Word 7, bits 8 through 15, indicates the number of 256 word blocks to be transferred from the controller buffer memory to main memory. Figure 3-23 illustrates the Read Buffer command.



<table>
<tr><th>WORD</th><th colspan="16">BIT</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>0</td><td colspan="16">0 ——— 0</td></tr>
<tr><td>1</td><td colspan="16">MEMORY DESTINATION ADDRESS</td></tr>
<tr><td>2</td><td colspan="16">0 ——— 0</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td colspan="8">0 ——— 0</td></tr>
<tr><td>4</td><td colspan="16">0 ——— 0</td></tr>
<tr><td>5</td><td colspan="16">0 ——— 0</td></tr>
<tr><td>6</td><td colspan="16">BUFFER SOURCE ADDRESS</td></tr>
<tr><td>7</td><td colspan="8">0 ——— 0</td><td colspan="8">BLOCK COUNT</td></tr>
<tr><td>8</td><td colspan="8">DISK CONTROLLER STATUS</td><td colspan="8">0 ——— 0</td></tr>
<tr><td>9</td><td colspan="16">0 ——— 0</td></tr>
</table>

128-09

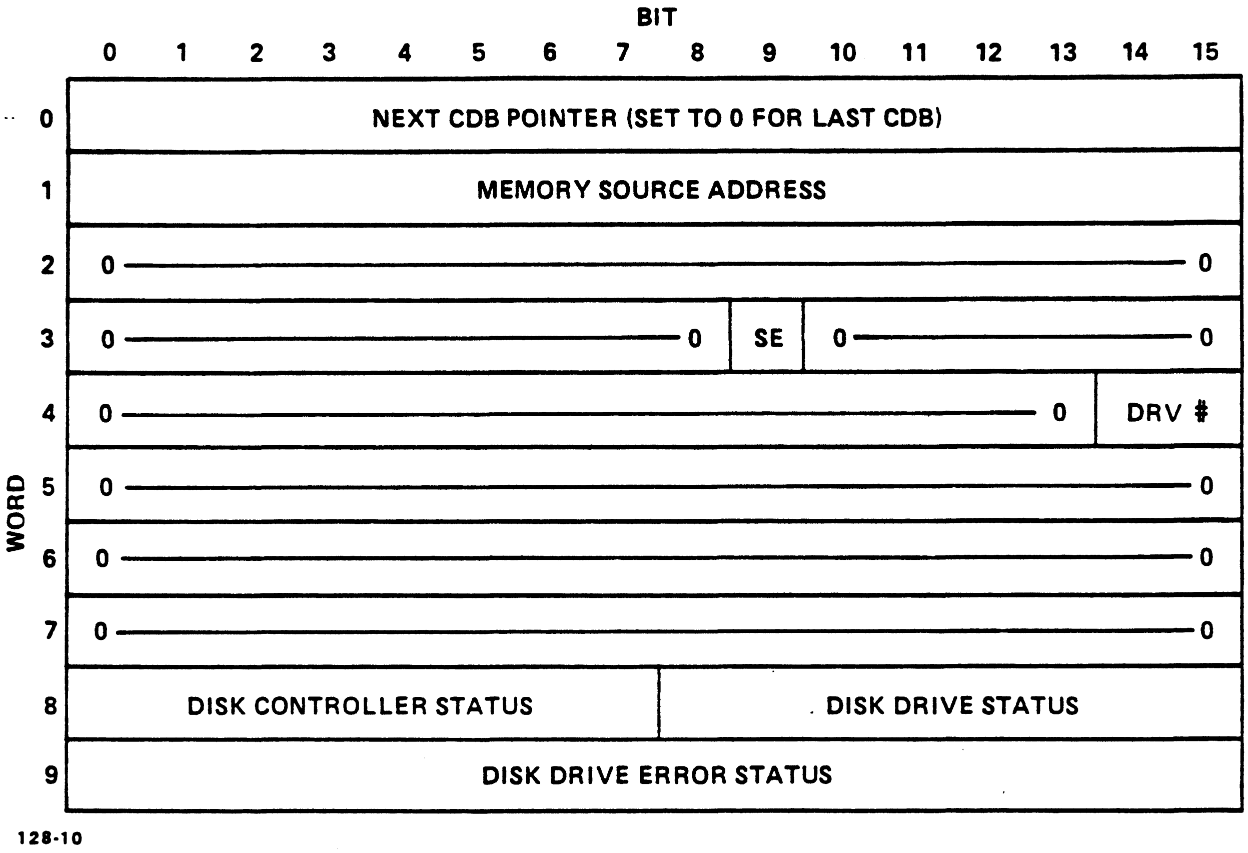
**Figure 3-23. Read Buffer Command**

## 3.4.10 Write Drive Parameter Block Command

The Write Drive Parameter Block command writes a 12-word Drive Parameter Block for the selected drive from the memory source address to the hard disk controller. Figure 3-24 illustrates the Write Drive Parameter Block command, and Figure 3-25 illustrates the Drive Parameter Block.

Although the drive parameter block can be reloaded to alter drive parameters, only the sectors per track field should be altered unless the block is reloaded for diagnostic purposes. Otherwise, altering other fields may result in drive errors.

At power-up/reset, controller firmware automatically sets all parameters in the drive parameter blocks. Therefore, it is only necessary for software to set sectors per track (excluding spares) after formatting, surface analysis, and remapping.

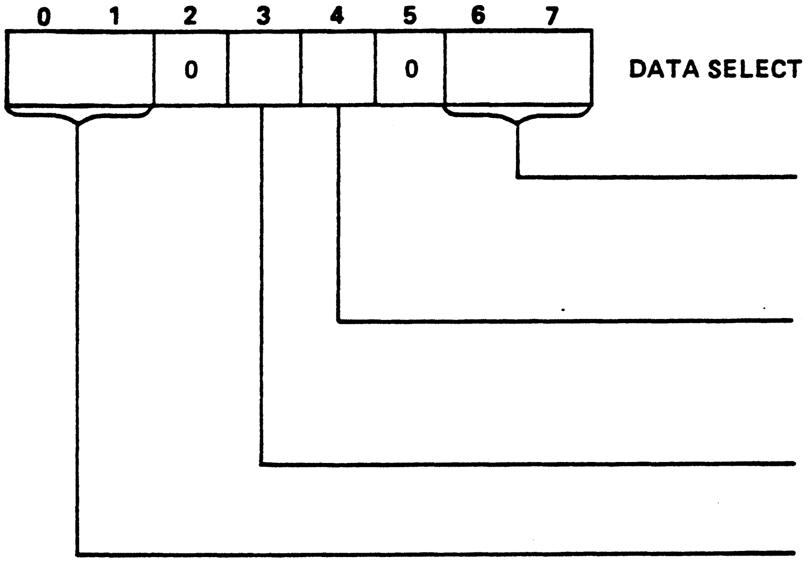


| WORD | | | |
|---|---|---|---|
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | |
| 1 | MEMORY SOURCE ADDRESS | | |
| 2 | 0 ——— 0 | | |
| 3 | 0 ——— 0 | SE | 0 ——— 0 |
| 4 | 0 ——— 0 | DRV # | |
| 5 | 0 ——— 0 | | |
| 6 | 0 ——— 0 | | |
| 7 | 0 ——— 0 | | |
| 8 | DISK CONTROLLER STATUS | DISK DRIVE STATUS | |
| 9 | DISK DRIVE ERROR STATUS | | |

