



**INTERFACE SPECIFICATION  
0664  
2.0 GB - SCSI - 3.5 INCH DRIVE  
RELEASE 6.01**

July 12, 1994

---

# Preface

---

## 0664 S10 FUNCTIONAL FEATURES

---

<b>"On the Fly" error correction capabilities</b>	Refer to the product Functional Specification for reliability features
<b>LRC protection for data integrity</b>	Refer to the product Functional Specification for reliability features
<b>Self-optimizing buffer ratios</b>	Refer to 1.6.4, "Page 2 - Disconnect/Reconnect Parameters" on page 74
<b>Variable Block Lengths</b>	Refer to the product Functional Specification
<b>Sector Slipping Defect Management</b>	Refer to 1.16, "Reassign Blocks" on page 114
<b>Optional ANSI SCSI commands supported</b>	Refer to 1.0, "SCSI Commands" on page 19
<b>Command Queuing (both Tagged and Untagged)</b>	Refer to 4.3, "Command Queuing" on page 204
<b>Command Reordering</b>	Refer to 4.3, "Command Queuing" on page 204
<b>Back to Back Writes</b>	Refer to 4.4.1, "Back to Back Writes" on page 208
<b>Write Caching (Volatile)</b>	Refer to 1.6.8, "Page 8h - Read Caching Parameters" on page 85 regarding the WCE bit
<b>Motor (Spindle) Synchronization</b>	Refer to 4.5, "Motor Synchronization" on page 208
<b>Automatic Rewrite/Reallocate</b>	Refer to 4.7, "Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign" on page 211
<b>Automatic Recommend Rewrite/Reallocate</b>	Refer to 4.7, "Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign" on page 211
<b>Predictive Failure Analysis</b>	Refer to 4.8, "Predictive Failure Analysis" on page 213
<b>Segmented Caching</b>	Refer to 4.9, "Segmented Caching" on page 214
<b>Adaptive Caching</b>	Refer to 1.6.2, "Page 0 - Vendor Unique Parameters" on page 59
<b>Read Ahead support</b>	Refer to 4.9.3, "Read-Ahead" on page 214
<b>Down-loadable SCSI Firmware</b>	Refer to 1.34, "Write Buffer" on page 142

---

---

<b>LED support</b>	Refer to 4.16, “LED Pin” on page 223
<b>Basic Assurance Tests</b>	Refer to 4.19, “Basic Assurance Tests (BATS)” on page 226
<b>Full 32 byte SCSI Sense Data Format</b>	Refer to Appendix A, “SCSI Sense Data Format” on page 229
<b>Synchronous/wide data transfer support</b>	Refer to 3.1, “Supported Messages” on page 149 and 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59
<b>Performance counter information via Log Sense</b>	Refer to 1.5, “Log Sense” on page 44
<b>Selectable reporting of Errors (QPE etc.)</b>	Refer to 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59
<b>Multiple Initiator Support</b>	Refer to 4.14, “Multiple Initiator Systems” on page 222

---

---

## Data Representation

This document uses ASCII notation. Hex values are indicated by a trailing 'h'. Binary values are indicated by a trailing 'b'. The following notations are equivalent:

- 07h or 07H
- '07'x or '07'X
- 0x07 (except in the case where "x" is defined as "don't care")

In ASCII, the high order bit in a byte is seven, the low order bit is zero.

# Contents

<b>1.0 SCSI Commands</b>	19
1.1.1 CDB Control Byte	20
1.1.2 Reserved field definitions	20
1.1.3 Reserved value definitions	20
1.1.4 Invalid field Errors	20
1.2 Format Unit	21
1.3 Inquiry	28
1.3.1 Inquiry Data Format - EVPD = 0	30
1.3.2 Inquiry Data Format - EVPD = 1 - Page Code = 00	33
1.3.3 Inquiry Data Format - EVPD = 1 - Page Code = 01	34
1.3.4 Inquiry Data Format - EVPD = 1 - Page Code = 02	36
1.3.5 Inquiry Data Format - EVPD = 1 - Page Code = 03	38
1.3.6 Inquiry Data Format - EVPD = 1 - Page Code = 80	39
1.3.7 Inquiry Data Format - EVPD = 1 - Page Code = 82	40
1.4 Log Select	42
1.5 Log Sense	44
1.5.1 Log Page Parameters	46
1.5.2 Log Sense Page 00h	46
1.5.3 Log Sense Page 30h	47
1.5.4 Log Sense Page 32h	50
1.5.5 Log Sense Page 33h	52
1.5.6 Log Sense Page 35h	54
1.6 Mode Select (6)	55
1.6.2 Page 0 - Vendor Unique Parameters	59
1.6.3 Page 1 - Error Recovery Parameters	66
1.6.4 Page 2 - Disconnect/Reconnect Parameters	74
1.6.5 Page 3 - Format Device Parameters	77
1.6.6 Page 4h - Rigid Disk Drive Geometry Parameters	80
1.6.7 Page 7h - Error Recovery Parameters for Verify	83
1.6.8 Page 8h - Read Caching Parameters	85
1.6.9 Page 0Ah - Control Mode Parameters	88
1.7 Mode Sense (6)	90
1.7.1 Mode Sense Data Format (of Header & Block Descriptor)	92
1.7.2 Mode Sense Data Format (of All Pages)	94
1.8 Pre-Fetch	100
1.9 Read (6)	102
1.10 Read (10)	103
1.11 Read Buffer	104
1.12 Read Capacity	106
1.13 Read Defect Data (10)	107
1.14 Read Defect Data (12)	110
1.15 Read Long	113
1.16 Reassign Blocks	114
1.17 Receive Diagnostic	116
1.17.1 Receive Diagnostics Page 40h	117
1.17.2 Translate Address Page - Receive Diagnostics	117
1.18 Release (6)	119
1.19 Release (10)	120
1.20 Request Sense	121
1.21 Reserve (6)	123
1.22 Reserve (10)	124

