# EMULEX SCSI

# DISK CONTROLLER

# PROGRAMMING REFERENCE MANUAL


3545 Harbor Boulevard
Costa Mesa, California 92626
(714) 662-5600  TWX 910-595-2521

MD2352501-00 Rev E
March, 1988

Printed in U.S.A.

Table of Contents

**SIX SCSI COMMAND DESCRIPTIONS**

LIST OF APPENDICES

## 1.1 INTRODUCTION

The Emulex family of SCSI disk controllers are designed to interface Small Computer System Interface (SCSI) host adapters and other peripheral controllers to 5.25-inch Winchester disk drives. These disk drives support one of two industry-standard disk drive interfaces:

- The Enhanced Small Device Interface (ESDI)
- The Storage Module Drive (SMD) interface

This manual provides programming information for Emulex SCSI disk controllers. Currently, only the ESDI interface is supported; SMD programming information will be added when it is available.

In addition to this manual, Emulex also offers technical manuals that document installation procedures and describe the hardware for each SCSI disk controller in the Emulex family. The other manuals are listed in subsection 1.1.1.

It is assumed the user's system configuration includes a well-defined SCSI host adapter and the appropriate driver for interfacing to the disk controller. The contents of the six sections and two appendices in this manual are described briefly below.

Section 1 General Description: This section contains an overview of the controller with regards to its compatibility with the defined SCSI standard.

Section 2 Controller Architecture: This section describes the architecture of the controller.

Section 3 Disk Drive Organization: This section describes controller disk operations, including sector and track format and bad sector format operations.

Section 4 Controller Initialization and Self-Test Procedures: This section describes controller initialization and self-test procedures.

Section 5 SCSI Bus Operations: This section describes controller reset and initialization procedures, and SCSI bus phases, conditions, timing, and command queuing.

Section 6 SCSI Command Descriptions: This section describes SCSI commands and their corresponding Command Descriptor Blocks which are supported by the controller. It includes an overview of the SCSI command execution sequence.

Appendix A SCSI Error Codes: This section lists the error codes by the different classes of error conditions that can occur during controller disk operations.

Appendix B SCSI Sense Keys: This section describes the sense keys.

For reference convenience, Section 1 is divided into three subsections, as listed in the following table:

| Subsection | Title |
|---|---|
| 1.1 | Introduction |
| 1.2 | Controller SCSI Bus Features |
| 1.3 | Compatibility |

### 1.1.1 RELATED DOCUMENTS

This manual is designed to be used by system programmers who are writing operating system drivers and support utilities. This manual assumes familiarity with the SCSI standard, the SCSI Common Command Set (CCS) for direct-access devices, and the ESDI disk drive interface specification.

The Small Computer System Interface (SCSI) command set for the controller is based on the ANSC X3T9.2/82-2 SCSI Specification, Revision 17B (16 DEC 1985). Copies of the ANSC SCSI Specification can be obtained from the following publisher:

American National Standard Task Group X3T9.2/82-2
Computer and Business Equipment Manufacturers Association
311 First Street, NW Suite 500
Washington, DC 20001

The CCS document is an industry standard that documents a common command set of SCSI commands for direct-access devices.

The ESDI interface standard for 5.25-inch Winchester disk drives is described in the Enhanced Small Device Interface specification. This specification is available from:

Maxtor Corporation
c/o ESDI Committee
150 River Oaks Pkwy.
San Jose, CA 95134

The other manuals in the set which document the Emulex SCSI ESDI disk controllers describe the physical and electrical characteristics of the disk controllers. Currently the following manuals are available from Emulex:

- MD23/24 Installation and Technical Reference Manual (P/N MD2351001)
- MD21/S2 Installation and Technical Reference Manual (P/N MD2151002)

### 1.1.2 PROGRAMMING MANUAL CONVENTIONS

To avoid possible confusion with other uses of the same words, throughout this manual we use the following conventions:

- All SCSI commands (such as **READ**, **MODE SELECT**, and **INQUIRY**) and diagnostic subcommands (such as **READ BAD SECTOR FILE** and **WRITE LONG**) are printed in uppercase boldface.
- All SCSI status and error messages (such as CHECK CONDITION and DRIVE NOT READY) are printed in uppercase.
- All SCSI bus phases and conditions (such as Arbitration Phase) and SCSI Command Descriptor Block names (such as Extended Sense Byte) are printed in initial caps.
- All SCSI command and message codes are given in their hexadecimal values.

## 1.2 CONTROLLER SCSI BUS FEATURES

In combination with an independent host adapter, the SCSI bus allows a wide variety of computers to interface with the controller. Compatible computers include IBM Personal Computer systems such as the IBM PC-XT and Multibus-based computers. Up to eight bus devices, in any combination of host systems and intelligent controllers, can be supported by the SCSI bus. Each controller can support up to four drives (depending on the controller model), providing a low-cost, compact storage subsystem in a SCSI environment.

The controller's architecture and supported SCSI features make it an ideal building block for use by OEMs and system integrators. The controller supports a powerful set of SCSI commands. By using those commands, an efficient multiple-Initiator configuration can be constructed with the support of the disconnect function. (The disconnect function allows the controller, when it is performing a time-consuming task, to release the SCSI bus temporarily and reconnect at a later time when the task is complete.) The controller may be considered a SCSI extended-bus device because it uses all standard and extended SCSI commands. It also supports the SCSI Common Command Set defined for direct-access devices.

The controller features are summarized below. More details on these features are given in subsequent sections.

- SCSI Implementation based on ANSC X3T9.2/82-2 Rev 17 B (16 DEC 85) Specification
- Support of standard 5.25-inch Winchester disk drives with ESDI interfaces
- Powerful SCSI command set including:

  -Common Commands
  -Random Access Device Commands
  -Linked Command Support

- Support of disconnect/reconnect function
- Hard Reset
- Buffered operation to optimize performance
- On-board self-test and diagnostic procedures

## 1.3 COMPATIBILITY

Compatibility of the controller with specific disk drives and SCSI host adapter systems is described in the following subsections.

### 1.3.1 SCSI COMPATIBILITY

The controller contains an on-board SCSI protocol controller that controls SCSI protocol and the SCSI bus. The controller supports SCSI arbitration and reselection capabilities.

The hexadecimal codes for the SCSI commands supported by the controller are shown in Table 1-1. Detailed command descriptions are given in Section 6.

Table 1-1. Controller SCSI Command Set

| Group 0 Command | Hex Code | Group 0 Command | Hex Code |
|---|---|---|---|
| **TEST DRIVE READY** | 00 | **REZERO UNIT** | 01 |
| **REQUEST SENSE** | 03 | **FORMAT UNIT** | 04 |
| **REASSIGN BLOCK** | 07 | **READ** | 08 |
| **WRITE** | 0A | **SEEK** | 0B |
| **INQUIRY** | 12 | **MODE SELECT** | 15 |
| **RESERVE UNIT** | 16 | **RELEASE UNIT** | 17 |
| **MODE SENSE** | 1A | **RECEIVE DIAGNOSTIC** | 1C |
| **SEND DIAGNOSTIC** | 1D | **START/STOP UNIT** | 1B |
| Group 1 Command | Hex Code | Group 1 Command | Hex Code |
| **READ BUFFER** | 3C | **READ DEFECT LIST** | 37 |
| **READ CAPACITY** | 25 | **SEEK (EXTENDED)** | 2B |
| **READ (EXTENDED)** | 28 | **VERIFY** | 2F |
| **WRITE (EXTENDED)** | 2A | **WRITE BUFFER** | 3B |
| Group 6 Command | Hex Code | | |
| **READ REVISION LEVEL** | C1 | | |
| Group 7 Command | Hex Code | Group 7 Command | Hex Code |
| **FORMAT TRACK** | E4 | **READ LONG** | E8 |
| **WRITE LONG** | EA | | |

### 1.3.2 ESDI DISK DRIVE COMPATIBILITY

The MD21 disk controller supports two ESDI 5.25-inch Winchester disk drives; the MD23/24 disk controller supports up to four ESDI 5.25-inch Winchester disk drives. Both controllers support ESDI disk drives that have clocks up to 15 Megahertz and can be used with hard-sectored, soft-sectored, and embedded servo disk drives.

BLANK

## 2.1 OVERVIEW

This section describes Emulex SCSI Disk Controller architecture. For reference convenience, this section is divided into two subsections, as listed in the following table:

| Subsection | Title |
|---|---|
| 2.1 | Overview |
| 2.2 | Controller Elements |

## 2.2 CONTROLLER ELEMENTS

Figure 2-1 is a block diagram that shows the major functional elements of the controller. The controller is organized around the 8031 microprocessor, the SCSI protocol controller, the disk formatter, and the buffer controller. The disk formatter and the buffer controller are custom VLSI chips designed by Emulex.

Two buses are used in the controller: the data bus and the microprocessor bus.

The data bus is connected directly to the disk formatter, SCSI protocol controller, buffer memory, and buffer controller. The buffer controller is connected directly to the data bus and the microprocessor bus, providing an interface between them. The buffer controller provides a data path between the 8031 microprocessor, the buffer memory, the disk formatter, and SCSI protocol controller.

The microprocessor bus provides a path for transmission of control and status information between the 8031 microprocessor, EPROM, buffer controller, and disk interface. This bus is completely separate from the data bus. The microprocessor may access the data bus via the buffer controller interface.

The controller SCSI Interface is implemented using a single LSI chip on the controller. In response to commands from the Initiator, the chip establishes and monitors SCSI bus phases appropriate to the command. It performs SCSI signal control and timing functions.

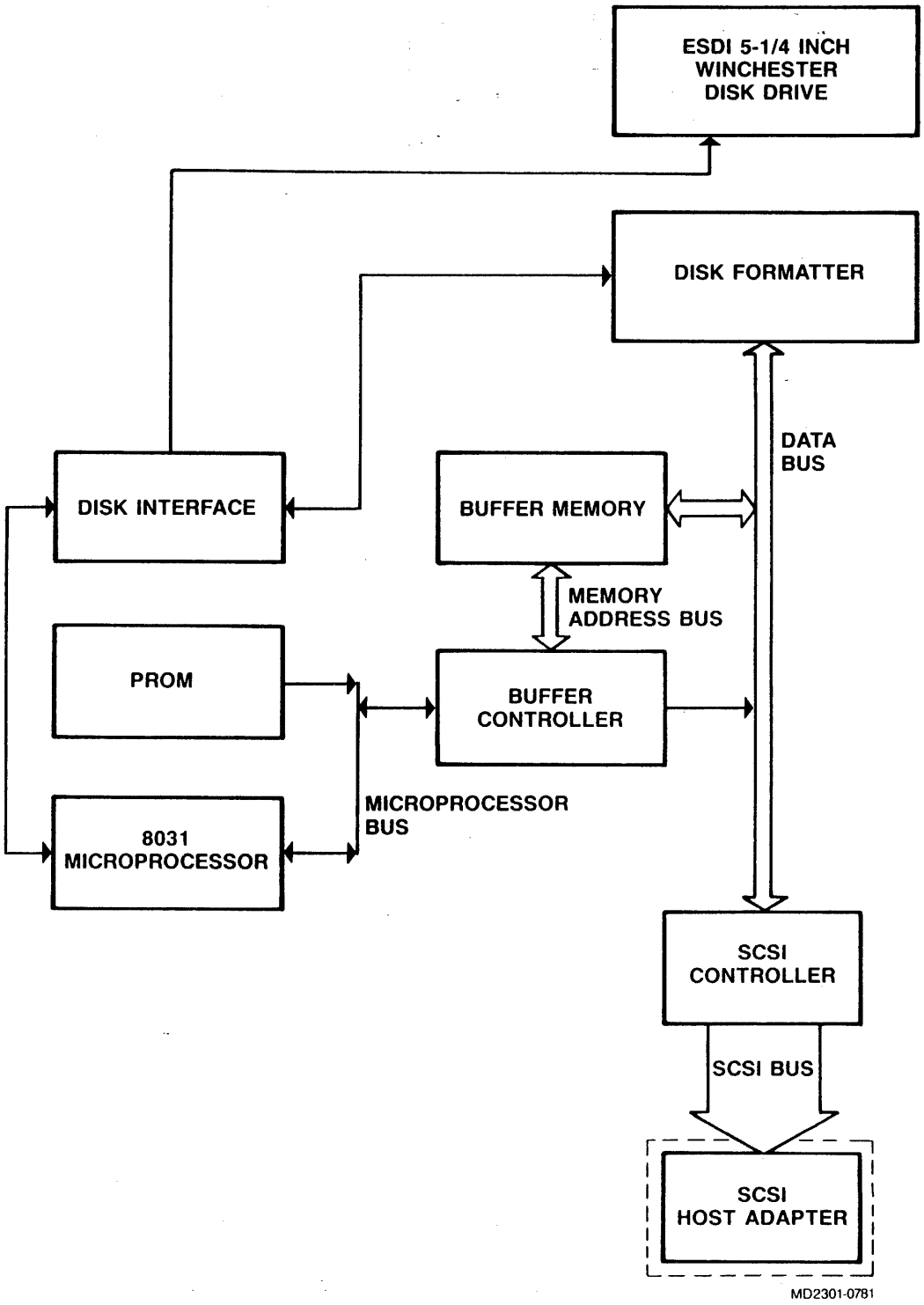



Figure 2-1. Controller Block Diagram

### 2.2.1 8031 MICROPROCESSOR

The 8031 microprocessor, in conjunction with the disk formatter, controls all disk drive operations. These disk operations include drive control, head positioning, and reading drive status.

During disk operations, the disk formatter controls formatting of the data that is written to, and read from, the disk drive. The 8031 microprocessor issues read and write commands that are executed by the disk formatter. All read and write commands involve operations only on a single data block (256 or 512 bytes).

### 2.2.2 DISK FORMATTER

The disk formatter is a 40-pin VLSI IC fabricated with CMOS gate array technology. This circuit, in conjunction with the 8031 microprocessor, handles the read and write operations of the disk drives.

### 2.2.3 BUFFER CONTROLLER

The buffer controller is a 68-pin VLSI IC fabricated with CMOS gate-array technology. The circuit is basically a three-channel DMA controller. The buffer controller controls data movement in or out of a dynamic buffer memory and provides the interface between the microprocessor bus and the data bus.

The buffer controller circuit provides the address and control for multiple controller activities that access a dynamic buffer memory. The buffer controller performs the following operations:

- Handles addressing and control operations for the disk formatter
- Handles addressing and control operations for the SCSI protocol controller
- Handles dynamic memory timing and refresh
- Performs parity checking and generation for the buffer memory
- Connects the microprocessor bus to the data bus
- Decodes the microprocessor address for the buffer memory and the internal input/output (I/O) space in the controller
- Determines priority of buffer memory access

BLANK

## 3.1 CYLINDER ORGANIZATION

Figure 3-1 shows the organization of cylinders on the disk drive.

Details of sector format, cylinder 0 format, and defect list format are discussed in the following subsections.

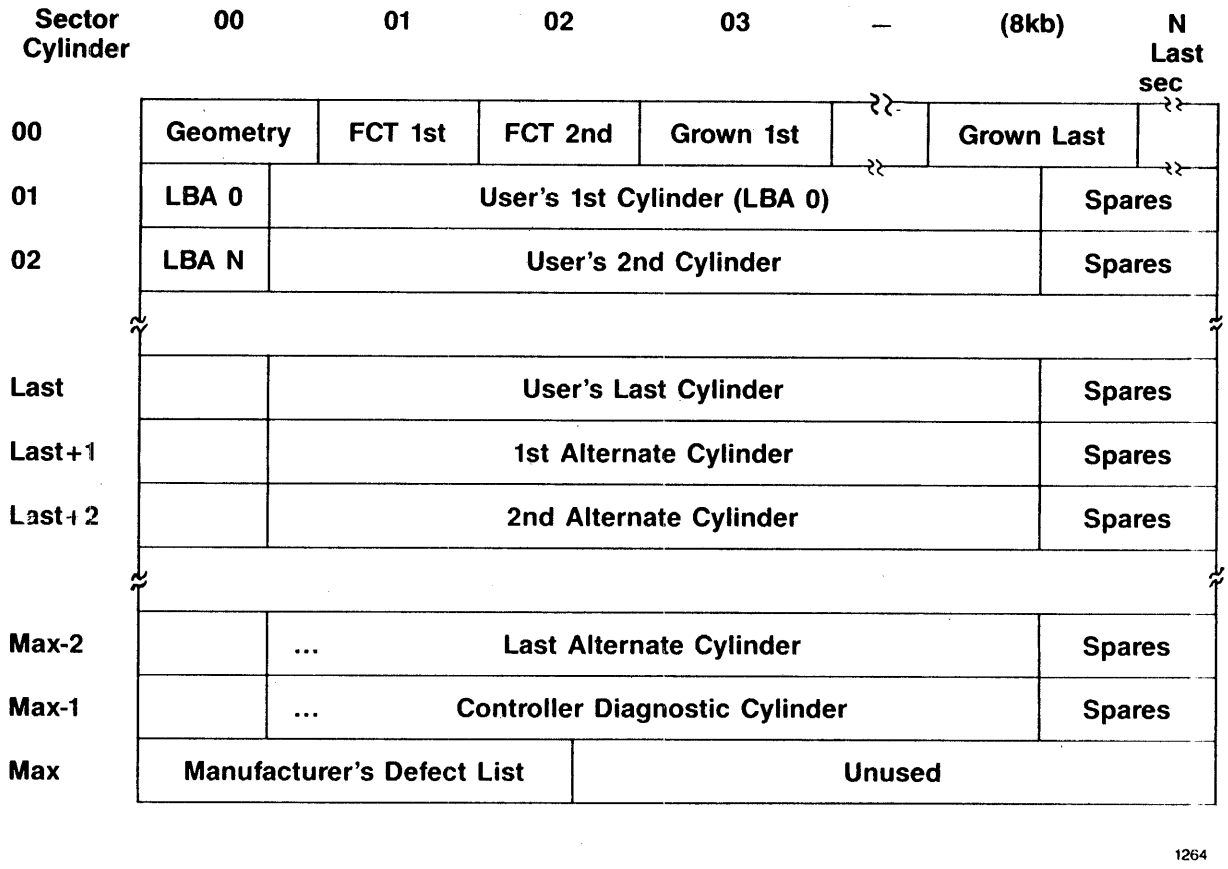
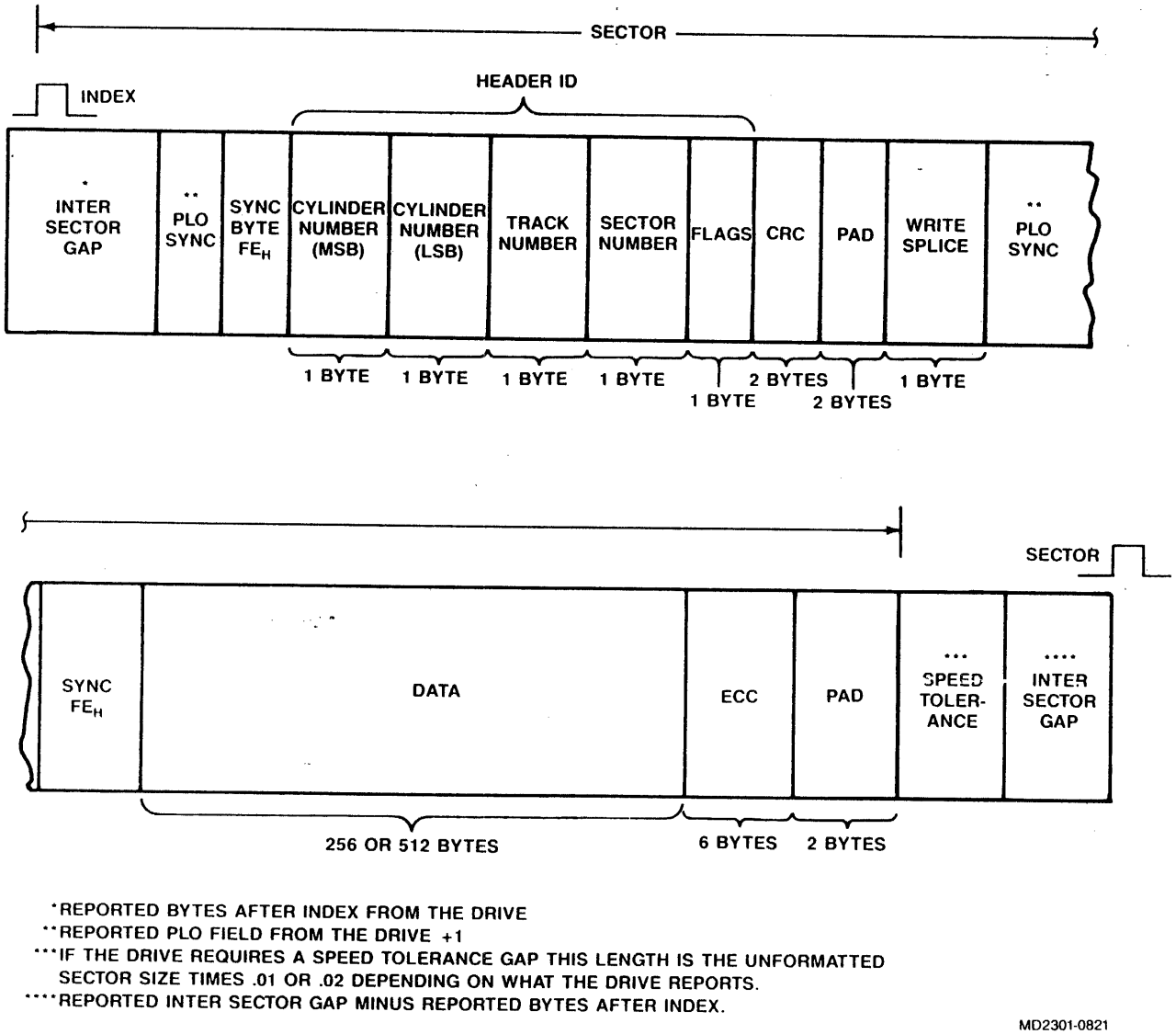



Figure 3-1. Organization of Cylinders

## 3.2 SECTOR AND TRACK FORMAT

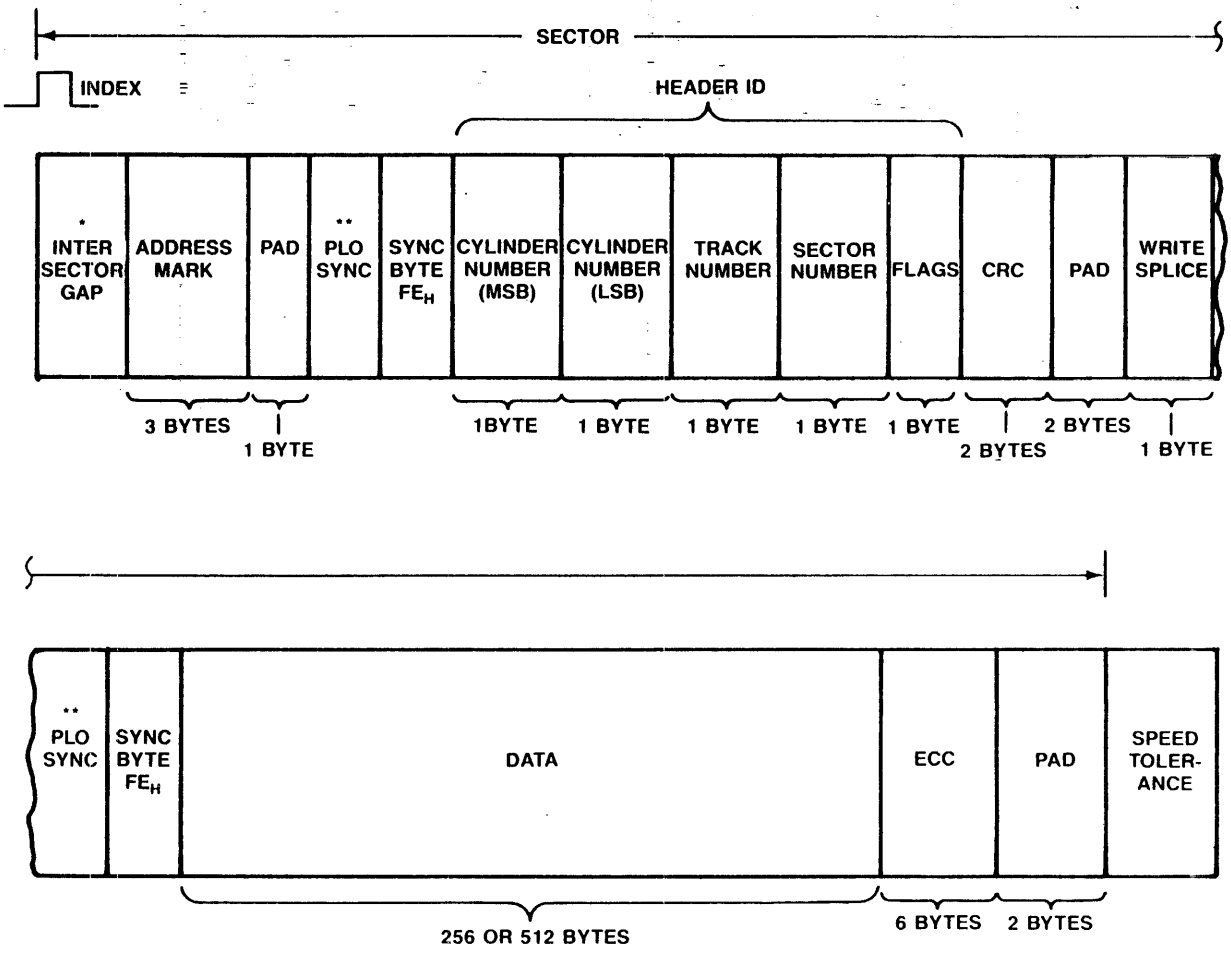
For hard-sectored disk drives, the controller formats each sector as shown in Figure 3-2. For soft-sectored disk drives, the controller formats each sector as shown in Figure 3-3.




*REPORTED BYTES AFTER INDEX FROM THE DRIVE
**REPORTED PLO FIELD FROM THE DRIVE +1
***IF THE DRIVE REQUIRES A SPEED TOLERANCE GAP THIS LENGTH IS THE UNFORMATTED SECTOR SIZE TIMES .01 OR .02 DEPENDING ON WHAT THE DRIVE REPORTS.
****REPORTED INTER SECTOR GAP MINUS REPORTED BYTES AFTER INDEX.

MD2301-0821

Figure 3-2. Sector Format for Hard-Sectored Disk Drives




*REPORTED BYTES AFTER INDEX FROM THE DRIVE
**REPORTED PLO FIELD FROM THE DRIVE +1.

MD2301-0822

Figure 3-3. Sector Format for Soft-Sectored Disk Drives

The **Inter Sector Gap (ISG)** varies between disks and is returned as Reported Bytes after Index by the disk drive over the serial communicatons interface.

The **PLO Sync Byte** varies between disks and is returned by the disk drive over the serial communications interface. The value used is PLO + 1 because the reported value is the minimum required after Read Gate has been asserted.

The size of the **speed tolerance gap** depends on values returned by the drive over the serial communications interface, as follows:

If Cnfig(14) and -Cnfig(11), gap = unformatted sector size x .01
If Cnfig(14) and Cnfig(11), gap = unformatted sector size x .02
If -Cnfig(14), gap = 0

The header ID on each track has the format shown below.

**Disk Sector Header Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Cylinder Number (MSB) | | | | | | | |
| 1 | Cylinder Number (LSB) | | | | | | | |
| 2 | Head Number | | | | | | | |
| 3 | Sector Number | | | | | | | |
| 4 | DT | SP | RP | 0 | 0 | 0 | 0 | 0 |

The **cylinder number, head number,** and **sector number** are physical addresses. The interleave factor may be specified during a format operation (see the **FORMAT UNIT** command).

The **DT flag** indicates that the entire track has been replaced. If the DT flag is set, the cylinder number and head number in the header contain the address of the new track.

The **SP flag** indicates that the sector is a spare. The number of spare sectors per track is specified at format time. These sectors are not available from any SCSI command.

The **RP flag** indicates that the sector has been replaced. If the RP flag is set, the sector number in the header contains the address of the new sector.

### 3.2.1 LOGICAL TO PHYSICAL CONVERSION

The following algorithms are used to convert logical block addresses to and from physical disk addresses.

The following definitions are used:

$B_l$ = Logical Block Number of operation

$S_t$ = Physical Sectors per Track - Spare Sectors per Track
$H_l$ = Logical Number of Heads
$C_l$ = Number of Data Cylinders

$S_p$ = Destination Physical Sector
$H_p$ = Destination Physical Head
$C_p$ = Destination Physical Cylinder

The following three formulae convert a logical block number to a cylinder-head-sector address:

$$S_p = B_l \text{ Modulo } S_t$$

$$H_p = ((B_l - S_p) / S_t)) \text{ Modulo } H_l$$

$$C_p = \text{INTEGER } (((B_l - S_p) / S_t) / H_l) + 1$$

The following formula converts a cylinder-head-sector address into a logical block number:

$$B_l = ((((C_p - 1) * H_l) + H_p) * S_t) + S_p$$

## 3.3 CYLINDER ZERO FORMAT

The format for cylinder 0 is repeated for each head. The format is shown in Figure 3-1.

The following subsections discuss cylinder 0 in detail.

### 3.3.1 GEOMETRY

The first sector of cylinder 0 contains information about the geometry of the drive. Table 3-1 descibes each byte in the geometry sector. The far left column indicates the page number in the **FORMAT UNIT** command in which each piece of information is found.

Table 3-1. Sector 0 Geometry Information

| Page | Byte | Description | Contents |
|---|---|---|---|
| | 00-0A | Validity Code | 'Emulex MDxx' |
| | 0B-0F | Version at Format | 'DnnRv' |
| 4 | 10 | Heads | Last Track |
| 4 | 11-12 | Cylinders | Last Cylinder |
| 3 | 13 | Sectors per Track | Last Sector |
| 3 | 14-15 | Bytes per Sector | Physical size |
| 3 | 16-17 | Logical Sector Size | Logical size |
| 3 | 18 | Tracks per Zone | |
| 3 | 19 | Spare Sectors | |
| 3 | 1A | Alternate Cylinders | |
| 3 | 1B | Interleave | |
| 3 | 1C | Track Skew | |
| 3 | 1D | Cylinder Skew | |
| | 1E-1F | Default Bytes per Sector | |
| | 20-3F | -- Table of Error Flags -- | |
| 1 | +00 | Recovery Flags | hostdata(0*4+0) |
| 1 | +01 | Retry Count | hostdata(0*4+1) |
| 1 | +02 | Corrections Span | hostdata(0*4+2) |
| 1 | +03 | Spare | hostdata(0*4+3) |
| | . | | |
| | . | | |
| | . | | |
| 1 | +1C | Recovery Flags | hostdata(7*4+0) |
| 1 | +1D | Retry Count | hostdata(7*4+1) |
| 1 | +1E | Correction Span | hostdata(7*4+2) |
| 1 | +1F | Spare | hostdata(7*4+3) |
| | 40-4F | -- Table of Bus Limits -- | |
| 2 | +00 | Bus Inactivity Limit | Idle Limit (0) |
| 2 | +02 | Bus Inactivity Limit | Idle Limit (1) |
| | . | | |
| | . | | |
| | . | | |
| 2 | +0E | Bus Inactivity Limit | Idle Limit (7) |
| | 50-7F | Unused | |

### 3.3.2 GROWN DEFECT LIST

When a data sector develops unrecoverable errors, the host may command the controller to copy the data from the sector to a spare sector. The old sector is reformatted with the RP bit set and the address of the new sector in the sector header.

The controller maintains a list of all bad sectors in the Grown Defect List that starts on sector 3 of cylinder 0. The Grown Defect List is duplicated on each track of the cylinder and may not be larger than 8K bytes. The controller returns the contents of the Grown Defect List during the Data Out phase of a **READ DEFECT LIST** command.

The Grown Defect List contains an 8-byte header followed by 6-byte entries for spare sectors and spare tracks. The entries for spare sectors list the location of the defect and the length of the defect. The address of the replacement sector is contained in the header of the bad sector.

When a data track develops multiple error conditions that cannot be resolved by the use of spare sectors, the host may command the controller to save the data from the track and reformats it as a flawed data track. The defective data track is reformatted with the Defective Track bit and the track address of the allocated alternate track in the header of each sector. The data saved from the defective track is written to the alternate track.

An alternate track is a track in the controller-reserved storage area on the disk drive. When an alternate track is so allocated, it is formatted with the sector address of the defective track. The controller considers the interleave factor when it formats an alternate track. Spare sectors are also maintained on the alternate tracks.

The following three figures show the format of the Grown Defect List header and the entries for spare sectors and spare tracks.

**Grown Defect List Header Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 2 | Number of Entries (MSB) | | | | | | | |
| 3 | Number of Entries (LSB) | | | | | | | |
| 4 | Address of Next Unused Alternate Cylinder (MSB) | | | | | | | |
| 5 | Address of Next Unused Alternate Cylinder (LSB) | | | | | | | |
| 6 | Head Number of Next Unused Alternate Cylinder | | | | | | | |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | MDL |

The **Manufacturer's Defect List (MDL)** bit indicates that the Grown Defect List contains the Manufacturer's Defect List.

| Grown Defect List Entry for Spare Sector | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | Type | | 0 | Head Number | | | |
| 1 | Cylinder Number (MSB) | | | | | | | |
| 2 | Cylinder Number (LSB) | | | | | | | |
| 3 | Bytes from Index (MSB) | | | | | | | |
| 4 | Bytes from Index (LSB) | | | | | | | |
| 5 | Length of Defect in Bits | | | | | | | |

| Grown Defect List Entry for Spare Track | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | Type | | 0 | Head Number | | | |
| 1 | Defective Cylinder Number (MSB) | | | | | | | |
| 2 | Defective Cylinder Number (LSB) | | | | | | | |
| 3 | 0 | 0 | 0 | 0 | Head Number | | | |
| 4 | Replacement Cylinder Number (MSB) | | | | | | | |
| 5 | Replacement Cylinder Number (LSB) | | | | | | | |

The **Type Code** indicates the source of the entry, as follows:

| Value | Type | Description |
|---|---|---|
| 00000000 | Grown | Entry included from **REPLACE BLOCK** command |
| 01000000 | Host | Entry included from Format Host Defect List |
| 10000000 | MDL | Entry included from Manufacturer's Defect List |
| 11000000 | ? | Unknown |

### 3.3.3 UNFORMATTED MEDIA

When the controller is powered up with a drive that has not been formatted, all the current **MODE SENSE** parameters will be set to the default parameters returned by the drive and/or controller switches. The Initiator will not be allowed to access the media until either a **FORMAT** fommand or a **MODE SELECT** command with both pages 3 (format parameters) and 4 (drive geometry parameters) has been sent to the controller. Any command sent by the Initiator which accesses the media, before the **FORMAT** or **MODE SELECT** commands have been sent, will be terminated with a CHECK STATUS and a DRIVE NOT READY/ILLEGAL FUNCTION FOR DEVICE TYPE (02/22h) sense key/error code.