**Figure 3-24. Write Drive Parameter Block Command**

<table>
<tr><th></th><th colspan="16">BIT</th></tr>
<tr><th>WORD</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>0</td><td colspan="8">DATA SELECT BYTE</td><td colspan="8">GENERAL SELECT BYTE</td></tr>
<tr><td>1</td><td colspan="16">TRACKS/SURFACE</td></tr>
<tr><td>2</td><td colspan="8">SECTORS/TRACK</td><td colspan="8">HEADS/DRIVE</td></tr>
<tr><td>3</td><td colspan="16">N/A</td></tr>
<tr><td>4</td><td colspan="16">N/A</td></tr>
<tr><td>5</td><td colspan="8">HEAD SETTLE</td><td colspan="8">N/A</td></tr>
<tr><td>6</td><td colspan="16">N/A</td></tr>
<tr><td>7</td><td colspan="8">N/A</td><td colspan="8">RETRY POLICY BYTE</td></tr>
<tr><td>8</td><td colspan="8">HEADER PLO SYNC LENGTH</td><td colspan="8">ISG DLY AFTER INDX/SCTR</td></tr>
<tr><td>9</td><td colspan="8">WRITE SPLICE LENGTH</td><td colspan="8">HEADER PAD LENGTH</td></tr>
<tr><td>10</td><td colspan="8">ECC LENGTH</td><td colspan="8">DATA PLO SYNC LENGTH</td></tr>
<tr><td>11</td><td colspan="8">SECTOR END GAP LENGTH</td><td colspan="8">DATA PAD LENGTH</td></tr>
</table>

128-11



DATA SELECT BYTE

DATA SIZE
00 – 128 BYTES
01 – 256 BYTES
10 – 512 BYTES (DEFAULT)
11 – RESERVED

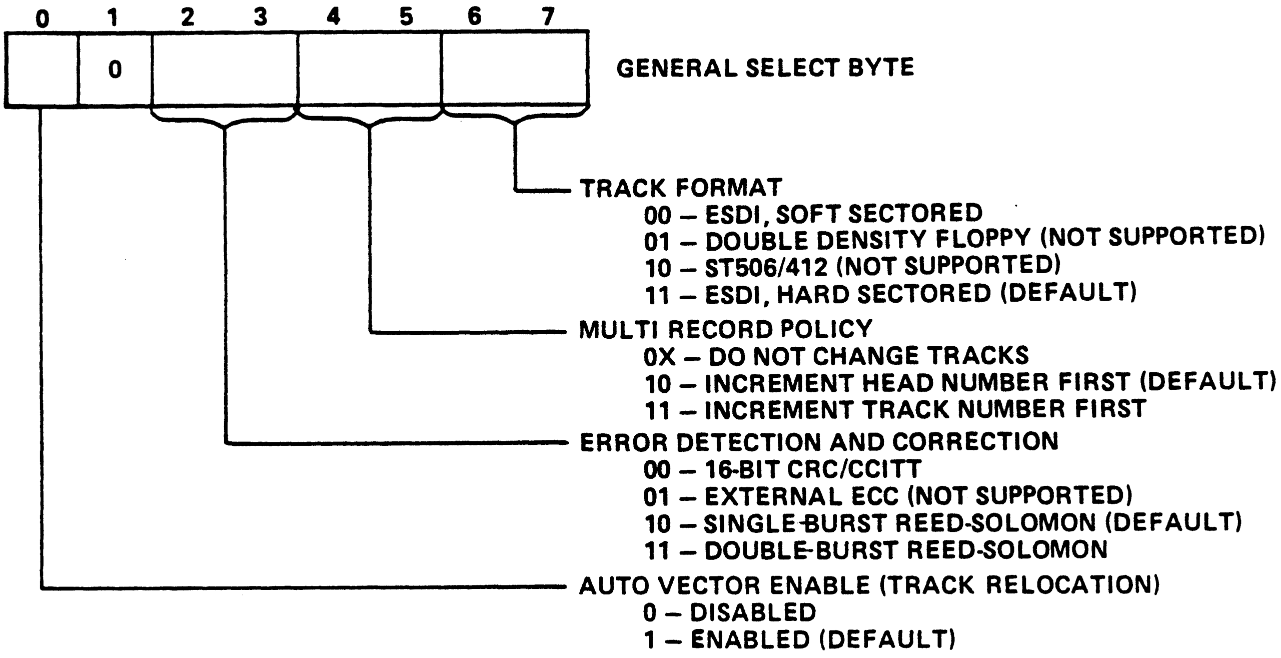
START SECTOR OPTION
0 – SECTOR NUMBERING STARTS WITH 1
1 – SECTOR NUMBERING STARTS WITH 0 (DEFAULT)

FLOPPY MOTOR ON DELAY
0 – READY TIME OUT (DEFAULT)
1 – MOTOR ON DELAY

DISK CONTROL INTERFACE TYPE
00 – SMD
01 – DOUBLE DENSITY FLOPPY
10 – ST506/412
11 – ESDI (DEFAULT)

128-12

Figure 3-25. Drive Parameter Block (1 of 2)



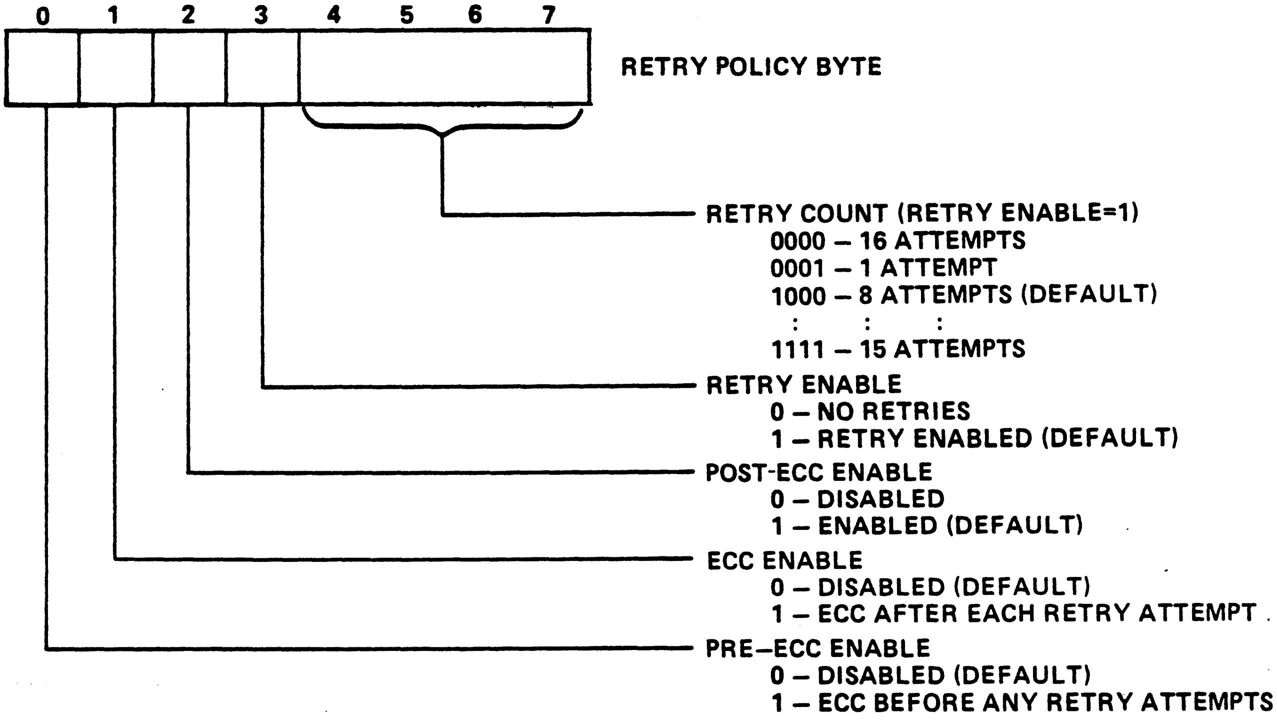
TRACK FORMAT
00 – ESDI, SOFT SECTORED
01 – DOUBLE DENSITY FLOPPY (NOT SUPPORTED)
10 – ST506/412 (NOT SUPPORTED)
11 – ESDI, HARD SECTORED (DEFAULT)