1.23	Rezero Unit	126
1.24	Seek (6)	127
1.25	Seek (10)	128
1.26	Send Diagnostic	129
1.26.1	Send Diagnostics Page 40h	131
1.26.2	Translate Address Page - Send Diagnostics	131
1.27	Start/Stop Unit	132
1.28	Synchronize Cache	134
1.29	Test Unit Ready	135
1.30	Verify	136
1.31	Write (6)	138
1.32	Write (10)	139
1.33	Write and Verify	140
1.34	Write Buffer	142
1.35	Write Long	145
1.36	Write Same	146
<b>2.0</b>	<b>SCSI Status Byte</b>	<b>147</b>
<b>3.0</b>	<b>SCSI Message System</b>	<b>149</b>
3.1	Supported Messages	149
3.1.1	Command Complete (00h)	150
3.1.2	Save Data Pointer (02h)	150
3.1.3	Restore Pointers (03h)	150
3.1.4	Disconnect (04h)	151
3.1.5	Initiator Detected Error (05h)	151
3.1.6	Abort (06h)	151
3.1.7	Message Reject (07h)	152
3.1.8	No Operation (08h)	152
3.1.9	Message Parity Error (09h)	152
3.1.10	Linked Command Complete (0Ah)	152
3.1.11	Linked Command Complete with Flag (0Bh)	153
3.1.12	Bus Device Reset (0Ch)	153
3.1.13	Abort Tag (0Dh)	153
3.1.14	Clear Queue (0Eh)	153
3.1.15	Continue I/O Process (12h)	154
3.1.16	Target Transfer Disable (13h)	154
3.1.17	Queue Tag Messages (20h, 21h, 22h)	155
3.1.18	Ignore Wide Residue (23h)	156
3.1.19	Identify (80h or C0h)	156
3.1.20	Extended Messages (01h,xxh)	157
3.1.21	Synchronous Data Transfer Request (01h,01h)	157
3.1.22	Wide Data Transfer Request (01h,03h)	161
3.2	Supported Message Functions	163
3.3	Attention Condition	164
3.4	SCSI Bus Parity Error Detection	167
3.5	Message State/Action Table	168
3.6	SCSI Bus Related Error Handling Protocol	174
3.6.1	Unexpected Bus Free Phase Error Condition	174
3.6.2	Message Out Phase Retry	175
3.6.3	Message In Phase Retry	176
3.6.4	Command Phase Retry	176
3.6.5	Status Phase Retry	177
3.6.6	Data Out Phase Retry	177
3.6.7	Data In Phase Retry	177