Section 4
# CONTROLLER INITIALIZATION AND SELF-TEST PROCEDURES

## 4.1 CONTROLLER RESET/POWER UP INITIALIZATION

This section describes the sequence of events during controller initialization and self-test sequences. The self-test sequence occurs before the initialization sequence occurs.

### 4.1.1 SELF-TEST SEQUENCE

The self-test sequence will be executed only if a controller power-up condition occurs. The self-test sequence verifies the integrity of the hardware. This test is not an exhaustive hardware diagnostic, but simply checks the major components for full functionality. If the self-test fails, the controller will light the Error Indicator and will stop any further initialization. If the self-test fails, only a SCSI Bus Reset or Power On Reset condition will restart the controller. During the self-test, the controller will not respond to a Selection Phase on the SCSI Bus.

The self-test sequence consists of the following events:

- **Hardware Reset Test** - This routine tests the 8031 microprocessor, buffer controller, disk formatter, and SCSI reset latch for the proper power-up condition. If any of these tests fail, the controller can only be reset by a power-up condition.

- **8031 Test** - This routine tests the 8031 internal memory, timers, and register bank switching for proper operation.

- **PROM Checksum Test** - This routine performs a checksum calculation on the controller firmware PROM and compares it against the checksum stored in the PROM.

- **Buffer Controller Test** - This routine tests the buffer controller for proper operation. All the registers are tested and the chip is engaged to access RAM memory. Other portions of the self-test check parts of the buffer controller, which cannot be tested at this time.

- **Dynamic RAM Test** - This routine tests the dynamic RAM memory by writing and reading different patterns to memory. In addition, the buffer controller is tested for proper refresh operation and parity detection. This test also tests the memory parity interrupt.

- **Disk Formatter Test** - This routine tests the disk formatter chip by writing and reading all possible patterns to each of the disk formatter chip registers. After the registers are tested, the interrupts are tested to ensure the formatter chip generates an interrupt when a command completes.

- **SCSI Controller Test** - This routine tests the SCSI controller chip by executing the chip diagnostic command. After the diagnostic test completes, the interrupts are tested to ensure the SCSI chip generates an interrupt when a command completes. Finally, the registers are tested by writing and reading all possible patterns to each of the SCSI controller chip registers.

If any portion of the self-test fails, except the hardware reset test, the controller can be reset by a SCSI bus reset condition or a power-up reset condition. The failure of the hardware reset test is considered a catastrophic failure and the controller can only be reset from such a failure by a power-up reset condition.

During the self-test, the onboard LEDs will indicate which test(s) are in progress as shown in Table 4-1.

Table 4-1. LED Sequences for Self-Test Procedure

| Red LED | Green LED | Description |
|---|---|---|
| OFF | OFF | Hardware Reset Test |
| OFF | ON | 8031 Test<br>PROM Checksum Test<br>Buffer Controller Test<br>Dynamic RAM Test |
| ON | OFF | Disk Formatter Test<br>SCSI Controller Test |
| ON | ON | Self-Test Passed |

If any of the tests fail, the pattern (ON or OFF) displayed by the LEDs indicate which portion of the self-test failed

**NOTE**

Due to the large amount of RAM memory on some controllers, the power up self-test may take up to 10 seconds to complete. During this time, the controller will not respond to a SCSI Bus Selection Phase.

### 4.1.2 INITIALIZATION SEQUENCE

The initialization sequence will be executed for any one of the following three reasons:

- Controller Power-Up condition occurs
- SCSI Bus Reset (-RST) signal is asserted
- BUS DEVICE RESET message (on the SCSI bus) is received

The initialization sequence consists of the following steps:

1. Initialize SCSI firmware.
   a. Set status for all LUNs to BUSY.
   b. Initialize SCSI interface.
   c. Enable SCSI interrupts.
2. At this point, the controller will respond to a Selection Phase from the Initiator but will return a BUSY status until the initialization sequence has been completed.
3. Initialize the disk firmware for each LUN supported by the controller.
   a. If a drive is not connected to this LUN, stop initialization for this LUN and go to the next LUN.
   b. Read the default parameters from the drive and/or switches.
   c. If this is a power-up condition and the disable spin-up switch is OFF, a **START UNIT** command will be sent to the drive to start the spin up operation. The controller will not wait for the spin up to complete before continuing the initialization sequence. If this is a SCSI bus Reset or Bus Device Message Reset, no action is taken.
   d. Turn on the user panel Ready LED.

4. At this point, BUSY is removed from all LUNs and the controller will accept commands from the Initiator.

Once the initialization sequence has completed, the controller will enter the idle state and flash the green LED. As long as the green LED is flashing, the controller is in its normal state of operation.

**NOTE**

The green LED may stop flashing momentarily while executing a command.

After this sequence is completed, the first command sent by an Initiator will be terminated with a CHECK STATUS and a UNIT ATTENTION Sense Key.

Until the drives have been spun up, any command sent by an Initiator which accesses the media will be terminated with a CHECK STATUS and DRIVE NOT READY Sense Key.

After the drive has spun up, the first command sent by the Initiator which accesses the media will load the saved **MODE SENSE** parameters. Prior to the drive being spun up, if an Initiator requests the current **MODE SENSE** parameters the controller will return the default **MODE SENSE** parameters. An Initiator should not request the current **MODE SENSE** parameters until the drive is spun up.

## 5.1 SCSI BUS PHASES

Communication on the SCSI bus occurs between an Initiator and a Target. The Initiator (usually a host adapter) originates an operation, and the Target (usually the controller) performs the operation. Throughout this section, the terms "Target" and "controller" are used interchangeably. The terms "Initiator" and "host" are also interchangeable.

The activities on the SCSI bus can be divided into the following phases of operation:

- Arbitration
- Selection
- Reselection
  - Command
  - Data
  - Status
  - Message

These phases are supported as specified by the ANSI SCSI specification (listed in subsection 1.1.1). The phases are individually discussed in subsequent subsections. The last four phases (Command, Data, Status, and Message) are grouped together as Information Transfer Phases.

When the SCSI bus is not involved in one of the SCSI bus phases, it is in a Bus Free Phase. The Bus Free Phase indicates that no host adapter or controller is actively using the SCSI bus and the SCSI bus is available for subsequent users.

The SCSI bus activities, implemented by the controller, include the disconnect function and reselection function. Overlapped operations on multiple controllers and multiple logical units are also supported.

In the following subsections, no attempt is made to detail the SCSI bus signal sequences. If detailed signal sequence information is required, refer to the SCSI Specification.

### 5.1.1 ARBITRATION PHASE

The Arbitration Phase is an optional implementation on the SCSI bus. This phase is used when multiple controllers or processors vie for SCSI bus ownership. Since multiple hosts adapters and/or controllers may desire control of the SCSI bus concurrently, arbitration for the SCSI bus is supported by the controller.

### 5.1.2 SELECTION AND RESELECTION PHASES

The SCSI bus Selection and Reselection phases provide methods for establishing a link between the Initiator and a desired Target.

Usually the controller is selected by an Initiator to perform some function (e.g., read or write data). The controller then has the option of disconnecting from the SCSI bus. When the controller needs to re-establish the link to its original Initiator, it reselects that Initiator.

The SCSI Selection and Reselection Phases can be terminated for any one of three conditions:

1. The preceding Selection or Reselection Phase is successfully completed by using the Selection/Reselection handshake protocol.

2. A Selection/Reselection timeout occurs. The timeout results if any Target or Initiator does not respond to the Selection/Reselection Phase within a selection timeout time period.

3. A Reset (-RST) signal occurs on the SCSI bus. When this signal is asserted, all SCSI bus sequences are immediately terminated and the SCSI bus signals are released by all Initiators and Targets.

The Initiator can use the Attention (-ATN) signal to notify the controller that a message from the Initiator is ready. To guarantee that the Target recognizes the Attention condition before the Command Phase is entered, the -ATN signal level must be true before the Selection or Reselection Phase is completed.

If an IDENTIFY message is used during the Selection Phase sequence, the specified Logical Unit Number (LUN) has precedence over the LUN field in the Command Descriptor Block (CDB). (CDBs are described in detail in Section 6.)

### 5.1.3 INFORMATION TRANSFER PHASES

The Command, Data, Status, and Message Phases are grouped together as Information Transfer Phases because they are all used to transfer data or control information via the SCSI data bus. The Information Transfer Phases are described in the following subsections.

### 5.1.4 COMMAND PHASE

The Command Phase allows the Target to request command information from the Initiator. An Initiator issues SCSI commands to a Target by transferring a command packet, called a Command Descriptor Block (CDB). The length of the SCSI command and the meaning of the information in the command packet depends on which command is being transferred. (See Section 6 for definitions of SCSI commands and all SCSI CDBs supported by the controller.)

The Command Phase is interrupted only for the following exception conditions:

- **Reset Condition.** This condition can occur when the SCSI Reset (-RST) signal is asserted or a power fail or power-off condition in the Target occurs. In this case, the Command Phase and the connection established during the Selection/Reselection Phase is terminated by the Target with the release of the -BSY signal.

- **Parity Error Condition.** This condition occurs if the controller detects a parity error on the SCSI bus during the command transfer operation. At this time, the controller sends a RESTORE POINTERS message and attempts to receive the command a second time. If the controller again detects a parity error, it will either go to the Bus Free state or terminate the command. For more information, see subsection 5.5, SCSI Error Conditions.

### 5.1.5 DATA PHASE

The Data Phase of a connection controls the transfer of data between the Initiator and Target devices. The Data Phase includes both the Data In Phase and the Data Out Phase. The Data In Phase allows the Target to request sending of data to the Initiator from the Target. The Data Out Phase allows the Target to request sending of data to the Target from the Initiator. The direction of the data transfer operation depends on the command being processed. Some commands may have no data to be transferred and therefore have a null Data Phase. Only the asynchronous data transfer mode is supported by the controller.

The Data Phase is interrupted only for the following exception conditions:

- **Reset Condition.** This condition can occur when the SCSI Reset (-RST) signal is asserted or when a power fail or power-off condition in the Target occurs. In this condition, the Data Phase and the connection established during the Selection/Reselection Phase are terminated by the Target with the release of the -BSY signal.

- **Data Out Parity Error Condition.** The Target detects a parity error on the SCSI bus during the data transfer operation from the Initiator to the Target. For more information, see subsection 5.5, SCSI Error Conditions.

- **Data In Parity Error Condition.** The Initiator detects a parity error on the SCSI bus during the data transfer operation from the Target to the Initiator. The Initiator can then assert the -ATN signal along with the Acknowledge (-ACK) signal. The Target detects this condition and enters the message out phase to receive a message. The Initiator sends an Initiator-detected error message in response. For more information, see subsection 5.5, SCSI Error Conditions.

### 5.1.6 STATUS PHASE

The Status Phase is used by the controller to send command completion information to the Initiator. The status is sent in a single byte, the format of which is defined in this subsection.

The controller will initiate the Status Phase when any one of the following conditions occur:

- **Busy Status**-- The Selection Phase is completed and the specified LUN has already received a command from the same Initiator, or the command queue for the specified LUN is full. The Status Byte transferred has the BUSY status code set.

- **Reservation Conflict Status** - The Command Phase is completed and the specified LUN is reserved for another Initiator. The Status Byte transferred has the RESERVATION CONFLICT status code set.

- **Terminated Status** - At the termination of a command, the Status Byte transferred has either the GOOD STATUS, the CHECK CONDITION, or the INTERMEDIATE STATUS code set to indicate the command completion status.

### 5.1.6.1 Status Byte Format

The format of the Status Byte containing the command completion information is defined below.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | Rsrv | Vendor Unique | | Status Code | | | | V |

The **Reserved** field <Bit 7> is set aside for future standardization and will always be set to zero.

The **Vendor Unique** field <Bits 5:6 and Bit 0> is reserved and will always be set to zero.

The **Status Code** field <Bits 1:4> are used to specify the status of the completed command. Table 5-1 describes the status codes returned by the controller.

Table 5-1. Status Codes

| Status Codes Bits 4 | 3 | 2 | 1 | Status Description |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **GOOD STATUS.** The controller successfully completed the command. |
| 0 | 0 | 0 | 1 | **CHECK CONDITION.** An error, exception, or abnormal condition occurred during the command execution. |
| 0 | 1 | 0 | 0 | **BUSY.** The controller already has a command for the specified LUN or the command queue is full. |
| 1 | 0 | 0 | 0 | **INTERMEDIATE STATUS.** The controller sends this status code for every command in a series of linked commands unless a CHECK CONDITION, BUSY, or RESERVATION CONFLICT status code is detected. |
| 1 | 1 | 0 | 0 | **RESERVATION CONFLICT.** Sent to an Initiator that attempts to access a LUN that is reserved by another Initiator. |
| 1=Set 0=Cleared | | | | |

### 5.1.7 MESSAGE PHASE

The Message Phase is used to transfer information about exception conditions between the Initiator and the Target. The Message Phase includes both the Message In and the Message Out Phases. The Message In Phase allows a Target to request that messages be sent to the Initiator from the Target. The Message Out Phase allows a Target to request that messages be sent from the Initiator to the Target. Messages from the controller are a single byte in length; they (and their corresponding hexadecimal codes) are listed and described in Section 6.

#### 5.1.7.1 SCSI Messages

This subsection describes the SCSI messages, and their hexadecimal codes, that are supported by the controller and under what conditions the controller will send the message. Message error conditions are discussed in the section on SCSI Error Conditions.

**COMMAND COMPLETE MESSAGE (00H)** - The COMMAND COMPLETE message is sent by the controller to the Initiator to indicate that the execution of a command (or a series of linked commands) has completed and that a valid status has been sent to the Initiator. Note: this message does not indicate the execution status of the command, but simply that the command has completed execution.

**EXTENDED MESSAGE (01H)** - The following three EXTENDED Messages will be received by the controller, with the specified action taken after the message has been received. Any other reserved or vendor unique extended messages will be rejected by the controller.

**MODIFY DATA POINTER (00H)** - The controller will receive the entire MODIFY DATA POINTERS message, then send a MESSAGE REJECT message to the Initiator.

**SYNCHRONOUS DATA TRANSFER REQUEST (01H)** - The controller will receive the entire SYNCHRONOUS DATA TRANSFER REQUEST message, then send a SYNCHRONOUS DATA TRANSFER REQUEST message back to the Initiator, with a REQ/ACK offset of zero. The REQ/ACK offset of zero specifies that the controller does not support synchronous transfers and the Initiator should use Asynchronous mode.

**EXTENDED IDENTIFY (02H)** - The controller will receive the entire EXTENDED IDENTIFY message, then send a MESSAGE REJECT message to the Initiator.

**SAVE DATA POINTERS MESSAGE (02H)** - The SAVE DATA POINTER message is sent by the controller to direct the Initiator to save a copy of the present active data pointer. This message is sent just before the controller issues the DISCONNECT message and disconnects from the bus. If the data phase of the command has not been started yet (i.e., just after the command phase), the controller may not send this message before disconnecting. Note: The controller will not send this message to an Initiator which does not support the disconnect option.

**RESTORE POINTERS MESSAGE (03H)** - The RESTORE POINTERS message is sent by the controller to direct the Initiator to restore the most recently saved set of pointers. This message will be issued by the Controller if it detects a parity error during the Command Phase or receives an Initiator Detected Error message after the Status Phase.

**DISCONNECT MESSAGE (04H)** - The DISCONNECT message is sent by the controller just before releasing the BUSY signal to indicate that the present physical connection will be temporarily broken. A reconnection will be attempted later to complete the command. Note: The controller will not send this message to an Initiator which does not support the disconnect option.

**INITIATOR DETECTED ERROR MESSAGE (05H)** - Issued by an Initiator to inform the controller that an error has occurred during an operation. This message will be sent by the controller, while in Initiator mode, when it detects a parity error during a Data In phase.

**ABORT MESSAGE (06H)** - Issued by the Initiator to the controller to clear the specified LUN and cause the SCSI bus to go to the Bus Free Phase. If the Logical Unit has been identified, all pending data and status from the identified LUN, for the issuing Initiator, will be cleared. If the Logical Unit has not been identified, the controller will not clear any information. After receiving the Abort Message, the controller will go straight to the Bus Free phase and will not return any status information for the current command.

**MESSAGE REJECT MESSAGE (07H)** - Issued by the Initiator or Target Controller in response to a received message that was undefined or not implemented. An EXTENDED message is the only valid SCSI message which will be rejected by the controller.

**NO OPERATION MESSAGE (08H)** - A null message issued by the Initiator if the controller requests a message from the Initiator but the Initiator has no message to send.

**MESSAGE PARITY ERROR MESSAGE (09H)** - Issued by the Initiator to inform the controller that a parity error has occurred on a message receive operation from the controller to the Initiator.

**LINKED COMMAND COMPLETE MESSAGE (0AH)** - Issued by the controller to the Initiator to indicate the completion of a linked command.

**LINKED COMMAND COMPLETE WITH FLAG MESSAGE (OBH)** - Issued by the controller to the Initiator to indicate the completion of a linked command that had the Flag bit set.

**BUS DEVICE RESET MESSAGE (0CH)** - Issued by the Initiator to the Target to reset all current I/O activities on the controller. This message generates a hard Reset Condition and initializes the controller to a power-up reset state. Any **MODE SELECT** information must be re-issued by the Initiator after this message is sent.

**IDENTIFY MESSAGE (80H)** - Issued by the controller, or Initiator, to establish a connection to a particular LUN. In addition, this message also specifies that the sender supports some or all of the optional messages. The following bits have particular meaning:

| | |
|---|---|
| Bit 07 | Always set to one. |
| Bit 06 | Set if the Initiator can support Disconnect and Reconnect sequences. Note: The controller will never set this bit when the IDENTIFY message is used during a reconnection operation. |
| Bits 0:2 | Specify LUN address (hexadecimal) in a Target. |

The controller will send this message during a Reselection Phase as a Target, or during a selection Phase as an Initiator.

### 5.1.8 SCSI BUS PHASE SEQUENCING

The status of the SCSI bus is a function of the SCSI bus control signals. These signals place the SCSI bus in one of four phases: Arbitration, Selection/Reselection, Information Transfer, or Bus Free. The order in which SCSI bus phases are used follows the prescribed sequence shown in Figure 5-1.

All SCSI command sequences start with the Bus Free Phase. The normal progression is from the Bus Free Phase to the Arbitration Phase. During arbitration, host adapters or controllers contest for control of the SCSI bus; priority is given to the contestant that has the highest SCSI bus address.

Once a host adapter or controller has control (i.e., is the bus master) of the SCSI bus, the SCSI bus enters the Selection/Reselection Phase. This phase allows the bus master to select a specific device for communication. An Initiator can select a Target to initiate an operation, or a Target can reselect an Initiator to continue an operation.

After a physical path between an Initiator and a Target is established, the SCSI bus enters one of the Information Transfer Phases. These phases include six types of information exchange:

- Data Out Phase
- Data In Phase
- Command Phase
- Status Phase
- Message In Phase
- Message Out Phase

Normally the Information phase follows this sequence:

- Message Out Phase (IDENTIFY message)
- Command Phase
- Data In or Out Phase (if required)
- Status Phase
- Message In Phase (COMMAND COMPLETE message)

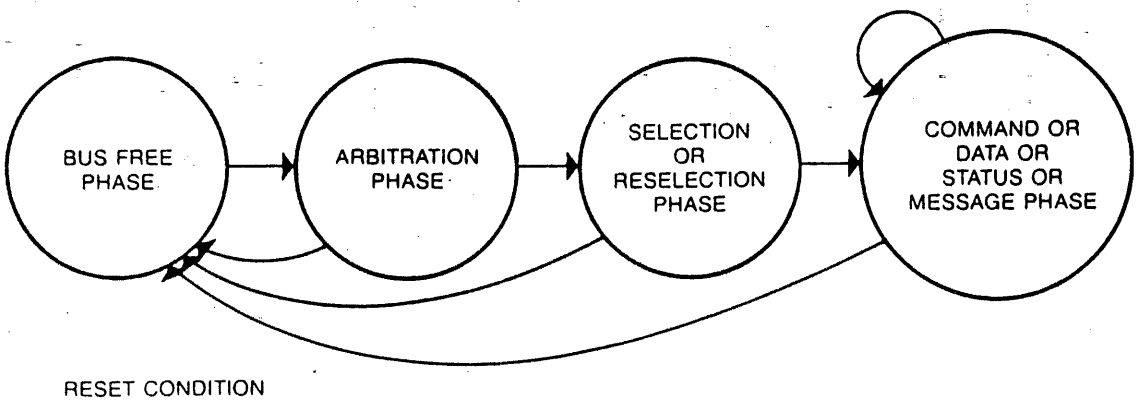



Figure 5-1. SCSI Bus Phase Sequences

## 5.2 SCSI BUS EVENTS

The SCSI bus has the following asynchronous events:

- Reset Condition
- Attention (-ATN) Condition

These events cause certain SCSI device actions and can alter the phase sequence. The two events are described in the following subsections.

### 5.2.1 RESET

The Reset event is used to immediately clear all bus masters from the SCSI bus. This condition takes precedence over all other SCSI bus phases and conditions. During the Reset event, no SCSI bus signal except -RST is guaranteed to be valid.

The controller supports the SCSI Hard Reset option (the SCSI Soft Reset option is not supported). When it detects a Reset event, the controller performs the following:

- Goes to Bus Free state
- Clears all uncompleted commands

- Releases device reservations
- Returns device operating modes (such as the **MODE SELECT** command) to their default conditions

The Hard Reset condition has the same effect on the controller as power-on; therefore, all Initiator-defined parameters must be re-submitted to the controller.

The controller includes a watchdog timer feature. If a SCSI Hard Reset Condition occurs, a latch is set. If the controller microprocessor cannot service the reset latch within 40 milliseconds, the timer times out and the controller executes a power-on reset.

### 5.2.2 ATTENTION (-ATN)

The Attention event allows an Initiator to inform a Target that the Initiator has a message ready. The Target will respond with a Message Out Phase in an timely manner. However, it may have to complete the current Information Transfer Phase before it responds with the Message Out Phase. This means that the Target may continue to transfer data after the Initiator has asserted -ATN.

## 5.3 SCSI COMMAND TIMING

This subsection provides information on the timing for the SCSI commands. The timing information is broken down into two areas: phase change timing and command timing. The phase change timing defines the times required for the controller to change phases during the execution of a command. The command timing defines the overall time required to complete certain commands.

### 5.3.1 SCSI PHASE CHANGE TIMING

This subsection provides information on the time required for the controller to change phases during the course of command execution. The times listed are the maximum time it will take for the controller to complete the defined operation. This information is best case timing which assumes only one command is being executed by the controller.

| | |
|---|---|
| Connect Latency ..... | 125us |
| Command Transfer Latency ..... | 125us |
| Command Disconnect Latency ..... | 150us |
| Command Decode Time ..... | 1.5ms |
| Reconnect Latency ..... | 100us |
| Data Transfer Latency ..... | 225us |
| Data Transfer Disconnect Latency ... | 150us |
| SCSI Write Data Latency ..... | 750us |
| SCSI Read Data Latency ..... | 400us |
| Status Latency ..... | 150us |
| Command Complete Latency ..... | 100us |
| Bus Free Latency ..... | 75us |

**Start of Phase** - The start or beginning of phase is defined as the leading edge of the first Request (-REQ) signal after the phase change.

**End of Phase** - The end of phase is defined as the trailing edge of the last Acknowledge (-ACK) signal before the next phase change.

**Connect Latency** - The Connect Latency time is defined as the time between Selection and the start of the Message Out Phase for the IDENTIFY message.

**Command Transfer Latency** - The Command Transfer Latency time is defined as the time between the end of the Message Out Phase, for the IDENTIFY message, and the start of the Command Phase.

**Command Disconnect Latency** - The Disconnect Latency time is defined as the time between the end of the Command Phase and the Message In Phase for the Disconnect message, when a disk **READ** command has been received.

**Command Decode Time** - The Command Decode Time is defined as the time between the end of the Command Phase and the start of the operation. In the case of commands which require a seek operation to be performed, this time is between the end of the Command Phase and the start of the seek operation.

**Reconnect Latency** - The Reconnect Latency is defined as the time between the Reconnect sequence and the start of the Message In Phase for the IDENTIFY message.

**Data Transfer Latency** - The Data Transfer Latency is defined as the time between the end of the Message In Phase, for the IDENTIFY message during a reconnect, and the start of the Data In/Out Phase.

**Data Transfer Disconnect Latency** - The Data Transfer Disconnect Latency is defined as the time between the end of the Data In/Out Phase and the start of the first Message In Phase for the disconnect sequence.

**SCSI Write Data to Host Latency** - The Write Data Latency Time is defined as the time between the end of the first sector read off the disk and the start of the SCSI Data In transfer. This measurement does not include the time required to transfer the IDENTIFY message during the reconnection sequence.

**SCSI Read Data from Host Latency** - The Read Data Latency Time is defined as the time between the end of the Command Phase and the start of the SCSI Data Out Phase.

**Status Latency** - The Status Latency time is defined as the time between the end of the Data In or Message In Phase, and the start of the Status Phase.

**Command Complete Latency** - The Command Complete Latency is defined as the time between the end of the Status Phase and the start of the Message In Phase for the COMMAND COMPLETE message.

**Bus Free Latency** - The Bus Free Latency is defined as the time between the end of the Message In Phase, for the COMMAND COMPLETE or DISCONNECT message, and the Bus Free Phase.

### 5.3.2 SCSI COMMON COMMAND TIMING

This subsection provides information on the time required for the controller to complete common commands. These times are specified for a **READ** and a **WRITE** command which transfers 1024 bytes of data. Most other commands will also fall within these times except for the **FORMAT UNIT, COPY, COMPARE, RE-ASSIGN BLOCK, READ DEFECT LIST, SEND DIAGNOSTIC** and **RECEIVE DIAGNOSTIC RESULTS** commands. The times listed are the maximum time it will take for the controller to complete the defined operation. This information is best case timing which assumes only one command is being executed by the controller.

| Description | Read Time | Write Time |
|---|---|---|
| Total Controller Latency..... | 2.5ms | 2.5ms |
| Total Connect Time..... | 1.2ms | 1.8ms |

**Total Controller Latency** - Total Controller Latency is defined as the amount of processing time required by the controller to complete a command. This measurement does not include the time required for data transfers, host latency, device latency, command transfer, message transfer and status transfer.

**Total Connect Time** - Total Connect Time is defined as the amount of time the controller is connected to the Initiator on the SCSI bus in order to complete a command. This measurement does not include the time required for host latency, data transfers or device latency.

## 5.4 SCSI COMMAND QUEUING

The controller will queue up to two commands per LUN, in addition to the command currently being executed. All commands queued for a single LUN must be from different Initiators. If the same Initiator attempts to send more than one command to the same LUN, the second command will return a BUSY status. For any command that is placed in the LUN command queue, the controller will disconnect after receiving the command. The controller will not queue commands for Initiators that do not support the disconnect function.

## 5.5 SCSI ERROR CONDITIONS

This subsection describes the various SCSI bus related errors which can occur during the execution of a command and the actions taken by the controller in response to these errors.

### 5.5.1 SENSE DATA

The controller maintains Sense Information for each LUN supported by the controller. The LUN Sense Information is further divided into Sense data for each of the eight possible Initiators. Each command sent to the controller is identified by the LUN it was sent to and the Initiator which sent it. When an error occurs, the Sense data for that command is saved for the corresponding LUN/Initiator combination. When the Initiator sends a **REQUEST SENSE** command, that saved Sense data is returned to the Initiator. The command Sense data for any LUN/Initiator combination is saved until the next command (except for a **REQUEST SENSE** command) is received for the same LUN, from the same Initiator.

### 5.5.2 UNIT ATTENTION CONDITION

The controller maintains Unit Attention data for each LUN supported by the controller. The LUN Unit Attention data is further divided into Unit Attention Flags for each of the eight possible Initiators. The Unit Attention Flag will be set for all Initiators for any of the following reasons:

- When the controller has been reset by a SCSI Bus Reset, BUS DEVICE RESET message, or Power On Reset. The Sense Key/Error Code will be set to UNIT ATTENTION/POWER UP OR RESET (06h/29h).

- When an Initiator issues a **MODE SELECT** command which changes the parameters that may affect another Initiator. The Sense Key/Error Code will be set to UNIT ATTENTION/MODE SELECT CHANGED CONDITION (06h/2Ah).

- When the controller detects a media changed condition, it will set the Sense Key/Error Code to UNIT ATTENTION/MEDIA CHANGED (06h/28h). A Media Changed condition occurs when the controller detects a drive offline condition and then detects the disk drive come online.

The Unit Attention Flags for each Initiator remain set until the controller returns a CHECK CONDITION status for a command sent from that Initiator. Once the first CHECK CONDITION is returned, the controller will clear the Unit Attention Flag for that Initiator and set the Sense data as noted above. If the next command sent by that Initiator is a **REQUEST SENSE** command, the Unit Attention Sense data will be returned, otherwise the Unit Attention Sense data is lost.

The **INQUIRY** and **REQUEST SENSE** commands are the only commands which will not generate a CHECK CONDITION status when a Unit Attention condition is pending. The **REQUEST SENSE** command will clear the Unit Attention Flag and return an UNIT ATTENTION Sense Key. The **INQUIRY** command will execute normally and will not clear the Unit Attention Flag.

### 5.5.3 TARGET MODE ERROR CONDITIONS

Under several error conditions, the controller will change the phase to Bus Free without correctly terminating the command (i.e., no DISCONNECT or COMMAND COMPLETE message will be sent). The controller will clear all information regarding the command, except sense data (if any), and will not attempt to reconnect or in any other way terminate the command. The Initiator must assume this is a catastrophic failure and return the error to the host software.

Sense data may or may not be valid when this condition occurs. If the initiator issues a **REQUEST SENSE** command and the returned Sense Key/Error Code is anything other than 0/0, the sense data is valid.

#### 5.5.3.1 Message Out Phase Parity Error

If the controller detects a parity error during the Message Out Phase, it will retry the Message Out Phase one time using the following sequence:

1. Continue -REQ/-ACK Handshake until the Initiator negates ATN (receive all of the message bytes).
2. Notify the Initiator to resend all of the message bytes sent during the previous Message Out Phase by not changing the phase and asserting -REQ.
3. The Initiator will then resend all of the previous message bytes.

If after one retry, the message is still not received correctly, the controller will process the error using one of the following sequences:

- If an IDENTIFY message has not been received yet (the addressed LUN is unknown), the controller will immediately go to the Bus Free Phase. No Sense Key/Error Code information will be set for this type of error.
- If the IDENTIFY message has been received (the addressed LUN is known), the controller will terminate the present command with a CHECK CONDITION status and set the Sense Key/Error Code to ABORTED COMMAND/PARITY ERROR (0Bh/47h). This error does not prevent the Initiator from retrying the command.

#### 5.5.3.2 Command Phase Parity Error

When the controller detects a parity error during the Command Phase, it will retry the Command Phase one time using the following sequence:

1. Send the Initiator a RESTORE POINTERS message to reset the pointers to the start of the command.
2. Attempt to receive the entire command over.

If after one retry the command is still not received successfully, the controller will abort the command using one of the following sequences:

- If the Initiator did not send an IDENTIFY message (the addressed LUN is unknown), the controller will immediately go to the Bus Free Phase. No Sense Key/Error Code information will be set for this type of error.

- If the controller received an IDENTIFY message (the addressed LUN is known), it will terminate the command with a CHECK CONDITION status and set the Sense Key/Error Code to ABORTED COMMAND/PARITY ERROR (0Bh/47h). This error does not prevent the Initiator from retrying the command.

#### 5.5.3.3 Data Out Phase Parity Error

If the controller detects a parity error during the Data Out Phase, it will terminate the command with a CHECK CONDITION status and set the Sense Key/Error Code to ABORTED COMMAND/PARITY ERROR (0Bh/47h). This error does not prevent the Initiator from retrying the command.

#### 5.5.3.4 Initiator Detected Error

At any time during the command, except during the status phase or command complete message, if the controller receives an initiator-detected error message it will terminate the current command with a CHECK CONDITION status and set the Sense Key/Error Code to Aborted Command/Initiator Detected Error (0Bh/48h). This error does not prevent the initiator from retrying the command.

If the initiator sends an initiator-detected error message immediately after the status phase, the controller will send a Restore Pointers message and resend the status. If the initiator sends another initiator-detected error message in response to the resent status the controller will immediately go to the Bus Free phase. The Sense Key/Error Code will be set to Aborted Command/Initiator Detected Error (0Bh/48h). This error does not prevent the initiator from retrying the command.

If the initiator sends an initiator-detected error message immediately after the command complete message is sent, the controller will immediately go to the Bus Free phase. The Sense Key/Error Code will be set to Aborted Command/Initiator Detected Error (0Bh/48h). This error does not prevent the initiator from retrying the command.

### 5.5.3.5 Rejected Message

When the controller receives a MESSAGE REJECT message from the Initiator, it will resend the original message one more time. If the message is rejected again, the controller will take the following action, based on which message was rejected.

**COMMAND COMPLETE** - The controller will go to the Bus Free phase and not consider this an error.

**DISCONNECT** - The controller will not disconnect from the Initiator and will continue the current command. This condition does not preclude the controller from attempting to disconnect at a later time. Note: The controller will not send a DISCONNECT message to an Initiator which does not support the disconnect/reconnect option.

**IDENTIFY (Reconnect)** - The controller will immediately go to the Bus Free Phase and abort the current SCSI command. No further reconnection will be attempted and no status or COMMAND COMPLETE message will be sent for the command. The Sense Key/Error Code will be set to HARDWARE ERROR/MESSAGE REJECT ERROR (04h/43h).

**LINKED COMMAND COMPLETE** - The controller will immediately go to the Bus Free phase and will not read the next command in the linked list. The Sense Key/Error Code will be set to HARDWARE ERROR/MESSAGE REJECT ERROR (04h/43h).

**LINKED COMMAND COMPLETE WITH FLAG** - The controller will immediately go to the Bus Free phase and will not read the next command in the linked list. The Sense Key/Error Code will be set to HARDWARE ERROR/MESSAGE REJECT ERROR (04h/43h).

**MESSAGE REJECT** - The controller will immediately terminate the present command with a CHECK CONDITION status and set the Sense Key/Error Code to HARDWARE ERROR/MESSAGE REJECT ERROR (04h/43h).

**RESTORE POINTERS** - Since the Restore Pointers message is only used in an error recovery or retry situation, the controller will abort the recovery or retry attempt, assume the error is unrecoverable, and complete the command according to the error condition.

**SAVE DATA POINTER** - The controller assumes the Initiator does not support this message and will not attempt to disconnect from the bus during this command.