MULTI RECORD POLICY
0X – DO NOT CHANGE TRACKS
10 – INCREMENT HEAD NUMBER FIRST (DEFAULT)
11 – INCREMENT TRACK NUMBER FIRST

ERROR DETECTION AND CORRECTION
00 – 16-BIT CRC/CCITT
01 – EXTERNAL ECC (NOT SUPPORTED)
10 – SINGLE-BURST REED-SOLOMON (DEFAULT)
11 – DOUBLE-BURST REED-SOLOMON

AUTO VECTOR ENABLE (TRACK RELOCATION)
0 – DISABLED
1 – ENABLED (DEFAULT)

128-13



RETRY COUNT (RETRY ENABLE=1)
0000 – 16 ATTEMPTS
0001 – 1 ATTEMPT
1000 – 8 ATTEMPTS (DEFAULT)
: : :
1111 – 15 ATTEMPTS

RETRY ENABLE
0 – NO RETRIES
1 – RETRY ENABLED (DEFAULT)

POST-ECC ENABLE
0 – DISABLED
1 – ENABLED (DEFAULT)

ECC ENABLE
0 – DISABLED (DEFAULT)
1 – ECC AFTER EACH RETRY ATTEMPT

PRE–ECC ENABLE
0 – DISABLED (DEFAULT)
1 – ECC BEFORE ANY RETRY ATTEMPTS

128-14

**Figure 3-25. Drive Parameter Block (2 of 2)**

## 3.4.11 Read Drive Parameter Command

The Read Drive Parameter command transfers the 12-word drive parameter block (see Figure 3-25) for the selected drive from the hard disk controller to memory starting at the Memory Destination Address. Figure 3-26 illustrates the Read Drive Parameter command.

BIT

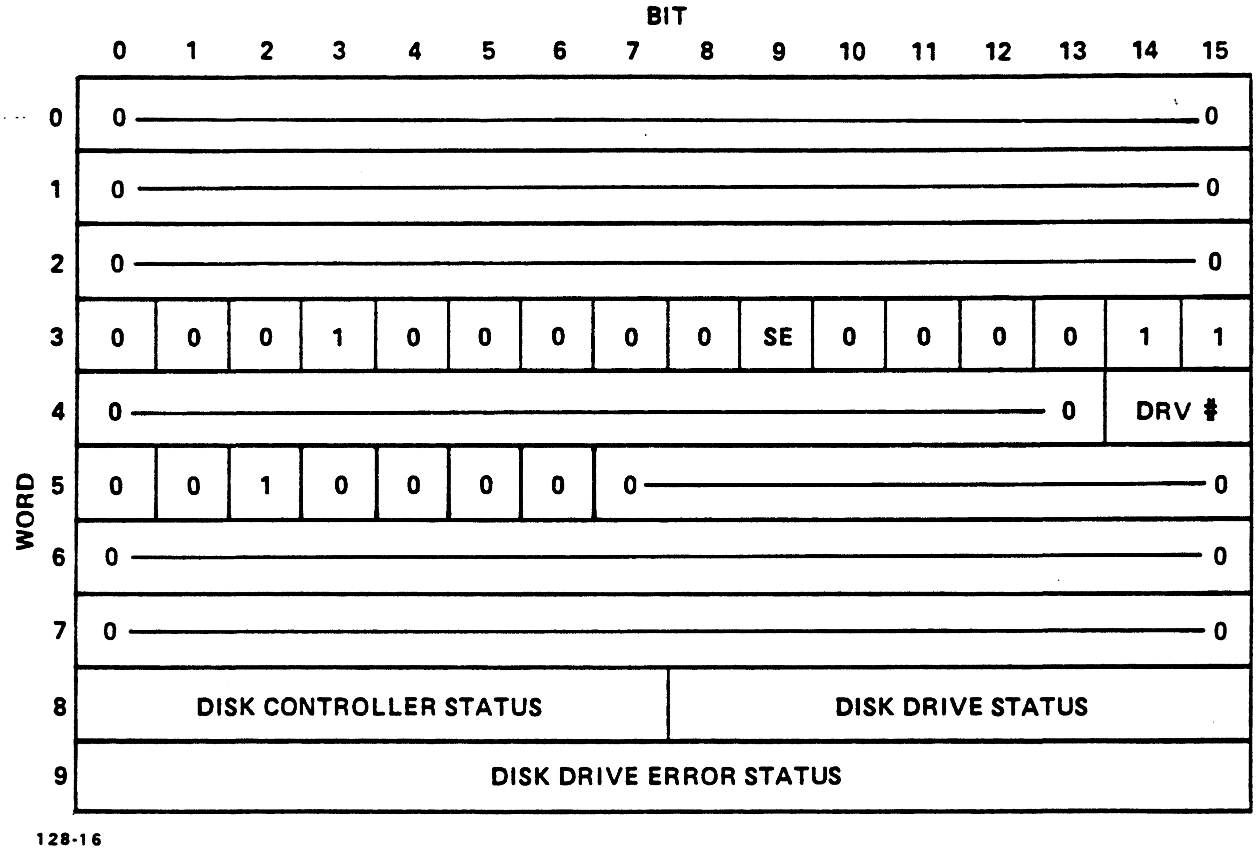
| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | MEMORY DESTINATION ADDRESS | | | | | | | | | | | | | | | |
| 2 | 0 | | | | | | | | | | | | | | | 0 |
| 3 | 0 | | | | | 0 | 1 | 1 | 0 | SE | 0 | | | | | 0 |
| 4 | 0 | | | | | | | | | | | | | 0 | DRV # | |
| 5 | 0 | | | | | | | | | | | | | | | 0 |
| 6 | 0 | | | | | | | | | | | | | | | 0 |
| 7 | 0 | | | | | | | | | | | | | | | 0 |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