3.6.8 Identify Message In Phase Retry . . . . .	178
3.6.9 Simple Queue Tag Message In Phase Retry . . . . .	178
<b>4.0 Additional Information . . . . .</b>	<b>179</b>
4.1 SCSI Protocol . . . . .	179
4.1.1 Priority of SCSI Status Byte Reporting . . . . .	179
4.1.2 Invalid LUN Processing . . . . .	180
4.1.3 Incorrect Initiator Connection . . . . .	181
4.1.4 Command Processing During Execution of Active I/O Process . . . . .	182
4.1.5 Unit Attention Condition . . . . .	184
4.1.6 Command Processing During Start-up and Format Operations . . . . .	187
4.1.7 Internal Error Condition . . . . .	188
4.1.8 Deferred Error Condition . . . . .	189
4.1.9 Degraded Mode . . . . .	190
4.1.10 Command Processing While Reserved . . . . .	202
4.2 Priority Commands . . . . .	203
4.3 Command Queuing . . . . .	204
4.3.1 Untagged Queuing . . . . .	204
4.3.2 Tagged Queuing . . . . .	205
4.3.3 Queue Full . . . . .	205
4.3.4 Rules for Queuing commands . . . . .	206
4.3.5 Effects of errors on Command Queuing . . . . .	206
4.3.6 Termination of I/O processes . . . . .	207
4.4 Concurrent Command Processing . . . . .	207
4.4.1 Back to Back Writes . . . . .	208
4.5 Motor Synchronization . . . . .	208
4.6 Error Logs . . . . .	211
4.7 Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign . . . . .	211
4.8 Predictive Failure Analysis . . . . .	213
4.9 Segmented Caching . . . . .	214
4.9.1 Overview . . . . .	214
4.9.2 Options Supported . . . . .	214
4.9.3 Read-Ahead . . . . .	214
4.9.4 Segment Selection . . . . .	218
4.9.5 Segment Prioritization . . . . .	219
4.10 Reselection Timeout . . . . .	221
4.11 Single Initiator Selection . . . . .	221
4.12 Non-arbitrating Systems . . . . .	221
4.13 Selection without ATN . . . . .	221
4.14 Multiple Initiator Systems . . . . .	222
4.14.1 Initiator Sense Data . . . . .	222
4.14.2 Initiator Mode Select/Mode Sense Parameters . . . . .	222
4.14.3 Initiator Data Transfer Mode Parameters . . . . .	222
4.15 Option Pin . . . . .	223
4.16 LED Pin . . . . .	223
4.17 Reset . . . . .	224
4.17.1 Reset Sources . . . . .	224
4.17.2 Reset Actions . . . . .	224
4.18 Bring-Up Sequence . . . . .	225
4.19 Basic Assurance Tests (BATS) . . . . .	226
4.19.1 BATS1 Tests . . . . .	226
4.19.2 BATS2 Tests . . . . .	226
4.20 Reserved Area . . . . .	227
4.21 Microcode Unique Function . . . . .	227

<b>Appendix A. SCSI Sense Data Format</b> .....	229
A.1.1 Sense Data Description .....	230
<b>Appendix B. Recovery Procedures</b> .....	253
B.1 Data Recovery Procedure for Data Field Errors .....	253
B.1.1 Read Commands .....	254
B.1.2 Verify Commands .....	260
B.1.3 Write Commands (No Sector Found) .....	262
B.2 Error Recovery for Non-Data Errors .....	263
B.2.1 Position Error (Servo Error Recovery) .....	263
B.2.2 Drive Fault .....	263
B.2.3 Spindle Motor Error .....	263
B.2.4 Spindle Synchronization Error .....	263
B.3 Priority of Error Reporting .....	263
<b>Appendix C. Recommended Initiator Error Recovery Procedures</b> .....	265
C.1 Drive service strategy .....	265
C.2 Recommendations for System Error Log .....	266
C.3 Data Recovery Procedure .....	266
C.3.1 Rewrite an LBA .....	267
C.3.2 Reassign a physical sector .....	268
C.3.3 Data error logging .....	269
C.3.4 Reassign Blocks Recovery .....	270
C.4 Non-Data Error Recovery Procedure .....	270
C.4.1 Drive busy .....	270
C.4.2 Unrecovered drive error .....	271
C.4.3 Recovered drive error .....	271
C.4.4 Drive not ready .....	271
C.4.5 No defect spare .....	271
C.4.6 Degraded Mode .....	272
C.4.7 Reserved Area Hard Error .....	273
C.4.8 Interface Protocol .....	273
C.4.9 Aborted Command .....	274
C.4.10 Unit Attention .....	275
C.4.11 Components Mismatch .....	276
C.4.12 Self Initiated Reset .....	278
C.4.13 Defect List Recovery .....	278
C.4.14 Miscompare recovery .....	278
C.4.15 Microcode error .....	279
C.4.16 Predictive failure analysis .....	280
<b>Appendix D. UEC List</b> .....	281
<b>Glossary</b> .....	289
<b>Index</b> .....	291



# **Figures**

1. Attention Condition during Asynchronous Information Transfer phase	165
2. Attention Condition during Synchronous Data Transfer phase . . . . .	166