#### 5.5.3.6 Initiator Message Parity Error

When the controller receives a Message Parity Error message from the Initiator, it will retry the operation by resending the original message. If the message cannot be sent successfully, the controller will immediately go to the Bus Free Phase and abort the current SCSI command. No further reconnection will be attempted and no status or Command Complete message will be returned for the command. The Sense Key/Error Code will be set to ABORTED COMMAND/PARITY ERROR (0Bh/47h).

#### 5.5.3.7 Reselection Timeout

When the controller attempts to reselect the Initiator and it does not respond within a Selection Timeout Delay (as defined in the SCSI standard), the reselection will be aborted. The controller will attempt the reselection one more time, and if both attempts fail, the controller will abort the current SCSI command. No further reconnection will be attempted and no status or COMMAND COMPLETE message will be returned for the command. The Sense Key/Error Code will be set to HARDWARE ERROR/SELECT-RESELECT TIMEOUT (04h/45h). Note: The Initiator must have an overall command timeout delay to detect this error.

#### 5.5.3.8 Internal Controller Errors

If an error occurs within the controller which is related to the SCSI hardware or firmware, the controller will terminate the present command with a CHECK CONDITION status and set the Sense Key/Error Code to HARDWARE ERROR/SCSI HARDWARE ERROR (04h/44h). This error does not prevent the Initiator from retrying the command.

### 5.5.4 INITIATOR MODE ERROR CONDITIONS

The controller will only operate in the Initiator Mode when it is performing a **COPY** or **COMPARE** command to or from an external device. This section describes the controller actions when is detects an error in the Initiator mode.

#### 5.5.4.1 Selection Timeout

When the controller attempts to select a Target and it does not respond within a Selection Timeout Delay (as defined in the SCSI standard), the selection will be aborted. The controller will attempt the selection one more time, and if both attempts fail, the controller will abort the **COPY/COMPARE** command. The Sense Key/Error Code will be set to HARDWARE ERROR/SELECT-RESELECT TIMEOUT (04h/44h).

#### 5.5.4.2 Parity Error on Received Data

When the controller detects a parity error during the Data In Phase, it will send an INITIATOR DETECTED ERROR message to the Target.

#### 5.5.4.3 Parity Error on Received Message

When the controller detects a parity error while receiving a message, it will send a MESSAGE PARITY ERROR message to the Target.

#### 5.5.4.4 Rejected Message

When the controller receives a MESSAGE REJECT message from the Target, it will resend the the original message one more time. If the message is rejected again, the controller will take the following action, based on the message which was rejected.

**Initiator Detected Error** - The controller will send an ABORT MESSAGE message.

**Abort** - The controller will immediately go to the Bus Free phase and abort the **COPY/COMPARE** command. The Sense Key/Error Code will be set to HARDWARE ERROR/MESSAGE REJECT ERROR (04h/43h).

**Message Parity Error** - The controller will send an ABORT MESSAGE message.

**Identify** - The controller will send an ABORT MESSAGE message.

**No Operation** - The controller will ignore the message and not consider this an error condition.

**Message Reject** - The controller will ignore the message and not consider this an error condition.

**Bus Device Reset** - The controller will never send this message while operating in the Initiator mode.

#### 5.5.4.5 Internal Controller Errors

If an error occurs within the controller which is related to the SCSI hardware or firmware, the controller will send an ABORT message to the Target to terminate the **COPY** or **COMPARE** command. The Sense Key/Error Code will be set to HARDWARE ERROR/SCSI HARDWARE ERROR (04h/44h).

## 5.6 SCSI COMMAND DISCONNECTS

This section describes the conditions under which the controller will disconnect/reconnect with the SCSI bus. These are the default options provided by the controller but may be modified by the Initiator using the **MODE SELECT** command.

### 5.6.1 NORMAL DISCONNECT/RECONNECT CONDITIONS

The normal controller Disconnect/Reconnect options stress maximum throughput, yet do not monopolize the SCSI bus, providing maximum bus utilization. These options would be used in a multiple Initiator environment, or where the Initiator can support simultaneous commands on multiple LUNS. Also, the Initiator should be capable of sustaining a SCSI bus throughput which is equal to or greater than the transfer rate from the disk drive to the controller.

To disable disconnects, the Initiator should not set Bit 6 in the ID message that it sends at the start of a command. If the controller is processing a command which does not support disconnects (Bit 6 in the ID message equals 0) and any other command is active, the controller will immediately terminate the command that cannot disconnect with a BUSY status. If normal system operation includes multiple commands sent to the same controller simultaneously, the Initiator should enable the disconnect/reconnect option (Bit 6 in the ID message equals 1).

The controller disconnect/reconnect conditions are as follows:

1. When the SCSI bus is idle (no -REQ is active) for the amount of time specified in the **MODE SELECT** command Bus Inactivity Limit Field (see subsection 6.2.5.5.5), the controller will automatically disconnect from the SCSI bus.

2. When the controller receives a command while another command is active, it will disconnect from the bus as soon as the entire command has been received. This disconnect allows the other command access to the bus if it needs it, providing maximum SCSI bus utilization.

3. If the command received is for a LUN which is already executing a command, the controller will queue the command just received and disconnect from the Bus.

4. Disk **READ** commands will always disconnect immediately after receipt of the command. This allows other commands access to the SCSI bus during the disk latency and/or seek time.

5. When the controller is reading data from the disk, and no data is ready to be transferred (the bus is idle), the controller will disconnect from the bus when the time specified in the **MODE SELECT** Bus Inactivity Limit Field has expired. The reconnect will occur when data is available from the disk. This condition may occur for the following reasons:

   - The disk read operation crosses a cylinder boundary and a seek operation is required.
   - A seek operation to an alternate track is required.
   - The SCSI data transfer rate is greater than the data transfer rate from the disk.

6. When the controller receives a disk **WRITE** command, it will read data from the Initiator until all the internal buffers available for the LUN are filled. Once the buffers are filled, or all the data has been transferred, the controller will disconnect from the bus. The reconnect will occur when the controller starts to write data to the disk and internal buffers become available for more data to be read from the Initiator. A reconnect will occur after all the data has been written to the disk to return the status and terminate the command.

## 6.1 SCSI COMMAND OVERVIEW

This section describes all the SCSI commands implemented by Emulex SCSI disk controllers. The command descriptions are listed alphabetically within each command group, starting with group 0. Each command description is self-contained and explains the command function, command descriptor block, data returned and error conditions that may arise.

## 6.2 SCSI GROUP CODE 0 COMMANDS

This subsection provides a detailed description of the SCSI Group Code 0 commands, including Command Descriptor Block formats, hexadecimal operation code, byte and bit functions, and any necessary effects produced by the commands. Each SCSI command is described in a separate subsection. Figure 6-1 shows a sample Group 0 Command Descriptor Block. Explanations of those fields which are common among all commands follow.

| Typical Group 0 Command Descriptor Block | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Group Code | | | Command Code | | | | |
| 1 | Logical Unit Number | | | Command Dependent Parameters | | | | |
| 2 | Command Dependent Parameters | | | | | | | |
| 3 | Command Dependent Parameters | | | | | | | |
| 4 | Command Dependent Parameters | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

Figure 6-1. Sample Group 0 Command Descriptor Block

The **Reserved** bits, bytes, fields and code values are set aside for future standardization. All Reserved bits, bytes or fields are checked by the controller and **must** be set to 0. If a Reserved field is not set to 0, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL FIELD IN CDB (05h/24h) or ILLEGAL REQUEST/ILLEGAL FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

The **Operation Code** is the first byte <Byte 0> of a Command Descriptor Block. The Operation Code contains two fields: the Group Code in the high-order three bits <Bits 05:07>, and the Command Code in the low-order five bits <Bits 00:04>. The

Group Code specifies the length of the Command Descriptor Block and, together with the Command Code, determines the operation to be performed. If the specified Operation Code is invalid or not implemented, the controller returns a CHECK CONDITION status with an ILLEGAL REQUEST/INVALID COMMAND (05h/20h) Sense Key/Error Code.

The **Logical Unit Number** field <Byte 01, Bits 05:07> contains the number of the device being addressed. The controller, acting as a SCSI bus Target, supports up to four LUNs. Therefore, the value for the LUN field (in byte 01 of the CDB) is limited to 000, 001, 010, or 011. If an invalid LUN is specified, the controller returns a CHECK CONDITION status with an ILLEGAL REQUEST/INVALID LUN (05h/25h) Sense Key/Error Code.

The controller provides this method of addressing the device for Initiators that do not implement the IDENTIFY message. A LUN specified in the IDENTIFY message will override any LUN specified in the Command Descriptor Block.

The **Control Byte** is the last byte <Byte 05> in every Command Descriptor Block. The Control Byte is separated into four fields:

- The Vendor Unique control bits <Bits 06:07> are used as Command-Dependent Parameters. Unless otherwise specified in the command description, these bits are to be treated as reserved.
- Bits <05:02> are reserved.
- Bit <01> is defined as the Flag bit and is only checked when the Link bit is set to 1. When the Flag bit is 0, the controller will send a LINKED COMMAND COMPLETE message when the command completes successfully. If the Flag bit is set to 1, the controller sends a LINKED COMMAND COMPLETE (WITH FLAG) message when the command completes successfully.
- Bit <00> is defined as the Link bit. When the current command completes successfully and the Link bit is set to 1, the controller will return an INTERMEDIATE status, followed by one of the two messages defined by the Flag bit above. The controller will then automatically link to the next command. If a linked command is not completed successfully, the controller will return a CHECK CONDITION status and will not link to the next command.

The remaining bytes in the Command Descriptor Block are primarily **Command Dependent** and are described in the individual command sections.

### 6.2.1 COPY - 18H

The controller does not currently support the **COPY** command. However, it will be supported in future releases.

If any attempt is made to execute the **COPY** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL COMMAND (05h/20h) Sense Key/Error Code.

### 6.2.2 FORMAT UNIT - 04H

The **FORMAT UNIT** command ensures that the media is formatted so that all data blocks can be accessed. The controller maintains a defective sector and track file on the disk on a cylinder that is inaccessible to the Initiator. During the formatting process, the Initiator may specify a set of defective blocks or tracks to be reassigned using spare blocks or alternate tracks as appropriate.

The **FORMAT UNIT** command uses four different sets of defect information:

**Manufacturers Defect List** - The Manufacturers Defect List is supplied by the manufacturer and is resident on the disk drive.

**NOTE**

Usually, the user cannot directly access the Manufacturer's Defect List. If you need to do so, use the **READ DEFECT LIST** command (see subsection 6.).

**Certification Defect List** - The Certification Defect List is built by the controller while it is certifying the drive after the format operation. The controller does not perform a certification step during the format operation so this defect list will always be empty or null.

**Initiator Defect List** - The Initiator Defect List is supplied by the Initiator during the Data Out Phase of the **FORMAT UNIT** command. When the format operation is completed, this list becomes part of the Grown Defect List.

**Grown Defect List** - The Grown Defect List contains any defects which were re-assigned using the **RE-ASSIGN BLOCK** command.

The **FORMAT UNIT** command uses the drive geometry and format information read from a reserved area on the disk during power up to format the disk drive. These parameters may be changed using the **MODE SELECT** command just prior to issuing the **FORMAT UNIT** command. If the information contained in the reserved area is invalid or cannot be read, the **FORMAT UNIT** command will use the following drive geometry/format information (this is the default information returned by the **MODE SENSE** command):

**Number of Heads** - The number of heads will be obtained from the drive or controller switches.

**Number of Logical Cylinders** - The number of logical cylinders is equal to the number of physical cylinders minus three, minus the number of alternate cylinders. The number of physical cylinders will be obtained from the drive or controller switches.

**Number of Logical Sectors per Track** - The number of Logical Sectors per Track is equal to the number of physical sectors per track, minus the number of Alternate Sectors. The number of physical sectors per track will be obtained from the drive or controller switches.

**Physical Sector Size** (For soft-sectored drives only) - The Physical Sector Size will default to 256 or 512 bytes per sector depending on the setting of the controller switches. For hard sectored drives the physical sector size is obtained from the drive.

**Number of Alternate Cylinders** - The number of Alternate Cylinders defaults to 3.

**Number of Alternate Sectors per Track** - The number of Alternate Sectors per Track defaults to 1.

**Track-to-Track Sector Skew** - The controller uses the default head skew value of 0 if the disk drive indicates a head switch time that is less than 15 microseconds. When the disk drive indicates a head switch time that is greater than 15 microseconds, the head skew default value is 0Ah.

**Cylinder-to-Cylinder Sector Skew** - The Cylinder-to-Cylinder sector skew defaults to 0.

The **FORMAT UNIT** command writes all of the **MODE SELECT** parameters, including those mentioned above, to a reserved area on the disk inaccessible to the Initiator.

### 6.2.2.1 FORMAT UNIT Command Performance

The **FORMAT UNIT** command requires two disk revolutions to format each track.

### 6.2.2.2 FORMAT UNIT SCSI Deviations

The **FORMAT UNIT** command does not certify the drive after the format operation is completed.

The Vendor Unique Byte in the Command Descriptor Block is used to specify the data pattern to be written in the sector data area during the format operation.

### 6.2.2.3 FORMAT UNIT Command Parameters

The Command Descriptor Block for the **FORMAT UNIT** command is formatted as shown below.

| FORMAT UNIT Command Descriptor Block | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | Logical Unit Number | | | FmtDat | CmpLst | Defect List Format | | |
| 2 | Format Data Pattern | | | | | | | |
| 3 | Interleave (MSB) | | | | | | | |
| 4 | Interleave (LSB) | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Format Data (FmtDat)** bit <Byte 01, Bit 04>, when set to 1, indicates an Initiator Defect List will be supplied during the Data Out Phase of the command. The defect list specifies the defects to be entered into the Defect Map. When the FmtDat bit is set to 0, the Data Out Phase of the command does not occur and no defect information is supplied by the Initiator. If FmtDat is set to 0, both CmpLst and the Defect List Format must also be set to 0.

The **Complete List (CmpLst)** bit <Byte 01, Bit 03>, when set to 1, indicates the Initiator Defect List is the complete list of known defects. All previous defect information (if any) is erased (This action does not destroy the Manufacturers Defect List). The CmpLst bit is set to 1 to purge any previously specified defect list and to build a new defect list. If this bit is 0 (reformat mode), the set of known defects (all four defect lists) are combined with the Initiator Defect List supplied during the Data Out phase to form the new defect list.

The **Defect List Format** field <Byte 01, Bits 00:02> specifies the format of the Initiator Defect List, which is sent by the Initiator to the controller during the Data Out Phase. Table 6-1 lists the possible formats for the Initiator Defect List.

Table 6-1. Initiator Defect List Formats

| Defect List Fmt Bits 2 | 1 | 0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | **Format with Block Format.** The Initiator Defect List is in Block Format. The location of the blocks in the Defect List refers to the current block length and block addresses. |
| 1 | 0 | 0 | **Format with Bytes from Index Format.** The Initiator Defect List is in Bytes from Index Format. |
| 1 | 0 | 1 | **Format with Physical Sector Format.** The Initiator Defect List is in Physical Sector Format. |
| 1 | 1 | 0 | Vendor Unique (Reserved). |
| 1 | 1 | 1 | Reserved |

The **Format Data Pattern** field <Byte 2> specifies the pattern to be written in the data area when the drive is formatted. A value of 0 in this field specifies that the controller should use the default pattern of 43h.

The **Interleave** field <Bytes 03 through 04> specifies the interleave factor to be used for this format operation. The most significant byte of the Interleave field (Byte 03) must be 0. An interleave value of 0 or 1 requests that the Target use its default interleave (1:1 sequential).

### 6.2.2.4 Format Modes and Defect List Header

If the FmtDat bit is set to 1, the controller transfers the Defect List Header from the Initiator to the controller during the Data Out Phase of the **FORMAT UNIT** command. The Defect List Header is four-bytes long followed by zero or more Defect Descriptors. The header specifies the total number of bytes in the defect list and several parameters for the format mode.

| FORMAT UNIT Defect List Header Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Enable | DMDL | DCERT | SERR | Reserved | | | VU |
| 2 | Defect List Length (MSB) | | | | | | | |
| 3 | Defect List Length (LSB) | | | | | | | |

The **Enable** bit <Byte 1, Bit 7>, when set to 1, indicates the other fields in this byte are valid and may be set to 1 also. If this bit is set to 0, bits 1 through 6 must also be set to 0 and the controller will format the drive using the Manufacturers Defect List.

The **Disable Manufacturers Defect List (DMDL)** bit <Byte 1, Bit 6>, when set to 1, specifies the controller is to format the disk without using the Manufacturers Defect List. If this bit is 0, the controller will use the Manufacturers Defect List. This bit is only valid if the Enable bit is set to 1.

The **Disable Certification (DCERT)** field <Byte 1, Bit 5> is not supported by the controller and should be 0. The controller will not report an error if this field is set to 1 when the Enable field set to 1.

The **Stop on Error (SERR)** bit <Byte 1, Bit 4>, when set to 1, specifies that the controller should terminate the command when it encounters an unrecoverable error while accessing any of the defect lists. When this bit is 0, the controller will continue the format operation if any of the above errors occur. This bit is only valid if the Enable field is set to 1.

The **Vendor Unique (VU)** field <Byte 1, Bit 00> is reserved and must be set to 0.

The **Defect List Length** field <Bytes 02 through 03> specifies the total number of bytes (not the total number of defect descriptors) in the defect list. This length does not include the four bytes in the header. A Defect List Length of 0 is not considered an error by the controller.

### 6.2.2.5 Format Modes

The format mode for the controller is specified with the following bits: FmtDat, CmpLst, DMDL, and DCERT. If the Enable bit in the Defect List Header is 0 (not enabled), the controller treats the DMDL and DCERT fields as if they were set to 0 and 1 respectively. Table 6-2 lists the format modes supported by the controller and their corresponding states for the four bits.

Table 6-2. Controller Format Modes

| Fmt Dat | Cmp Lst | DMDL | DCERT | Format Mode |
|---|---|---|---|---|
| 0 | 0 | 0 | X | The controller formats the drive with no Initiator Defect List. The previous Grown Defect List is used. There is no Data Phase for this format mode. |
| 1 | 0 | 0 | X | The controller reformats the drive using the Manufacturers Defect List, Initiator Defect List and the current Grown Defect List to produce the full set of known defects. |
| 1 | 1 | 0 | X | The controller formats the drive using the Manufacturers Defect List and Initiator Defect List to produce the full set of known defects. The previous Grown Defect List is purged. |
| 1 | 0 | 1 | X | The controller reformats the drive using the Initiator Defect List and the current Grown Defect List to produce the full set of known defects. |
| 1 | 1 | 1 | X | The controller formats the drive using the Initiator Defect List to produce the full set of known defects. The previous Grown Defect List is purged. |

### 6.2.2.6 Defect List Block Format

The Defect List is transferred from the Initiator to the controller during the Data Out Phase of the **FORMAT UNIT** command.

**NOTE**

This Defect List Format is not a well-defined format by the SCSI standard and may vary between manufacturers. It is recommended that the Bytes from Index or Physical Sector Defect List formats be used during the format operation.

The defect list begins with a 4-byte header followed by zero or more four-byte Defect Descriptors. When the defect list is specified using Block Addresses, the controller uses following format:

| FORMAT UNIT Defect List Block Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Enable | DMDL | DCERT | SERR | Reserved | | | VU |
| 2 | Defect List Length (MSB) | | | | | | | |
| 3 | Defect List Length (LSB) | | | | | | | |
| | Defect Descriptor(s) | | | | | | | |
| 0 | Defect Logical Block Address (MSB) | | | | | | | |
| 1 | Defect Logical Block Address | | | | | | | |
| 2 | Defect Logical Block Address | | | | | | | |
| 3 | Defect Logical Block Address (LSB) | | | | | | | |

The **Defect List Logical Block Address** field <Bytes 0 through 3 of the Defect Descriptor) specify the address of the logical block that contains the defect. The location of the blocks in the Defect List refers to the current block length and block addresses.

Each defect descriptor for the Block Format mode specifies a 4-byte defect logical block address that contains the defect. The defect descriptors must be in ascending order.

### 6.2.2.7 Defect List Bytes From Index Format

The Defect List is transferred from the Initiator to the controller during the Data Out Phase of the **FORMAT UNIT** command. The defect list begins with a 4-byte header followed by zero or more 8-byte Defect Descriptors. When the defect list is specified using the number of Bytes from the Index, the controller uses the following format:

| FORMAT UNIT Defect List Bytes From Index Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Enable | DMDL | DCERT | SERR | Reserved | | | VU |
| 2 | Defect List Length (MSB) | | | | | | | |
| 3 | Defect List Length (LSB) | | | | | | | |
| | Defect Descriptor(s) | | | | | | | |
| 0 | Cylinder Number of Defect (MSB) | | | | | | | |
| 1 | Cylinder Number of Defect | | | | | | | |
| 2 | Cylinder Number of Defect (LSB) | | | | | | | |
| 3 | Head Number of Defect | | | | | | | |
| 4 | Defect Bytes from Index (MSB) | | | | | | | |
| 5 | Defect Bytes from Index | | | | | | | |
| 6 | Defect Bytes from Index | | | | | | | |
| 7 | Defect Bytes from Index (LSB) | | | | | | | |

The **Cylinder Number of Defect** field <Bytes 0 through 2 of the Defect Descriptor> specifies the physical cylinder number which contains the defect.

The **Head Number of Defect** field <Byte 3 of the Defect Descriptor> specifies the head number which contains the defect.

The **Defect Bytes from Index** field <Bytes 4 through 7 of the Defect Descriptor> specifies the number of bytes between the index and the defect on the specified track. A value of 0FFFFFFFF (hexadecimal) in the Bytes from Index field indicates that the entire track is to be reassigned.

The defect descriptors must be in ascending order. For determining ascending order, the Cylinder Number of Defect is considered the most significant part of the address and the Defect Bytes from Index is considered the least significant part of the address.

### 6.2.2.8 Defect List Physical Sector Format

The Defect List is transferred from the Initiator to the controller during the Data Out Phase of the **FORMAT UNIT** command. The defect list begins with a 4-byte header followed by zero or more 8-byte Defect Descriptors. When the defect list is specified using Physical Sector addresses, the controller uses the following format:

| FORMAT UNIT Defect List Physical Sector Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Enable | DMDL | DCERT | SERR | Reserved | | | VU |
| 2 | Defect List Length (MSB) | | | | | | | |
| 3 | Defect List Length (LSB) | | | | | | | |
| | Defect Descriptor(s) | | | | | | | |
| 0 | Cylinder Number of Defect (MSB) | | | | | | | |
| 1 | Cylinder Number of Defect | | | | | | | |
| 2 | Cylinder Number of Defect (LSB) | | | | | | | |
| 3 | Head Number of Defect | | | | | | | |
| 4 | Defect Sector Number (MSB) | | | | | | | |
| 5 | Defect Sector Number | | | | | | | |
| 6 | Defect Sector Number | | | | | | | |
| 7 | Defect Sector Number (LSB) | | | | | | | |

The **Cylinder Number of Defect** field <Bytes 0 through 2 of the Defect Descriptor> specifies the physical cylinder number which contains the defect.

The **Head Number of Defect** field <Byte 3 of the Defect Descriptor> specifies the head number which contains the defect.

The **Defect Sector Number** field <Bytes 4 through 7 of the Defect Descriptor> specifies the sector number which contains the defect. A value of 0FFFFFFFF (hexadecimal) in the Defect Sector Number field indicates that the entire track is to be reassigned.

The defect descriptors must be in ascending order. For determining ascending order, the Cylinder Number of Defect is considered the most significant part of the address and the Defect Sector Number is considered the least significant part of the address.

### 6.2.2.9 FORMAT UNIT Error Conditions

If the Interleave value exceeds the maximum number of sectors per track, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the Format Mode is invalid, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the Stop on Error bit is 1 and the controller encounters an unrecoverable error reading or accessing a defect list, the controller will terminate the command with a CHECK CONDITION status and a MEDIUM ERROR/DEFECT LIST ERROR (03h/19h) Sense Key/Error Code.

If the controller has insufficient capacity to re-assign all the defective blocks, it will terminate the command with a CHECK CONDITION status and a MEDIUM ERROR/NO DEFECT SPARE LOCATION AVAILABLE (03h/32h) Sense Key/Error Code. This error indicates that there were not enough spare tracks available for the requested format. To correct this, increase the number of spare tracks as needed.

If the Stop on Error bit is 0 and the controller encounters an error while accessing a defect list, it will continue the **FORMAT UNIT** command. When the command is completed, and no other errors occurred, it will terminate with a CHECK CONDITION status and an RECOVERED ERROR/DEFECT LIST ERROR (01h/19h) Sense Key/Error Code.

If the Enable bit in the Defect List Header is 0 and the DMDL, DCERT, and SERR bits are not 0, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the SERR bit is 1, the DMDL bit is 0, and the controller cannot locate the manufacturer's defect list, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/PRIMARY DEFECT LIST NOT FOUND (05h/1Ch) Sense Key/Error Code.

If the SERR bit is 0, the DMDL bit is 0, and the controller cannot locate the manufacturer's defect list, it will continue the **FORMAT UNIT** command. When the command is completed, and no other errors have occurred, the controller will terminate with a CHECK CONDITION status and an RECOVERABLE ERROR/PRIMARY DEFECT LIST NOT FOUND (01h/1Ch) Sense Key/Error Code.

### 6.2.2.10 **FORMAT UNIT** Sense Information

Errors detected by the controller during the **FORMAT UNIT** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

| FORMAT UNIT Nonextended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

| FORMAT UNIT Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Logical Block Address (MSB) | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address | | | | | | | |
| 6 | Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to 1, indicates the Logical Block Address information is valid. If the Valid bit is not 1, the Logical Block information should be ignored.

The **Logical Block Address** field <Bytes 3 through 6> is the first invalid block address detected by the **FORMAT UNIT** command. This address will be specified in the same format as the supplied Defect List.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **FORMAT UNIT** command the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | Recoverable Read Error With Retries (17h) |
| | Recoverable Read Error With ECC (18h) |
| | Defect List Error (19h) |
| | Primary Defect List Not Found (1Ch) |
| Not Ready (02h) | Drive Not Ready (04h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID field Address Mark (12h) |
| | No Data field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| | Format Failed (31h) |
| | No Defect Spare Location Available (32h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Write Fault (03h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Primary Defect List Not Found (1Ch) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| | Invalid Field in Parameter List (26h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Data Protect (07h) | Write Protected (27h) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.2.3 INQUIRY - 12H

The **INQUIRY** command provides a means by which the Initiator may request information regarding the controller and its attached peripheral device(s).

If an **INQUIRY** command is received from an Initiator with a pending Unit Attention Condition, the controller will execute the **INQUIRY** command, return a GOOD status and will **not** clear the Unit Attention Condition.

#### 6.2.3.1 INQUIRY Command Operation

When the **INQUIRY** command is sent to a nonexistent LUN, the controller will transfer the **INQUIRY** data back to the Initiator and will terminate the command with a GOOD status. The Initiator must examine the Device Type Qualifier field to determine if it is a valid LUN.

#### 6.2.3.2 INQUIRY SCSI Deviations

There are no deviations.

#### 6.2.3.3 INQUIRY Command Parameters

The **INQUIRY** Command Descriptor Block is formatted as shown below.

**INQUIRY Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Allocation Length | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Allocation Length** field <Byte 4> specifies the number of bytes that the Initiator has allocated for the returned **INQUIRY** data. An allocation length of 0 indicates no data will be transferred to the Initiator and is not considered an error. Any other value indicates the maximum number of bytes that will

be transferred. The controller will terminate the data transfer when the number of bytes specified in the Allocation Length field are transferred or all available **INQUIRY** data has been transferred, whichever is less.

### 6.2.3.4 INQUIRY Data Format

The **INQUIRY** command will return 36 bytes of data to the Initiator. This data is formatted as follows:

**INQUIRY Command Data Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Peripheral Device Type | | | | | | | |
| 1 | RMB | Device Type Qualifier | | | | | | |
| 2 | ISO Version | | ECMA Version | | | ANSI Version | | |
| 3 | Reserved | | | | Response Data Format | | | |
| 4 | Additional Length (1Fh) | | | | | | | |
| 5 | Request Sense Length | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Reserved | | | | | | | |
| 8 thru 15 | Vendor Identification | | | | | | | |
| 16 thru 31 | Product Identification | | | | | | | |
| 32 thru 35 | Revision Level | | | | | | | |

The **Peripheral Device Type** field <Byte 0> will be 0 to indicate this is a direct-access device (disk drive). If the specified LUN is nonexistent, this field will be set to 07Fh. The controller does not support other types of devices. A nonexistent LUN is a LUN not supported by the controller. For example, on a controller which supports 4 LUNs (0-3), only LUNs 4 through 7 would return a nonexistent LUN device type qualifier. An **INQUIRY** command to LUNs 0 through 3 would always return a Peripheral Device Type of 0, even if no device is attached to the controller.

The **Removable Medium (RMB)** bit <Byte 1, bit 7> will always be set to 0 because the controller does not support removable media.

The **Device Type Qualifier** field <Byte 1, Bits 6:0> will always be 0 because the controller supports a direct-access device.

The **ISO Version** field <Byte 2, Bits 6:7> will always be 0 since the controller does not comply with the ISO SCSI standard.

**NOTE**

The ISO SCSI standard is currently a draft proposal and is not a standard.

The **ECMA Version** <Byte 2, Bits 3:5> will always be 0 to indicate the controller does not comply with the ECMA SCSI standard.

The **ANSI Version** <Byte 2, Bits 0:2> is currently 0 to indicate the controller does not comply with the ANSI SCSI standard. This field will be changed to a 1 when the ANSI SCSI Standard is published.

The **Response Data Format** field <Byte 3, Bits 0:3> will always be set to 01h, indicating this controller conforms to the SCSI Common Command Set for direct-access devices.

The **Additional Length** field <Byte 4> defines the number of parameter bytes which follow the 5-byte header. The controller will always set this byte to 31 (1Fh).

The **Request Sense Length** field <Byte 05> indicates the number of Extended Sense bytes normally returned by the controller, not including the **COPY** and **COMPARE** commands. This field will always be set to 18 (12h).

The **Vendor Identification** field <Bytes 8 through 15> will always be set to the ASCII string "EMULEX " (8 characters). The string will be left justified and filled with ASCII spaces (20h).

The **Product Identification** field <Bytes 16 through 31> will contain the Emulex Product Identifier in ASCII followed by the product name, left justified and filled with ASCII spaces (20h). There are 16 characters total in the string. Current Emulex products will have the following character strings:

- The MD21 Controller will have the ASCII string "MD21/S2 ESDI" in this field
- The MD23 Controller will have the ASCII string "MD23/S2 ESDI" in this field
- The MD24 Controller will have the ASCII string "MD24/S2 ESDI" in this field.
- The MD32 Controller will have the ASCII string "MD32/S2 SMD" in this field
- The MD33 Controller will have the ASCII string "MD33/S2 SMD" in this field.

The **Revision Level** field <Bytes 32 through 35> will contain the Emulex Firmware Revision Level in ASCII hex. There are 4 characters total in the string. For example, "A0F " specifies revision "A", engineering release "0F".

### 6.2.3.5 INQUIRY Sense Information

Errors detected by the controller during the **INQUIRY** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**INQUIRY Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**INQUIRY Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Additional Sense Length** field <Byte 7> specifies the number of additional sense bytes which follow. For the **INQUIRY** command the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Hardware Error (04h) | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.2.4 MODE SELECT - 15H

The **MODE SELECT** command provides a means by which the Initiator may specify medium, logical unit and/or peripheral device parameters to the controller. Any changes in the **MODE SELECT** parameters take effect immediately after the **MODE SELECT** command has terminated. The **MODE SELECT** is a complementary command to the **MODE SENSE** command, which allows the Initiator to request that the controller send it the current values for the parameters.

The Initiator can send the controller optional blocks of parameters that are separated into categories (or pages). The individual pages specify various options and features which the Initiator may change.

Pages 1 (Error Recovery Parameters), 2 (Disconnect/Reconnect Parameters), 3 (Direct-Access Format Parameters), and 4 (Disk Drive Geometry Parameters) of the **MODE SELECT** data are written to the disk when the **FORMAT UNIT** command is executed. Subsequent controller power-up or reset conditions will cause the controller to read this information from the disk drive.

The Initiator should send to the controller those pages for which it requests parameters to be changed. The Initiator may send all the pages at one time and they do not need to be sent in any particular order.

#### 6.2.4.1 MODE SELECT Command Usage

It is recommended that prior to issuing a **MODE SELECT** command to the controller, the Initiator issue a **MODE SENSE** command with a page code of 3Fh in byte 2 of the CDB and a Page Code Field (PCF) of 00, to determine which pages are implemented by the controller for that particular LUN and the length of the pages. (The PCF value of 00 allows the Initiator to determine what current values are defined in the **MODE SENSE** command).

When a **MODE SELECT** command is issued that changes any parameters in Pages 3 (Direct-Access Format Parameters) and 4 (Disk Drive Geometry Parameters), the controller will issue a CHECK CONDITION status with a Sense Key/Error Code of UNIT ATTENTION/MODE SELECT CHANGED CONDITION (06h/2Ah) to the first command received from all Initiators, except the one that issued the **MODE SELECT** command.

### 6.2.4.2 MODE SELECT Parameter Values

The controller maintains the following three different sets of **MODE SELECT** data:

- **Default values:** The default values are stored in the controller PROM, the disk drive, and the controller switches.
- **Saved values:** The saved values are all the changeable **MODE SELECT** parameters saved by the controller on the disk drive when it performs a **FORMAT UNIT** command or a **MODE SELECT** command with the SMP bit (in the CDB) set to 1.
- **Current values:** The current values are all the changeable **MODE SELECT** parameters used by the controller during normal controller operations. Any **MODE SELECT** command issued to the controller changes the current values.

At initialization time (a power-on or reset condition has occurred, see Section 4), the controller copies the default values into the current values. After the disk drive spins up, the controller will read the saved values from the drive and copy them into the current values.

When the controller completes a **FORMAT UNIT** command, it writes all supported pages to the disk drive. On an unformatted disk drive, the current parameters are the same as the default parameters. The Initiator may then change the current parameters using a **MODE SELECT** command prior to formatting the disk drive. When the **FORMAT UNIT** command completes, the controller writes the current values (which may have been changed by a **MODE SELECT** command) to the disk drive as the saved values.

The Initiator may change the saved values of pages 1 (Error Recovery Parameters) and 2 (Disconnect/Reconnect Parameters) by issuing a **MODE SELECT** command with the SMP bit in the CDB set to 1. This action will not change the saved values of Pages 3 and 4.

**NOTE**

If the drive has never been formatted, or the Saved **MODE SELECT** configuration information has been destroyed, and the Initiator issues a **MODE SELECT** command with the SMP bit set to 1, the controller will terminate the command with a CHECK CONDITION status.

### 6.2.4.3 MODE SELECT SCSI Deviations

Only a single Block Descriptor may be sent to a specified Logical Unit.

The controller ignores the Number of Blocks field in the Block Descriptor.