128-15

**Figure 3-26. Read Drive Parameter Command**

### 3.4.12 Read Status Command

The Read Status command generates the controller status and drive status for the selected drive. It should not be used in a linked command descriptor block (CDB) list. Figure 3-27 illustrates the Read Status command.



| | BIT | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 0 | | | | | | | | | | | | | | | 0 |
| 1 | 0 | | | | | | | | | | | | | | | 0 |
| 2 | 0 | | | | | | | | | | | | | | | 0 |
| 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | SE | 0 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | | | | | | | | | | | | | 0 | DRV # | |
| 5 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | | | 0 |
| 6 | 0 | | | | | | | | | | | | | | | 0 |
| 7 | 0 | | | | | | | | | | | | | | | 0 |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

128-16

Figure 3-27. Read Status Command

## 3.4.13 Write Hard Disk Controller (HDC) Buffer Command

The Write HDC Buffer command is a diagnostic command to write to either one of two 256-word HDC internal buffers. When the command is given, the Memory Source Address points to data which is then transferred to the HDC internal sector buffer. Unless the data size field is changed in the Drive Parameter Block, the number of words transferred is always 256. If the toggle buffer (TB) is set, the HDC toggles to the second sector buffer after the first has been written. This means that two consecutive Write HDC Buffer commands will write to both sector buffers. The toggle buffer option is Word 3, Bit 12 of the Command Descriptor Block (CDB) command option field. Figure 3-28 illustrates the Write HDC Buffer command.

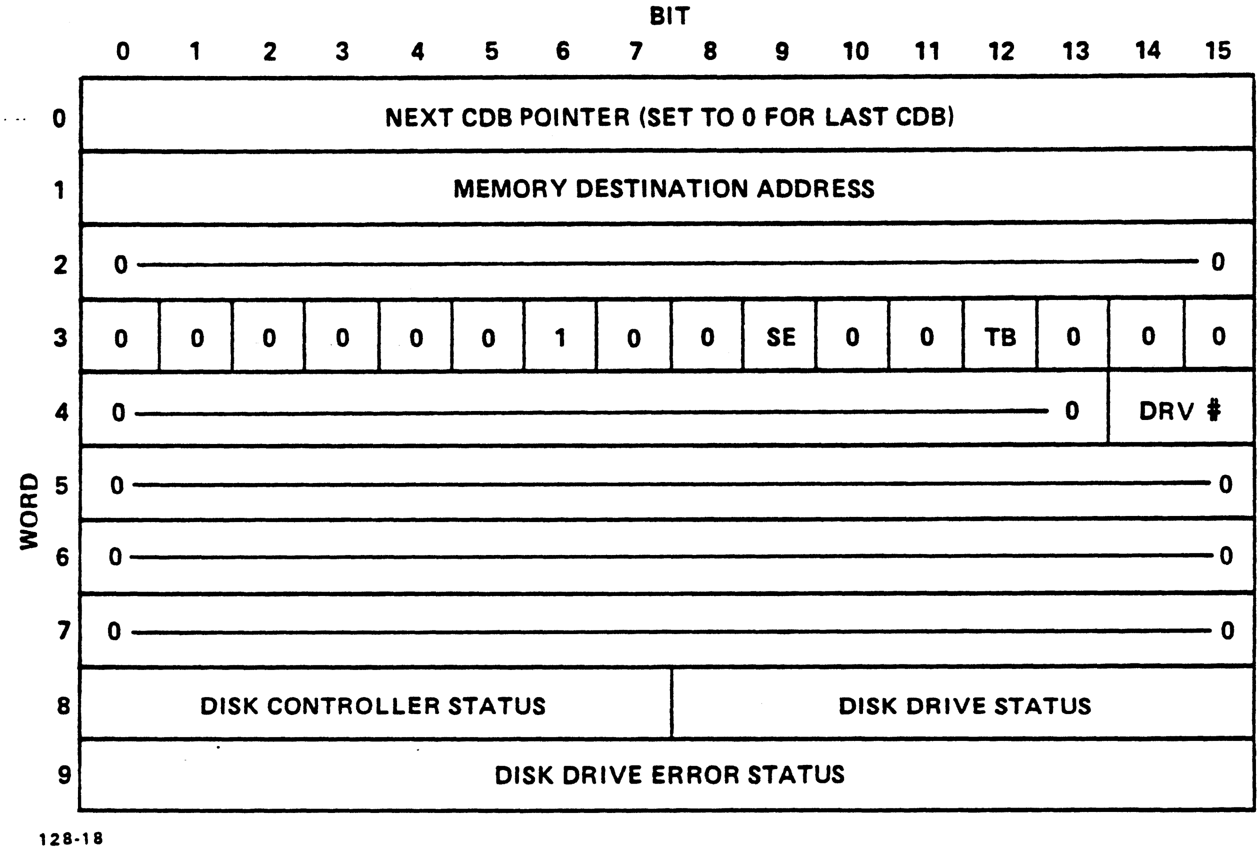
| | BIT | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | MEMORY SOURCE ADDRESS | | | | | | | | | | | | | | | |
| 2 | 0 | | | | | | | | | | | | | | | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | SE | 0 | 0 | TB | 0 | 0 | 0 |
| 4 | 0 | | | | | | | | | | | | | 0 | DRV # | |
| 5 | 0 | | | | | | | | | | | | | | | 0 |
| 6 | 0 | | | | | | | | | | | | | | | 0 |
| 7 | 0 | | | | | | | | | | | | | | | 0 |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

128-17

**Figure 3-28. Write Hard Disk Controller Buffer Command**

## 3.4.14 Read Hard Disk Controller (HDC) Buffer Command

The Read HDC Buffer command is a diagnostic command to read either one of two 256-word internal HDC buffers. When the command is given, the data in the HDC sector buffer is transferred to main memory starting at the Memory Destination Address. As in the Write HDC Buffer command, the number of words transferred is specified by the Drive Parameter Block. The default is 256 words. If the toggle buffer (TB) is set, the HDC toggles to the other sector buffer after the current sector buffer has been read. This means that two consecutive Read HDC Buffer commands will cause both HDC buffers to be read. Figure 3-29 illustrates the Read HDC Buffer command.



| WORD \ BIT | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | MEMORY DESTINATION ADDRESS | | | | | | | | | | | | | | | |
| 2 | 0 ——— 0 | | | | | | | | | | | | | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | SE | 0 | 0 | TB | 0 | 0 | 0 |
| 4 | 0 ——— 0 | | | | | | | | | | | | | | DRV # | |
| 5 | 0 ——— 0 | | | | | | | | | | | | | | | |
| 6 | 0 ——— 0 | | | | | | | | | | | | | | | |
| 7 | 0 ——— 0 | | | | | | | | | | | | | | | |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

128-18

Figure 3-29. Read Hard Disk Controller Buffer Command

## 3.4.15 Read Header Command

The Read Header command reads the 6-byte header from the specified physical sector. The 6-byte header field, which is organized into six words, is transferred to the Memory Destination Address. If the arbitrary sector (AS) option is enabled, the hard disk controller (HDC) reads the first header that it encounters and will not do a physical search for the specified physical sector. Figure 3-30 illustrates the Read Header command, and Figure 3-31 illustrates the 6-byte header field dumped by the Read Header command.