# Tables

1.	SCSI Commands Supported	19
2.	Format Unit Command Descriptor Block	21
3.	Format of Defect List - Header	23
4.	Format of Defect List - Descriptor(s)	23
5.	Effect of Options on Format operations	25
6.	Defect Descriptor - Bytes From Index format	26
7.	Defect Descriptor - Physical Sector Format	26
8.	Inquiry Command Descriptor Block	28
9.	Standard Inquiry Data - EVPD = 0	30
10.	Inquiry data - EVPD = 1 - Page Code = 00	33
11.	Inquiry data - EVPD = 1 - Page Code = 01	34
12.	Inquiry data - EVPD = 1 - Page Code = 02	36
13.	Inquiry data - EVPD = 1 - Page Code = 03	38
14.	Inquiry data - EVPD = 1 - Page Code = 80	39
15.	Inquiry data - EVPD = 1 - Page Code = 82	40
16.	Log Select Command Descriptor Block	42
17.	Log Sense Command Descriptor Block	44
18.	Log Sense Page 00h	46
19.	Log Sense Page 30h	47
20.	Log Sense Page 32h	50
21.	Log Parameter Format - Page 32h	51
22.	Log Sense Page 33h	52
23.	Log Parameter Format - Page 33h	52
24.	Log Sense Page 35h	54
25.	Mode Select Command Descriptor Block	55
26.	Mode Select Parameter List - Header	57
27.	Mode Select Parameter List - Block Descriptor	58
28.	Mode Select Parameter List - Page Descriptor(s)	58
29.	Mode Select Data Format Page 0	59
30.	TCHD and TCC bit combinations.	63
31.	Mode Select Data Format Page 1	66
32.	Mode Select Data Format Page 2	74
33.	Mode Select Data Format Page 3	77
34.	Mode Select Data Format Page 4h	80
35.	SCSI/ESDI Synchronized Spindle Modes	81
36.	Mode Select Data Format Page 7h	83
37.	Mode Select Data Format Page 8h	85
38.	Mode Select Data Format Page 0Ah	88
39.	Mode Sense Command Descriptor Block	90
40.	Mode Sense Data Format - Header	92
41.	Mode Sense Data Format - Block Descriptor	92
42.	Mode Sense Data Page 00h	94
43.	Mode Sense Data Page 01h	94
44.	Mode Sense Data Page 02h	95
45.	Mode Sense Data Page 03h	96
46.	Mode Sense Data Page 04h	97
47.	Mode Sense Data Page 07h	97
48.	Mode Sense Data Page 08h	98
49.	Mode Sense Data Page 0Ah	98
50.	Pre-Fetch Command Descriptor Block	100
51.	Read (6) Command Descriptor Block	102

52.	Read (10) Command Descriptor Block	103
53.	Read Buffer Command Descriptor Block	104
54.	Read Capacity Command Descriptor Block	106
55.	Format of Read Capacity Data Returned	106
56.	Read Defect Data Command Descriptor Block	107
57.	Read Defect Data - Header	108
58.	Read Defect Data - Defect Descriptor(s)	108
59.	Defect Descriptor - Bytes From Index Format	109
60.	Defect Descriptor - Physical Sector Format	109
61.	Read Defect Data Command Descriptor Block	110
62.	Read Defect Data - Header	111
63.	Read Defect Data - Defect Descriptor(s)	111
64.	Defect Descriptor - Bytes From Index Format	112
65.	Defect Descriptor - Physical Sector Format	112
66.	Read Long Command Descriptor Block	113
67.	Reassign Blocks Command Descriptor Block	114
68.	Format of Reassign Blocks Data - Header	115
69.	Format of Reassign Blocks Data - Descriptor	115
70.	Receive Diagnostic Command Descriptor Block	116
71.	Translate Address Page - Receive Diagnostics	117
72.	Release Command Descriptor Block	119
73.	Release Command Descriptor Block	120
74.	Request Sense Command Descriptor Block	121
75.	Reserve Command Descriptor Block	123
76.	Reserve Command Descriptor Block	124
77.	Rezero Unit Command Descriptor Block	126
78.	Seek (6) Command Descriptor Block	127
79.	Seek (10) Command Descriptor Block	128
80.	Send Diagnostic Command Descriptor Block	129
81.	Translate Address Page - Send Diagnostics	131
82.	Start/Stop Unit Command Descriptor Block	132
83.	Synchronize Cache Command Descriptor Block	134
84.	Test Unit Ready Command Descriptor Block	135
85.	Verify Command Descriptor Block	136
86.	Write (6) Command Descriptor Block	138
87.	Write (10) Command Descriptor Block	139
88.	Write and Verify Command Descriptor Block	140
89.	Write Buffer Command Descriptor Block	142
90.	Write Long Command Descriptor Block	145
91.	Write Same Command Descriptor Block	146
92.	SCSI Status Byte	147
93.	Status Code Bit Definitions	147
94.	Supported One-Byte Messages	149
95.	Supported Two-Byte Messages	149
96.	Supported Multiple-Byte Messages	150
97.	Queue Tag Messages	155
98.	Ignore Wide Residue Message Format	156
99.	Extended Message Format	157
100.	Synchronous Data Transfer Request Message Definition	158
101.	Initiator Request/Target Response	158
102.	Target Response to Initiator's Transfer Period	160
103.	Wide Data Transfer Request Message Definition	161
104.	Initiator Request/Target Response	162
105.	Target Response to Initiator's Transfer Period	163
106.	Action Table Example	169