### 6.2.4.4 MODE SELECT Command Parameters

The **MODE SELECT** Command Descriptor Block is formatted as shown below.

**MODE SELECT Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | Logical Unit Number | | | PF | Reserved | | | SMP |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Parameter List Length | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Page Format (PF)** bit <Byte 1, Bit 04> is set to 1 to indicate that the data sent by the initiator after the **MODE SELECT** Header and the Block Descriptors (if any) complies to the Page Format. It is set to 0 to indicate that the data sent by the initiator after the **MODE SELECT** Header and the Block Descriptors (if any) is vendor unique. The controller does not support vendor unique **MODE SENSE** data.

The **Save Mode Parameters (SMP)** bit <Byte 1, Bit 0>, when set to 1, specifies that the controller should take the current values for Pages 1 (Error Recovery Parameters) and 2 (Disconnect/ Reconnect Parameters) and write them to the disk as the saved values. Before the controller saves the parameters, it will make any changes to these pages as specified in the current **MODE SELECT** command. If the controller encounters an error during the **MODE SELECT** command, it will terminate the command without writing the parameters to the disk as the saved values. If the SMP bit is 0, the controller will update the current values and will not modify the saved values.

The **Parameter List Length** field <Byte 4> specifies the length in bytes of the parameters that will be sent from the Initiator to the controller during the data phase of the **MODE SELECT** command. A Parameter List Length of 0 indicates that no data will be transferred and is not considered an error by the controller.

### 6.2.4.5 MODE SELECT Parameter List Format

The **MODE SELECT** Parameter List is sent by the Initiator to the controller during the Data Out Phase. This list consists of a Parameter List Header, zero or one Block Descriptor and zero or more Page Descriptors. The entire length of the Parameter List is specified in the **MODE SELECT** Command Descriptor Block.

The Parameter List Header is four bytes and specifies the media type and the length of the Block Descriptor.

The Block Descriptor consists of eight bytes and specifies the medium density, the number of blocks, and the logical block length.

The Page descriptors contains various parameters separated into categories (or pages). These parameters specify various options and features which the Initiator may change. One or more pages may be sent during the **MODE SELECT** command.

6.2.4.5.1 Parameter List Header Format. The **MODE SELECT** Parameter List Header is the first part of the Parameter List. The Header is formatted as shown below.

| **MODE SELECT Parameter List Header** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Medium Type | | | | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Block Descriptor Length | | | | | | | |

The **Media Type** field <Byte 1> must be set to 0 (00h) to indicate the controller should use the Default Media Type.

The **Block Descriptor Length** field <Byte 3> specifies the length of the Block Descriptor in bytes, starting at Byte 04 of the parameter list. The controller will support zero or one Block Descriptor per **MODE SELECT** command, therefore, the only valid Block Descriptor lengths are 0 or 8 bytes.

6.2.4.5.2 Parameter List Block Descriptor Format. The **MODE SELECT** Parameter List Block Descriptor immediately follows the Parameter List Header. The controller will not report an error if the Block Descriptor is not in the Parameter List (Block Descriptor Length equals 0). The Block Descriptor is formatted as shown below.

| MODE SELECT Parameter List Block Descriptor Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Density Code | | | | | | | |
| 1 | Number of Logical Blocks (MSB) | | | | | | | |
| 2 | Number of Logical Blocks | | | | | | | |
| 3 | Number of Logical Blocks (LSB) | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Logical Block Length (MSB) | | | | | | | |
| 6 | Logical Block Length | | | | | | | |
| 7 | Logical Block Length (LSB) | | | | | | | |

The **Density Code** field <Byte 0> defines the density of the media on the addressed LUN. The Density Code has a value of 0 to indicate the LUN is a hard disk drive.

The **Number of Logical Blocks** field <Bytes 1 through 3> specifies the number of logical blocks in the user's address space. The controller ignores this field.

The **Logical Block Length** field <Bytes 5 through 7> specifies the length of the logical block in bytes. The Block Length must be equal to, or an exact multiple of, the physical sector size. The controller supports logical block sizes of 256, 512, 1024, 2048, and 4096 bytes.

Due to the constraints of the **RE-ASSIGN BLOCK** command, it is strongly recommended that the logical block size be the same as the physical sector size. When the logical block size exceeds the physical sector size, the **RE-ASSIGN BLOCK** command must re-assign all the physical sectors contained in the logical block. If the number of sectors per logical block exceeds the number of spare sectors per track, the entire track will be re-assigned whenever a **RE-ASSIGN BLOCK** command is executed.

**NOTES**

Data transfers have a maximum length of 65,535 **physical** sectors. If the logical block size is greater than the physical sector size, be sure the maximum transfer length is correspondingly smaller.

If page 3 is sent without a Block Descriptor Format, Logical Block Length defaults to the physical sector size.

6.2.4.5.3 Format of the Page Header. Each of the optional Page Descriptors is preceded by a Page Header. The Page Header is two bytes long and identifies the page type and the length and is formatted as shown below. The Page Header is immediately followed by the Page Parameters.

| MODE SELECT Page Header Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | Page Code | | | | | |
| 1 | Page Length | | | | | | | |

The **Page Code** field <Byte 0, Bits 0:5> identifies the page type. Table 6-3 lists the page codes and their corresponding page descriptions.

Table 6-3. **MODE SELECT** Page Codes

| Page Code | Page Description |
|---|---|
| 00h | Vendor Unique (not used by the controller) |
| 01h | Error Recovery Parameters |
| 02h | Disconnect/Reconnect Control Parameters |
| 03h | Direct Access Device Format Parameters |
| 04h | Rigid Disk Drive Drive Geometry Parameters |
| 05-1Fh | Reserved |
| 20-39h | Vendor Unique |
| 3A-3Bh | Reserved |
| 3C-3Eh | Vendor Unique (reserved for use in the **MODE SENSE** command) |
| 3Fh | Reserved for use in **MODE SENSE** Command |

The **Page Length** field <Byte 1> specifies the number of bytes in the page, not including the Page Length byte. The Initiator **must** send the entire page to the controller.

6.2.4.5.4 Format of Error Recovery Parameter Page. This section specifies the **MODE SELECT** error recovery options supported by the controller. The format for the Error Recovery Parameters Page is shown below. A copy of each of these parameters is saved for each Initiator. This allows any Initiator to change its own parameters without affecting the parameters of any other Initiator.

**NOTE**

The controller will save this page whenever the Initiator issues a **FORMAT UNIT** command or a **MODE SELECT** command with the SMP bit set to 1.

| MODE SELECT Error Recovery Parameter Page Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | Page Code = 1h | | | | | |
| 1 | Maximum Page Length = 6h | | | | | | | |
| 2 | EARW | EARR | TB | RC | EEC | RRERR | TRERR | DCOR |
| 3 | Retry Count | | | | | | | |
| 4 | Correction Span | | | | | | | |
| 5 | Head Offset Count | | | | | | | |
| 6 | Data Strobe Offset Count | | | | | | | |
| 7 | Recovery Time Limit | | | | | | | |

The **Enable Automatic Reallocation on Write (EARW)** field <Byte 2, Bit 7>, when set to 1, instructs the controller to automatically re-assign bad blocks when it encounters them during a write operation. This function is similar to that of the **RE-ASSIGN BLOCK** command, but is initiated by the controller. When the EARW bit is 0, the controller will not automatically re-assign bad blocks when it encounters them. **This option is not supported by the controller and must be set to 0.**

The **Enable Automatic Reallocation on Read (EARR)** field <Byte 2, Bit 6>, when set to 1, instructs the controller to automatically re-assign bad blocks when it encounters them during a read operation. This function is similar to that of the **RE-ASSIGN BLOCK** command, but is initiated by the controller. When the EARR bit is 0, the controller will not automatically re-assign bad blocks when it encounters them. **This option is not supported by the controller and must be set to 0.**

The **Transfer Block (TB)** field <Byte 2, Bit 5>, when set to 1, specifies that the controller should transfer the block with the data error before terminating the command. This bit is only applicable when a hard error is encountered or the TRERR bit is set to 1 and a recoverable error is encountered. If this field is 0, the controller will not transfer the block with the data error. In both cases, the controller reports the block address of the block with the error, rather than that of the preceding block, in the sense data. The Transfer Block field defaults to 1. If the transfer terminates with other than a data error (i.e., data not found), the controller will not transfer the block.

The **Read Continuous (RC)** field <Byte 2, Bit 4>, when set to 1, overrides the Enable Early Correction (EEC), Terminate on Recoverable Error (TRERR), Report Recoverable Error (RRERR), and Disable Correction (DCOR) fields and disables all retries and data correction. The Transfer Block (TB) field is not applicable. When the RC bit is set to 1, the controller transfers the entire requested length of data without adding

delays that are caused by its error recovery schemes. The controller will send data which may be erroneous or fabricated to maintain a continuous flow of data and avoid delays. The RC field defaults to 0.

The **Enable Early Correction (EEC)** field <Byte 2, Bit 3> specifies that the controller should perform a minimum number of retries before applying any correction algorithm. When this bit is set to 1, the controller will not exhaust the retry count before attempting any ECC correction. When this bit is 0, the controller will exhaust the retry count (as specified in Byte 3) before it attempts any ECC correction. Also, the controller will attempt to recover the data using head offset and data strobe offset before attempting ECC correction. The Enable Early Correction field defaults to 0.

**NOTE**

This field does not disable retries during seek operations. For any seek or positioning error, the controller will issue a recalibrate command to the drive then retry the seek operation. If the second attempt also fails, the controller will terminate the command.

The **Report Recoverable Error (RRERR)** field <Byte 2, Bit 2>, when set to 1, instructs the controller to report any recoverable errors to the Initiator. This error will either be reported immediately, or at the normal completion of the command depending on the state of the TRERR field. The error actually reported to the Initiator will be the last error encountered during the data transfer. If multiple errors occur, the controller reports (in the sense information) the block address of either: (1) the last block where the recovered error occurred or, (2) the block with the first unrecoverable error. The RRERR field defaults to 1. The controller may terminate the data transfer before all data has been transferred, depending on the error encountered and the states of the other error recovery bits in Byte 2.

The **Terminate on Recoverable Error (TRERR)** field <Byte 2, Bit 1>, when set to 1 and the RRERR bit is set to 1, instructs the controller to terminate the command immediately when a recoverable error is encountered and create the CHECK CONDITION status. The controller may or may not transfer the data contained in the block in error, depending on the setting of the TB bit. The Initiator can only set the TRERR bit to 1 if it has set the RRERR bit to 1. If the TRERR bit is 0, the controller will continue the data transfer when a recoverable error is encountered. The Terminate on Recoverable Error field defaults to 0.

The **Disable Correction (DCOR)** field <Byte 2, Bit 0>, when set to 1, disables ECC correction when reading a sector from the disk drive. No correction is attempted and if a correctable ECC error occurs, it is treated as a recoverable error. This field defaults to 0.

The **Retry Count** field <Byte 3> specifies the maximum number of retries to attempt when an error is encountered. The controller supports Retry Counts from 0 to 255 (decimal) and defaults to 10.

The **Correction Span** field <Byte 4> specifies the largest read data error, in bits, on which correction may be attempted. The controller supports a Correction Span of 0 to 19 decimal bits and defaults to 12 if the sector size is less than 768 bytes or 10 if the sector size is greater than or equal to 768 bytes.

The **Head Offset Count** field <Byte 5> specifies the incremental offset, from the center of the track, to use when performing a disk operation. When set to 0, the controller will not apply any offset. When set to a positive value, the controller will apply the offset in an increasing distance in the direction of the next physical track, which could be toward the innermost or outermost track (depending on whether track 0 was located on the innermost or outermost track of the drive). When set to a two's complement, the controller will apply the offset in an increasing distance in the opposite direction of a positive value. **This option is not supported by the controller and must be set to 0.**

The **Data Strobe Offset Count** field <Byte 6> specifies an incremental position the controller will use to adjust the recovered data strobe from the nominal. When this field is 0, the controller will make no adjustment to the nominal position. When set to a positive value, the controller will adjust the recovered data strobe in the positive direction as defined by the device (e.g., moving the data strobe out in time by some number of nanoseconds). When set to a two's complement, the controller will adjust the recovered data strobe in the negative direction. **This option is not supported by the controller and must be set to 0.**

The **Recovery Time Limit** field <Byte 7> specifies the maximum amount of time that the controller will attempt to recover an error. The field is defined in increments of 10 milliseconds. When set to 0, the controller will use an unlimited time. **This option is not supported by the controller and must be set to 0.**

6.2.4.5.4.1 Error Recovery Modes. Table 6-4 lists the possible error recovery modes which may occur using the above parameters. Those combinations which do not provide any useful function, i.e. terminate on errors but do not report them, are marked as Invalid Mode and should not be selected by the Initiator. The Enable Early Correction (EEC) field modifies each one of these modes to use a limited set of retries before applying any correction algorithm.

Table 6-4. **MODE SELECT** Error Recovery Modes

| EEC | RRERR | TRERR | DCOR | Description |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **Retries Then Correction.** Retries will be exhausted, ECC correction will be attempted, and recovered errors will not be reported. The transfer will terminate prematurely if an unrecoverable error is encountered. The Transfer Block (TB) field specifies whether or not to transfer the block which contained the data error. |
| 0 | 0 | 0 | 1 | **No Correction.** Retries will be exhausted, no ECC correction is attempted and recovered errors will not be reported. The transfer will terminate prematurely if an unrecoverable error is encountered. The Transfer Block (TB) field specifies whether or not to transfer the block which contained the data error. |
| 0 | 0 | 1 | 0 | **Invalid Mode.** The Initiator should not use this mode. |
| 0 | 0 | 1 | 1 | **Invalid Mode.** The Initiator should not use this mode. |

(continued on next page)

Table 6-4. **MODE SELECT** Error Recovery Modes (continued)

| EEC | RRERR | TRERR | DCOR | Description |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | **Report Recovered Errors.** Retries will be exhausted, ECC correction will be attempted, and recovered errors will be reported. The transfer will terminate prematurely if an unrecoverable error is encountered. The Transfer Block (TB) field specifies whether or not to transfer the block which contained the data error. This is the default error recovery mode for the controller. |
| 0 | 1 | 0 | 1 | **Report Errors, No Correction.** Retries will be exhausted, and no ECC correction is attempted and recovered errors will be reported. The transfer will terminate prematurely if an unrecoverable error is encountered. The Transfer Block (TB) field specifies whether or not to transfer the block which contained the data error. |
| 0 | 1 | 1 | 0 | **Report Errors and Stop.** Retries will be exhausted, ECC correction will be attempted, and recovered errors will be reported. The transfer will terminate prematurely if any error occurs. The Transfer Block (TB) field specifies whether or not to transfer the block which contained the data error. |

(continued on next page)

Table 6-4. **MODE SELECT** Error Recovery Modes (continued)

| EEC | RRERR | TRERR | DCOR | Description |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | **Report Errors, No Correction, Stop.** Retries will be exhausted, no ECC correction will be attempted, and recovered errors will be reported. The transfer will terminate prematurely if any error occurs. The Transfer Block (TB) field specifies whether or not to transfer the block which contained the data error. |
| 1 | 0 | 0 | 0 | **Early Correction.** ECC correction will be attempted before the controller attempts to recover data with retries. Any recovered errors will not be reported. The transfer will terminate prematurely if an unrecoverable data error is encountered. The Transfer Block (TB) field specifies whether or not to transfer the block which contained the data error. |
| 1 | 0 | 0 | 1 | **Invalid Mode.** The Initiator should not use this mode. |
| 1 | 0 | 1 | 0 | **Invalid Mode.** The Initiator should not use this mode. |
| 1 | 0 | 1 | 1 | **Invalid Mode.** The Initiator should not use this mode. |
| 1 | 1 | 0 | 0 | **Report Recovered Errors with ECC First.** ECC correction will be attempted before the controller attempts to recover data with retries. Any recovered errors will be reported. The transfer will terminate prematurely if an unrecoverable error is encountered. The Transfer Block (TB) field specifies whether or not to transfer the block which contained the data error. |

Table 6-4. **MODE SELECT** Error Recovery Modes (continued)

| EEC | RRERR | TRERR | DCOR | Description |
|---|---|---|---|---|
| 1 | 1 | 0 | 1 | **Invalid Request.** The Initiator should not use this mode. |
| 1 | 1 | 1 | 0 | **Report Errors and Stop with ECC First.** ECC correction will be attempted before the controller attempts to recover data with retries. Any recovered errors will be reported. The transfer will terminate prematurely if any error occurs. The Transfer Block (TB) field specifies whether or not to transfer the block which contained the data error. |
| 1 | 1 | 1 | 1 | **Invalid Mode.** The Initiator should not use this mode. |

6.2.4.5.5 Format of Disconnect/Reconnect Parameter Page. This section specifies the **MODE SELECT** Disconnect/Reconnect Parameter options supported by the controller. These options modify the Disconnect/Reconnect strategy built into the controller. Refer to subsection 5.6, Disconnect/Reconnect Strategy, for more information. The format for the Disconnect/Reconnect Parameter Page is shown below. The controller saves a copy of each of these parameters for each Initiator. This action allows any Initiator to change its own parameters without affecting the parameters of any other Initiator.

**NOTE**

The controller will save this page whenever the Initiator issues a **FORMAT UNIT** command or a **MODE SELECT** command with the SMP bit set to 1.

| MODE SELECT Disconnect/Reconnect Parameter Page Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | Page Code = 2h | | | | | |
| 1 | Maximum Page Length = 0Ah | | | | | | | |
| 2 | Buffer Full Ratio | | | | | | | |
| 3 | Buffer Empty Ratio | | | | | | | |
| 4 | Bus Inactivity Limit (MSB) | | | | | | | |
| 5 | Bus Inactivity Limit (LSB) | | | | | | | |
| 6 | Disconnect Time Limit (MSB) | | | | | | | |
| 7 | Disconnect Time Limit (LSB) | | | | | | | |
| 8 | Connect Time Limit (MSB) | | | | | | | |
| 9 | Connect Time Limit (LSB) | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |

The **Buffer Full Ratio** field <Byte 1> specifies how full the internal buffer should be before the controller reconnects to transfer the data to the Initiator. **This option is not supported by the controller and must be set to 0.**

The **Buffer Empty Ratio** field <Bytes 2 and 3> specifies how empty the internal buffer should be before the controller reconnects to transfer more data from the Initiator. **This option is not supported by the controller and must be set to 0.**

The **Bus Inactivity Limit** field <Bytes 4 and 5> specifies the length of time, in 100 microsecond increments, that the controller is allowed to stay connected to the SCSI bus without any bus activity. The controller supports a range of 1 (100 microseconds) to 650 (65000 microseconds). A value of zero in this field specifies that the controller may stay connected to the bus indefinitely. The default value for this field is 5 (500 microseconds).

The **Disconnect Time Limit** field <Bytes 6 and 7> specifies the length of time, in 10 microsecond increments, that the controller should remain disconnected before attempting to reconnect. **This option is not supported by the controller and must be set to 0.**

The **Connect Time Limit** field <Bytes 8 and 9> specifies the maximum length of time, in 10 increments, the controller should remain connected before it attempts to disconnect. **The controller does not support this option and the field must be set to 0.**

6.2.4.5.6 Format of Direct-Access Device Format Parameter Page. This section specifies the **MODE SELECT** Direct Access Device Format Parameters supported by the controller. The format for the Disk Format Parameter Page is shown below. For the Handling of Defects fields (Bytes 2 through 9), the controller may not accept the exact value requested by the Initiator for these fields. Instead, it may round to its nearest most convenient value (which may be 0).

**NOTE**

The controller will save this page whenever the Initiator issues a **FORMAT UNIT** command.

**MODE SELECT Direct Access Device Format Parameter Page**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | Page Code = 3h | | | | | |
| 1 | Maximum Page Length = 16h | | | | | | | |
| 2 | Tracks per Zone (MSB) | | | | | | | |
| 3 | Tracks per Zone (LSB) | | | | | | | |
| 4 | Alternate Sectors per Zone (MSB) | | | | | | | |
| 5 | Alternate Sectors per Zone (LSB) | | | | | | | |
| 6 | Alternate Tracks per Zone (MSB) | | | | | | | |
| 7 | Alternate Tracks per Zone (LSB) | | | | | | | |
| 8 | Alternate Tracks per Volume (MSB) | | | | | | | |
| 9 | Alternate Tracks per Volume (LSB) | | | | | | | |
| 10 | Sectors per Track (MSB) | | | | | | | |
| 11 | Sectors per Track (LSB) | | | | | | | |
| 12 | Data Bytes per Physical Sector (MSB) | | | | | | | |
| 13 | Data Bytes per Physical Sector (LSB) | | | | | | | |
| 14 | Interleave Value (MSB) | | | | | | | |
| 15 | Interleave Value (LSB) | | | | | | | |
| 16 | Track Skew (MSB) | | | | | | | |
| 17 | Track Skew (LSB) | | | | | | | |
| 18 | Cylinder Skew (MSB) | | | | | | | |
| 19 | Cylinder Skew (LSB) | | | | | | | |
| 20 | SSEC | HSEC | RMB | SURF | INS | Reserved | | |
| 21 | Reserved | | | | | | | |
| 22 | Reserved | | | | | | | |
| 23 | Reserved | | | | | | | |

The **Tracks per Zone** field <Bytes 2 through 3> specifies the number of tracks per zone. The controller only supports one track per zone. This field may be set to either 1 or 0. A zero in this field specifies the default number of tracks per zone, which is one.

The **Alternate Sectors per Zone** field <Bytes 4 and 5> specifies the number of alternate sectors per track to allocate during format. The controller supports from zero to three alternate sectors per track and defaults to one sector per track.

The **Alternate Tracks per Zone** field <Bytes 6 and 7> specifies the number of alternate tracks to allocate during format for the zone specified in the Tracks per Zone field. **This field is not used by the controller and must be set to 0.**

The **Alternate Tracks per Volume** field <Bytes 8 and 9> specifies the number of alternate tracks to allocate, for the entire disk, during a format operation. Bad tracks are mapped onto the alternate tracks by the **FORMAT UNIT** or **RE-ASSIGN BLOCK** commands. The controller will round this value up to the nearest cylinder boundary. The controller supports from 0 to 255 alternate tracks per volume and defaults to 3 cylinders of tracks per volume (the actual number of tracks is dependent on the number of heads).

The **Sectors per Track** field <Bytes 10 and 11> specifies the number of physical sectors per track. This value, minus the number of Alternate Sectors per Zone, equals the number of sectors per track available to the user. The controller supports from 1 to 255 sectors per track. If this field is 0, the default sector size (obtained from the disk drive) is used. This field must be supplied if a non-default sector size is used and must correspond to the value in the **Data Bytes per Physical Sector** field.

**NOTE**

This field can only be changed using the **MODE SELECT** command when the disk drive is configured for soft sector operation. If the drive is configured for hard sector operation and this field does not match the value returned by the **MODE SENSE** command, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION.

The **Data Bytes per Physical Sector** field <Bytes 12 and 13> specifies the number of bytes per each physical sector. The controller only supports sector sizes of 256 and 512 bytes and defaults to 512 bytes per sector. The host is responsible for calculating the proper number of sectors per track for any sector size that does not match the default value.

**NOTE**

This field can only be changed using the **MODE SELECT** command when the disk drive is configured for soft sector operation. If the drive is configured for hard sector operation and this field does not match the value returned by the **MODE SENSE** command, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION.

The **Interleave Value** field <Bytes 14 and 15> is ignored by the controller and may be set to any value. The interleave value is set in the **FORMAT UNIT** command.

The **Track Skew** field <Bytes 16 and 17> specifies the number of physical sectors between the last logical block of one track and the first logical block on the next sequential track of the same cylinder. The controller uses the default head skew value of 0 if the disk drive indicates a head switch time that is less than 15 microseconds. When the disk drive indicates a head switch time that is greater than 15 microseconds, the default head skew value is 10. The track skew must be less than the value in the sectors per track field.

The **Cylinder Skew** field <Bytes 18 and 19> specifies the number of physical sectors between the last logical block of one cylinder and the first logical block on the next sequential cylinder. The Cylinder Skew field defaults to 0. The cylinder skew cannot exceed the value in the sectors per track field.

The **Soft Sector Format (SSEC)** bit <Byte 20, Bit 7> cannot be set by the **MODE SELECT** command and the controller ignores this field (see the **MODE SENSE** command).

The **Hard Sector Format (HSEC)** bit <Byte 20, Bit 6> cannot be set by the **MODE SELECT** command and the controller ignores this field (see the **MODE SENSE** command).

The **Removable Media (RMB)** bit <Byte 20, Bit 5>, when set to 1, indicates the media is removable. When the RMB bit is 0, the media is not removable. **This field is not used by the controller and must be set to 0.**

The **Surface (SURF)** bit <Byte 20, Bit 4>, when set to 1, indicates the controller should format by surface rather than cylinder. **This option is not supported by the controller and must be set to 0.**

The **Inhibit Save (INS)** bit <Byte 20, Bit 3>, when set to 1, indicates the controller will inhibit the saving of any parameters during the successful completion of the next **FORMAT UNIT** command. **This field is not used by the controller and must be 0.** The controller will always save pages 1 through 4 after successful completion of the **FORMAT UNIT** command.

6.2.4.5.7 Format of Rigid Disk Drive Geometry Page. This subsection specifies the **MODE SELECT** rigid disk drive geometry options supported by the controller. The format for the Rigid Disk Drive Geometry Page is shown below.

**NOTE**

The controller will save this page whenever the Initiator issues a **FORMAT UNIT** command.

| MODE SELECT Rigid Drive Geometry Page Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | Page Code = 4h | | | | | |
| 1 | Maximum Page Length = 12h | | | | | | | |
| 2 | Maximum Number of Cylinders (MSB) | | | | | | | |
| 3 | Maximum Number of Cylinders | | | | | | | |
| 4 | Maximum Number of Cylinders (LSB) | | | | | | | |
| 5 | Maximum Number of Heads | | | | | | | |
| 6 | Starting Cylinder - Write Precompensation (MSB) | | | | | | | |
| 7 | Starting Cylinder - Write Precompensation | | | | | | | |
| 8 | Starting Cylinder - Write Precompensation (LSB) | | | | | | | |
| 9 | Starting Cylinder - Reduced Write Current (MSB) | | | | | | | |
| 10 | Starting Cylinder - Reduced Write Current | | | | | | | |
| 11 | Starting Cylinder - Reduced Write Current (LSB) | | | | | | | |
| 12 | Drive Step Rate (MSB) | | | | | | | |
| 13 | Drive Step Rate (LSB) | | | | | | | |
| 14 | Landing Zone Cylinder (MSB) | | | | | | | |
| 15 | Landing Zone Cylinder | | | | | | | |
| 16 | Landing Zone Cylinder (LSB) | | | | | | | |
| 17 | Reserved | | | | | | | |
| 18 | Reserved | | | | | | | |
| 19 | Reserved | | | | | | | |

The **Maximum Number of Cylinders** field <Bytes 2 through 4> specifies the maximum number of cylinders available on the disk drive. This value minus the Alternate Tracks per Volume value (converted to cylinders) **and minus the three** cylinders reserved by the controller, equals the number of cylinders addressable by the user. The MSB <Byte 2> must be set to 0. The default value for this field is obtained from the disk drive.

The **Maximum Number of Heads** field <Bytes 5> specifies the number of data heads on the disk drive. The controller supports from 1 to 15 heads. The default value for this field is obtained from the disk drive.

The **Starting Cylinder - Write Precompensation** field <Bytes 6 through 8> specifies the starting cylinder at which the controller will apply precompensation when writing data to the disk. **This option is not supported by the controller and must be set to 0.**

The **Starting Cylinder - Reduced Write Current** field <Bytes 9 through 11> specifies the starting cylinder at which the controller will reduce the write current to the head during a disk write operation. **This option is not supported by the controller and must be set to 0.**

The **Drive Step Rate** field <Byte 12 and 13> specifies the time in 100 nanosecond increments between step pulses. **This option is not supported by the controller and must be set to 0.**

The **Landing Zone Cylinder** field <Bytes 14 through 16> specifies the cylinder number to which the heads will be moved when the **STOP/START UNIT** command is executed. **This field is not supported by the controller and must be set to 0.**

### 6.2.4.6 MODE SELECT Error Conditions

If any field not used or supported by the controller is not 0, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the Media Type is not set to 0, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If a Block Descriptor Length of other than 0 or 8 is specified, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05H/26H) Sense Key/Error Code.

If the Density Code is not set to 0, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the Bytes per Physical Sector is not 256 or 512, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the Track Skew parameter exceeds the number of sectors per track, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the Cylinder Skew parameter exceeds the number of sectors per track, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the Tracks per Zone field is not set to 0 or 1, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the drive is configured for hard sector operation and the values in the Sectors Per Track field and Data Bytes Per Physical Sector do not match the values returned by the **MODE SENSE** command, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the drive has never been formatted, or the Saved **MODE SELECT** configuration information has been destroyed, and the Initiator issues a **MODE SELECT** command with the SMP bit set to 1, the controller will terminate the command with a CHECK CONDITION status and a NOT READY/ILLEGAL FUNCTION FOR DEVICE TYPE (02h/22h) Sense Key/Error Code.

If the Alternate Sectors per Zone is not in the range 0 to 3, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the Initiator specifies an invalid mode in the error recovery bits, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If, in the Rigid Disk Drive Geometry Parameters Page, the controller receives a value in the Maximum Number of Heads field that is greater than either the default values returned by the MODE SENSE command or the actual values, the controller terminates the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

If the Page Length byte in each page header does not match the page length as specified in this document and returned by the **MODE SENSE** command, the controller will terminate the **MODE SELECT** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

### 6.2.4.7 **MODE SELECT** Sense Information

Errors detected by the controller during the **MODE SELECT** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

| MODE SELECT Nonextended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| MODE SELECT Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **MODE SELECT** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is

valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h)<br>Seek Error (15h)<br>Recoverable Read Error With Retries (17h)<br>Recoverable Read Error With ECC (18h) |
| Not Ready (02h) | Drive Not Ready (04h)<br>Drive Not Selected (05h)<br>Illegal Function For Device Type (22h)<br>Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h)<br>Uncorrectable Data Error (11h)<br>No ID field Address Mark (12h)<br>No Data field Address Mark (13h)<br>Block Not Found (14h)<br>Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h)<br>Drive Not Ready (04h)<br>Drive Not Selected (05h)<br>No Track Zero (06h)<br>ID Field Address Mark Not Found (12h)<br>SCSI Msg Reject Error (43h)<br>SCSI Hardware Error (44h)<br>Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h)<br>Invalid Field in CDB (24h)<br>Invalid LUN (25h)<br>Invalid Field in Parameter List (26h) |
| Unit Attention (06h) | Power Up or Reset (29h)<br>Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h)<br>Initiator Detected Error (48h)<br>Illegal Message (49h) |

### 6.2.5 MODE SENSE - 1AH

The **MODE SENSE** command provides a means by which the Initiator may receive the medium, logical unit and peripheral device parameters from the controller. **MODE SENSE** is a complementary command to the **MODE SELECT** command.

The controller will send blocks of parameters that are separated into categories (called pages). These parameters specify various options and features which the Initiator may change. Each page is preceded by a Page Code and the length of the page. The Page Length value does not include Bytes 0 or 1 of the page.

#### 6.2.5.1 MODE SENSE SCSI Deviations

There are no deviations.

#### 6.2.5.2 MODE SENSE Parameter Values

The controller maintains the following four different sets of **MODE SENSE** data:

- **Default values:** The default values are stored in the controller PROM, the disk drive, and the controller switches.
- **Saved values:** The saved values are all the changeable **MODE SELECT** parameters saved by the controller on the disk drive when it performs a **FORMAT UNIT** command or a **MODE SELECT** command with the SMP bit (in the CDB) set to 1.
- **Current values:** The current values are all the changeable **MODE SELECT** parameters used by the controller during normal controller operations. Any **MODE SELECT** command issued to the controller changes the current values.
- **Changeable values:** The changeable values are those parameters supported by the controller that can be changed by the **MODE SELECT** command.

At initialization time (a power-on or reset condition has occurred, see Section 4), the controller copies the default values into the current values. After the disk drive spins up, the controller will read the saved values from the drive and copy them into the current values.

When the controller completes a **FORMAT UNIT** command, it writes all supported pages to the disk drive. On an unformatted disk drive, the current parameters are the same as the default parameters. The Initiator may then change the current parameters using a **MODE SELECT** command prior to formatting the disk drive. When the **FORMAT UNIT** command completes, the controller writes the current values (which may have been changed by a **MODE SELECT** command) to the disk drive as the saved values.

**NOTE**

The Initiator can determine if the drive has been formatted by attempting to read the Saved values with the **MODE SENSE** command. If any error is returned, the drive has never been formatted or the Saved configuration information has been destroyed and the drive must be formatted.

The Initiator may change the saved values of pages 1 (Error Recovery Parameters) and 2 (Disconnect/Reconnect Parameters) by issuing a **MODE SELECT** command with the SMP bit in the CDB set to 1. This action will not change the saved values of Pages 3 (Direct-Access Format Parameters) and 4 (Disk Drive Geometry Parameters).

### 6.2.5.3 MODE SENSE Command Parameters

The **MODE SENSE** Command Parameter Block is formatted as shown below.

**MODE SENSE Command Descriptor Block**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | PCF | | Page Code | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Allocation Length | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Page Control Field (PCF)** <Byte 2, Bits 6:7> specifies the type of page values the controller will return: current, changeable, default, or saved. Table 6-5 lists and describes the page control fields.

Table 6-5. **MODE SENSE** Page Control Fields

| Bit 7 | Bit 6 | Page Control Field |
|---|---|---|
| 0 | 0 | **Report Current Values.** If the Page Code is 3Fh, the controller will return all the pages it implements with the field and bits set to their current values. If the Page Code is not 3Fh and the controller implements the page specified (only pages 1 through 4), the controller will return the page with the field and bits set to their current values. Any fields or bits not supported by the controller will be set to 0. The controller will return the length specified in the Page Length field for each page. |
| 0 | 1 | **Report Changeable Values.** If the Page Code is 3Fh, the controller will return all the pages it implements with the field and bits that can be changed set to 1. Fields and bits that are not changeable will be set to 0. For fields where the entire value is not supported by the controller (for example, only two bytes out of three may be changed), only those bits which can be changed by the Initiator will be set to 1. If the Page Code is not 3Fh and the controller implements the page specified (only pages 1 through 4), the controller will return the page with the field and bits that can be changed set to 1. Fields and bits that are not changeable will be set to 0. The controller will return the length specified in the Page Length field for each page. |
| 1 | 0 | **Report Default Values.** If the Page Code is 3Fh, the controller will return all the pages it implements with the field and bits set to the controller's or the device's default values. If the Page Code is not 3Fh and the controller implements the page specified (only pages 1 through 4), the controller will return the page with the field and bits set to the controller's or the device's default values. Any fields or bits not supported by the controller will be set to 0. The controller will return the length specified in the Page Length field for each page. |

(continued on next page)

Table 6-5. **MODE SENSE** Page Control Fields (continued)

| Bit 7 | Bit 6 | Page Control Field |
|---|---|---|
| 1 | 1 | **Report Saved Values.** If the Page Code is 3Fh, the controller will return all the pages it implements with the field and bits set to their saved values. If the Page Code is not 3Fh and the controller implements the page specified (only pages 1 through 4), it will return the page with the field and bits set to their saved values. Any fields or bits not supported by the controller will be set to 0. The controller will return the length specified in the Page Length field for each page. |

The **Page Code** field <Byte 2, Bits 0:5> specifies the page(s) to be returned in the **MODE SENSE** data. Table 6-6 lists and describes the page codes. If a single page is requested (i.e., the page code is set to 1, 2, 3, or 4), the controller will only return the requested page. The Block Descriptor information will only be sent if multiple pages are requested (i.e., the Page Code field equal to 3Ch, 3Dh, 3Eh, or 3Fh).