| | BIT | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | NEXT CDB POINTER (SET TO 0 FOR LAST CDB) | | | | | | | | | | | | | | | |
| 1 | MEMORY DESTINATION ADDRESS | | | | | | | | | | | | | | | |
| 2 | 0 | | | | | | | | | | | | | | | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | SE | 0 | 0 | AS | 0 | 0 | 0 |
| 4 | 0 | | | | | | | | | | | | | 0 | DRV # | |
| 5 | CYLINDER | | | | | | | | | | | | | | | |
| 6 | 0 | | | 0 | HEAD | | | | PHYSICAL SECTOR # | | | | | | | |
| 7 | 0 | | | | | | | | | | | | | | | 0 |
| 8 | DISK CONTROLLER STATUS | | | | | | | | DISK DRIVE STATUS | | | | | | | |
| 9 | DISK DRIVE ERROR STATUS | | | | | | | | | | | | | | | |

128-19

**Figure 3-30. Read Header Command**

| WORD | BIT 0 1 2 3 4 5 6 7 | BIT 8 9 10 11 12 13 14 15 |
|---|---|---|
| 0 | 0 ——————— 0 | SYNC BYTE (376 OR 375)* |
| 1 | 0 ——————— 0 | CYLINDER # (MSBYTE) |
| 2 | 0 ——————— 0 | CYLINDER # (LSBYTE) |
| 3 | 0 ——————— 0 | HEAD # |
| 4 | 0 ——————— 0 | SECTOR # |
| 5 | 0 ——————— 0 | FLAG (0) |

*375 (OCTAL) WILL ONLY OCCUR ON ANY SECTOR OF A RELOCATED TRACK

128-20

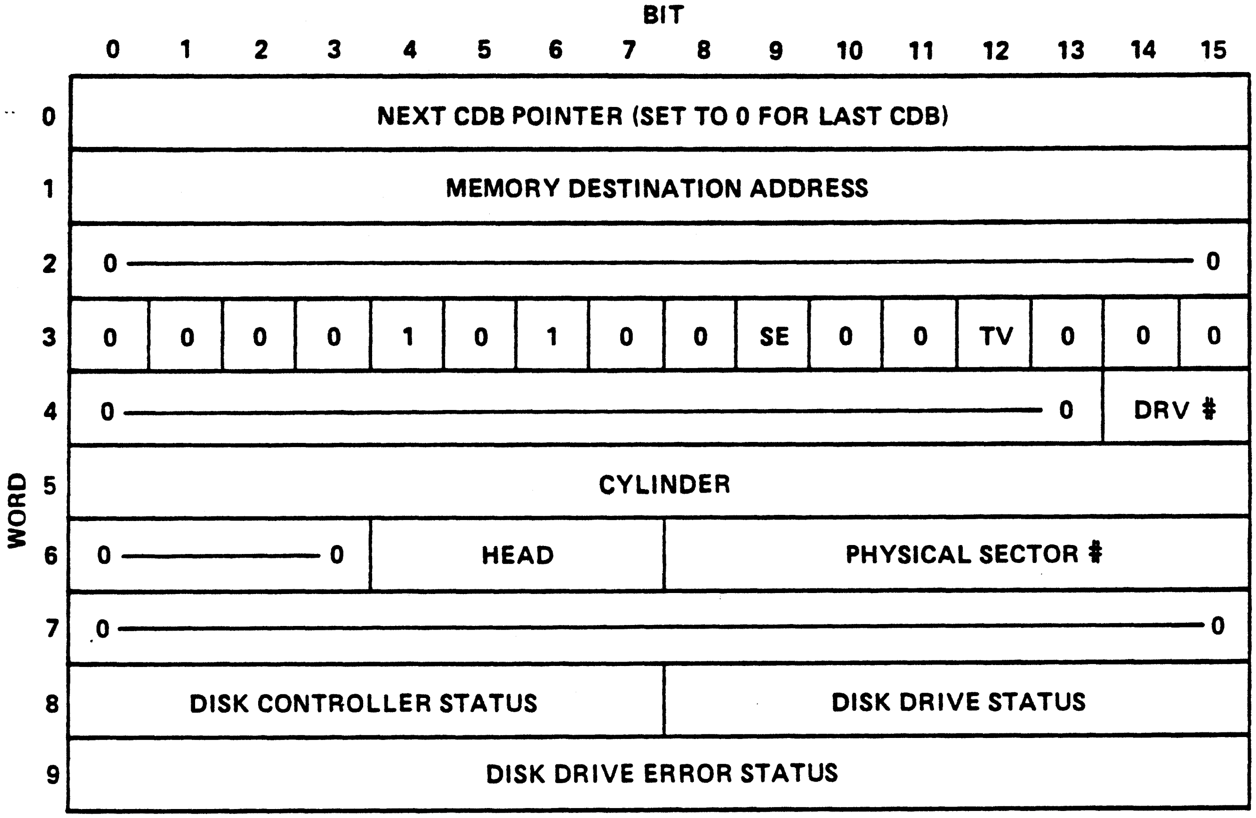
**Figure 3-31. 6-Byte Header Field Dumped by the Read Header Command**

### 3.4.16 Read Physical Sector Command

The Read Physical Sector command is a diagnostic command that reads the data field of the selected physical sector. If the track verify (TV) bit is set, the hard disk controller (HDC) reads a number of headers on the specified track to verify that it is on the correct track. The HDC then does a physical search for the specified physical sector and transfers the data field to the controller buffer, which in turn transfers it to the memory destination address in main memory. Figure 3-32 illustrates the Read Physical Sector command.

**NOTE**

> Physical sector 1 corresponds to the first sector after the Index mark.



<table>
<tr><th>WORD</th><th colspan="16">BIT</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>0</td><td colspan="16">NEXT CDB POINTER (SET TO 0 FOR LAST CDB)</td></tr>
<tr><td>1</td><td colspan="16">MEMORY DESTINATION ADDRESS</td></tr>
<tr><td>2</td><td colspan="16">0 ——— 0</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>SE</td><td>0</td><td>0</td><td>TV</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>4</td><td colspan="14">0 ——— 0</td><td colspan="2">DRV #</td></tr>
<tr><td>5</td><td colspan="16">CYLINDER</td></tr>
<tr><td>6</td><td colspan="4">0 ——— 0</td><td colspan="4">HEAD</td><td colspan="8">PHYSICAL SECTOR #</td></tr>
<tr><td>7</td><td colspan="16">0 ——— 0</td></tr>
<tr><td>8</td><td colspan="8">DISK CONTROLLER STATUS</td><td colspan="8">DISK DRIVE STATUS</td></tr>
<tr><td>9</td><td colspan="16">DISK DRIVE ERROR STATUS</td></tr>
</table>

128-21

**Figure 3-32. Read Physical Sector Command**

### 3.4.17 Read Hard Disk Controller (HDC) Registers Command

The Read HDC Registers command is a diagnostic command that dumps the seven HDC registers to the Memory Destination Address in main memory. Figure 3-33 illustrates the Read HDC Registers command, and Figure 3-34 illustrates the 7-word HDC registers.