107. Message Action Legend . . . . .	170
108. Message State/Action Table (Part 1) . . . . .	171
109. Message State/Action Table (Part 2) . . . . .	172
110. Message State/Action Table (Part 3) . . . . .	173
111. Auto/Recommend Rewrite/Reallocate PER/QPE and RPF AE combinations. . . . .	212
112. Auto/Recommend Rewrite/Reallocate Sense data combinations. . . . .	213
113. Format of Sense Data . . . . .	229
114. Valid Sense Key, Code, Qualifier Combinations used by the drive. . . . .	232
115. Actual Retry Count Bytes . . . . .	250
116. Progress Indication Bytes . . . . .	250
117. Field Pointer Bytes . . . . .	251
118. Read Recovery Steps . . . . .	254
119. Write and Verify Recovery Steps . . . . .	261
120. Verify Recovery Steps . . . . .	261
121. Write Recovery Steps . . . . .	262
122. Servo Error Recovery Steps . . . . .	263
123. Drive Fault Recovery Steps . . . . .	263
124. Recommend Rewrite errors . . . . .	267
125. Recommend Reassign errors . . . . .	268
126. Log Only errors . . . . .	269
127. Unit Error Codes with Descriptions. . . . .	281



# 1.0 SCSI Commands

Following is a summary of SCSI commands supported. The column “SCSI” refers to revision 10h of the ANSI version 2 standard.

Table 1. SCSI Commands Supported		
SCSI	Code	Description
M	04h	Format Unit
M	12h	Inquiry
O	4Ch	Log Select
O	4Dh	Log Sense
O	15h	Mode Select (6)*
O	1Ah	Mode Sense (6)*
O	34h	Pre-Fetch
M	08h	Read (6)*
M	28h	Read (10)*
O	3Ch	Read Buffer
M	25h	Read Capacity
O	37h	Read Defect Data (10)*
O	B7h	Read Defect Data (12)*
O	3Eh	Read Long
O	07h	Reassign Blocks
O	1Ch	Receive Diagnostic
M	17h	Release (6)
M	57h	Release (10)
M	03h	Request Sense
M	16h	Reserve
M	56h	Reserve (10)
O	01h	Rezero Unit
O	0Bh	Seek (6)*
O	2Bh	Seek (10)*
M	1Dh	Send Diagnostic
O	1Bh	Start/Stop Unit
O	35h	Synchronize Cache
M	00h	Test Unit Ready
O	2Fh	Verify
O	0Ah	Write (6)*
O	2Ah	Write (10)*
O	2Eh	Write and Verify
O	3Bh	Write Buffer
O	3Fh	Write Long
O	41h	Write Same

M - Mandatory  
O - Optional  
V - Vendor Unique  
\* Command Descriptor Block Length

### 1.1.1 CDB Control Byte

These bits are in the control byte, which is the last byte of every command descriptor block. The meaning of these fields is defined below.

VU	VU stands for Vendor Unique.
Flag	If Link is zero, Flag must also be zero. If Link is one and the command terminates successfully, the Target sends either the Linked Command Complete message (Flag=0) or the Linked Command Complete (With Flag) message (Flag=1). Typically this bit is used to cause an interrupt in the Initiator between linked commands.
Link	This bit is set to one to indicate that the Initiator desires an automatic link to the next command upon successful completion of the current command. Upon successful completion of the command, the Drive returns Intermediate status and then sends one of the two messages defined under Flag above.

### 1.1.2 Reserved field definitions

A Reserved field is defined as being required to be filled with binary 0. It may be shown in tables as any of the following:

- Reserved
- Reserved = 0
- RSVD
- RSVD = 0

### 1.1.3 Reserved value definitions

A value may be defined to be 'Reserved'. This means that the value is not valid for the field being described.

### 1.1.4 Invalid field Errors

If a value is invalid in one of the Command Descriptor Blocks, a *Check Condition Status* will result with a sense key of ***Illegal Request*** and additional sense code/qualifier dependent upon which field was invalid. The following are self explanatory:

- ***Invalid Command Operation Code***
- ***Logical Block Address out of Range***
- ***Logical Unit Not Supported***

***Invalid Field in CDB*** is used for fields other than the above.

***Invalid Field in Parameter List*** is used for invalid fields in parameter data.

***Invalid Bits in Identify Message*** is used when the Reserved bits in a message are non-zero.



## 1.2 Format Unit

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 04h							
1	LUN			FmtData	CmpLst	Defect List Format		
2	Vendor Unique = 0							
3	(MSB) Interleave Factor (LSB)							
4								
5	VU = 0		Reserved = 0			Flag	Link	

The Format Unit command performs a physical formatting of the Drive. This includes a rewrite of all the sector IDs, handling of defective sectors, and the overwriting of all data areas with binary zeros. The formatting process includes a verification of access positioning only and does not include a verification of ID or data fields.

The Drive manages two internal defect lists and one external. The primary defect list (PList) is created at time of manufacture and cannot be altered. The grown defect list (GList) is built after time of manufacture by the Initiators use of the Reassign Blocks command and the Automatic Reallocate function(see 4.7, “Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign” on page 211), The data defect list (DList) is an external list. It is supplied by the initiator in the DATA OUT phase of the Format Unit command.