Table 6-6. **MODE SENSE** Page Codes

| Page Code | Page Description |
|---|---|
| 00 | **Emulex SCSI Rev. 15 MODE SENSE.** This page code causes the controller to return the **MODE SENSE** parameters as implemented by Emulex in previous versions of the MD21 Controller. This format consists of the **MODE SENSE** Parameter List Header, a single block descriptor, followed by 9 bytes of vendor-unique parameters. This document does not show the format of this data. This format should not be used for any future products. The Page Control Field (PCF) must be 00.<br><br>This page code returns on the current information (such as defaults after power-up); the drive will not be read to obtain saved values. Once a media acess has been done, the saved values will be used as current information. |
| 01 | **Error Recovery Parameters.** This Page Code will cause the **MODE SENSE** command to return a single page which contains the Error Recovery Parameters. |
| 02 | **Disconnect/Reconnect Control Parameters.** This Page Code will cause the **MODE SENSE** command to return a single page which contains the Disconnect/Reconnect Control Parameters. |
| 03 | **Direct-Access Device Format Parameters.** This Page Code will cause the **MODE SENSE** command to return a single page which contains the Direct-Access Device Format Parameters. |
| 04 | **Rigid Disk Drive Geometry Parameters.** This Page Code will cause the **MODE SENSE** command to return a single page which contains the Rigid Disk Drive Geometry Parameters. |
| 3Ch | **Vendor Unique - Report Saved Values.** This Page Code will cause the **MODE SENSE** command to return pages 1 through 4 with each field set to the controller's saved value. Fields and bits not supported by the controller are set to 0. The controller always returns all the fields for each of the pages. This is a vendor-unique page code used in previous releases of the firmware and should not be used in future product releases. Initiators should specify a Page Code of 3Fh and the PCF bits set to 1 1 to determine the saved values of the page fields and bits. The Page Control Field (PCF) must be 00. |

(continued on next page)

Table 6-6. **MODE SENSE** Page Codes (continued)

| Page Code | Page Description |
|---|---|
| 3Dh | **Vendor Unique - Report Default Values.** This Page Code is vendor unique and will cause the **MODE SENSE** command to return pages 1 through 4 with each field set to the controller's default value. Fields and bits not supported by the controller are set to 0. The controller always returns all the fields for each of the pages. This is a vendor-unique page code used in previous releases of the firmware and should not be used in future product releases. Initiators should specify a Page Code of 3Fh and the PCF bits set to 1 0 to determine the default values of the page fields and bits. The Page Control Field (PCF) must be 00. |
| 3Eh | **Vendor Unique - Report Changeable Values.** This Page Code is vendor unique and will cause the **MODE SENSE** command to return pages 1 through 4 with fields changeable by the Initiator set to 1 and all other fields set to 0. For fields where the entire value is not supported by the controller (for example, only two bytes out of three may be changed), only those bits which can be changed by the Initiator will be set to 1. The controller always returns all the fields for each of the pages. This is a vendor-unique page code used in previous releases of the firmware and should not be used in future product releases. Initiators should specify a Page Code of 3Fh and the PCF bits set to 0 1 to determine the changeable values of the page fields and bits. The Page Control Field (PCF) must be 00. |
| 3Fh | **Report All Pages.** This Page Code will cause the **MODE SENSE** command to return pages 1 through 4. The value of each field is set to the values specified by the PCF bits. The controller returns all of the fields and bits for each of the pages. |

The **Allocation Length** field <Byte 4> specifies the number of bytes the Initiator has allocated for returned **MODE SENSE** data. If the controller receives a zero value in Byte 4, it does not transfer any data and does not treat this condition as an error. A non-zero Allocation Length value indicates the maximum number of bytes to be transferred. The controller terminates the Data In Phase when the Allocation Length bytes

have been transferred or all available **MODE SENSE** data has been transferred to the Initiator, whichever is less.

**NOTE**

The controller will return the entire page or up to the value specified by the Initiator in the Allocation Length field, whichever value is less. If the Initiator does not know the number of bytes returned by the controller, it can issue a **MODE SENSE** command with an Allocation Length value of 1. The first byte of data returned by the controller specifies the number of bytes of **MODE SENSE** data available. The Initiator should then re-issue the **MODE SENSE** command with the Allocation Length field value equal to the contents of the first byte plus one.

### 6.2.5.4 MODE SENSE Parameter List Format

The **MODE SENSE** Parameter List is sent by the controller to the Initiator during the Data In Phase. This list consists of a Parameter List Header, one Block Descriptor (only if multiple pages are requested), and one to four Page Descriptors.

The Parameter List Header is four bytes long and specifies the media type and the length of the Block Descriptor.

The Block Descriptor is eight bytes long and specifies the medium density, the number of blocks, and the block length.

The Page descriptors contain various parameters separated into categories (or pages). These parameters specify various options and features which the Initiator may change with the **MODE SELECT** command. The type of page data returned is specified with the Page Code and Page Control field in the Command Descriptor Block.

Each defined page is preceded by a header of two bytes that specifies the **Page Code** and the **Page Length**. The **Page Code** identifies the meaning of the bytes which follow it. The **Page Length** field indicates the number of bytes supported by the controller for that page. The Page Length value does not include the Page Length byte. After the header, the pages are separated into sub-blocks that contain a list of related flags and/or values.

6.2.5.4.1 Parameter List Header Format. The **MODE SENSE** Parameter List Header is the first part of the Parameter List. The header is formatted as shown below.

**MODE SENSE Parameter List Header**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Data Length | | | | | | | |
| 1 | Medium Type | | | | | | | |
| 2 | WP | Reserved | | | | | | |
| 3 | Block Descriptor Length | | | | | | | |

The **Data Length** field <Byte 0> specifies the length of the data that is returned when a **MODE SENSE** command is issued. This length does not include the Data Length field itself. The Parameter List length varies depending on which page(s) are requested.

The **Media Type** field <Byte 1> indicates the media type on the disk drive. The controller will always return a Media type of 0.

The **Write Protect (WP)** field <Byte 2, Bit 7>, when set to 1, indicates the disk drive is write protected. When the WP bit is set to 0, the disk drive is not write protected.

The **Block Descriptor Length** <Byte 3> specifies the length of the Block Descriptor and will be set to 8, if the Block Descriptor is sent. If the Block Descriptor is not sent, this field will be set to 0.

6.2.5.4.2 Parameter List Block Descriptor Format. The **MODE SENSE** Parameter List Block Descriptor (if it is sent) immediately follows the Parameter List Header. The Block Descriptor is formatted as shown below.

**NOTE**

The controller will calculate the information in the Block Descriptor from the parameters (Current, Saved, or Default) specified by the Initiator in the Page Control Field.

| MODE SENSE Parameter List Block Descriptor Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Density Code | | | | | | | |
| 1 | Number of Logical Blocks (MSB) | | | | | | | |
| 2 | Number of Logical Blocks | | | | | | | |
| 3 | Number of Logical Blocks (LSB) | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Logical Block Length (MSB) | | | | | | | |
| 6 | Logical Block Length | | | | | | | |
| 7 | Logical Block Length (LSB) | | | | | | | |

The **Density Code** field <Byte 0> defines the density of the media on the addressed LUN. The Density Code has a value of 0 to indicate only the default density of the hard disk drive is supported.

The **Number of Logical Blocks** field <Bytes 1 through 3> specifies the total number of logical blocks in the user's address space. This field is calculated using the following algorithm:

((# of Cylinders - # of Alternate Cylinders - 3)*(# of Sectors per Track - # Spare Sectors per Track)*(# of Heads))/(Logical Block Length/Physical Sector Size)

where - means minus, * means multiplied by, and / means divided by

The **Logical Block Length** field <Bytes 5 through 7> specifies the length of the logical block in bytes. The controller only supports logical block sizes of 256, 512, 1024, 2048, and 4096 bytes.

6.2.5.4.3 Format of Error Recovery Parameter Page. This subsection specifies the format of the Error Recovery Parameter Page as returned by the **MODE SENSE** command. A copy of each of these parameters is saved for each Initiator. The values returned will be for the Initiator which sent the **MODE SENSE** command.

| MODE SENSE Error Recovery Parameter Page Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | PS | Rsrv | Page Code = 1h | | | | | |
| 1 | Page Length = 6h | | | | | | | |
| 2 | EARW | EARR | TB | RC | EEC | RRERR | TRERR | DCOR |
| 3 | Retry Count | | | | | | | |
| 4 | Correction Span | | | | | | | |
| 5 | Head Offset Count | | | | | | | |
| 6 | Data Strobe Offset Count | | | | | | | |
| 7 | Recovery Time Limit | | | | | | | |

The **Parameter Savable (PS)** bit <Byte 0, Bit 7>, when set to 1, indicates the controller will save the parameters supported in this page. When the PS bit is 0, the controller does not save the page parameters. The controller will always return a 1 in this field.

The **Enable Automatic Reallocation on Write (EARW)** field <Byte 2, Bit 7>, when set to 1, instructs the controller to automatically re-assign bad blocks when it encounters them during a write operation. This function is similar to that of the **RE-ASSIGN BLOCK** command, but is initiated by the controller. **The controller does not support this option and will always return a 0 in this field.**

The **Enable Automatic Reallocation on Read (EARR)** field <Byte 2, Bit 6>, when set to 1, instructs the controller to automatically re-assign bad blocks when it encounters them during a read operation. This function is similar to that of the **RE-ASSIGN BLOCK** command, but is initiated by the controller. **The controller does not support this option and will always return a 0 in this field.**

The **Transfer Block (TB)** field <Byte 2, Bit 5>, when set to 1, specifies that the controller should transfer the block with the data error before terminating the command. This bit is only applicable when a hard error is encountered or the TRERR bit is set to 1 and a recoverable error is encountered. If this field is zero, the controller will not transfer the block with the data error. In both cases, the controller reports the block address of the block with the error, rather than that of the preceding block, in the sense data. The Transfer Block field defaults to 1. If the transfer terminates with other than a data error (i.e., data not found) the block will not be transferred. If the Initiator requested the changeable values, this field will be set to 1.

The **Read Continuous (RC)** field <Byte 2, Bit 4>, when set to 1, overrides the Enable Early Correction (EEC), Terminate on Recoverable Error (TRERR), Report Recoverable Error (RRERR), and Disable Correction (DCOR) fields and disables all retries and data correction. The Transfer Block (TB) field is not applicable. When the RC bit is set to 1, the controller transfers the entire requested length of data without adding delays that are caused by its error recovery schemes. The controller will send data which may be erroneous or fabricated to maintain a continuous flow of data and avoid delays. The RC field defaults to 0. If the Initiator requested the changeable values, this field will be set to 1.

The **Enable Early Correction (EEC)** field <Byte 2, Bit 3> specifies that the controller should perform a minimum number of retries before applying any correction algorithm. When this bit is set to 1, the controller will not exhaust the retry count before attempting any ECC correction. When this bit is 0, the controller will exhaust the retry count, as specified in Byte 03, before it attempts any ECC correction. The Enable Early Correction field defaults to 0. If the Initiator requested the changeable values, this field will be set to 1.

**NOTE**

This field does not disable retries during seek operations. For any seek or positioning error, the controller will issue a recalibrate command to the drive then retry the seek operation. If the second attempt also fails, the controller will terminate the command.

The **Report Recoverable Error (RRERR)** field <Byte 2, Bit 2>, when set to 1, instructs the controller to report any recoverable errors to the Initiator. This error will either be reported immediately, or at the normal completion of the command depending on the state of the TRERR field. The error actually reported to the Initiator will be the last error encountered during the data transfer. If multiple errors occur, the controller reports (in the sense information) the block address of either (1) the last block where the recovered error occurred or (2) the block with the first unrecoverable error. The RRERR field defaults to 1. If the Initiator requested the changeable values, this field will be set to 1.

The **Terminate on Recoverable Error (TRERR)** field <Byte 2, Bit 1>, when set to 1 and the RRERR bit is set to 1, instructs the controller to terminate the command immediately when a recoverable error is encountered and create the CHECK CONDITION status. The controller may or may not transfer the data contained in the block in error, depending on the setting of the TB bit. The Initiator can only set the TRERR bit to 1 if it has set the RRERR bit to 1. If the TRERR bit is set to 0,

the controller will continue the data transfer when a recoverable error is encountered. The Terminate on Recoverable Error field defaults to 0. If the Initiator requested the changeable values, this field will be set to 1.

The **Disable Correction (DCOR)** field <Byte 2, Bit 0>, when set to 1, disables ECC correction when reading a sector from the disk drive. No correction is attempted and if a correctable ECC error occurs, it is treated as a recoverable error. This field defaults to 0. If the Initiator requested the changeable values, this field will be set to 1.

The **Retry Count** field <Byte 3> specifies the maximum number of retries to attempt when an error is encountered. The controller supports Retry Counts from 0 to 255 (decimal) and defaults to 10. If the Initiator requested the changeable values, this field will be set to FFh.

The **Correction Span** field <Byte 4> specifies the largest read data error, in bits, on which correction may be attempted. The controller supports a Correction Span of 0 to 19 decimal bits and defaults to 11. If the Initiator requested the changeable values, this field will be set to 1Fh.

The **Head Offset Count** field <Byte 5> specifies the incremental offset, from the center of the track, to use when performing a disk operation. **The controller does not support this option and will always return a 0 in this field.**

The **Data Strobe Offset Count** field <Byte 6> specifies an incremental position the controller will use to adjust the recovered data strobe from the nominal. **The controller does not support this option and will always return a 0 in this field.**

The **Recovery Time Limit** field <Byte 7> specifies the maximum amount of time that the controller will attempt to recover an error. The field is defined in increments of 10 milliseconds. When set to 0, the controller will use an unlimited time. **The controller does not support this option and will always return a 0 in this field.**

6.2.5.4.4 Format of Disconnect/Reconnect Parameter Page. This section specifies the format of the Disconnect/Reconnect Parameter Page as returned by the **MODE SENSE** command. The controller saves a copy of each of these parameters for each Initiator. This allows any one Initiator to examine its own parameters without affecting the parameters of any other Initiator.

| MODE SENSE Disconnect/Reconnect Parameter Page Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | PS | Rsrv | Page Code = 2h | | | | | |
| 1 | Page Length = 0Ah | | | | | | | |
| 2 | Buffer Full Ratio | | | | | | | |
| 3 | Buffer Empty Ratio | | | | | | | |
| 4 | Bus Inactivity Limit (MSB) | | | | | | | |
| 5 | Bus Inactivity Limit (LSB) | | | | | | | |
| 6 | Disconnect Time Limit (MSB) | | | | | | | |
| 7 | Disconnect Time Limit (LSB) | | | | | | | |
| 8 | Connect Time Limit (MSB) | | | | | | | |
| 9 | Connect Time Limit (LSB) | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |

The **Parameter Savable (PS)** bit <Byte 0, Bit 7>, when set to 1, indicates the controller will save the parameters supported in this page. When the PS bit is 0, the controller does not save the page parameters. The controller will always return a 1 in this field.

The **Buffer Full Ratio** field <Byte 1> specifies how full the internal buffer should be before the controller reconnects to transfer the data to the Initiator. **The controller does not support this option and will always return a 0 in this field.**

The **Buffer Empty Ratio** field <Byte 2> specifies how empty the internal buffer should be before the controller reconnects to transfer more data from the Initiator. **The controller does not support this option and will always return a 0 in this field.**

The **Bus Inactivity Limit** field <Bytes 4 and 5> specifies the length of time, in 100 microsecond increments, that the controller is allowed to stay connected to the SCSI bus without any bus activity. The controller supports a range of 1 (100 microseconds) to 650 (65000 microseconds) and defaults to 5 (500 microseconds). If the Initiator requested the changeable values, this field will be set to FFFFh.

The **Disconnect Time Limit** field <Bytes 6 and 7> specifies the length of time, in 10 microsecond increments, that the controller should remain disconnected before attempting to reconnect. **The controller does not support this option and will always return a 0 in this field.**

The **Connect Time Limit** field <Bytes 8 and 9> specifies the maximum length of time the controller should remain connected (in 10 microsecond increments) before it attempts to disconnect. **The controller does not support this option and will always return a 0 in this field.**

6.2.5.4.5 Format of Direct-Access Device Format Parameter Page. This section specifies the format of the Direct-Access Device Format Parameters Page as returned by the **MODE SENSE** command.

| MODE SENSE Direct Access Device Format Parameter Page | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | PS | Rsrv | Page Code = 3h | | | | | |
| 1 | Page Length = 16h | | | | | | | |
| 2 | Tracks per Zone (MSB) | | | | | | | |
| 3 | Tracks per Zone (LSB) | | | | | | | |
| 4 | Alternate Sectors per Zone (MSB) | | | | | | | |
| 5 | Alternate Sectors per Zone (LSB) | | | | | | | |
| 6 | Alternate Tracks per Zone (MSB) | | | | | | | |
| 7 | Alternate Tracks per Zone (LSB) | | | | | | | |
| 8 | Alternate Tracks per Volume (MSB) | | | | | | | |
| 9 | Alternate Tracks per Volume (LSB) | | | | | | | |
| 10 | Sectors per Track (MSB) | | | | | | | |
| 11 | Sectors per Track (LSB) | | | | | | | |
| 12 | Bytes per Physical Sector (MSB) | | | | | | | |
| 13 | Bytes per Physical Sector (LSB) | | | | | | | |
| 14 | Interleave Value (MSB) | | | | | | | |
| 15 | Interleave Value (LSB) | | | | | | | |
| 16 | Track Skew (MSB) | | | | | | | |
| 17 | Track Skew (LSB) | | | | | | | |
| 18 | Cylinder Skew (MSB) | | | | | | | |
| 19 | Cylinder Skew (LSB) | | | | | | | |
| 20 | SSEC | HSEC | RMB | SURF | Reserved | | | |
| 21 | Reserved | | | | | | | |
| 22 | Reserved | | | | | | | |
| 23 | Reserved | | | | | | | |

The **Parameter Savable (PS)** bit <Byte 0, Bit 7>, when set to 1, indicates the controller will save the parameters supported in this page. When the PS bit is 0, the controller does not save the page parameters. The controller will always return a 1 in this field.

The **Tracks per Zone** field <Bytes 2 through 3> specifies the number of tracks per zone. The controller only supports 1 track per zone and will return a 1 in this field if the Initiator requests the Saved, Current, or Default values. If the Initiator requested the changeable values, this field will be set to 0.

The **Alternate Sectors per Zone** field <Bytes 4 and 5> specifies the number of alternate sectors per track to allocate during format. The controller supports from zero to three alternate sectors per track and defaults to 1 sector per track. If the Initiator requested the changeable values, this field will be set to FFh.

The **Alternate Tracks per Zone** field <Bytes 6 and 7> specifies the number of alternate tracks to allocate during format for the zone specified in the Tracks per Zone field. **The controller does not support this option and will always return a 0 in this field.**

The **Alternate Tracks per Volume** field <Bytes 8 and 9> specifies the number of alternate tracks to allocate, for the entire disk, during format. Bad tracks are mapped onto the Alternate tracks by the **FORMAT UNIT** or **RE-ASSIGN BLOCK** commands. If the Initiator requested the changeable values, this field will be set to FFFFh.

**NOTE**

This value will be rounded up or down to the nearest cylinder boundary. The controller defaults to 3 cylinders of tracks per volume (the actual number of tracks is dependent on the number of heads).

The **Sectors per Track** field <Bytes 10 and 11> specifies the number of physical sectors per track. This value, minus the number of Alternate Sectors per Zone, equals the number of sectors per track available to the user. The default value for this field is obtained from the disk drive. If the Initiator requested the changeable values and the drive is configured for soft sector operation, this field will be set to FFFFh. For hard sector operation, this field will be set to 0000h.

The **Bytes per Physical Sector** field <Bytes 12 and 13> specifies the number of bytes per each physical sector. The controller only supports sectors sizes of 256 and 512 bytes and defaults to 512 bytes per sector. If the Initiator requested the changeable values and the drive is configured for soft sector operation, this field will be set to FFFFh. For hard sector operation, this field will be set to 0000h.

The **Interleave Value** field <Bytes 14 and 15> returns the interleave value specified in the **FORMAT UNIT** command when the drive was formatted. This field defaults to 1. If the Initiator requested the changeable values, this field will be set to 0h.

The **Track Skew** field <Bytes 16 and 17> specifies the number of physical sectors between the last logical block of one track and the first logical block on the next sequential track of the same cylinder. The controller uses the default head skew value is 0 if the disk drive indicates a head switch time that is less than 15 microseconds. When the disk drive indicates a head switch time that is greater than 15 microseconds, the default head skew value is 10. If the Initiator requested the changeable values, this field will be set to FFFFh.

The **Cylinder Skew** field <Bytes 18 and 19> specifies the number of physical sectors between the last logical block of one cylinder and the first logical block on the next sequential cylinder. The Cylinder Skew defaults to 0. If the Initiator requested the changeable values, this field will be set to FFFFh.

The **Soft Sector Format (SSEC)** field <Byte 20, Bit 7>, when set to 1, specifies the controller should use soft sector formatting. The default value for this field is obtained from the disk drive. If the Initiator requested the changeable values, this field will be set to 0.

The **Hard Sector Format (HSEC)** field <Byte 20, Bit 6>, when set to 1, specifies the controller should use hard sector formatting. The default value for this field is obtained from the disk drive. If the Initiator requested the changeable values, this field will be set to 0.

The **Removable Media (RMB)** field <Byte 20, Bit 5> is not supported by the controller and will always be set to 0.

The **Surface (SURF)** field <Byte 20, Bit 4>, when set to 1, indicates the controller should format by surface rather than cylinder. **The controller does not support this option and will always return a 0 in this field.**

6.2.5.4.6 Format of Rigid Disk Drive Geometry Page. This section specifies the format of the Rigid Disk Drive Geometry Page as returned by the **MODE SENSE** command.

| MODE SENSE Rigid Drive Geometry Page Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | PS | Rsrv | Page Code = 4h | | | | | |
| 1 | Page Length = 12h | | | | | | | |
| 2 | Maximum Number of Cylinders (MSB) | | | | | | | |
| 3 | Maximum Number of Cylinders | | | | | | | |
| 4 | Maximum Number of Cylinders (LSB) | | | | | | | |
| 5 | Maximum Number of Heads | | | | | | | |
| 6 | Starting Cylinder - Write Precompensation (MSB) | | | | | | | |
| 7 | Starting Cylinder - Write Precompensation | | | | | | | |
| 8 | Starting Cylinder - Write Precompensation (LSB) | | | | | | | |
| 9 | Starting Cylinder - Reduced Write Current (MSB) | | | | | | | |
| 10 | Starting Cylinder - Reduced Write Current | | | | | | | |
| 11 | Starting Cylinder - Reduced Write Current (LSB) | | | | | | | |
| 12 | Drive Step Rate (MSB) | | | | | | | |
| 13 | Drive Step Rate (LSB) | | | | | | | |
| 14 | Landing Zone Cylinder (MSB) | | | | | | | |
| 15 | Landing Zone Cylinder | | | | | | | |
| 16 | Landing Zone Cylinder (LSB) | | | | | | | |
| 17 | Reserved | | | | | | | |
| 18 | Reserved | | | | | | | |
| 19 | Reserved | | | | | | | |

The **Parameter Savable (PS)** bit <Byte 0, Bit 7>, when set to 1, indicates the controller will save the parameters supported in this page. When the PS bit is 0, the controller does not save the page parameters. The controller will always return a 1 in this field.

The **Maximum Number of Cylinders** field <Bytes 2 through 4> specifies the maximum number of cylinders available on the disk drive. This value minus the Alternate Tracks per Volume (converted to cylinders) and the three cylinders reserved by the controller equals the number of cylinders addressable by the user. The default value for this field is obtained from the disk drive. If the Initiator requested the changeable values, this field will be set to FFFFFFh.

The **Maximum Number of Heads** field <Bytes 5> specifies the number of data heads on the disk drive. The default value for this field is obtained from the disk drive. If the Initiator requested the changeable values, this field will be set to 0Fh.

The **Starting Cylinder - Write Precompensation** field <Bytes 6 through 8> specifies the starting cylinder at which the controller will apply precompensation when writing data to the disk. **The controller does not support this option and will always return a 0 in this field.**

The **Starting Cylinder - Reduced Write Current** field <Bytes 9 through 11> specifies the starting cylinder at which the controller will reduce the write current to the head during a disk write operation. **The controller does not support this option and will always return a 0 in this field.**

The **Drive Step Rate** field <Byte 12 and 13> specifies the time in 100 nanosecond increments between step pulses. **The controller does not support this option and will always return a 0 in this field.**

The **Landing Zone Cylinder** field <Bytes 14 through 16> specifies the cylinder number to which the heads will be moved when the **START/STOP UNIT** command is executed. **The controller does not support this option and will always return a 0 in this field.**

### 6.2.5.5 MODE SENSE Error Conditions

If the Page Code is not valid, the controller will terminate the **MODE SENSE** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the PCF field is not 0 for page codes 0, 3Ch, 3Dh, and 3Eh, the controller will terminate the **MODE SENSE** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the drive has never been formatted, or the Saved **MODE SENSE** configuration information has been destroyed, and the Initiator requests the Saved values (Page Control Field set to 1 1 or Page Code set to 3Ch), the controller will terminate the command with a CHECK CONDITION status.

If the controller cannot read the default information from the disk drive and the Initiator requests the Current or Default values (Page Control Field set to 0 0 or 1 0 or Page Code set to 3Dh or 3Fh), the controller will terminate the command with a CHECK CONDITION status and a NOT READY/ILLEGAL FUNCTION FOR DEVICE TYPE (02h/22h) Sense Key/Error Code.

### 6.2.5.6 MODE SENSE Sense Information

Errors detected by the controller during the **MODE SENSE** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**MODE SENSE Nonextended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**MODE SENSE Extended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **MODE SENSE** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| | Recoverable Read Error With ECC (18h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID field Address Mark (12h) |
| | No Data field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| | Invalid Field in Parameter List (26h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.2.6 PREVENT/ALLOW MEDIA REMOVAL - 1EH

The **PREVENT/ALLOW MEDIA REMOVAL** command will not be implemented.

If any attempt is made to execute the **PREVENT/ALLOW MEDIA REMOVAL** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL COMMAND (05h/20h) Sense Key/Error Code.

### 6.2.7 READ - 08H

The **READ** command requests that the controller transfer data from the logical unit to the Initiator.

#### 6.2.7.1 READ SCSI Deviations

The Vendor Unique bits in the Command Descriptor Block Control Byte are used to disable ECC checking and retries.

#### 6.2.7.2 READ Command Parameters

The **READ** Command Descriptor Block is formatted as shown below.

| READ Command Descriptor Block | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | Logical Unit Number | | | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |
| 4 | Transfer Length | | | | | | | |
| 5 | ECC | ERTY | Reserved | | | | Flag | Link |

The **Logical Block Address** field <Bytes 1 through 3> specifies the logical block at which the read operation will begin.

The **Transfer Length** field <Byte 4> specifies the number of contiguous logical blocks of data to be transferred. A Transfer Length of 0 indicates that 256 logical blocks will be transferred.

The **Error Correction Code (ECC)** field <Byte 5, Bit 7> is vendor unique and indicates if ECC checking is disabled. If this bit is set to 1, ECC checking is disabled during the current **READ** command. Setting the ECC bit produces the same condition as setting the DCOR bit in the **MODE SELECT** Error Recovery Page to 1. If the ECC bit is 0, ECC checking is enabled.

**NOTE**

It is not recommended that the Initiator use the ECC bit as it is used only to maintain compatibility with previous products and will not be used in any future products.

The **Error Retry (ERTY)** field <Byte 5, bit 6> is vendor unique and indicates if Error Retries are disabled. If this bit is set to 1, no retries will be attempted when an error is encountered during the current **READ** command. Setting the ERTY bit produces the same condition as setting the DDR bit (in the **MODE SELECT** Error Recovery Page) to 1 and setting the Retry Count field (in the **MODE SELECT** Error Recovery Page) to 0. When the ERTY bit is 0, the controller will use the **MODE SELECT** error recovery flags when an error is encountered during the current **READ** command.

**NOTE**

It is not recommended that the Initiator use the ERTY bit as it is used only to maintain compatibility with previous products and will not be used in any future products.

### 6.2.7.3 READ Error Conditions

If the Logical Block Address is invalid, the controller will terminate the **READ** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

If the Logical Block Address plus the Transfer Length results in an invalid block address, the controller will terminate the **READ** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code. No data will be transferred if this condition occurs.

### 6.2.7.4 READ Sense Information

Errors detected by the controller during the **READ** command will cause it to terminate the **READ** command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**READ Nonextended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

| READ Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Logical Block Address (MSB) | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address | | | | | | | |
| 6 | Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to 1, indicates the Logical Block Address information is valid. If the valid bit is 0, the Logical Block information should be ignored.

The **Logical Block Address** field <Bytes 3 through 7> specifies the first invalid block address detected by the **READ** command.

The **Additional Sense Length** field <Byte 7> specifies the number of additional sense bytes which follow. For the **READ** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| | Recoverable Read Error With ECC (18h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID field Address Mark (12h) |
| | No Data field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| | Invalid Field in Parameter List (26h) |
| Unit Attention (06h) | Medium Change (28h) |
| | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

## 6.2.8 RE-ASSIGN BLOCK - 07H

### 6.2.8.1 RE-ASSIGN BLOCK Command Description

The **RE-ASSIGN BLOCK** command requests the controller to re-assign the defective logical block(s) to an area on the logical unit reserved for this purpose.

During the Data Out Phase, the Initiator transfers a defect list that contains the logical block(s) to be re-assigned. The controller will re-assign the physical medium used for each logical block specified by the Initiator. The data contained in those blocks specified by the Initiator may be altered, but the data in all other blocks will be preserved.

### 6.2.8.2 RE-ASSIGN BLOCK Command Operation

During the re-assign operation, if all available spare sectors on a track become allocated, the entire track is marked as defective and an alternate track is used. The controller will move all data blocks on the defective track to the alternate track. If all the spare sectors on an alternate track have been used, the alternate track will be revectored to another alternate track.

### 6.2.8.3 RE-ASSIGN BLOCK SCSI Deviations

There are no deviations.

### 6.2.8.4 RE-ASSIGN BLOCK Command Parameters

The **RE-ASSIGN BLOCK** Command Descriptor Block is formatted as shown below.

**RE-ASSIGN BLOCK Command Descriptor Block**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

### 6.2.8.5 RE-ASSIGN BLOCK Parameter List Format

The **RE-ASSIGN BLOCK** Parameter List consists of a four-byte header, which contains the Defect List Length, followed by zero or more Defect Descriptors. The length of each descriptor is four bytes. This information is transferred from the Initiator to the controller during the Data Out Phase of the **RE-ASSIGN BLOCK** command.

**RE-ASSIGN BLOCK Parameter List Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | Defect List Length (MSB) | | | | | | | |
| 3 | Defect List Length (LSB) | | | | | | | |
| | Defect Descriptor(s) | | | | | | | |
| 0 | Defect Logical Block Address (MSB) | | | | | | | |
| 1 | Defect Logical Block Address | | | | | | | |
| 2 | Defect Logical Block Address | | | | | | | |
| 3 | Defect Logical Block Address (LSB) | | | | | | | |

The **Defect List Length** specifies the total length, in bytes, of the Defect Descriptors that follow. The length is equal to four times the number of Defect Descriptors. A Defect List Length of 0 is not considered an error by the controller.

The **Defect Descriptor** contains the four-byte Defect Logical Block Address that specifies the location of the defect. The Defect Descriptors must be in ascending order.

### 6.2.8.6 RE-ASSIGN BLOCK Error Conditions

If the LUN has insufficient capacity to re-assign all the defective blocks, the controller will terminate the **RE-ASSIGN BLOCK** command with a CHECK CONDITION status and a MEDIUM ERROR/NO DEFECT SPARE LOCATION AVAILABLE (03h/32h) Sense Key/Error Code.

If the Defect Logical Block Address is invalid, the controller will terminate the **RE-ASSIGN BLOCK** command with a CHECK CONDITION status and a ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

### 6.2.8.7 RE-ASSIGN BLOCK Sense Information

Errors detected by the controller during the **RE-ASSIGN BLOCK** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**RE-ASSIGN BLOCK Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Defect Logical Block Address (MSB) | | | | |
| 2 | Defect Logical Block Address | | | | | | | |
| 3 | Defect Logical Block Address (LSB) | | | | | | | |

**RE-ASSIGN BLOCK Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Defect Logical Block Address (MSB) | | | | | | | |
| 4 | Defect Logical Block Address | | | | | | | |
| 5 | Defect Logical Block Address | | | | | | | |
| 6 | Defect Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to 1, indicates the Defect Logical Block Address information is valid. If the valid bit is not 1, the Defect Logical Block information should be ignored.

The **Defect Logical Block Address** is the address of the first logical block that was **not** re-assigned.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **RE-ASSIGN BLOCK** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| | Recoverable Read Error With ECC (18h) |
| | Defect List Error (19h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID field Address Mark (12h) |
| | No Data field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| | Format Failed (31h) |
| | No Defect Spare Location Available (32h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Write Fault (03h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| | Invalid Field in Parameter List (26h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Data Protect (07h) | Write Protected (27h) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.2.9 RECEIVE DIAGNOSTIC RESULTS - 1CH

The **RECEIVE DIAGNOSTIC RESULTS** command requests analysis data be sent to the Initiator after completion of a **SEND DIAGNOSTIC** command.

#### 6.2.9.1 RECEIVE DIAGNOSTIC RESULTS Command Limitations

If the **RECEIVE DIAGNOSTIC RESULTS** command was not preceded by a **SEND DIAGNOSTIC** command, the controller will terminate the **RECEIVE DIAGNOSTICS RESULTS** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID COMMAND (05h/20h) Sense Key/Error Code.

#### 6.2.9.2 RECEIVE DIAGNOSTIC RESULTS SCSI Deviations

There are no deviations.