### 3.4.18 Write Hard Disk Controller (HDC) Registers Command

The Write HDC Registers command is a diagnostic command that writes to the seven HDC registers. The 7-word HDC register set is transferred from the memory source address to the HDC.

Care should be taken when writing to the Status/Command register, because it is a read/write register and undefined HDC operation can be initiated by writing to the command field.

Figures 3-33 and 3-34 also apply to the Write HDC Register command and its 7-word HDC registers.

| | BIT | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 0 — | | | | | | | | | | | | | | | 0 |
| 1 | MEMORY DESTINATION ADDRESS | | | | | | | | | | | | | | | |
| 2 | 0 — | | | | | | | | | | | | | | | 0 |
| 3 | 0 | 0 | 1 | 0 | 0 | 0 | * | 1 | 0 — | | | | | | | 0 |
| 4 | 0 — | | | | | | | | | | | | | | | 0 |
| 5 | 0 — | | | | | | | | | | | | | | | 0 |
| 6 | 0 — | | | | | | | | | | | | | | | 0 |
| 7 | 0 — | | | | | | | | | | | | | | | 0 |
| 8 | DISK CONTROLLER STATUS | | | | | | | | 0 — | | | | | | | 0 |
| 9 | 0 — | | | | | | | | | | | | | | | 0 |

*1=READ HDC REGISTERS COMMAND
0=WRITE HDC REGISTERS COMMAND

128-22

**Figure 3-33. Read/Write Hard Disk Controller (HDC) Registers Command**

| | BIT | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WORD | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | STATUS/CMD REGISTER | | | | | | | | | | | | | | | |
| 1 | MODE REGISTER | | | | | | | | | | | | | | | |
| 2 | NEXT BLK PTR LOW | | | | | | | | | | | | | | | |
| 3 | NEXT BLK PTR HIGH | | | | | | | | | | | | | | | |
| 4 | STAT RESULT PTR LOW | | | | | | | | | | | | | | | |
| 5 | STAT RESULT PTR HIGH | | | | | | | | | | | | | | | |
| 6 | STAT RESULT LENGTH | | | | | | | | | | | | | | | |

128-23

**Figure 3-34. 7-Word Hard Disk Controller (HDC) Register Set**

# Section 4

# INSTALLATION

This section provides the information and instructions required to install the LOTUS 800 Controller in a POINT 4 MARK system that uses ESDI hard disk drives. Installation of the controller should be done by a technician who is familiar with the system hardware as well as the disk and tape drives that are installed.

Installation of the LOTUS 800 Controller includes the following steps:

- Inspecting the controller
- Accessing the system
- Configuring the controller
- Configuring the drive(s)
- Installing the controller
- Running diagnostics

These steps are described in the following subsections. Before beginning, have at hand both Phillips and flat-head screwdrivers. Technicians who are not well acquainted with MARK computers, should also have the following reference documents available: the IRIS R9 User Reference Manual, the appropriate System Installation and Maintenance Manual, and the LOTUS 800 Controller Diagnostic Document. These documents provide comprehensive information about particular procedures and components that are part of the installation process.

## 4.1 INSPECTING THE CONTROLLER

If the controller has not already been removed from the shipping carton, inspect the carton for damage caused by dropping, puncturing, or crushing. If damage is evident, contact the shipping company and the POINT 4 Data Corporation Sales Representative for further instructions.

Once unpacked, check the contents against the packing slip to ensure that all listed items are present. Visually inspect the board, the cables, and the connectors to make certain that all components have arrived intact. Again, if damage is evident, report it to the shipping company and to the POINT 4 Sales Representative.

## 4.2 ACCESSING THE SYSTEM

The preliminary steps listed below are necessary to gain access to the interior of the system and to prepare it for the installation of the LOTUS 800 Controller.

1. If the system is currently in use, shut down and back up the system (refer to the IRIS R9 User Reference Manual).

2. Make certain that the control panel keyswitch and the main power switch of the computer are turned OFF.

3. Disconnect the AC power cord.

4. Move the system to an open space.

5. Remove the exterior panels required to gain access to the interior of the computer (refer to the appropriate System Installation and Maintenance Manual).

6. If a controller is currently installed and is to be removed and replaced, remove it as follows:

   a. If present, disconnect cables that prohibit the removal of the controller board from the card cage. Be sure to note where each cable is connected so that it can be reconnected when the new controller is in place.

   b. Pull up the tabs that are located on the outside edges of the controller board; pull on them to release the controller from the backplane; and slide the controller out of the guide rails.

## 4.3 CONFIGURING THE CONTROLLER

Before the LOTUS 800 Controller can be installed, it must be configured for the particular hardware system, disk, and tape drives. The controller is shipped set to the standard configuration.

Figure 4-1 illustrates the LOTUS 800 Controller. The subsections that follow provide information on the switch settings and the controls used in configuration.