Several options for managing the GList are supported.

FmtData set to one specifies that a Data Out phase follows the Command phase. The Data Out phase consists of a defect list header followed by zero or more defect descriptors (see Table 3 on page 23 and Table 4 on page 23).

FmtData set to zero specifies that no Data Out phase follows.

CmpLst set to one specifies that the GList existing prior to the format not be used and is discarded. The Drive is formatted with the PList and DList(if specified). The DList becomes the new GList.

CmpLst set to zero specifies that the GList existing prior to the format be used. The Drive is formatted with the PList, the GList, and the DList (if specified). The DList and the GList are combined and become the new GList.

The Defect List Format specifies the format of the defect descriptor transferred to the Target when the FmtData bit is set to one (see Table 5 on page 25).

Interleave Factor must be either zero or one; the target formats the file with interleave of one.

**Note:** It is recommended that the Mode Select command be issued prior to the Format Unit command to specify parameters that affect the formatting process.

The Block Length parameter (see 1.6.1.2, “Block Descriptor” on page 58) is used during formatting and is saved following a successful format operation. If a Mode Select command has not been issued since the last reset or start-up (bring-up) sequence, then the Block Length from the previous format operation is used.

The RPL parameter in Mode Select page 04h, affects the way that track and cylinder skew are physically applied. See 1.6.6, “Page 4h - Rigid Disk Drive Geometry Parameters” on page 80, 1.6.5, “Page 3 - Format Device Parameters” on page 77, and 4.5, “Motor Synchronization” on page 208 for more information. The current RPL parameter is used to control the desired track and cylinder skew definitions for the format operation, regardless of the state of motor synchronization.

Subsequent to receiving a Format Unit command the Target responds to commands as follows.

- All commands except Request Sense and Inquiry return *Check Condition Status* while the format operation is an active I/O process (see 4.1.6, “Command Processing During Start-up and Format Operations” on page 187).
- When tagged queuing is enabled (DQue = 0), all commands except Request Sense and Inquiry return *Queue Full Status* while the Format Unit command is a queued I/O process (see 4.3, “Command Queuing” on page 204).
- When tagged queuing is disabled (DQue = 1), all commands except Request Sense and Inquiry return *Busy Status* while the Format Unit command is a queued I/O process.
- If a Request Sense command is received while a format operation is an active I/O process, the Target returns *Good Status*. The sense key is set to Not Ready and the additional sense code and qualifier is set to Format In Progress.
- If an Inquiry command is received while a format operation is an active I/O process, the Target returns *Good Status* and Inquiry data as requested.

The format operation must complete successfully for the Drive to be usable. If the command is interrupted by a reset, power down, or an unrecoverable error, the Drive enters a degraded mode of operation in which reading and writing are prohibited (see 4.1.9, “Degraded Mode” on page 190). To exit the degraded mode, another Format Unit command must be sent by the Initiator and completed successfully by the Target.

The Format Unit command sets the Unit Attention condition for all Initiators except the one that issued the Format Unit command (see 4.1.5, “Unit Attention Condition” on page 184).

**Note:** While the Format Unit command effectively erases the entire data area of the Drive, this use is not recommended. To erase data areas on the Drive and save time, use the Write Same command.

Table 3. Format of Defect List - Header								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0							
1	FOV	DPRY	DCRT	STPF	IP = 0	DSP = 0	Immed	RSVD
2	(MSB) Defect list length (LSB)							
3								

Table 4. Format of Defect List - Descriptor(s)								
Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 7	Defect Descriptor 0							
8n - 8n+7	Defect Descriptor n							
<b>Note:</b> Format of the defect list sent during the Data Out phase.								

The Target has a limited implementation of the Format Option bits located in Bits 2 through 7 of Byte 1 of the Defect List Header (see Table 3). If the Initiator attempts to select any function not implemented by the Target, the Target terminates the command with *Check Condition Status*. The sense key is set to Illegal Request and the additional sense code is set to Invalid Field In Parameter List.

An FOV (Format Options Valid) bit of zero causes the Target to verify that the settings for the DPRY (Disable PRimary), DCRT (Disable CeRTification), STPF (SToP Format) IP (Initialization Pattern), and DSP (Disable Saving Parameters) bits are zero. If any of these bits are not zero, the Target terminates the command with *Check Condition Status*. The sense key is set to Illegal Request and the additional sense code is set to Invalid Field In Parameter List.

When the FOV bit is one there are two combinations of the DPRY, DCRT, STPF, IP and DSP bits allowed. Any other combinations return a *Check Condition Status*. With a sense key of Illegal Request and an additional sense code of Invalid Field In Parameter List. The supported combinations are:

DPRY=0 DCRT=1 STPF=1 IP=0 DSP=0

DPRY=1 DCRT=1 STPF=1 IP=0 DSP=0

The DPRY (disable primary) bit set to zero indicates that the Target does not use portions of the medium identified as defective in the primary defect PList for Initiator addressable logical blocks. If the Target cannot locate the PList or it cannot determine whether a PList exists, the Target terminates the Format Unit command as described for STPF=1. The DPRY bit set to one indicates that the target shall not use the Plist to identify defective areas of the medium. The Plist is not deleted.