#### 6.2.9.3 RECEIVE DIAGNOSTIC RESULTS Command Parameters

The **RECEIVE DIAGNOSTIC RESULTS** Command Descriptor Block is formatted as shown below.

**RECEIVE DIAGNOSTIC RESULTS Command Descriptor Block**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Allocation Length (MSB) | | | | | | | |
| 4 | Allocation Length (LSB) | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Allocation Length** field <Bytes 3 and 4> specifies the number of bytes that the Initiator has allocated for the returned diagnostic data. An allocation length of 0 indicates that no diagnostic data will be returned. Any other value indicates the maximum number of bytes the controller will transfer. The data transfer will terminate when the number of bytes in the allocation length have been transferred, or when all the available diagnostic data has been transferred, whichever is less.

### 6.2.9.4 RECEIVE DIAGNOSTIC RESULTS Data Format

See the **SEND DIAGNOSTIC** command.

### 6.2.9.5 RECEIVE DIAGNOSTIC RESULTS Sense Information

Errors detected by the controller during the **RECEIVE DIAGNOSTIC RESULTS** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**RECEIVE DIAGNOSTIC RESULTS Nonextended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | Vendor Unique | | | | Information Byte (MSB) | | | |
| 2 | Information Byte | | | | | | | |
| 3 | Information Byte (LSB) | | | | | | | |

**RECEIVE DIAGNOSTIC RESULTS Extended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Information Byte (MSB) | | | | | | | |
| 4 | Information Byte | | | | | | | |
| 5 | Information Byte | | | | | | | |
| 6 | Information Byte (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to 1, indicates the data in the Information Bytes is valid. If the valid bit is not 1, the Information Byte field should be ignored.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **RECEIVE DIAGNOSTIC RESULTS** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| Not Ready (02h) | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Hardware Error (04h) | Drive Not Ready (04h) |
| | ID Field Address Mark Not Found (12h) |
| | RAM Failure (40h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.2.10 RELEASE UNIT - 17H

The **RELEASE UNIT** command causes the LUN (connected to the controller and previously reserved by the **RESERVE UNIT** command) to be released. Once the **RELEASE UNIT** command is issued, other Initiators can access the LUN.

It is not an error to release any LUN which is not currently reserved.

#### 6.2.10.1 RELEASE UNIT Command Options

The controller does not support the Extent Release option.

#### 6.2.10.2 RELEASE UNIT SCSI Deviations

There are no deviations.

#### 6.2.10.3 RELEASE UNIT Command Parameters

The **RELEASE UNIT** Command Descriptor Block is formatted as shown below.

| RELEASE UNIT Command Descriptor Block | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | Logical Unit Number | | | THPR | 3rd Party Device ID | | | Extent |
| 2 | Reservation Identification | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

If the **Third Party Reservation Release (THPR)** field <Byte 01, bit 04> is set to 1, the controller releases the LUN provided the following conditions are true:

- The unit was originally reserved using the third party option in the **RESERVE UNIT** command.
- The same Initiator that issued the **RESERVE UNIT** command is requesting the release of the LUN.
- The Initiator specifies that is has the same SCSI bus device in the Third Party ID field as was specified in that field by the Initiator in the **RESERVE UNIT** command.

The **Third Party Device ID** field <Byte 1, bits 1:3> specifies the SCSI bus device ID for which the LUN was reserved.

The **Extent** bit <Byte 1, Bit 0> indicates if the Extent Reservation option is implemented. **The controller does not support this option and the field must be set to 0.**

**Reservation Identification** field <Byte 2> allows the Initiator to identify each extent reservation (which allows an Initiator to have multiple reservations outstanding in multi-tasking environments). **The controller does not support this option and the field must be set to 0.**

### 6.2.10.4 **RELEASE UNIT** Error Conditions

If the Extent Reservation option is specified, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the Third Party Release option is specified, and the unit was not originally reserved with the third party option, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the Third Party Release option is specified, and the Third Party Device ID is not the same as the one specified in the original **RESERVE UNIT** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the Third Party Release option is specified, and the Initiator SCSI ID is not the same as that of the Initiator which originally reserved the unit, the controller will terminate the command with a RESERVATION CONFLICT status.

### 6.2.10.5 **RELEASE UNIT** Sense Information

Errors detected by the controller during the **RELEASE UNIT** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

| RELEASE UNIT Nonextended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| RELEASE UNIT Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **RELEASE UNIT** command the Additional Sense Length will always be set to ten decimal.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Hardware Error (04h) | SCSI Msg Reject Error (43h)<br>SCSI Hardware Error (44h)<br>Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h)<br>Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h)<br>Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h)<br>Initiator Detected Error (48h)<br>Illegal Message (49h) |

### 6.2.11 REQUEST SENSE - 03H

The **REQUEST SENSE** command provides a means for the Initiator to obtain more detailed information after execution of a command. Typically, a **REQUEST SENSE** command is issued after the previous command has completed and a CHECK CONDITION status returned to the Initiator.

An Initiator should issue a **REQUEST SENSE** command as soon as it receives a CHECK CONDITION status code to obtain the Sense data saved by the controller. The Initiator can issue several **REQUEST SENSE** commands at this time to obtain the Extended Sense data as well as the Nonextended Sense data. However, when the controller receives a command, other than a **REQUEST SENSE**, from the same Initiator for the same LUN, it clears the Sense data for the previous command.

Although Emulex supports the Nonextended Sense format, it is **not recommended** that this format be used in any future products. All new development should use the Extended Sense format and the Sense Key to process any errors.

To determine the maximum length of the extended sense information returned, refer to the value for the **REQUEST SENSE** Length field in the **INQUIRY** command data format (see subsection 6.2.4)

#### 6.2.11.1 REQUEST SENSE SCSI Deviations

The **REQUEST SENSE** command does not clear sense information. This allows the host to issue multiple **REQUEST SENSE** commands (for example, extended following non-extended). All other commands clear sense information.

#### 6.2.11.2 REQUEST SENSE Command Parameters

The **REQUEST SENSE** Command Descriptor Block is formatted as shown below.

**REQUEST SENSE Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Allocation Length | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Allocation Length** field <Byte 5> specifies the number of bytes of data the Initiator has allocated for the sense information. The controller will transfer Sense Data until the Allocation Length is exhausted or all the Sense Data has been transferred, whichever is less. The count supplied determines the format of the returned sense data as explained in the following section.

### 6.2.11.3 REQUEST SENSE Data Format

The format of the returned Sense Data depends on the number of bytes specified in the Allocation Length field of the command descriptor block. Two Sense Data formats are supported:

- An Allocation Length of zero results in a transfer of four bytes in the Nonextended Sense Data format.
- An Allocation Length greater than zero, results in a transfer of up to the requested number of sense bytes, or until all the sense data has been transferred, in the Extended Sense Data format.

6.2.11.3.1 **REQUEST SENSE** Nonextended Sense Data Format. When the Allocation Length is 0, the controller returns the Nonextended Sense Data for all commands, except **COPY** and **COMPARE**, which only returns the Extended Sense format.

Although Emulex supports the Nonextended Sense format, it is not recommended that this format be used in any future products. All new development should use the Extended Sense format and the Sense Key to process any errors.

**REQUEST SENSE Nonextended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | Vendor Unique | | | | Logical Block Address (MSB) | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

The **Valid** field <Byte 0, Bit 7>, when set, indicates that the Logical Block Address field contains valid information related to the error condition.

The **Logical Block Address** field <Bytes 2 and 3> field specifies the logical block address associated with the Error Class and Error Code. For the **FORMAT UNIT** command, these bytes can be either logical or physical block addresses, depending on the format specified for the defect list.

Together the **Error Class** field <Byte 0, Bits 4:6> and the **Error Code** field <Byte 0, Bits 00:03> are called the Nonextended Sense Error Code and indicate the type of error that occurred.

6.2.11.3.2 Nonextended Sense Error Codes Table 6-7 lists and describes the nonextended sense error codes supported by the controller.

Table 6-7. Nonextended Sense Error Codes

| Hex Code | Error | Description |
|---|---|---|
| 00 | NO SENSE | The controller detected no error during execution of the previous command. |
| 01 | RESERVED | This code is reserved. |
| 02 | NO SEEK COMPLETE | The controller could not seek to the specified logical block within an allotted time. |
| 03 | WRITE FAULT | The controller determined that the Attention line from the disk drive was asserted during a write operation. |
| 04 | DRIVE NOT READY | The disk drive is not ready. |
| 05 | DRIVE NOT SELECTED | The drive associated with the specified LUN could not be addressed. |
| 06 | NO TRACK ZERO | The controller could not rezero the disk drive. |
| 07-0F | RESERVED | These codes are reserved. |
| 10 | ID FIELD CRC ERROR | The sector ID field could not be read after the number of retry attempts specified in the **MODE SELECT** command. |
| 11 | UNCORRECTABLE DATA ERROR | A block could not be written or read after the number of retry attempts specified in the **MODE SELECT** command. |

(continued on next page)

Table 6-7. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 12 | ID FIELD ADDRESS MARK NOT FOUND | The controller could not locate the address mark for a sector header. |
| 13 | DATA ADDRESS MARK NOT FOUND | The controller could not locate the address mark for the sector data area. |
| 14 | BLOCK NOT FOUND | The block sequence is improper, or a block is missing. |
| 15 | SEEK ERROR | A miscompare occurred between the cylinder address of the data header and the address specified in the CDB of the command. |
| 16 | RESERVED | This code is reserved. |
| 17 | RECOVERED READ ERROR (WITH RETRIES) | The controller encountered an error. It recovered the data using retries. |
| 18 | RECOVERED READ ERROR (WITH ECC) | The controller encountered an error. It recovered the data using ECC correction. |
| 19 | DEFECT LIST ERROR | The controller encountered an error while accessing one of the Defect Lists. |
| 1A | PARAMETER OVERRUN | The Parameter List Length specified in the CDB by the Initiator is too large for the controller. |

(continued on next page)

Table 6-7. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 1B | RESERVED | This code is reserved. |
| 1C | PRIMARY DEFECT LIST NOT FOUND | The controller could not locate the manufacturer's defect list. |
| 1D | COMPARE ERROR | One or more bytes did not compare when the **VERIFY** or **READ BUFFER** command was issued. |
| 1E-1F | RESERVED | These codes are reserved. |
| 20 | INVALID COMMAND | The Initiator issued a command that cannot be executed, or is not applicable. |
| 21 | INVALID BLOCK ADDRESS | The addressed block was not valid. |
| 22 | ILLEGAL FUNCTION FOR DEVICE TYPE | The addressed LUN is unable to perform the requested function. |
| 23 | RESERVED | This code is reserved. |
| 24 | ILLEGAL FIELD IN CDB | A field in the Command Descriptor Block is reserved and contains a value other than zero or, the value in the field is incorrect. |
| 25 | INVALID LUN | The LUN specified in the Command Descriptor Block or the SCSI IDENTIFY message is not supported by the controller. |
| 26 | ILLEGAL FIELD IN PARAMETER LIST | A field in the Parameter List is reserved and contains a value other than 0 or, the value in the field is incorrect. |

(continued on next page)

Table 6-7. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 27 | WRITE PROTECTED | The disk is write protected. The outstanding WRITE command is aborted. |
| 28 | MEDIUM CHANGE | When the controller detects a media changed condition, it will set the Sense Key/Error Code to UNIT ATTENTION/MEDIA CHANGED (06h/28h). A media changed condition is defined as when the controller detects a drive offline condition then detects the drive come online. |
| 29 | POWER UP OR RESET | The controller has been reset by a SCSI Bus Reset, Bus Device Reset Message, or Power On Reset condition. After the controller detects the condition, it reports this error when the first command (except the **INQUIRY** command) is issued to it. The controller clears this condition when the next command is issued to it by the same Initiator. The controller reports the UNIT ATTENTION Sense Key to all SCSI devices that subsequently issue a command to it. |

(continued on next page)

Table 6-7. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 2A | **MODE SELECT** CHANGED CONDITION | The **MODE SELECT** parameters for this device have been changed by another Initiator and may affect current operations. After the controller detects the condition, it reports the error when the first command (except the **INQUIRY** command) is issued to it. The controller clears this condition when the next command is issued to it by the same Initiator. The controller reports the UNIT ATTENTION Sense Key to all SCSI devices that subsequently issue a command to it. |
| 2B-2F | RESERVED | These codes are reserved. |
| 30 | RESERVED | This code is reserved. |
| 31 | FORMAT FAILED | The **FORMAT UNIT** command encountered an error while attempting to access on the defect lists. |
| 32 | NO DEFECT SPARE LOCATION AVAILABLE ERROR | There are no remaining alternate tracks on the addressed LUN. This error condition may occur during the processing of a **FORMAT UNIT** or **RE-ASSIGN BLOCK** command. |
| 33-3F | RESERVED | These codes are reserved. |

(continued on next page)

Table 6-7. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 40 | RAM FAILURE | The controller detected a RAM memory error during a **SEND DIAGNOSTIC** test operation. |
| 41-42 | RESERVED | These codes are reserved. |
| 43 | MESSAGE REJECT ERROR | The Initiator responded with a MESSAGE REJECT message to a message sent by the controller. |
| 44 | SCSI HARDWARE/ FIRMWARE ERROR | The SCSI Firmware detected an internal firmware or hardware error and was unable to complete the current command. |
| 45 | SELECT/RESELECT FAILED ERROR | The SCSI firmware detected a timeout error while attempting a Selection or Reselection. |
| 46 | RESERVED | This code is reserved. |
| 47 | PARITY ERROR | A parity error occurred on the SCSI Bus and the controller was unable to recover the data. |
| 48 | INITIATOR DETECTED ERROR | The Initiator sent an INITIATOR DETECTED ERROR message and the controller was unable to recover from the error. |
| 49 | INAPPROPRIATE/ ILLEGAL MESSAGE ERROR | The Initiator sent an Inappropriate or Illegal SCSI Message to the controller. |

(continued on next page)

Table 6-7. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 4A-4F | RESERVED | These codes are reserved. |
| 50-5F | RESERVED | These codes are reserved. |
| 60-6F | RESERVED | These codes are reserved. |
| 70-7F | RESERVED | These codes are reserved. |
| 80-8F | RESERVED (Vendor Unique) | Reserved for future use by EMULEX. |

6.2.11.3.3 **REQUEST SENSE** Extended Sense Data Format. The Extended Sense Data format is available for all commands and will be returned by the controller when the Allocation Length specified in the **REQUEST SENSE** command is greater than 0 bytes.

**REQUEST SENSE Extended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | Segment Number | | | | | | | |
| 2 | FMark | EOM | ILI | Rsrv | Sense Key | | | |
| 3 | Information Byte (MSB) | | | | | | | |
| 4 | Information Byte | | | | | | | |
| 5 | Information Byte | | | | | | | |
| 6 | Information Byte (LSB) | | | | | | | |
| 7 | Additional Sense Length | | | | | | | |
| 8 | Reserved for Copy/Compare Command | | | | | | | |
| 9 | Reserved for Copy/Compare Command | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Additional Sense Code | | | | | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set, indicates that the Information Byte field contains valid information related to the error condition.

The **Segment Number** field <Byte 1> contains the current segment number if the Extended Sense information is in response to a COPY or COMPARE command. For all other commands this field will be set to 0.

The **Filemark (FMark)** field <Byte 2, Bit 7>, **End Of Media (EOM)** field <Byte 2, Bit 6>, and **Incorrect Length Indicator (ILI)** field <Byte 2, Bit 5> are for sequential-access devices and are not used in the controller. These bits will always be set to 0.

The **Information Bytes** <Bytes 3 through 6> are command specific and are only valid when the Valid field bit is set to 1. Refer to the individual command for the information returned in this field.

The **Additional Sense Length** field <Byte 7> specifies the number of bytes of Additional Sense Data that follows. If the Allocation Length specified in the CDB is too small to transfer the Additional Sense Data, the Additional Sense Length field is not adjusted to reflect the truncation. The Additional Sense Data contains information that further defines the nature of the CHECK CONDITION status code. The Additional Sense Length byte will be set to ten (10) for all commands, except the **COPY** and COMPARE commands.

Bytes 8 and 9 are reserved for the **COPY** and **COMPARE** commands. Please refer to the **COPY** or **COMPARE** command for additional information on these two bytes.

The **Additional Sense Code** field <Byte 12> contains additional information about the error that occurred. This field contains the Error Class and Code that are returned when the Nonextended Sense Data format is used. Please refer to the Nonextended Sense Data format for a complete listing of these error codes. When this field is set to 0, the controller does not have any additional sense information.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an

ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. It points to the most significant bit in error if multiple bits are detected in error. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** field <Byte 2, Bits 0:3> indicates status information about any errors detected during the operation. The errors are listed and defined below.

6.2.11.3.4 **REQUEST SENSE** Extended Sense Key Definitions Table 6-8 lists and describes the extended sense keys supported by the controller.

Table 6-8. Extended Sense Key Codes

| Hex Code | Error | Description |
|---|---|---|
| 00 | NO SENSE | There is no Sense Key information to be reported for the designated LUN. This code occurs for a successfully completed command. |
| 01 | RECOVERED ERROR | The last command was completed successfully, but with some recovery action performed by the Target. |
| 02 | NOT READY | The addressed LUN cannot be accessed. Operator intervention may be required. |
| 03 | MEDIUM ERROR | The command terminated with a nonrecoverable-error condition which was probably caused by a flaw in the media or by an error in the recorded data. |
| 04 | HARDWARE ERROR | A nonrecoverable hardware error (e.g., controller failure, device failure, parity error, etc.) was detected while the Target was performing the command or while the Target was performing a Self-Test operation. |

(continued on next page)

Table 6-8. Extended Sense Key Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 05 | ILLEGAL REQUEST | There was an illegal parameter in the command or in the additional required parameters supplied as data for some related commands. If the error is detected in the Command Descriptor Block, the controller will not alter the Media. |
| 06 | UNIT ATTENTION | The addressed LUN has been reset. This error is reported the first time any command is issued after the condition is detected; then the requested command is not performed. This condition is cleared when the next command is issued by the same Initiator. UNIT ATTENTION is reported to all SCSI devices that subsequently issue a command to the LUN. |
| 07 | DATA PROTECT | A write operation was attempted on a write-protected device. |
| 08 | RESERVED | This key is reserved. |
| 09 | VENDOR UNIQUE | A Vendor-Unique error condition occurred. The corresponding nonextended error class and code are specified in Byte 12 of the Extended Sense Byte. This code is currently not returned by the controller. |
| 0A | COPY/COMPARE ABORTED | A COPY or COMPARE command was aborted because an error condition was detected on the source and/or destination device. |

(continued on next page)

Table 6-8. Extended Sense Key Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 0B | ABORTED COMMAND | The Target aborted the command. The Initiator may recover by trying to execute the command again. |
| 0C | RESERVED | This key is reserved. |
| 0D | RESERVED | This key is reserved. |
| 0E | MISCOMPARE | Used by the **VERIFY** command to indicate the source data did not match the data read from the disk. |
| 0F | RESERVED | This key is reserved. |

### 6.2.11.4 REQUEST SENSE Error Conditions

If a CHECK CONDITION status is received on a **REQUEST SENSE** command, any Sense Data returned by the Target is invalid.

### 6.2.12 RESERVE UNIT - 16H

The **RESERVE UNIT** command is used to reserve the specified LUN for exclusive use by the Initiator. This reservation remains in effect until one of the following conditions releases the reservation:

- A **RELEASE UNIT** command from the same Initiator is received by the controller.
- Another **RESERVE UNIT** command from the same Initiator is received by the controller.
- A BUS DEVICE RESET message is received by the controller from any Initiator.
- A SCSI Bus Reset occurs.

If a **RESERVE UNIT** command, or any other command, is received for a Logical Unit which is reserved by another Initiator, the controller will return a RESERVATION CONFLICT status.

An Initiator that holds a current reservation may modify that reservation by issuing another **RESERVE UNIT** command. The superseding RESERVE command shall release the previous reservation when the new reservation is granted. The previous reservation will not be modified if the new reservation cannot be granted and the controller will return a RESERVATION CONFLICT status.

#### 6.2.12.1 RESERVE UNIT Command Options

The controller does not support the Extent Reservation Option.

The controller does not support Reservation Queuing. If a **RESERVE UNIT** command is received for a Logical Unit which is already reserved, the controller will return a RESERVATION CONFLICT status.

#### 6.2.12.2 RESERVE UNIT SCSI Deviations

There are no deviations.

### 6.2.12.3 RESERVE UNIT Command Parameters

The **RESERVE UNIT** Command Descriptor Block is formatted as shown below.

| RESERVED UNIT Command Descriptor Block | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | Logical Unit Number | | | 3rdpty | 3rd Party Device ID | | | Extent |
| 2 | Reservation Identification | | | | | | | |
| 3 | Extent List Length (MSB) | | | | | | | |
| 4 | Extent List Length (LSB) | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Third Party Reservation (3rdpty)** field <Byte 1, Bit 4> allows an Initiator to reserve a LUN for another device on the SCSI bus. If the 3rdpty bit is set to 1, an Initiator is allowed to reserve the specified LUN for the SCSI bus device specified in the Third Party ID field <Byte 1, Bits 1:3>. This option is intended for use in multiple-Initiator systems in which the **COPY/COMPARE** commands are used. Any device that uses the 3rdpty option to reserve a LUN must also use the Third Party options to release the LUN (See the **RELEASE UNIT** command) before any other commands can be sent to the LUN.

The **Third Party Device ID** field <Byte 1, Bits 1:3> specifies the SCSI bus device ID for which the LUN is being reserved. The ID is valid only when the 3rdpty field <Byte 1, Bit 4> is set to 1.

The **Extent** bit <Byte 1, Bit 0> indicates if the Extent Reservation option is implemented. **The controller does not support this option and the field must be set to 0.**

**Reservation Identification** field <Byte 2> allows the Initiator to identify each extent reservation (which allows an Initiator to have multiple reservations outstanding in multi-tasking environments). **The controller does not support this option and the field must be set to 0.**

**Extent List Length** <Bytes 3 and 4> specifies the number of extents in the reservation request. **The controller does not support this option and the field must be set to 0.**

### 6.2.12.4 **RESERVE UNIT** Error Conditions

If the Extent, Reservation Identification or Extent List Length fields are not 0, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

### 6.2.12.5 **RESERVE UNIT** Sense Information

Errors detected by the controller during the **RESERVE UNIT** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**RESERVE UNIT Nonextended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**RESERVE UNIT Extended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7> will always be set to 0.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **RESERVE UNIT** command the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Hardware Error (04h) | SCSI Msg Reject Error (43h)<br>SCSI Hardware Error (44h)<br>Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h)<br>Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h)<br>Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h)<br>Initiator Detected Error (48h)<br>Illegal Message (49h) |

### 6.2.13 REZERO UNIT - 01H

The **REZERO UNIT** command requests that the controller set the logical unit to logical block address zero.

#### 6.2.13.1 REZERO UNIT SCSI Deviations

The **REZERO UNIT** command will always force the disk drive to position the heads on physical cylinder zero.

#### 6.2.13.2 REZERO UNIT Command Parameters

The **REZERO UNIT** Command Descriptor Block is formatted as shown below.

**REZERO UNIT Command Descriptor Block**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

#### 6.2.13.3 REZERO UNIT Sense Information

Errors detected by the controller during the **REZERO UNIT** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**REZERO UNIT Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| REZERO UNIT Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **REZERO UNIT** command the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| Not Ready (02h) | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Hardware Error (04h) | Drive Not Ready (04h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.2.14 SEEK - 0BH

The **SEEK** command causes the selected LUN to seek to the specified logical block location. If the logical block number specifies a block on a defective track, the seek to the alternate track is not performed until the controller receives and processes a command which accesses the media.

#### 6.2.14.1 SEEK Command Parameters

The **SEEK** Command Descriptor Block is formatted as shown below.

**SEEK Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | Logical Unit Number | | | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Logical Block Address** field <Bytes 1 through 3> specifies the Logical Block Address to seek to. When the **SEEK** command completes without any errors the LUN will be positioned at the specified block address.

#### 6.2.14.2 SEEK Error Conditions

If the Logical Block Address is invalid, the controller will terminate the **SEEK** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

#### 6.2.14.3 SEEK Sense Information

Errors detected by the controller during the **SEEK** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

| SEEK Nonextended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

| SEEK Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Logical Block Address (MSB) | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address | | | | | | | |
| 6 | Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7> when set to 1 indicates the Logical Block Address information is valid. If the valid bit is not 1, the Logical Block information should be ignored.

The **Logical Block Address** field <Bytes 3 through 6> is the first invalid block address detected by the **SEEK** command.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **SEEK** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | No ID field Address Mark (12h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

## 6.2.15 SEND DIAGNOSTIC - 1DH

The **SEND DIAGNOSTIC** command requests the controller to perform diagnostic tests on itself, on the attached peripheral device(s), or on both. This command is usually followed by the **RECEIVE DIAGNOSTIC RESULTS** command except when the Self Test bit (SlfTst) is set to 1.

### 6.2.15.1 SEND DIAGNOSTIC Command Limitations

Except for the Self-Test mode, any errors that occur during the execution of a diagnostic subcommand will be reported as a CHECK CONDITION status for the **RECEIVE DIAGNOSTIC RESULTS** command. This conditions occurs because the diagnostic subcommand is not actually executed until the **RECEIVE DIAGNOSTIC RESULTS** command is issued.

### 6.2.15.2 SEND DIAGNOSTIC SCSI Deviations

Except for Self-Test, the specified diagnostic command is not actually executed until the controller receives a **RECEIVE DIAGNOSTIC RESULTS** command from the same Initiator that issued the **SEND DIAGNOSTIC** command.

The Device Offline and Unit Offline fields are only used for the Self-Test mode of the **SEND DIAGNOSTIC** command.

### 6.2.15.3 SEND DIAGNOSTIC Command Parameters

The **SEND DIAGNOSTICS** Command Descriptor Block is formatted as shown below.

**SEND DIAGNOSTIC Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | Logical Unit Number | | | Reserved | | SlfTst | DevOfl | UntOfl |
| 2 | Reserved | | | | | | | |
| 3 | Parameter List Length (MSB) | | | | | | | |
| 4 | Parameter List Length (LSB) | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Self-Test (SlfTst)** field <Byte 1, Bit 2> directs the controller to perform the Self-Test on the LUN specified in the Command Descriptor Block. No device access will occur during this test.

The **Device Offline (DevOfl)** field <Byte 1, Bit 1> directs the controller to perform the controller diagnostic tests. This field is not supported by the controller and should always be set to zero.

The **Unit Offline (UntOfl)** field <Byte 1, Bit 0> directs the controller to perform the Device Diagnostics on the LUN specified in the Command Descriptor Block. This field is only used when the Self-Test bit is set to 1 and must be set to 0 at all other times.

The **Parameter List Length** field <Bytes 3 through 4> specifies the length (in bytes) of the Parameter List transferred from the Initiator to the controller during the Data Out Phase of the **SEND DIAGNOSTIC** command. The Parameter List contains a diagnostic subcommand and any additional information required. Each subcommand is described in a section below. When the Self-Test bit is set to 1, the Parameter List Length must be set to 0.

### 6.2.15.4 Controller Self-Test

The Unit Offline field, in conjunction with the Self-Test field, specifies the type of tests to be performed by the controller. Table 6-9 lists and describes the options and the tests performed.

**NOTE**

If the initiator of the SEND DIAGNOSTIC command does not support disconnect, the SCSI chip portion of the self-test will not be performed, as it requires that the controller be disconnected from the bus.

Table 6-9. Controller Self Test Options

| Bits SlfTst | DevOfl | UntOfl | Tests Performed |
|---|---|---|---|
| 1 | 0 | 0 | This mode directs the controller to perform a limited power up self-test on the controller without disturbing the operation of any other commands. The controller performs tests on the SCSI controller chip, buffer controller chip, disk formatter chip, and the RAM memory for the LUN specified in the CDB or SCSI ID message. |
| 1 | 0 | 1 | This mode directs the controller to perform the limited self-test and device diagnostics on the LUN specified in the CDB, without disturbing the operation of any other commands. The controller will perform tests on the SCSI controller chip, disk formatter chip, buffer controller chip, RAM memory, and the device diagnostics (described below) for the LUN specified in the CDB or SCSI ID message. |

6.2.15.4.1 Self Test - Device Diagnostics. The device diagnostics perform write and verify operations on each surface of the diagnostic cylinder of the specified disk drive.

If any sectors on the diagnostic cylinder cannot be written or verified by using two different bit patterns, the controller terminates the command with a CHECK CONDITION status code and sets the Sense Key/Error Code to MEDIA ERROR/UNCORRECTABLE DATA ERROR (03h/11h). The Sense Information bytes contain the number of bad sectors found on this cylinder.

### 6.2.15.5 SEND DIAGNOSTIC Error Conditions

If the Device Offline field is not 0, the controller terminates the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the Self-test field is 0 and the Unit Offline field is not 0, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the Self-test field is set to 1 and the Parameter List Length is not 0, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

If the diagnostic subcommand specified in the parameter list is not legal, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

### 6.2.15.6 SEND DIAGNOSTIC Sense Information

Errors detected by the controller during the **SEND DIAGNOSTIC** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**SEND DIAGNOSTIC Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**SEND DIAGNOSTIC Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7> will always be set to 0.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **SEND DIAGNOSTIC** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| | Recoverable Read Error With ECC (18h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID field Address Mark (12h) |
| | No Data field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Write Fault (03h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| | Invalid Field in Parameter List (26h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Data Protect (07h) | Write Protected (27h) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.2.15.7 GET DRIVE STATUS Subcommand

The GET DRIVE STATUS subcommand, shown below, causes the controller to return unmodified status from the disk drive. The controller begins to execute the subcommand and transfer the status data when it has received the appropriate **RECEIVE DIAGNOSTIC RESULTS** command from the Initiator.

| GET DRIVE STATUS Subcommand Parameter List | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | Number of Bytes to Transfer | | | | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Reserved | | | | | | | |

The **Subcommand Code** <Byte 0> is set to 03 (hexadecimal) to specify the GET DRIVE STATUS diagnostic subcommand.

The **Number of Bytes to Transfer** <Byte 1> specifies the number of status bytes to transfer to the Initiator during the **RECEIVE DIAGNOSTIC RESULTS** command. The GET DRIVE STATUS subcommand will always return 5 bytes or the number specified in this field, whichever is less.

The remaining bytes are reserved and must be set to 0.

6.2.15.7.1 GET DRIVE STATUS Data Format. Data for the GET DRIVE STATUS diagnostic subcommand is transferred in the following format:

| GET DRIVE STATUS Data Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Command Status | | | | | | | |
| 1 | ESDI Drive Status (MSB) | | | | | | | |
| 2 | ESDI Drive Status (LSB) | | | | | | | |
| 3 | Vendor Unique Status (MSB) | | | | | | | |
| 4 | Vendor Unique Status (LSB) | | | | | | | |

The **Command Status** field <Byte 0> indicates the status of the command. Table 6-10 lists the possible values for this field.

**NOTE**

If the last ESDI command to the disk drive caused a fault condition, the controller will return the status of the previous command. Otherwise, it will get the current status from the drive.

Table 6-10. GET DRIVE STATUS Command Status

| Status Code | Description |
|---|---|
| 00h | **No Status.** The four bytes in the status data format are not valid. |
| 01h | **Error Condition.** The controller encountered an error. The information in the status bytes is not valid. |
| 02h | **ESDI Status Valid.** The information in the ESDI Drive Status field is valid. |
| 04h | **ESDI and Vendor Unique Status Valid.** The information in the ESDI Drive Status and the Vendor Unique Status fields is valid. |

The **ESDI Drive Status** field <Bytes 1 and 2> indicates the status of the ESDI drive. The status returned is specific to the particular drive connected to the controller.

**NOTE**

The controller clears the ESDI Drive Status information after it has processed the GET DRIVE STATUS subcommand.

The **Vendor Unique Status** field <Bytes 3 and 4> specifies vendor unique status. The status returned is specific to the drive.

### 6.2.15.8 PASS DRIVE COMMAND Subcommand

The PASS DRIVE COMMAND subcommand, shown below, causes the controller to pass the disk drive command from the Initiator to the disk drive. The controller begins to execute the subcommand and transfer status data when it has received the appropriate **RECEIVE DIAGNOSTIC RESULTS** command from the Initiator.

| PASS DRIVE COMMAND Subcommand Parameter List | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | Number of Bytes to Transfer | | | | | | | |
| 2 | Drive Command (MSB) | | | | | | | |
| 3 | Drive Command (LSB) | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Reserved | | | | | | | |

The **Subcommand Code** <Byte 0> is set to 04 (hexadecimal) to specify the PASS DRIVE COMMAND diagnostic subcommand.

The **Number of Bytes to Transfer** <Byte 1> specifies the number of status bytes to transfer to the Initiator during the **RECEIVE DIAGNOSTIC RESULTS** command. The Initiator may request up to 5 bytes of drive status. The PASS DRIVE COMMAND subcommand will always return 5 bytes or the number specified in this field, whichever is less.

The **Drive Command** field <Bytes 2 through 3> specifies a command specific to the ESDI disk drive connected to the controller.

The remaining bytes are reserved and must be set to 0.

6.2.15.8.1 PASS DRIVE COMMAND Status Data Format. The format for the status data returned from the PASS DRIVE COMMAND diagnostic subcommand is shown below.

| PASS DRIVE COMMAND Status Data Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Command Status | | | | | | | |
| 1 | ESDI Drive Status (MSB) | | | | | | | |
| 2 | ESDI Drive Status (LSB) | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |

The **Command Status** field <Byte 0> indicates either the status of the command or the number of status bytes returned by the controller. Table 6-11 lists the possible values for this field.

Table 6-11. PASS DRIVE COMMAND Status

| Status Code | Description |
|---|---|
| 00h | **No Status.** The four bytes in the status data format are not valid. |
| 01h | **Error Condition.** The controller encountered an error. The information in the status bytes is not valid. |
| 02h | **ESDI Status Valid.** The information in the ESDI Drive Status field is valid. |

**NOTE**

The controller clears the ESDI Drive Status information after it has processed the PASS DRIVE COMMAND subcommand.

The **ESDI Drive Status** field <Bytes 1 and 2> indicate the status of the ESDI drive. The status returned is specific to the particular drive connected to the controller.

### 6.2.15.10 READ DISK PARTITIONS Subcommand

The READ DISK PARTITIONS diagnostic subcommand, shown below, causes the controller to return the block addresses of the bad sector file, alternate track area, and diagnostic area. The controller begins to execute the subcommand and transfer data when it has received the appropriate **RECEIVE DIAGNOSTIC RESULTS** command from the Initiator.

| **READ DISK PARTITIONS Subcommand Parameter List** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | Reserved | | | | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Reserved | | | | | | | |

The **Subcommand Code** <Byte 0> is set to 06 (hexadecimal) to specify the READ DISK PARTITIONS diagnostic subcommand.

The remaining bytes are reserved and must be set to 0.

6.2.15.10.1 READ DISK PARTITIONS Data Format. Data for the READ DISK PARTITIONS diagnostic subcommand is transferred in the format shown below.