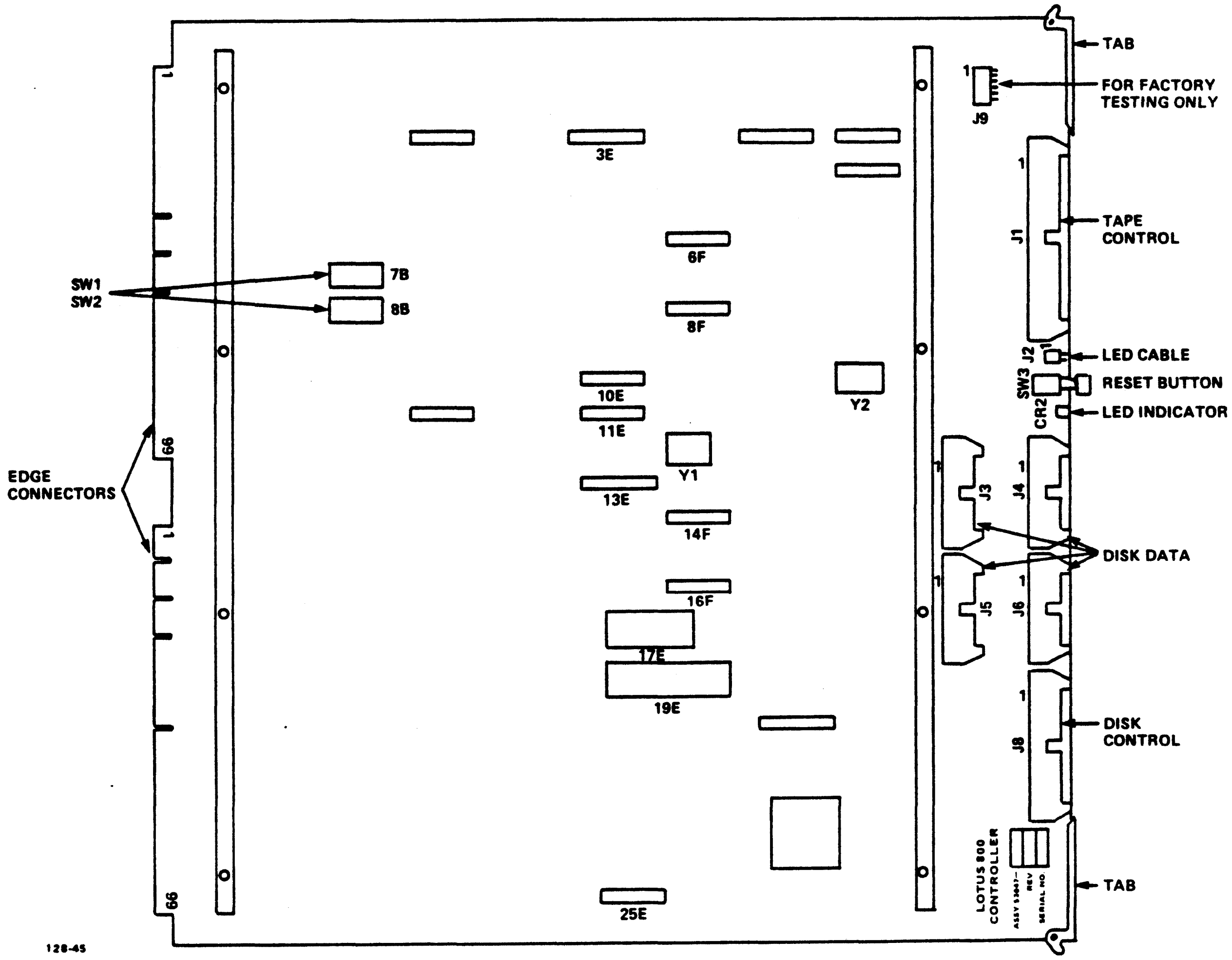


**Figure 4-1. LOTUS 800 Controller**

### 4.3.1 Disk Device Code and Error Correction Select Switch

Switch SW1 is an 8-position switch that is used to set the disk device and error correction codes. It is located on the controller at location 7B.

The default disk device code is 45 (octal); it is set with keys 1 through 6. The default Error Correction Code (ECC) is single-burst Reed Solomon; it is set with key 8. After power-up or an IORST instruction, double-burst Reed Solomon can be implemented through software by using the Write Drive Parameter Block command (see Section 3.4.10). Figure 4-2 illustrates the default setting for SW1.

### 4.3.2 Tape Device Code Select Switch

Switch SW2 is an 8-position switch that is used to set the tape device code. It is located on the controller at location 8B. The default tape device code is 42 (octal). Figure 4-2 illustrates the default setting for SW2.

### 4.3.3 Reset

Switch SW3 is a reset switch or button that is pressed to reset the controller if self-test fails at power-up. It is located on the front edge of the controller board.

Self-test failure is indicated if the LED adjacent to SW3 remains lighted after power-up. Normally, the LED turns on immediately after power-up and remains on until self-test has completed successfully and all drives have spun up; then it goes off. If the LED remains on, the power-up procedure should be repeated. If self-test fails a second time, check SW1, SW2, and disk drive jumper settings.

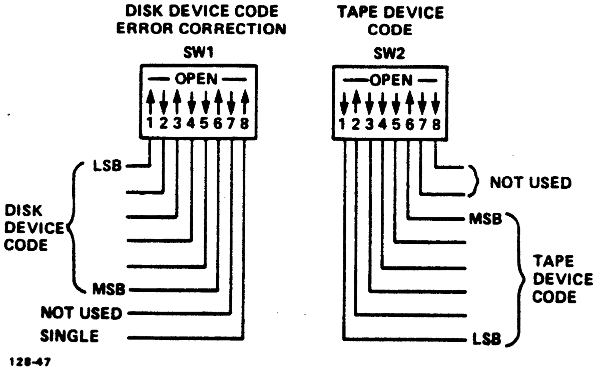


**Figure 4-2. LOTUS 800 Device Code and Switch Settings**

## 4.4 CONFIGURING THE DRIVE

The LOTUS 800 Controller supports a maximum of four 5-1/4-inch ESDI hard disk drives. These drives must also be configured to be compatible with the controller. The information in this section assumes that the system is using Micropolis drives in 170 or 382MB capacities. If other drives are used, refer to documentation for that drive. Figure 4-3 illustrates the component side of each disk drive where the configuration process occurs. The following subsections provide information on configuring the disk drive. If disk drives are added or need to be reconfigured, refer to the appropriate System Installation and Maintenance Manual.

### 4.4.1 Terminating the Last Disk Drive

The last physical drive on the 34-pin daisy-chained control cable must have a terminator attached to it. All other drives must have the terminators removed. The terminator is located at RN1.

### 4.4.2 Installing the Jumpers

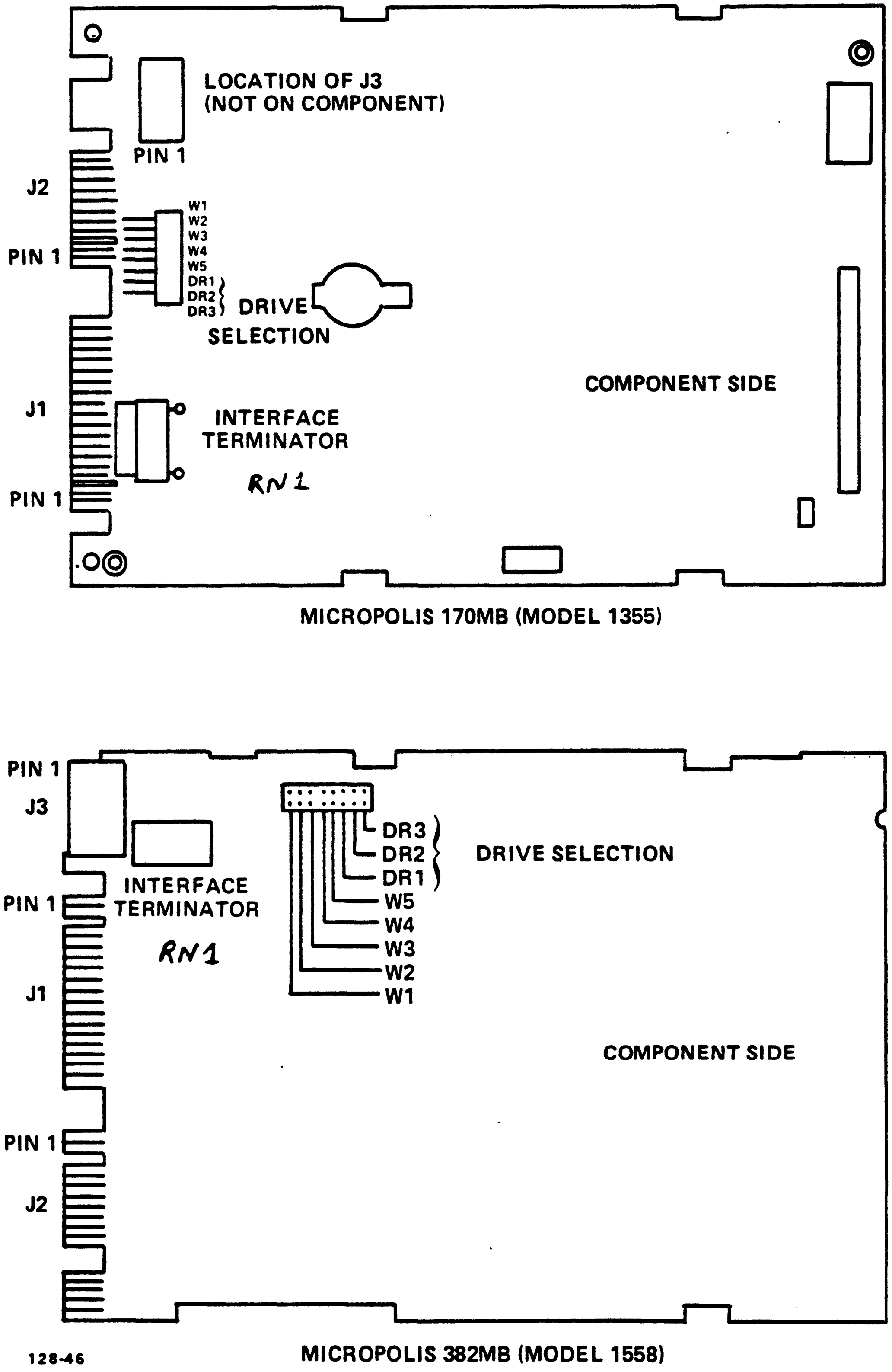
Jumpers must be positioned to set the options for drive select and spindle control.