The DCRT (disable certification) bit must be set to one. The Target does not generate a CList (certification list) or perform a certification process while executing the Format Unit Command.

The STPF (stop format) bit controls the behavior of the Target when one of the following events occurs:

- The Target cannot locate a required defect list nor determine that the list exists.
- The Target encounters an unrecoverable error while accessing a required defect list.

The STPF bit must be set to one. If one or both of the above conditions occurs, the Target terminates the Format Unit command with *Check Condition Status*. The sense key is set to Medium Error and the additional sense code is set to either Defect List Not Found if the first condition occurred, or Defect List Error if the second condition occurred.

The IP (initialization pattern) bit must be set to zero. The Target initializes all data with zeros.

The DSP (disable saving parameters) bit must be zero. The Target saves all the Mode Select savable parameters during the format operation.

An Immed bit set to zero requests that status be returned at the end of the format operation.

An Immed bit set to one requests that status be returned immediately. *Good Status* is returned following the CDB validation and transfer of data in the Data Out phase. If the immediate format operation terminates in error, Deferred Error Sense data is generated. See 4.1.8, “Deferred Error Condition” on page 189 for more information regarding the immediate format operation. With the Immed bit set to one, the Link bit must be set to zero.

The Defect List Length field specifies the total length in bytes of the defect descriptors that follow. The defect list length must be equal to eight times the number of defect descriptors to follow, otherwise the command is terminated with *Check Condition Status*. The sense key is set to Illegal Request and the additional sense code is set to Invalid Field In Parameter List. The defect descriptors must specify the defect based on the current Format Device parameters reported by the Mode Sense command.

The following table describes the influence of the various options on the format operation.

Table 5. Effect of Options on Format operations						
<u>FmtData</u>	<u>CmpList</u>	<u>Defect List Format</u>	<u>Defect List Length</u>	<u>Command Type</u>	<u>Defect Sources</u>	<u>New GList Composition</u>
				M = Mandatory O = Optional	P = PLlist G = GLlist D = DLlist	E = Erase the GLlist K = Keep the GLlist A = Add the DLlist to the GLlist R = Replace the GLlist with DLlist
<b><u>No Defect List</u></b>						
0	0	000b	N/A	M	P,G	K
<b><u>Block Format</u></b>						
1	0	000b	Zero	M	P,G	K
1	1	000b	Zero	M	P	E
<b><u>Bytes From Index Format</u></b>						
1	0	100b	Zero	O	P,G	K
1	1	100b	Zero	O	P	E
1	0	100b	> 0	O	P,G,D	A
1	1	100b	> 0	O	P,D	R
<b><u>Physical Sector Format</u></b>						
1	0	101b	Zero	O	P,G	K
1	1	101b	Zero	O	P	E
1	0	101b	> 0	O	P,G,D	A
1	1	101b	> 0	O	P,D	R

The Target supports two defect descriptor formats for the Format Unit command and the Read Defect Data commands, Bytes From Index format and Physical Sector format (see Table 5).

The Target does not require that DLlist entries be in ascending order.

If the DLlist entry does not correspond to a valid user addressable media location, the command terminates with *Check Condition Status*. The sense key is set to Illegal Request and the additional sense code is set to Invalid Field In Parameter List.

Table 6. Defect Descriptor - Bytes From Index format								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB)							
1	Cylinder Number of Defect							
2	(LSB)							
3	Head Number of Defect							
4	(MSB)							
5	Defect Bytes From Index							
6								
7	(LSB)							

Each defect descriptor for the Bytes From Index format specifies that the sector containing this byte be marked defective. The defect descriptor is comprised of the cylinder number of the defect, the head number of the defect, and the defect bytes from index. The internal format of the Target's GList is not identical to the DList, thus the DList entry is interpreted as a defect which starts at the beginning of the sector pointed to by the DList entry, with a length equal to the sector length. If the Initiator sends more than one defect descriptor for the same sector, the Target creates one GList entry for each defect descriptor.

**Note:** For the specified block size, if the Dlist entry when converted to a physical sector, is equal to the physical sector of a Plist entry (DPRY = 1), that Dlist entry is not added to the Dlist.

If the Target receives a defect bytes from index value of FFFFFFFFh, indicating that the entire track be marked defective, the command is terminated with *Check Condition Status*. The sense key is set to Illegal Request and the additional sense code is set to Invalid Field In Parameter List.