**READ DISK PARTITIONS Data Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (Start of Bad Sector File) PHAD | | | Block Address (MSB) | | | | |
| 1 | | | | Block Address | | | | |
| 2 | | | | Block Address | | | | |
| 3 | | | | Block Address (LSB) | | | | |
| 4 | (Start of Alternate Track Storage) PHAD | | | Block Address (MSB) | | | | |
| 5 | | | | Block Address | | | | |
| 6 | | | | Block Address | | | | |
| 7 | | | | Block Address (LSB) | | | | |
| 8 | (Start of Diagnostic Cylinder) PHAD | | | Block Address (MSB) | | | | |
| 9 | | | | Block Address | | | | |
| 10 | | | | Block Address | | | | |
| 11 | | | | Block Address (LSB) | | | | |

The **Physical Address (PHAD)** field <Byte 0, Bit 7; Byte 4, Bit7; Byte 8, Bit 7> indicates if the controller interprets the address (specified in Bytes 0 through 3) as a physical block address or as a logical address. The PHAD bit is always set to 1 to indicate the controller interprets the address as a physical block address.

The **Bad Sector File Block Address** <Bytes 0 through 3> bytes specify the starting block address of the Bad Sector File. Note that the PHAD bit is always set to 1.

The **Alternate Track Storage Block Address** <Bytes 4 through 7> specify the starting logical block address of the Alternate Track Storage. The PHAD bit is always set to 1 to indicate the controller interprets the address as a physical block address.

The **Diagnostic Cylinder Block Address** <Bytes 8 through B> specify the starting block address of the Diagnostic Cylinder. The PHAD bit is always set to 1 to indicate the controller interprets the address as a physical block address.

### 6.2.15.11 READ HEADER Subcommand

The READ HEADER diagnostic subcommand, shown below, causes the controller to perform a read operation of the header address field for all blocks on a track. The controller begins to execute the subcommand and transfer data when it has received the appropriate **RECEIVE DIAGNOSTIC RESULTS** command from the Initiator.

**READ HEADER Subcommand Parameter List**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | PHAD | Block Address (MSB) | | | | | | |
| 2 | Block Address | | | | | | | |
| 3 | Block Address | | | | | | | |
| 4 | Block Address (LSB) | | | | | | | |
| 5 | Reserved | | | | | | | |

The **Subcommand Code** <Byte 0> is set to hexadecimal 07 to specify the READ HEADER diagnostic subcommand.

The **Physical Address (PHAD)** field <Byte 1, Bit 7> indicates if the controller interprets the address (specified in Bytes 2 through 5) as a physical block address or as a logical block address. If the PHAD bit is set to 1, the controller interprets the address as a physical block address. If the PHAD bit is 0, the controller interprets the address as a logical address.

The **Block Address** field <Bytes 1 through 4> specifies any block address on the track where the read header diagnostic operation is performed.

The remaining bytes are reserved and must be set to 0.

6.2.15.11.1 READ HEADER Data Format. The controller transfers data for the READ HEADER diagnostic subcommand during the Data Out phase in the format shown below. The data format is repeated for each block in the track.

**READ HEADER Data Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Cylinder Number (MSB) | | | | | | | |
| 1 | Cylinder Number (LSB) | | | | | | | |
| 2 | Head Number | | | | | | | |
| 3 | Sector Number | | | | | | | |
| 4 | DT | SP | 0 | 0 | 0 | 0 | 0 | 0 |

The **Cylinder Number** field <Bytes 1 through 2> indicate the cylinder number of the specified track.

The **Head Number** field <Byte 3> indicates the head number.

The **Sector Number** field <Byte 4> indicates the sector number.

The **Defective Track (DT)** field <Byte 5, Bit 7>, when set to 1, indicates the entire track is defective.

The **Spare Sector (SP)** field <Byte 5, Bit 6>, when set to 1, indicates this is a spare sector on the track

## 6.2.16 START/STOP UNIT - 1BH

The **START/STOP UNIT** command requests that the controller enable or disable the logical unit for further operations. The controller automatically performs a **STOP/START UNIT** function during its initialization operation.

### 6.2.16.1 START/STOP UNIT SCSI Deviations

There are no deviations.

### 6.2.16.2 START/STOP UNIT Command Parameters

The Command Descriptor Block for the **START/STOP UNIT** command is formatted as shown below.

**START/STOP UNIT Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | | Immed |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | Start |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Immediate** field <Byte 0, Bit 0>, when set to 1, indicates the status will be returned as soon as the operation is initiated. If the the immediate bit is 0, the controller will return the status when the operation is completed.

The **Start** field <Byte 5, Bit 0>, when set to 1, requests the logical unit be made ready for use. A Start field of 0 requests that the logical unit be stopped.

### 6.2.16.3 START/STOP UNIT Error Conditions

If a command which accesses the media is sent to the controller after a STOP UNIT command, the controller will terminate the command with a CHECK CONDITION status and set the Sense Key/Error Code to NOT READY/DRIVE NOT READY (02h/04h).

### 6.2.16.4 START/STOP UNIT Sense Information

Errors detected by the controller during the **START/STOP UNIT** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**START/STOP UNIT Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**START/STOP UNIT Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7> will always be set to a 0.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **START/STOP UNIT** command the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Not Ready (02h) | Drive Not Ready (04h) |
| Hardware Error (04h) | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

## 6.2.17 TEST UNIT READY - 00H

The **TEST UNIT READY** command provides a means for the Initiator to check if the logical unit is ready.

### 6.2.17.1 TEST UNIT READY SCSI Deviations

There are no deviations.

### 6.2.17.2 TEST UNIT READY Command Parameters

The Command Descriptor Block for the **TEST UNIT READY** command is formatted as shown below.

**TEST UNIT READY Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

### 6.2.17.3 TEST UNIT READY Error Conditions

If the logical unit is not ready, the controller will return a CHECK CONDITION or BUSY status code in response to this command. A **REQUEST SENSE** command can be issued to obtain detailed information about the reason the disk drive is not ready (unavailable).

### 6.2.17.4 TEST UNIT READY Sense Information

Errors detected by the controller during the **TEST UNIT READY** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

| TEST UNIT READY Nonextended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| TEST UNIT READY Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7> will always be set to 0.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **TEST UNIT READY** command the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an

ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Not Ready (02h) | Drive Not Ready (04h) |
| Hardware Error (04h) | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.2.18 WRITE - 0AH

The **WRITE** command requests that the controller write the data transferred by the Initiator to the logical unit.

#### 6.2.18.1 WRITE SCSI Deviations

A Vendor Unique bit, in the Command Descriptor Block Control Byte, is used to disable Retries.

#### 6.2.18.2 WRITE Command Parameters

The Command Descriptor Block for the **WRITE** command is formatted as shown below.

**WRITE Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | Logical Unit Number | | | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |
| 4 | Transfer Length | | | | | | | |
| 5 | VU | ERTY | Reserved | | | | Flag | Link |

The **Logical Block Address** field <Bytes 1 through 3> specifies the logical block at which the write operation will begin.

The **Transfer Length** field <Byte 4> specifies the number of contiguous logical blocks of data to be transferred. A Transfer Length of 0 indicates that 256 logical blocks will be transferred.

The **Vendor Unique (VU)** field <Byte 5, bit 7> is reserved and should be set to 0.

The **Error Retry (ERTY)** Field <Byte 5, bit 6> is vendor unique and indicates if Error Retries are disabled. If this bit is set to 1, no retries will be attempted when an error is encountered during the current **WRITE** command. Setting the ERTY bit produces the same condition as setting the DDR bit (in the **MODE SELECT** Error Recovery Page) to 1 and setting the Retry Count field (in the **MODE SELECT** Error Recovery Page) to 0. When the ERTY bit is 0, the controller will use the **MODE SELECT** error recovery flags when an error is encountered during the current **WRITE** command.

**NOTE**

It is not recommended that the Initiator use the ERTY bit as it is used only to maintain compatibility with previous products and will not be used in any future products.

### 6.2.18.3 WRITE Error Conditions

If the Logical Block Address is invalid, the controller will terminate the **WRITE** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

If the Logical Block Address plus the Transfer Length results in an invalid block address, the controller will terminate the **WRITE** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code. No data will be transferred if this condition occurs.

### 6.2.18.4 WRITE Sense Information

Errors detected by the controller during the **WRITE** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**WRITE Nonextended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

| WRITE Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Logical Block Address (MSB) | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address | | | | | | | |
| 6 | Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to 1, indicates the Logical Block Address information is valid. If the valid bit is not 1, the Logical Block information should be ignored.

The **Logical Block Address** field <Bytes 3 through 6> contains the first invalid block address, if any, detected by the **WRITE** command.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **WRITE** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | No ID field Address Mark (12h) |
| | No Data field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Write Fault (03h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| | Invalid Field in Parameter List (50h) |
| Unit Attention (06h) | Medium Change (28h) |
| | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Data Protect (07h) | Write Protected (27h) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

## 6.3 SCSI GROUP CODE 1 COMMAND DESCRIPTIONS

This subsection provides a detailed description of the SCSI Group Code 1 commands, including Command Descriptor Block formats, hexadecimal operation code, byte and bit functions, and any necessary effects produced by the commands. Each SCSI command is described in a separate subsection. A sample Group 1 Command Descriptor Block is shown below with an explanation of those fields which are common among all commands.

**Typical Group 1 Command Descriptor Block**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Group Code | | | Command Code | | | | |
| 1 | Logical Unit Number | | | Reserved | | | | RelAdr |
| 2 | Command Dependent Parameters | | | | | | | |
| 3 | Command Dependent Parameters | | | | | | | |
| 4 | Command Dependent Parameters | | | | | | | |
| 5 | Command Dependent Parameters | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Command Dependent Parameters | | | | | | | |
| 8 | Command Dependent Parameters | | | | | | | |
| 9 | Vendor Unique | | | Reserved | | | Flag | Link |

The **Reserved** bits, bytes, fields and code values are set aside for future standardization. All Reserved bits, bytes or fields are checked by the controller and **must** be set to 0. If a Reserved field is not set to 0 the command will be terminated with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL FIELD IN CDB (05h/24h) or ILLEGAL REQUEST/ILLEGAL FIELD IN PARAMETER LIST (05h/26h) Sense Key/Error Code.

The **Operation Code** is the first byte <Byte 00> of a Command Descriptor Block. The Operation Code contains two fields: the Group Code in the high-order three bits <Bits 05:07>, and the Command Code in the low-order five bits <Bits 00:04>. The Group Code specifies the length of the Command Descriptor Block, and together with the Command Code determines the operation to be performed. If the specified Operation Code is invalid or not implemented, a CHECK CONDITION status will be returned with an ILLEGAL REQUEST/INVALID COMMAND (05/20) Sense Key/Code.

The **Logical Unit Number** field <Byte 01, Bits 05:07> contains the number of the device being addressed. The controller, acting as a SCSI bus Target, supports up to four LUNs. Therefore, the value for the LUN field, in byte 01 of the CDB, is limited to 000, 001, 010 or 011. If an invalid LUN is specified, a CHECK CONDITION status will be returned with an ILLEGAL REQUEST/INVALID LUN (05h/25h) Sense Key/Error Code.

This method of addressing is provided for Initiators that do not implement the IDENTIFY message. A LUN specified in the IDENTIFY message will override any LUN specified in the Command Descriptor Block.

Setting the **Relative Address (RelAdr)** bit to 1 causes the Logical Block Address to be treated as a two's complement displacement. This displacement is added to the Logical Block Address last accessed on the LUN to form the Logical Block Address for this command. This feature is only available when the linking commands function is performed. The feature requires a previous command in the linked group to have accessed a Logical Block on the LUN; if it has not, the controller terminates the command, sends a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

The **Control Byte** is the last byte <Byte 09> in every Command Descriptor Block. The Control Byte is separated into four fields:

- The Vendor Unique control bits <Bits 06:07> are used as Command Dependent Parameters. Unless otherwise specified in the command description, these bits are to be treated as reserved.
- Bits <05:02> are reserved.
- Bit <01> is defined as the Flag bit and is only checked when the Link bit is set to a 1. When the Flag bit is 0, the controller will send a LINKED COMMAND COMPLETE Message when the command completes successfully. If the Flag bit is set to 1, a LINKED COMMAND COMPLETE (WITH FLAG) Message is sent when the command completes successfully.
- Bit <00> is defined as the Link bit. When the current command completes successfully and the Link bit is set to a 1, the controller will return an INTERMEDIATE status, followed by one of the two messages defined by the Flag bit above. The controller will then automatically link to the next command. If a linked command is not completed successfully, the controller will return a CHECK CONDITION status and will not link to the next command.

The remaining bytes in the Command Descriptor Block are primarily **Command Dependent** and are described in the individual command sections.

### 6.3.1 COMPARE - 39H

The controller does not currently support the **COMPARE** command. However, it will be supported in future releases.

If any attempt is made to execute the **COMPARE** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL COMMAND (05H/20H) Sense Key/Error Code.

### 6.3.2 COPY AND VERIFY - 3AH

The **COPY AND VERIFY** command will not be implemented.

If any attempt is made to execute the **COPY AND VERIFY** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL COMMAND (05H/20H) Sense Key/Error Code.

### 6.3.3 READ BUFFER - 3CH

The **READ BUFFER** command is used in conjunction with the **WRITE BUFFER** command as a diagnostic function for testing the controller's data buffer memory and the SCSI bus integrity. There is no media access with this command.

#### 6.3.3.1 READ BUFFER Command Limitations

The **READ BUFFER** command will only read the buffer area associated with the LUN specified in the Command Descriptor Block. Buffer areas reserved for other LUN's will not be affected by this command.

#### 6.3.3.2 READ BUFFER SCSI Deviations

There are no deviations.

#### 6.3.3.3 READ BUFFER Command Usage

It is recommended that the Initiator issue the **RESERVE UNIT** command before it issues the **READ BUFFER** command to ensure that no other Initiator sends data to the controller's data buffer. After the controller has completed the execution of the **READ BUFFER** command, the Initiator should issue a **RELEASE UNIT** command to release the LUN attached to the controller.

To determine the maximum amount of data that can be transferred with the **READ BUFFER** and **WRITE BUFFER** commands, the Initiator can issue a **READ BUFFER** command with the Allocation Length set to 4. Bytes 2 and 3 returned by the controller will contain the maximum buffer size for the specified LUN.

#### 6.3.3.4 READ BUFFER Command Parameters

The **READ BUFFER** Command Descriptor Block is formatted as shown below.

| READ BUFFER Command Descriptor Block | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | VU |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Reserved | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Allocation Length (MSB) | | | | | | | |
| 8 | Allocation Length (LSB) | | | | | | | |
| 9 | Vendor Unique | | | Reserved | | | Flag | Link |

The **Vendor Unique (VU)** field <Byte 1, Bit 0; Byte 9, Bits 7:6> must always be set to 0.

The **Allocation Length** field <Bytes 7 and 8> specifies the number of bytes the Initiator has allocated for the returned buffer data. An allocation length of 0 is not considered an error by the controller and no data will be sent to the Initiator. The Initiator may request up to 65,535 bytes to be transferred, including the four-byte header. If the number of bytes requested exceeds the controller buffer size, the controller will transfer the entire buffer and terminate the command without an error. Under this condition, the Initiator must check the value in the Available Length field in the **READ BUFFER** data format to determine the number of bytes returned.

### 6.3.3.5 READ BUFFER Data Format

The data returned from the **READ BUFFER** command during the Data In Phase consists of a four byte header immediately followed by the data bytes from the controller data buffer. This data is formatted as shown below.

| READ BUFFER Data Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | Maximum Available Length (MSB) | | | | | | | |
| 3 | Maximum Available Length (LSB) | | | | | | | |
| 4 | Data Byte 0 | | | | | | | |
| | . . | | | | | | | |
| n | Data Byte n | | | | | | | |

The **Maximum Available Length** field <Bytes 2 and 3> specifies the maximum amount of memory that the controller has available in its data buffer. This may or may not be the number of bytes actually transferred depending on the Allocation Length specified in the Command Descriptor Block.

The **Data Byte** field <Bytes 04 through "n"> contains the data from the controller's data buffer.

### 6.3.3.6 READ BUFFER Error Conditions

If the data in the buffer may have been modified since the last **WRITE BUFFER** command was issued, the **READ BUFFER** command will be terminated with a CHECK CONDITION status and a MISCOMPARE/COMPARE ERROR (0Eh/1Dh) Sense Key/Error Code. If the Allocation Length is set to 4 or less, the controller will not return this error.

### 6.3.3.7 READ BUFFER Sense Information

Errors detected by the controller during the **READ BUFFER** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**READ BUFFER Nonextended Sense Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | | | | | | | |
| 2 | 0 | | | | | | | |
| 3 | 0 | | | | | | | |

| READ BUFFER Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | 0 | Rsrv | Sense Key | | | |
| 3 | 0 | | | | | | | |
| 4 | 0 | | | | | | | |
| 5 | 0 | | | | | | | |
| 6 | 0 | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **READ BUFFER** command the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Hardware Error (04h) | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Illegal Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Select/Reselect Failed (45h) |
| | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |
| Miscompare (0Eh) | Compare Error (1Dh) |

### 6.3.4 READ CAPACITY - 25H

The **READ CAPACITY** command is used to determine the maximum logical block number on the specified LUN which can be accessed by the Initiator. This command also returns the size of logical block. The information is returned to the Initiator during the Data In phase.

#### 6.3.4.1 READ CAPACITY SCSI Deviations

There are no deviations.

#### 6.3.4.2 Read Capacity Command Parameters

The **READ CAPACITY** Command Descriptor Block is formatted as shown below.

**READ CAPACITY Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | | RelAdr |
| 2 | Logical Block Address (MSB) | | | | | | | |
| 3 | Logical Block Address | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address (LSB) | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Reserved | | | | | | | |
| 8 | Vendor Unique | | | Reserved | | | | PMI |
| 9 | Vendor Unique | | | Reserved | | | Flag | Link |

The **Logical Block Address** field <Bytes 2 through 5> is only used when the PMI bit is set to 1. This field specifies the block address to use when computing the last block before a substantial delay is encountered.

A **Partial Medium Indicator (PMI)** field <Byte 8, Bit 0> of 1 indicates that the information returned will be for the last full logical block which can be transferred before a substantial delay is encountered (e.g., a cylinder boundary). A PMI of 0 indicates that the information returned will be for the last logical block of the specified LUN.

### 6.3.4.3 READ CAPACITY Data Format

The eight bytes of **READ CAPACITY** data will be sent to the Initiator during the Data In phase and will be formatted as shown below.

**READ CAPACITY Data Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Logical Block Address (MSB) | | | | | | | |
| 1 | Logical Block Address | | | | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |
| 4 | Block Length (MSB) | | | | | | | |
| 5 | Block Length | | | | | | | |
| 6 | Block Length | | | | | | | |
| 7 | Block Length (LSB) | | | | | | | |

The **Logical Block Address** field <Bytes 0 through 3> specifies the last logical block on the unit (if the PMI bit is 0) or the last full logical block before a substantial delay is encountered (if the PMI bit is 1).

The **Block Length** field <Bytes 4 through 7> specifies the size of the logical block in bytes.

### 6.3.4.4 READ CAPACITY Error Conditions

If the PMI bit is 1 and the Logical Block Address is invalid, the controller will terminate the **READ CAPACITY** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

If the PMI bit is 0 and the Logical Block Address is not 0, the controller will terminate the **READ CAPACITY** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

### 6.3.4.5 READ CAPACITY Sense Information

Errors detected by the controller during the **READ CAPACITY** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

| READ CAPACITY Nonextended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| READ CAPACITY Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **READ CAPACITY** command the Additional Sense Length will always be set to ten (0Ah).

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Hardware Error (04h) | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| | SCSI Msg Reject Error (43h) |
| Illegal Request (05h) | Illegal Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.3.5 READ DEFECT LIST - 37H

The **READ DEFECT LIST** command requests that the controller transfer the defect list maintained by the controller to the Initiator.

#### 6.3.5.1 READ DEFECT LIST SCSI Deviations

There are no deviations.

#### 6.3.5.2 READ DEFECT LIST Command Parameters

The **READ DEFECT LIST** Command Descriptor Block is formatted as shown below.

**READ DEFECT LIST Command Descriptor Block**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | Reserved | | | MDL | GDL | Defect List Format | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Reserved | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Allocation Length (MSB) | | | | | | | |
| 8 | Allocation Length (LSB) | | | | | | | |
| 9 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Manufacturer's Defect List (MDL)** field <Byte 2, Bit 4>, when set to a 1, specifies the Manufacturer's Defect List should be returned by the controller during the Data In Phase. To request more than 1 list, this bit may be set in combination with the GDL bit. Note: This is a request by the Initiator and is not considered an error if the controller cannot return the requested defect list. When this bit is set to 0, the controller does not return the Manufacturer's Defect List.

The **Grown Defect List (GDL)** field <Byte 2, Bit 3>, when set to a 1, specifies the Grown Defect List should be returned by the controller during the Data In Phase. To request more than 1 list, this bit may be set in combination with the MDL bit. Note: This is a request by the Initiator and is not considered an error if the controller cannot return the requested defect list. When this bit is set to 0, the controller does not return the Grown Defect List.

When both the MDL and GDL bits are set to 1, the controller returns both the Manufacturers Defect List and the Growing Defect List. The controller will determine the order in which it returns the lists, and if it will merge the lists. When both the MDL and GDL bits are set to 0, the controller returns only the Defect List Header.

Neither the MDL or GDL will be returned before the drive has been formatted. Only an internally maintained list can be returned prior to formatting.

The **Defect List Format** field <Byte 1, Bits 0:2> specifies the preferred format of the returned defect list. The controller does not have to return the list in the requested format. Table 6-12 lists the types of formats supported by the controller.

Table 6-12. **READ DEFECT LIST** Formats

| Defect List Format Bits 2 1 0 | Description |
|---|---|
| 0 X X | **Block format.** The Initiator requests that the Defect List be returned in Block format. |
| 1 0 0 | **Bytes from Index format.** The Initiator requests that the defect list be returned in Bytes from Index format. This format is the same as used by the **FORMAT UNIT** command, so it is recommended when reading the list to save it for later use with the **FORMAT UNIT** command. |
| 1 0 1 | **Physical Sector format.** The Initiator requests that the defect list be returned in Physical Sector format. |
| 1 1 0 | Vendor Unique (Reserved). |
| 1 1 1 | Reserved |

The **Allocation Length** <Bytes 7 and 8> specifies the number of bytes the Initiator has allocated for the returned defect list. The controller will terminate the Data In Phase when Allocation Length bytes have been transferred or the entire Defect List has been transferred, whichever is less.

The **Vendor Unique** field <Byte 9, Bits 7:6> should always be set to 0.

### 6.3.5.3 READ DEFECT LIST Header

The Defect List Header is transferred from the controller to the Initiator during the Data In Phase of the **READ DEFECT LIST** command. The Defect List Header is four-bytes long followed by zero or more Defect Descriptors. The header specifies the format and the total number of bytes in the returned defect list.

**Defect List Header Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 1 | Reserved | | | MDL | GDL | Defect List Format | | |
| 2 | Defect List Length (MSB) | | | | | | | |
| 3 | Defect List Length (LSB) | | | | | | | |

The **Manufacturers Defect List (MDL)** field <Byte 02, Bit 05>, when set to a 1, indicates that the controller is returning the Manufacturers Defect List during the Data In Phase. This bit may be set in combination with the GDL bit.

The **Grown Defect List (GDL)** field <Byte 02, Bit 04>, when set to a 1, indicates the controller is returning the Grown Defect List during the Data In Phase. This bit may be set in combination with the MDL bit.

The **Defect List Format** field <Byte 1, Bits 00:02> specifies the format of the returned defect list. The controller supports the Block Address, Physical Sector, and Bytes from Index formats. If the Initiator requests the Bytes from Index format, the controller will return the list in the Physical Sector format.

The **Defect List Length** field <Bytes 02 through 03> specifies the total number of bytes (not the total number of defect descriptors) in the defect list. This length does not include the four bytes in the header.

### 6.3.5.4 READ DEFECT LIST Block Format

When the Block Address format is specified, the Defect List transferred to the Initiator from the controller during the Data In Phase, will be formatted as shown below.

| Defect List Block Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Reserved | | | | MDL | GDL | Defect List Format | |
| 2 | Defect List Length (MSB) | | | | | | | |
| 3 | Defect List Length (LSB) | | | | | | | |
| | Defect List Descriptor(s) | | | | | | | |
| 0 | Defect Block Address (MSB) | | | | | | | |
| 1 | Defect Block Address | | | | | | | |
| 2 | Defect Block Address | | | | | | | |
| 3 | Defect Block Address (LSB) | | | | | | | |

The **Defect List Format** <Byte 1, Bits 0:2> will be set to 0 to indicate the list is in Block format.

The **Defect List Block Address** <Bytes 0 through 4 of the Defect Descriptor) specifies the physical address of the block that contains the defect.

Each defect descriptor for the Block Format mode specifies a 4-byte defect physical block address that contains the defect.

### 6.3.5.5 READ DEFECT LIST Bytes From Index Format

When the Bytes from Index format is specified, the Defect List, transferred to the Initiator from the controller during the Data In Phase, will be formatted as shown below.

| Defect List Bytes From Index Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Reserved | | | | MDL | GDL | Defect List Format | |
| 2 | Defect List Length (MSB) | | | | | | | |
| 3 | Defect List Length (LSB) | | | | | | | |
| | Defect List Descriptor(s) | | | | | | | |
| 0 | Cylinder Number of Defect (MSB) | | | | | | | |
| 1 | Cylinder Number of Defect | | | | | | | |
| 2 | Cylinder Number of Defect (LSB) | | | | | | | |
| 3 | Head Number of Defect | | | | | | | |
| 4 | Defect Bytes from Index (MSB) | | | | | | | |
| 5 | Defect Bytes from Index | | | | | | | |
| 6 | Defect Bytes from Index | | | | | | | |
| 7 | Defect Bytes from Index (LSB) | | | | | | | |

The **Cylinder Number of Defect** field <Bytes 0 through 2 of the Defect Descriptor> specifies the physical cylinder number which contains the defect.

The **Head Number of Defect** field <Byte 3 of the Defect Descriptor> specifies the head number which contains the defect.

The **Defect Bytes from Index** field <Bytes 4 through 7 of the Defect Descriptor> specifies the number of bytes between the index and the defect on the specified track. A value of 0FFFFFFFF (hexadecimal) in the Bytes from Index field indicates that the entire track is to be reassigned.

### 6.3.5.6 READ DEFECT LIST Physical Sector Format

When the Physical Sector format is specified, the Defect List, transferred to the Initiator from the controller during the Data In Phase, will be formatted as shown below.

| Defect List Physical Sector Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Reserved | | | MDL | GDL | Defect List Format | | |
| 2 | Defect List Length (MSB) | | | | | | | |
| 3 | Defect List Length (LSB) | | | | | | | |
| | Defect List Descriptor(s) | | | | | | | |
| 0 | Cylinder Number of Defect (MSB) | | | | | | | |
| 1 | Cylinder Number of Defect | | | | | | | |
| 2 | Cylinder Number of Defect (LSB) | | | | | | | |
| 3 | Head Number of Defect | | | | | | | |
| 4 | Defect Sector Number (MSB) | | | | | | | |
| 5 | Defect Sector Number | | | | | | | |
| 6 | Defect Sector Number | | | | | | | |
| 7 | Defect Sector Number (LSB) | | | | | | | |

The **Defect List Format** <Byte 1, Bits 0:2> will be set to 05h to indicate the list is in Physical Sector format.

The **Cylinder Number of Defect** <Bytes 0 through 2 of the Defect Descriptor> specifies the cylinder number which contains the defect.

The **Head Number of Defect** <Byte 3 of the Defect Descriptor> specifies the head number which contains the defect.

The **Defect Sector Number** <Bytes 4 through 7 of the Defect Descriptor> specifies the sector number which contains the defect. A defect sector number of 0FFFFFFFF (hexadecimal) indicates that the entire track was reassigned.

### 6.3.5.7 **READ DEFECT LIST** Error Conditions

If the Preferred Defect List Format does not specify Block Address, Bytes from Index or Physical Format, the controller will terminate the **READ DEFECT LIST** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID FIELD IN CDB (05h/24h) Sense Key/Error Code.

### 6.3.5.8 READ DEFECT LIST Sense Information

Errors detected by the controller during the **READ DEFECT LIST** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**READ DEFECT LIST Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Residual Count (MSB) | | | | |
| 2 | Residual Count | | | | | | | |
| 3 | Residual Count (LSB) | | | | | | | |

| READ DEFECT LIST Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Residual Count (MSB) | | | | | | | |
| 4 | Residual Count | | | | | | | |
| 5 | Residual Count | | | | | | | |
| 6 | Residual Count (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 00, Bit 07>, when set to a 1, indicates the residual count data is valid. If this bit is a 0, the information in the Residual Count field should be ignored.

The **Residual Count** field <Bytes 03 through 06> contains the number of bytes not transferred, when the size of the Defect List exceeds the requested Allocation Length.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **READ DEFECT LIST** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h)<br>Seek Error (15h)<br>Recoverable Read Error With Retries (17h)<br>Recoverable Read Error With ECC (18h)<br>Defect List Error (19h)<br>Primary Defect List Not Found (1Ch) |
| Not Ready (02h) | Drive Not Ready (04h)<br>Drive Not Selected (05h)<br>Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h)<br>Uncorrectable Data Error (11h)<br>No ID field Address Mark (12h)<br>No Data field Address Mark (13h)<br>Block Not Found (14h)<br>Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h)<br>Drive Not Ready (04h)<br>Drive Not Selected (05h)<br>No Track 0 (06h)<br>ID Field Address Mark Not Found (12h)<br>SCSI Msg Reject Error (43h)<br>SCSI Hardware Error (44h)<br>Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h)<br>Invalid LUN (25h)<br>Primary Defect List Not Found (1Ch) |
| Unit Attention (06h) | Power Up or Reset (29h)<br>Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h)<br>Initiator Detected Error (48h)<br>Illegal Message (49h) |

### 6.3.6 READ (EXTENDED) - 28H

The **READ (EXTENDED)** command requests that the controller transfer data from the logical unit to the Initiator.

#### 6.3.6.1 READ (EXTENDED) Command Parameters

The **READ (EXTENDED)** Command Descriptor Block is formatted as shown below.

| READ (EXTENDED) Command Descriptor Block | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | RelAdr |
| 2 | Logical Block Address (MSB) | | | | | | | |
| 3 | Logical Block Address | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address (LSB) | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Transfer Length (MSB) | | | | | | | |
| 8 | Transfer Length (LSB) | | | | | | | |
| 9 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Logical Block Address** <Bytes 2 through 5> specifies the logical block at which the read operation will begin.

The **Transfer Length** <Bytes 7 and 8> specifies the number of contiguous logical blocks of data to be transferred. A Transfer Length of 0 indicates that no data will be transferred and is not considered an error by the controller.

The **Vendor Unique** field <Byte 9, Bits 7:6> should always be set to 0.

#### 6.3.6.2 READ (EXTENDED) Error Conditions

If the Logical Block Address is invalid, the controller will terminate the **READ** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

If the Logical Block Address plus the Transfer Length results in an invalid block address, the controller will terminate the **READ** command with a CHECK CONDITION status and an ILLEGAL REQUEST /ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code. No data will be transferred if this condition occurs.

### 6.3.6.3 READ (EXTENDED) Sense Information

Errors detected by the controller during the **READ (EXTENDED)** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**READ (EXTENDED) Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

**READ (EXTENDED) Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Logical Block Address (MSB) | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address | | | | | | | |
| 6 | Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to a 1, indicates the Logical Block Address information is valid. If the valid bit is not a 1, the Logical Block information should be ignored.

The **Logical Block Address** is the first invalid block address detected by the **READ** command.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **READ** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| | Recoverable Read Error With ECC (18h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID field Address Mark (12h) |
| | No Data field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track 0 (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.3.7 SEARCH DATA EQUAL - 31H

The **SEARCH DATA EQUAL** command will not be implemented.

If any attempt is made to execute the **SEARCH DATA EQUAL** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL COMMAND (05h/20h) Sense Key/Error Code.

### 6.3.8 SEARCH DATA HIGH - 30H

The **SEARCH DATA HIGH** command will not be implemented.

If any attempt is made to execute the **SEARCH DATA HIGH** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL COMMAND (05h/20h) Sense Key/Error Code.

### 6.3.9 SEARCH DATA LOW - 32H

The **SEARCH DATA LOW** command will not be implemented.

If any attempt is made to execute the **SEARCH DATA LOW** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL COMMAND (05h/20h) Sense Key/Error Code.

### 6.3.10 SEEK (EXTENDED) - 2BH

The **SEEK (EXTENDED)** command causes the selected LUN to begin a seek operation to the specified logical block location. If the logical block number specifies a block on a defective track, the seek to the alternate track is not performed until the controller receives and processes an I/O command.

#### 6.3.10.1 SEEK (EXTENDED) Command Parameters

The **SEEK (EXTENDED)** Command Descriptor Block is formatted as shown below.

**SEEK (EXTENDED) Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | | RelAdr |
| 2 | Logical Block Address (MSB) | | | | | | | |
| 3 | Logical Block Address | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address (LSB) | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Reserved | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Vendor Unique | | | Reserved | | | Flag | Link |

The **Logical Block Address** specifies the block address to seek to. When the **SEEK** command completes without any errors the LUN will be positioned at the specified block address.

#### 6.3.10.2 SEEK (EXTENDED) Error Conditions

If the Logical Block Address is invalid, the controller will terminate the **SEEK** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

### 6.3.10.3 SEEK (EXTENDED) Sense Information

Errors detected by the controller during the **SEEK** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**SEEK (EXTENDED) Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

**SEEK (EXTENDED) Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Logical Block Address (MSB) | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address | | | | | | | |
| 6 | Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to a 1, indicates the Logical Block Address information is valid. If the valid bit is not a 1, the Logical Block information should be ignored.

The **Logical Block Address** is the first invalid block address detected by the **SEEK** command.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **SEEK** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type |
| Medium Error (03h) | ID field CRC Error (10h) |
| | No ID field Address Mark (12h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track 0 (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.3.11 SET LIMITS - 33H

The **SET LIMITS** command will not be implemented.

If any attempt is made to execute the **SET LIMITS** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL COMMAND (05h/20h) Sense Key/Error Code.

### 6.3.12 VERIFY - 2FH

The **VERIFY** command requests that the controller verify the data written on the logical unit.

#### 6.3.12.1 VERIFY Command Parameters

The **VERIFY** Command Descriptor Block is formatted as shown below.

| VERIFY Command Descriptor Block | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | BytChk | RelAdr |
| 2 | Logical Block Address (MSB) | | | | | | | |
| 3 | Logical Block Address | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address (LSB) | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Verification Length (MSB) | | | | | | | |
| 8 | Verification Length (LSB) | | | | | | | |
| 9 | Vendor Unique | | | Reserved | | | Flag | Link |

The **Byte Check (BytChk)** field <Byte 1, Bit 1>, if set to 1, causes the specified logical blocks to be read from the disk and compared with the data transferred from the Initiator. Data is transferred from the Initiator just as in a write operation. If the BytChk bit is set to 0, the data is read from the disk and the ECC is checked for correctness; no Data Transfer operation occurs between the Initiator and the controller.