On all drives, set the spindle control option by installing a jumper at W5. Remove jumpers at locations W1 through W4.

On each drive, position the jumper(s) for the drive select option using one of the following as appropriate:

| | Jumper Location | | |
|---|---|---|---|
| Drive | DA3 | DA2 | DA1 |
| 0 | --- | --- | X* |
| 1 | --- | X | --- |
| 2 | --- | X | X |
| 3 | X | --- | --- |

X = jumper installed



MICROPOLIS 170MB (MODEL 1355)

MICROPOLIS 382MB (MODEL 1558)

Figure 4-3. Disk Drive Configuration Locations

## 4.5 INSTALLING THE CONTROLLER

The following instructions assume that the preliminary steps in Section 4.2 have been performed, and that **the power is turned OFF and the AC power cord is disconnected.**

1. Place the controller component side up on a flat surface and press down on all socketed components.

2. Hold the controller by its side tabs, slide it into the guide rails of the selected slot of the card cage; push until the board snaps into the backplane connectors; and close the side tabs.

3. Connect and route the cables according to the general instructions provided below. When connecting cables, make certain that pin 1 of each cable connector aligns with pin 1 of the matching edge or power connector. For specific instructions, refer to the appropriate System Installation and Maintenance Manual as routing cables differs from one system to another.

| Cable | Connector Locations | |
|---|---|---|
| Disk Control,<br>34-pin<br>(daisy chain) | Controller J8 | Disk drive J1 |
| Disk Data,<br>20-pin | | |
| Drive 0 or 1 | Controller J4 or J6 | Disk drive J2 |
| Drive 2 or 3 | Controller J3 or J5 | Disk drive J2 |
| Tape Control,<br>50-pin | Controller J1 | Tape drive, rear |
| Power, 4-pin | | |
| Drive 0 or 1 | Refer to appropriate | Disk drive J3 |
| Drive 2 or 3 | I&M Manual | Disk drive J3 |
| LED | Controller J2 | Control panel J1 |

4. Power up the system as described in the appropriate System Installation and Maintenance Manual.

5. Ensure that the diagnostic self-test that occurs at power-up has completed successfully. A successful self-test is indicated when the LED, adjacent to SW3, goes out.

## 4.6 RUNNING DIAGNOSTICS

POINT 4 recommends that testing be continued by running the LOTUS 800 Controller diagnostics. These diagnostics are available as stand-alone programs on streaming tape or on the IRIS Operating System. For more information on the diagnostics and instructions on how to use them, refer to the LOTUS 800 Controller Diagnostic Document.

# Section 5

# OPERATING PROCEDURES

TO BE SUPPLIED

## Appendix A

## SELF-TEST ERROR CODES

If the Failed Self-Test bit in the Disk Controller Status word indicates an error, an error code, which indicates the nature of the failure, is given in bits 10-15 of word 9 (disk drive error status) of the command descriptor block. The self-test error codes are defined as follows:

| Code | Definition |
|---|---|
| 1 | Memory address register LS byte zero test error |
| 2 | Memory address register MS byte zero test error |
| 3 | Memory address register LS byte one's test error |
| 4 | Memory address register MS byte one's test error |
| 5 | Buffer test LS byte error |
| 6 | Buffer test MS byte error |
| 7 | HDC 9590 ID error |
| 10 | HDC not in idle after reset |
| 11 | HDC CFT error after reset (reset not complete) |
| 12 | HDC CF bit not set |
| 13 | HDC null NBP: A resume or start chain command was given with NBP=0 |
| 14 | SRA overflow: SRB buffer overflow, SRL reg=0 |
| 15 | HDC forced idle: Execution terminated at end of current CDB due to Idle command issued during current CDB |
| 16 | HDC SRB error: SRB written when executing CDB with SSRB set; this option not used |
| 17 | HDC DM timeout: This option not used |
| 20 | HDC Illegal Resume: An attempt was made to resume an unresumable CDB |
| 21 | HDC reset complete in CDB_CMPLT |
| 22 | HDC wait stop: HDC stopped execution at end of current CDB with wait option set |
| 23 | HDC reserved status: Illegal CFT generated |
| 24 | HDC SRB: Multi-record overflow |
| 25 | HDC SRB: Sector size mismatch (floppy) |
| 26 | HDC SRB: ECC not selected |
| 27 | HDC SRB: End Data Map; this option not used |
| 30 | HDC SRB: Reserved; illegal SRB |
| 31 | HDC SRB: Data not corrected |
| 32 | HDC CFT error: Drive Parameter/Spinup CDB CFT error |
| 33 | Drv0 ESDI parameter get error |
| 34 | Drv1 ESDI parameter get error |
| 35 | Drv2 ESDI parameter get error |
| 36 | Drv3 ESDI parameter get error |
| 37 | Load Drive Parameter remap failed |
| 40 | ESDI gap calculation error (too big or negative) |
| 41 | Selected unformatted bytes/sector too small for sector overhead |

## COMMENT SHEET

MANUAL TITLE: LOTUS 800 Controller Manual

PUBLICATION NO. HM-128-0070 REVISION 01

FROM: NAME/COMPANY:__________

BUSINESS ADDRESS:__________

CITY/STATE/ZIP:__________

COMMENTS: Your evaluation of this manual will be appreciated by POINT 4 Data Corporation. Notation of any errors, suggested additions or deletions, or general comments may be made below. Please include page number references where appropriate.

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
Fold on Dotted Lines and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 1458 TUSTIN, CA

POSTAGE WILL BE PAID BY ADDRESSEE

**POINT 4 Data Corporation**
**PUBLICATIONS DEPARTMENT**
**15442 Del Amo Avenue**
**Tustin, CA 92680**

CUT ON T LINE