The **Logical Block Address** <Bytes 2 through 5> specifies the logical block at which the Verify operation will begin.

The **Verification Length** <Bytes 7 and 8> specifies the number of contiguous logical blocks of data to be verified. A Verification Length of 0 indicates that no data will be verified and is not considered an error by the controller.

#### 6.3.12.2 VERIFY Error Conditions

If the Logical Block Address is invalid, the controller will terminate the **VERIFY** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

If the Logical Block Address plus the Verification Length results in an invalid block address, the controller will terminate the **VERIFY** command with a CHECK CONDITION status and an "Illegal Request/Illegal Block Address (05h/21h)" Sense Key/Error Code.

### 6.3.12.3 **VERIFY** Sense Information

Errors detected by the controller during the **VERIFY** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**VERIFY Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

**VERIFY Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Logical Block Address (MSB) | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address | | | | | | | |
| 6 | Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to a 1, indicates the Logical Block Address information is valid. If the valid bit is not a 1, the Logical Block information should be ignored.

The **Logical Block Address** is the first invalid block address detected by the **VERIFY** command.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **VERIFY** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| | Recoverable Read Error With ECC (18h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID Field Address Mark (12h) |
| | No Data Field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track 0 (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |
| Miscompare (0Eh) | Compare Error (1Dh) |

### 6.3.13 WRITE BUFFER - 3BH

The **WRITE BUFFER** command is used in conjunction with the **READ BUFFER** command as a diagnostic function for testing the controller's data buffer memory and the SCSI bus integrity. There is no media access with this command.

#### 6.3.13.1 WRITE BUFFER Command Limitations

The **WRITE BUFFER** command will only write data to the buffer area associated with the LUN specified in the Command Descriptor Block. Buffer areas reserved for other LUN's will not be affected by this command.

#### 6.3.13.2 WRITE BUFFER SCSI Deviations

There are no deviations.

#### 6.3.13.3 WRITE BUFFER Command Usage

To determine the maximum amount of data that can be transferred with the **READ BUFFER** and **WRITE BUFFER** commands, the Initiator can issue a **READ BUFFER** command with the Allocation Length set to 4. Bytes 2 and 3 returned by the controller will contain the maximum buffer size for the specified LUN.

#### 6.3.13.4 WRITE BUFFER Command Parameters

The **WRITE BUFFER** Command Descriptor Block is formatted as shown below.

| WRITE BUFFER Command Descriptor Block | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | | VU |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Reserved | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Byte Transfer Length (MSB) | | | | | | | |
| 8 | Byte Transfer Length (LSB) | | | | | | | |
| 9 | Vendor Unique | | | Reserved | | | Flag | Link |

The **Vendor Unique (VU)** field <Byte 1, Bit 0; Byte 9, Bits 7:6> must always be set to 0.

The **Byte Transfer Length** field <Bytes 7 and 8> specifies the number of bytes the controller will transfer during the Data Out Phase to its internal buffer. The transfer length includes the four bytes of header information sent before the actual data. A transfer length of 0 is not considered an error by the controller and no data will be expected or read from the Initiator. If the number of bytes requested exceeds the controller buffer size, the controller will not read any data from the Initiator and it will terminate the command with a CHECK CONDITION (see the section on errors below).

### 6.3.13.5 WRITE BUFFER Data Format

The data sent by the Initiator during the Data Out Phase consists of a four-byte header immediately followed by the data bytes to write to the controller data buffer. This data is formatted as shown below.

| WRITE BUFFER Data Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Reserved | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Data Byte 0 | | | | | | | |
| | . | | | | | | | |
| | . | | | | | | | |
| n | Data Byte n | | | | | | | |

The Data Byte Field <Bytes 04 through n> contains the data to be written to the controller data buffer.

### 6.3.13.6 WRITE BUFFER Error Conditions

If the Byte Transfer Length exceeds the size of the controller's buffers, the **WRITE BUFFER** command will be terminated with a CHECK CONDITION status and a ILLEGAL REQUEST/ILLEGAL FIELD IN CDB (05h/24h) Sense Key/Error Code.

### 6.3.13.7 WRITE BUFFER Sense Information

Errors detected by the controller during the **WRITE BUFFER** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**WRITE BUFFER Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | | | | | | | |
| 2 | 0 | | | | | | | |
| 3 | 0 | | | | | | | |

**Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | 0 | Rsrv | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **WRITE BUFFER** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Hardware Error (04h) | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.3.14 WRITE (EXTENDED) - 2AH

The **WRITE (EXTENDED)** command requests that the controller write the data transferred by the Initiator to the specified logical unit.

#### 6.3.14.1 WRITE (EXTENDED) Command Parameters

The Command Descriptor Block for the **WRITE (EXTENDED)** command is formatted as shown below.

**WRITE (EXTENDED) Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | RelAdr |
| 2 | Logical Block Address (MSB) | | | | | | | |
| 3 | Logical Block Address | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address (LSB) | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Transfer Length (MSB) | | | | | | | |
| 8 | Transfer Length (LSB) | | | | | | | |
| 9 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Logical Block Address** <Bytes 2 through 5> specifies the logical block at which the write operation will begin.

The **Transfer Length** <Bytes 7 and 8> specifies the number of contiguous logical blocks of data to be transferred. A Transfer Length of 0 indicates that no data will be transferred and is not considered an error by the controller.

The **Vendor Unique** field <Byte 9, bits 7:6> is reserved and should be set to 0.

#### 6.3.14.2 WRITE (EXTENDED) Error Conditions

If the Logical Block Address is invalid, the controller will terminate the **WRITE** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

If the Logical Block Address plus the Transfer Length results in an invalid block address, the controller will terminate the **WRITE** command with a CHECK CONDITION status and an ILLEGAL REQUEST /ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

### 6.3.14.3 WRITE (EXTENDED) Sense Information

Errors detected by the controller during the **WRITE (EXTENDED)** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**WRITE (EXTENDED) Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

**WRITE (EXTENDED) Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Logical Block Address (MSB) | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address | | | | | | | |
| 6 | Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to a 1, indicates the Logical Block Address information is valid. If the valid bit is not a 1, the Logical Block information should be ignored.

The **Logical Block Address** is the first invalid block address detected by the **WRITE (EXTENDED)** command.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **WRITE** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID Field Address Mark (12h) |
| | No Data Field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Write Fault (03h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track 0 (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Data Protect (07h) | Write Protected (27h) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

### 6.3.15 **WRITE AND VERIFY** - 2EH

The **WRITE AND VERIFY** command will not be implemented.

If any attempt is made to execute the **WRITE AND VERIFY** command, the controller will terminate the command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL COMMAND (05h/20h) Sense Key/Error Code.

## 6.4 SCSI GROUP CODE 6 COMMAND DESCRIPTIONS

This subsection provides detailed descriptions of the SCSI Group Code 6 commands, which are vendor-unique commands. Each SCSI command is described in a separate subsection.

The bytes in the Command Descriptor Block for Group Code 6 are command-dependent and are defined in the subsections for each individual Group Code 6 command.

## 6.4.1 READ REVISION LEVEL - C1H

### 6.4.1.1 READ REVISION LEVEL Command Description

The **READ REVISION LEVEL** command returns the current revision level of the PROM residing on the controller to the Initiator during the Data In phase.

**NOTE**

This command is provided for compatibility with previous controllers and should not be used. This same information can be obtained using the **INQUIRY** command.

### 6.4.1.2 READ REVISION LEVEL Command Parameters

The **READ REVISION LEVEL** Command Descriptor Block is formatted as shown below.

**READ REVISION LEVEL Command Descriptor Block**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | Reserved | | | | | | | |
| 3 | Reserved | | | | | | | |
| 4 | Reserved | | | | | | | |
| 5 | Vendor Unique | | Reserved | | | | Flag | Link |

### 6.4.1.3 READ REVISION LEVEL Data Format

The **READ REVISION LEVEL** command will return 6 bytes of data to the Initiator during the Data In phase. This data is formatted as follows:

**READ REVISION LEVEL Data Format**

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Product Designation | | | | | | | |
| 1 | Product Designation | | | | | | | |
| 2 | Product Designation | | | | | | | |
| 3 | Major Revision Level | | | | | | | |
| 4 | Engineering Revision Level | | | | | | | |
| 5 | PROM Checksum | | | | | | | |

The **Product Designation** field <Bytes 0 through 2> contains the ASCII Emulex product designation, which indicates the model number for the controller firmware. For example, the MD21 Controller will return A98 (in ASCII) in these bytes.

The **Major Revision** field <Byte 3> indicates the major version number (in ASCII) of the product.

The **Engineering Revision** field <Byte 4> indicates the engineering revision (in hexadecimal) of the product. If the product is a released version, this byte is 0.

The **Checksum** field <Byte 5> contains the checksum of the firmware on the controller.

### 6.4.1.4 READ REVISION LEVEL Sense Information

Errors detected by the controller during the **READ REVISION LEVEL** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**READ REVISION LEVEL Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| READ REVISION LEVEL Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **READ REVISION LEVEL** command, the Additional Sense Length will always be set to ten decimal.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Hardware Error (04h) | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

## 6.5 SCSI Group Code 7 Command Descriptions

This subsection provides detailed descriptions of the SCSI Group Code 7 commands, which are vendor-unique commands. Each SCSI command is described in a separate subsection.

The bytes in the Command Descriptor Block for Group Code 7 are command-dependent and are defined in the subsections for each individual Group Code 7 command.

### 6.5.1 FORMAT TRACK - E4H

The **FORMAT TRACK** command formats a single physical track according to the current parameters established with the **MODE SELECT** command.

**CAUTION**

**Use extreme caution when issuing this command to the controller.** Since this command specifies a physical track, the Initiator can format any part of the disk drive. If the Initiator uses this command to format any areas of the disk that are reserved by the controller, critical information (such as defect lists or drive configuration information) could be lost. In this situation, the entire disk would have to be formatted for normal operation. Further, when the controller executes this command, it does not use the defect information when it formats a track. If this command is issued to a track which contains a defect, the data contained in the spare sector(s) will be lost.

#### 6.5.1.1 FORMAT TRACK Command Parameters

The **FORMAT TRACK** Command Descriptor Block is formatted as shown below.

**FORMAT TRACK Command Descriptor Block**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | Cylinder Number (MSB) | | | | | | | |
| 3 | Cylinder Number | | | | | | | |
| 4 | Cylinder Number (LSB) | | | | | | | |
| 5 | Head Number | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Reserved | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Vendor Unique | | Reserved | | | | Flag | Link |

The **Cylinder Number** field <Bytes 2 through 4> specifies the physical cylinder to be formatted. The most significant byte of this field must be 0.

The **Head Number** field <Byte 5> specifies which head of the cylinder is to be formatted.

### 6.5.1.2 **FORMAT TRACK** Error Conditions

If the value for the Cylinder Number exceeds the number of cylinders set by the **MODE SELECT** command, the controller will terminate the **FORMAT TRACK** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

If the value for the Head Number exceeds the number of heads set by the **MODE SELECT** command, the controller will terminate the **FORMAT TRACK** command with a CHECK CONDITION status and an ILLEGAL REQUEST/INVALID BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

### 6.5.1.3 **FORMAT TRACK** Sense Information

Errors detected by the controller during the **FORMAT TRACK** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**FORMAT TRACK Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Logical Block Address (MSB) | | | | |
| 2 | Logical Block Address | | | | | | | |
| 3 | Logical Block Address (LSB) | | | | | | | |

| FORMAT TRACK Extended Sense Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Logical Block Address (MSB) | | | | | | | |
| 4 | Logical Block Address | | | | | | | |
| 5 | Logical Block Address | | | | | | | |
| 6 | Logical Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to 1, indicates the Logical Block Address information is valid. If the valid bit is not 1, the Logical Block information should be ignored.

The **Logical Block Address** field <Bytes 3 through 6> is the first invalid block address detected by the **FORMAT TRACK** command. It is calculated as follows:

Logical Block Address = Cylinder * Head * Sectors per track
+ Sector # on the track that failed

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **FORMAT TRACK** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | No ID field Address Mark (12h) |
| | No Data field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Write Fault (03h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Data Protect (07h) | Write Protected (27h) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

## 6.5.2 READ LONG - E8H

### 6.5.2.1 READ LONG Command Description

The **READ LONG** command requests the controller to perform a read operation of one data block and the six ECC bytes associated with that block. The data from the block and the ECC bytes are transferred to the Initiator during the Data In phase.

**NOTE**

The READ LONG command does not perform any ECC correction when reading the disk.

### 6.5.2.2 READ LONG Command Parameters

The **READ LONG** Command Descriptor Block is formatted as shown below.

**READ LONG Command Descriptor Block**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | PHAD | Block Address (MSB) | | | | | | |
| 3 | Block Address | | | | | | | |
| 4 | Block Address | | | | | | | |
| 5 | Block Address (LSB) | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Reserved | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Vendor Unique | | | Reserved | | | Flag | Link |

The **Physical Address (PHAD)** field <Byte 02, Bit 07> specifies if the controller should interpret the address (specified in Bytes 2 through 5) as a physical address or as a logical address. If the PHAD bit is set to 1, the controller interprets the address as a physical address. If the PHAD bit is 0, the controller interprets the address as a logical address.

The **Block Address** <Bytes 2 through 5> specifies the block at which the read long operation will begin.

**CAUTION**

The physical location of the **READ LONG** and **WRITE LONG** block address is always computed using the physical sector size instead of the logical block size.

### 6.5.2.3 READ LONG Error Conditions

If the Block Address is invalid, the controller will terminate the **READ LONG** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

### 6.5.2.4 READ LONG Sense Information

Errors detected by the controller during the **READ LONG** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**READ LONG Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Block Address (MSB) | | | | |
| 2 | Block Address | | | | | | | |
| 3 | Block Address (LSB) | | | | | | | |

**READ LONG Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Block Address (MSB) | | | | | | | |
| 4 | Block Address | | | | | | | |
| 5 | Block Address | | | | | | | |
| 6 | Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to a one, indicates the Block Address information is valid. If the valid bit is not a one, the Block address should be ignored.

The **Block Address** is the address of the invalid block detected by the **READ LONG** command. This will either be a logical or physical address depending on the value of the PHAD field in the Command Descriptor Block.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **READ LONG** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID Field Address Mark (12h) |
| | No Data Field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

## 6.5.3 WRITE LONG - EAH

### 6.5.3.1 WRITE LONG Command Description

The **WRITE LONG** command requests the controller to perform a write operation of one data block and the six bytes of ECC information. The data and the six ECC bytes for the specified logical block are supplied by the Initiator during the Data Out phase.

### 6.5.3.2 WRITE LONG Command Parameters

The Command Descriptor Block for the **WRITE LONG** command is formatted as shown below.

**WRITE LONG Command Descriptor Block**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | Logical Unit Number | | | Reserved | | | | |
| 2 | PHAD | Block Address (MSB) | | | | | | |
| 3 | Block Address | | | | | | | |
| 4 | Block Address | | | | | | | |
| 5 | Block Address (LSB) | | | | | | | |
| 6 | Reserved | | | | | | | |
| 7 | Reserved | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Vendor Unique | | | Reserved | | | Flag | Link |

The **Physical Address (PHAD)** field <Byte 02, Bit 07> specifies if the controller should interpret the address (specified in Bytes 2 through 5) as a physical address or as a logical address. If the PHAD bit is set to 1, the controller interprets the address as a physical block address. If the PHAD bit is 0, the controller interprets the address as a logical block address.

The **Block Address** <Bytes 2 through 5> specifies the block at which the write long operation will begin.

**CAUTION**

The physical location of the **READ LONG** and **WRITE LONG** block address is always computed using the physical sector size instead of the logical block size.

### 6.5.3.3 WRITE LONG Error Conditions

If the Logical Block Address is invalid, the controller will terminate the **WRITE LONG** command with a CHECK CONDITION status and an ILLEGAL REQUEST/ILLEGAL BLOCK ADDRESS (05h/21h) Sense Key/Error Code.

### 6.5.3.4 WRITE LONG Sense Information

Errors detected by the controller during the **WRITE LONG** command will cause the controller to terminate the command with a CHECK CONDITION status. Any error information requested by the Initiator via the **REQUEST SENSE** command will be formatted as shown below.

**WRITE LONG Nonextended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | Error Class | | | Error Code | | | |
| 1 | 0 | 0 | 0 | Block Address (MSB) | | | | |
| 2 | Block Address | | | | | | | |
| 3 | Block Address (LSB) | | | | | | | |

**WRITE LONG Extended Sense Format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Valid | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Reserved | | | | Sense Key | | | |
| 3 | Block Address (MSB) | | | | | | | |
| 4 | Block Address | | | | | | | |
| 5 | Block Address | | | | | | | |
| 6 | Block Address (LSB) | | | | | | | |
| 7 | Additional Sense Length (0Ah) | | | | | | | |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Error Class | | | | Error Code | | | |
| 13 | Reserved | | | | | | | |
| 14 | Field Replaceable Unit (FRU) | | | | | | | |
| 15 | FPV | C/D | Reserved | | BPV | Bit Pointer | | |
| 16 | Field Pointer | | | | | | | |
| 17 | Field Pointer | | | | | | | |

The **Valid** field <Byte 0, bit 7>, when set to a one, indicates the Block Address information is valid. If the valid bit is not a one, the Block address should be ignored.

The **Block Address** is the address of the invalid block detected by the **WRITE LONG** command. This will either be a logical or physical address depending on the value of the PHAD field in the Command Descriptor Block.

The **Additional Sense Length** <Byte 7> specifies the number of additional sense bytes which follow. For the **WRITE LONG** command, the Additional Sense Length will always be set to ten.

The **Field Replaceable Unit (FRU)** field <Byte 14> is a numeric code which indicates which field replaceable unit in the controller failed. The controller does not use this field and it will always be set to 0.

The **Field Pointer Valid (FPV)** field <Byte 15, Bit 7>, when set to 1, indicates the information in the C/D bit and BPV bit fields, plus the Field Pointer (Bytes 16 and 17) field is valid. Normally the Field Pointer will only be valid when an ILLEGAL REQUEST Sense Key is returned. In this situation, the Field Pointer points to the byte which caused the error. When this field is set to 0, the controller does not have any information on an exact CDB or location in error.

The **Command/Data (C/D)** field <Byte 15, Bit 6>, when set to 1, indicates the Field Pointer is pointing to a byte in the Command Descriptor Block. When this bit is 0, it indicates the Field Pointer is pointing to a byte in the command parameters which were passed to the controller during the Data Out phase. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer Valid (BPV)** field <Byte 15, Bit 3>, when set to 1, indicates the information in the Bit Pointer field is valid. This bit is only valid if the Field Pointer Valid Bit is set to 1.

The **Bit Pointer** field <Byte 15, Bits 0:2> specifies the bit position which was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Bit Pointer Valid bit is set to 1.

The **Field Pointer** field <Bytes 16 and 17> specifies the byte position that was incorrect when an ILLEGAL REQUEST Sense Key is returned. This field is only valid when the Field Pointer Valid bit is set to 1. This pointer will point to either the Command Descriptor Block or the Command Parameters passed during the Data Out Phase depending on the value in the Command/Data field.

The **Sense Key** <Byte 2, bits 0:3>, **Error Code** <Byte 12, bits 0:3> and **Error Class** <Byte 12, bits 4:7> fields describe the type of error which occurred. The possible values in these fields are described below.

| Sense Key | Error Class/Code |
|---|---|
| Recoverable Error (01h) | ID Field CRC Error (10h) |
| | Seek Error (15h) |
| | Recoverable Read Error With Retries (17h) |
| Not Ready (02h) | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | Illegal Function For Device Type (22h) |
| Medium Error (03h) | ID field CRC Error (10h) |
| | Uncorrectable Data Error (11h) |
| | No ID Field Address Mark (12h) |
| | No Data Field Address Mark (13h) |
| | Block Not Found (14h) |
| | Seek Error (15h) |
| Hardware Error (04h) | No Seek Complete (02h) |
| | Write Fault (03h) |
| | Drive Not Ready (04h) |
| | Drive Not Selected (05h) |
| | No Track Zero (06h) |
| | ID Field Address Mark Not Found (12h) |
| | SCSI Msg Reject Error (43h) |
| | SCSI Hardware Error (44h) |
| | Select/Reselect Failed (45h) |
| Illegal Request (05h) | Invalid Block Address (21h) |
| | Invalid Field in CDB (24h) |
| | Invalid LUN (25h) |
| Unit Attention (06h) | Power Up or Reset (29h) |
| | Mode Select Changed Condition (2Ah) |
| Data Protect (07h) | Write Protected (27h) |
| Aborted Command (0Bh) | Parity Error (47h) |
| | Initiator Detected Error (48h) |
| | Illegal Message (49h) |

## A.1 SCSI ERROR CODES

Table A-1 lists and describes the nonextended error codes supported by the controller.

Table A-1. Nonextended Sense Error Codes

| Hex Code | Error | Description |
|---|---|---|
| 00 | NO SENSE | The controller detected no error during execution of the previous command. |
| 01 | RESERVED | This code is reserved. |
| 02 | NO SEEK COMPLETE | The controller could not seek to the specified logical block within an allotted time. |
| 03 | WRITE FAULT | The controller determined that the Attention line from the disk drive was asserted during a write operation. |
| 04 | DRIVE NOT READY | The disk drive is not ready. |
| 05 | DRIVE NOT SELECTED | The drive associated with the specified LUN could not be addressed. |
| 06 | NO TRACK ZERO | The controller could not rezero the disk drive. |
| 07-0F | RESERVED | These codes are reserved. |
| 10 | ID FIELD CRC ERROR | The sector ID field could not be read after the number of retry attempts specified in the **MODE SELECT** command. |

(continued on next page)

Table A-1. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 11 | UNCORRECTABLE DATA ERROR | A block could not be written or read after the number of retry attempts specified in the **MODE SELECT** command. |
| 12 | ID FIELD ADDRESS MARK NOT FOUND | The controller could not locate the address mark for a sector header. |
| 13 | DATA ADDRESS MARK NOT FOUND | The controller could not locate the address mark for the sector data area. |
| 14 | BLOCK NOT FOUND | The block sequence is improper, or a block is missing. |
| 15 | SEEK ERROR | A miscompare occurred between the cylinder address of the data header and the address specified in the CDB of the command. |
| 16 | RESERVED | This code is reserved. |
| 17 | RECOVERED READ ERROR (WITH RETRIES) | The controller encountered an error. It recovered the data using retries. |
| 18 | RECOVERED READ ERROR (WITH ECC) | The controller encountered an error. It recovered the data using ECC correction. |
| 19 | DEFECT LIST ERROR | The controller encountered an error while accessing one of the Defect Lists. |

(continued on next page)

Table A-1. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 1A | PARAMETER OVERRUN | The Parameter List Length specified in the CDB by the Initiator is too large for the controller. |
| 1B | RESERVED | This code is reserved. |
| 1C | PRIMARY DEFECT LIST NOT FOUND | The controller could not locate the manufacturer's defect list. |
| 1D | COMPARE ERROR | One or more bytes did not compare when the **VERIFY** or **READ BUFFER** command was issued. |
| 1E-1F | RESERVED | These codes are reserved. |
| 20 | INVALID COMMAND | The Initiator issued a command that cannot be executed, or is not applicable. |
| 21 | INVALID BLOCK ADDRESS | The addressed block was not valid. |
| 22 | ILLEGAL FUNCTION FOR DEVICE TYPE | The addressed LUN is unable to perform the requested function. |
| 23 | RESERVED | This code is reserved. |
| 24 | ILLEGAL FIELD IN CDB | A field in the Command Descriptor Block is reserved and contains a value other than zero or, the value in the field is incorrect. |
| 25 | INVALID LUN | The LUN specified in the Command Descriptor Block or SCSI IDENTIFY message is not supported by the controller. |

(continued on next page)

Table A-1. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 26 | ILLEGAL FIELD IN PARAMETER LIST | A field in the Parameter List is reserved and contains a value other than zero or the value in the field is incorrect. |
| 27 | WRITE PROTECTED | The disk is write protected. The outstanding WRITE command is aborted. |
| 28 | MEDIUM CHANGE | When the controller detects a media changed condition, it will set the Sense Key/Error Code to UNIT ATTENTION/MEDIA CHANGED (06h/28h). A media changed condition is defined as when the controller detects a drive offline condition then detects the drive come online. |
| 29 | POWER UP OR RESET | The controller has been reset by a SCSI Bus Reset, Bus Device Reset Message, or Power On Reset condition. After the controller detects the condition, it reports this error when the first command (except the **INQUIRY** command) is issued to it. The controller clears this condition when the next command is issued to it by the same Initiator. The controller reports the UNIT ATTENTION Sense Key to all SCSI devices that subsequently issue a command to it. |

(continued on next page)

Table A-1. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 2A | **MODE SELECT** CHANGED CONDITION | The **MODE SELECT** parameters for this device have been changed by another Initiator and may affect current operations. After the controller detects the condition, it reports the error when the first command (except the **INQUIRY** command) is issued to it. The controller clears this condition when the next command is issued to it by the same Initiator. The controller reports the UNIT ATTENTION Sense Key to all SCSI devices that subsequently issue a command to it. |
| 2B-2F | RESERVED | These codes are reserved. |
| 30 | RESERVED | This code is reserved. |
| 31 | FORMAT FAILED | The **FORMAT UNIT** command encountered an error while attempting to access on the defect lists. |
| 32 | NO DEFECT SPARE LOCATION AVAILABLE ERROR | There are no remaining alternate tracks on the addressed LUN. This error condition may occur during the processing of a **FORMAT UNIT** or **RE-ASSIGN BLOCK** command. |
| 33-3F | RESERVED | These codes are reserved. |

(continued on next page)

Table A-1. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 40 | RAM FAILURE | The controller detected a RAM memory error during a **SEND DIAGNOSTIC** test operation. |
| 41-42 | RESERVED | These codes are reserved. |
| 43 | MESSAGE REJECT ERROR | The Initiator responded with a MESSAGE REJECT message to a message sent by the controller. |
| 44 | SCSI HARDWARE/ FIRMWARE ERROR | The SCSI Firmware detected an internal firmware or hardware error and was unable to complete the current command. |
| 45 | SELECT/RESELECT FAILED ERROR | The SCSI firmware detected a timeout error while attempting a Selection or Reselection. |
| 46 | RESERVED | This code is reserved. |
| 47 | PARITY ERROR | A parity error occurred on the SCSI Bus and the controller was unable to recover the data. |
| 48 | INITIATOR DETECTED ERROR | The Initiator sent an INITIATOR DETECTED ERROR message and the controller was unable to recover from the error. |
| 49 | INAPPROPRIATE/ ILLEGAL MESSAGE ERROR | The Initiator sent an Inappropriate or Illegal SCSI Message to the controller. |

(continued on next page)

Table A-1. Nonextended Sense Error Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 4A-4F | RESERVED | These codes are reserved. |
| 50-5F | RESERVED | These codes are reserved. |
| 60-6F | RESERVED | These codes are reserved. |
| 70-7F | RESERVED | These codes are reserved. |
| 80-8F | RESERVED (Vendor Unique) | Reserved for future usage by EMULEX. |

BLANK

## B.1 SCSI SENSE KEYS

Table B-1 lists and describes the extended sense keys supported by the controller.

Table B-1. Extended Sense Key Codes

| Hex Code | Error | Description |
|---|---|---|
| 00 | NO SENSE | There is no Sense Key information to be reported for the designated LUN. This code occurs for a successfully completed command. |
| 01 | RECOVERED ERROR | The last command was completed successfully, but with some recovery action performed by the Target. |
| 02 | NOT READY | The addressed LUN cannot be accessed. Operator intervention may be required. |
| 03 | MEDIUM ERROR | The command terminated with a nonrecoverable-error condition which was probably caused by a flaw in the media or by an error in the recorded data. |
| 04 | HARDWARE ERROR | A nonrecoverable hardware error (e.g., controller failure, device failure, parity error, etc.) was detected while the Target was performing the command or while the Target was performing a Self-Test operation. |

(continued on next page)

Table B-1. Extended Sense Key Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 05 | ILLEGAL REQUEST | There was an illegal parameter in the command or in the additional required parameters supplied as data for some related commands. If the error is detected in the Command Descriptor Block, the controller will not alter the Media. |
| 06 | UNIT ATTENTION | The addressed LUN has been reset. This error is reported the first time any command is issued after the condition is detected; then the requested command is not performed. This condition is cleared when the next command is issued by the same Initiator. UNIT ATTENTION is reported to all SCSI devices that subsequently issue a command to the LUN. |
| 07 | DATA PROTECT | A write operation was attempted on a write-protected device. |
| 08 | RESERVED | This key is reserved. |
| 09 | VENDOR UNIQUE | A Vendor-Unique error condition occurred. The corresponding nonextended error class and code are specified in Byte 12 of the Extended Sense Byte. |
| 0A | COPY/COMPARE ABORTED | A COPY or COMPARE command was aborted because an error condition was detected on the source and/or destination device. |

(continued on next page)

Table B-1. Extended Sense Key Codes (continued)

| Hex Code | Error | Description |
|---|---|---|
| 0B | ABORTED COMMAND | The Target aborted the command. The Initiator may recover by trying to execute the command again. |
| 0C | RESERVED | This key is reserved. |
| 0D | RESERVED | This key is reserved. |
| 0E | MISCOMPARE | Used by the **VERIFY** command to indicate the source data did not match the data read from the disk. |
| 0F | RESERVED | This key is reserved. |

BLANK

## C.1 OVERVIEW

The installation of the SCSI controller, when used with the recommended devices, should run smoothly and problem-free. The diagnostic procedures described in this manual are intended to help you identify and resolve any problems you may encounter. However, because of the wide variety of host adapters, disk drives and other devices to which the controller could be connected, diagnostic procedures cannot be specific or all-inclusive. The following subsections explain how to obtain technical assistance or service for problems you cannot resolve.

## C.2 PROBLEM IDENTIFICATION

The self-test, described in Section Four, diagnoses problems within the controller itself. It does not diagnose problems with the host adapter, disk drive, or other devices to which the controller is connected.

If the controller does not pass the self-test when it is connected to the other devices in your system, remove the devices and try the self-test again. If the controller passes the self-test, the problem may be elsewhere in the system. See directions for obtaining help from Emulex's technical support personnel in the subsection on Technical Assistance.

If the controller does not pass the self-test when it is tested apart from the devices, it may be defective and should be returned to the factory for replacement. See directions for returning the unit in the subsection on Service.

## C.3 TECHNICAL ASSISTANCE

If the SCSI controller passed the self-test, but you believe it is not performing as expected, you can obtain assistance from Emulex's technical support personnel. The SCSI Product Performance Report (see last two pages of Appendix C) allows you to gather all the required information. Complete the form and mail it to Emulex at the address on the form. A technical support representative will contact you within five days of receipt of the form.

Note that this form is required in order for you to receive technical assistance regarding your SCSI controller. Please do not attempt to contact Technical Support by any other means. (If you have not heard from Emulex after five days, you may call in to check the status of your report.)

It is suggested that you use a photocopy of this form, so that the form will be available should you need to use it again.

## C.4 SERVICE

The components of your Emulex SCSI Controller have been designed to give years of trouble-free service, and they were thoroughly tested before leaving the factory.

If one of the diagnostic procedures described in this manual indicates that a component is not working properly, the controller must be returned to the factory, or to an Emulex authorized repair center, for service. Emulex products are not designed to be repaired in the field.

Before returning the component to Emulex, whether the product is or is not under warranty, you must contact Emulex's Repair Center for instructions and a Return Materials Authorization (RMA) number.

**DO NOT RETURN A SCSI CONTROLLER TO EMULEX WITHOUT AUTHORIZATION.** A controller returned for service without an authorization will be returned to the owner at the owner's expense.

In the continental United States, Alaska, and Hawaii contact:

Emulex Repair Center
3545 Harbor Boulevard
Costa Mesa, CA 92626
(714) 662-5600 TWX 910-595-2521

Outside the United States, contact the distributor from whom the SCSI Controller was initially purchased.

## PROBLEM DESCRIPTION

Give a complete description of the problem you are encountering. Provide details of the command packet and/or test loop/routine that you are using to troubleshoot the problem. Include a copy of the program(s) that demonstrates the problem on either hard copy or magnetic media (DOS or FILES 11 format).

*Please mail the completed form to:*


EMULEX CORPORATION
National Technical Support
3545 Harbor Boulevard
Costa Mesa, CA 92626

Attn: SCSI Support

# SCSI PRODUCT PERFORMANCE REPORT

*Please provide all information requested.*

## CUSTOMER INFORMATION

Name _____ Phone No. _____ Ext. _____
Position/Title _____
Company Name _____
Address _____
City _____ State _____ Zip _____

## PRODUCT INFORMATION

EMULEX Product in Use: MD _____ MT _____ IB _____ UC _____
Top Assembly Number _____
Serial Number _____ Date of Purchase _____
Purchased From _____ Sales Engineer _____

## PERIPHERAL INFORMATION

### DISK

Manufacturer _____
Model Number _____
Interface: ST506 _____ ESDI _____
Geometry: No. of Cylinders _____
Number of Heads _____ Sectors _____
Number of Bytes/Track _____
Number of Drives _____

### TAPE

Drive Interface: QIC36 _____ QIC44 _____
Media Interface: QIC11 _____ QIC24 _____
QIC120 _____ QIC150 _____

Tape Type: 300 _____ 450 _____ 600 _____
Tape Manufacturer: 3M _____ DEI _____
DYSAN _____ CARTREX _____ SCOTCH _____

## SYSTEM CONFIGURATION

Bus Type: IBM _____ DEC _____ VME _____
MULTIBUS _____ OTHER _____
Software: Diagnostics _____
Utilities _____

Amount of Memory _____
Host Adapter _____
Formatter _____
Operating System _____

## GRAPHIC REPRESENTATION OF YOUR SYSTEM

EMULEX

# Reader's Comments

**Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publication.**

**Manual Part Number** _____ **Rev.** _____

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults or errors have you found in the manual? _____

_____

_____

_____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

_____

_____

_____

☐ Please send me the current copy of the *Controller Handbook*, which contains the information on the remainder of EMULEX's controller products.

Name _____

Title _____

Company _____

Department _____

Street _____

City _____

State/Country _____

Zip _____

Additional copies of this document are available from:

Emulex Corporation
3545 Harbor Boulevard
P.O. Box 6725
Costa Mesa, CA 92626
Attention: Customer Services

Fold Here and Staple

Do Not Tear — Fold Here

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 202 COSTA MESA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

EMULEX CORPORATION
3545 HARBOR BOULEVARD
P.O. BOX 6725
COSTA MESA, CALIFORNIA 92626
ATTN: TECHNICAL PUBLICATIONS