Table 7. Defect Descriptor - Physical Sector Format								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB)							
1	Cylinder Number of Defect							
2	(LSB)							
3	Head Number of Defect							
4	(MSB)							
5	Defective Sector Number							
6								
7	(LSB)							

Each defect descriptor for the Physical Sector format specifies a defect that is the length of a sector. The defect descriptor is comprised of the cylinder number of the defect, the head number of the defect, and the defect sector number. The actual defect stored in the Glist points to the start of the sector and has a length equal to the sector size. If the Initiator sends more than one defect descriptor for the same sector, the Target creates one GList entry for each defect descriptor.

**Note:** For the specified block size, if the Dlist entry when converted to a physical sector, is equal to the physical sector of a Plist entry (DPRY = 1), that Dlist entry is not added to the Dlist.

If the Target receives a defect sector value of FFFFFFFFh, indicating that the entire track be marked defective, the command is terminated with *Check Condition Status*. The sense key is set to Illegal Request and the additional sense code is set to Invalid Field In Parameter List.

## 1.3 Inquiry

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 12h							
1	LUN			Reserved = 0				EVPD
2	Page Code							
3	Reserved = 0							
4	Allocation Length							
5	VU = 0		Reserved = 0				Flag	Link

The Inquiry command requests that information regarding parameters of the Target and its peripheral device(s) be sent to the Initiator. Several options allow the Initiator to request detailed Vital Product Data (VPD) about the Drive.

EVPD (Enable Vital Product Data) bit set to zero specifies the Target return standard Inquiry data. When the EVPD bit is set to zero, The Page Code field must also be zero. If the Page Code is not zero, the target reports Check Condition status with the sense key set to Illegal Request and the additional sense code set to Invalid Field in CDB (24 00).

EVPD (Enable Vital Product Data) bit set to one requests that the target will return the vital product data specified by the Page Code field. Page Code specifies which page of VPD information the target will return. The supported VPD pages are defined in EVPD page 00h. If the value in the Page Code Field is not a supported page the target reports Check Condition status with the sense key set to Illegal Request and the additional sense code set to Invalid Field in CDB (24 00).

Allocation Length specifies the number of bytes the Initiator allocated for returned Inquiry Data. Allocation length of zero is not an error and means that no data is returned. The target terminates the Data In phase when all requested Inquiry Data has been returned or when the number of bytes returned equals the allocation length, whichever is less.

The Inquiry command returns Check Condition status only when the target cannot return the requested Inquiry data.

If an Inquiry command is received from an Initiator with a pending Unit Attention (that is, before the target reports Check Condition status), the target performs the Inquiry command and does not clear the Unit Attention condition.

**Note:** The Inquiry Command is not queued. For more information see 4.3, “Command Queuing” on page 204.

**Note:** The inquiry data is set at the time of manufacture and will not change (without a FRU change), with the following exceptions:

- *ASCII RAM Revision Level (EVPD=0)* can be changed when the microcode is downloaded with the WRITE BUFFER command.
- *ASCII RAM uCode Load P/N (EVPD=0)* can be changed when the microcode is downloaded with the WRITE BUFFER command.



- *Load ID (EVPD= 1, Page Code= 3)* can be changed when the microcode is downloaded with the WRITE BUFFER command.
- *Release level/modification number (EVPD= 1, Page Code= 3)* can be changed when the microcode is downloaded with the WRITE BUFFER command.

**Note:** The Inquiry data returned when media is not available will not be complete.

### 1.3.1 Inquiry Data Format - EVPD = 0

Table 9. Standard Inquiry Data - EVPD = 0								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Peripheral Qualifier = 0			Peripheral Device Type = 0				
1	RMB = 0	Device Type Modifier = 0						
2	ISO = 0		ECMA = 0			ANSI = 2		
3	RSVD = 0	TrmIOP = 0	Reserved = 0		Response Data Format = 2			
4	Additional Length = 159 (9Fh)							
5-6	Reserved = 0							
7	RelAdr = 0	WBus32 = 0	WBus16 = 0	SYNC = 1	Linked = 1	TranDis = 1	CmdQue = 1	SftRe = 0
8-15	ASCII Vendor ID 'IBM OEM '							
16-19	ASCII Product Type ' <b>0664</b> '							
20-22	ASCII Model Number							
23-31	Unused							
32-33	ASCII ROM code Revision Level							
34-35	ASCII RAM Load Revision Level							
36-43	ASCII Unit Serial Number							
44-55	ASCII RAM uCode Load P/N							
56-95	Reserved = 0							
96-97	Vendor Unique Reserved = 0							
98-101	ASCII Plant of Manufacture							
102-106	ASCII Date of Manufacture							
107	Unused							
108-111	ASCII FRU Count '0002'							
112-113	ASCII FRU Field Length '22'							
114-125	ASCII Assembly P/N							
126-135	ASCII Assembly EC Level							
136-147	ASCII Card Assembly P/N							
148-157	ASCII Card Assembly EC							
158-163	Reserved = 0							

If the inquiry data contained on the media is not available, bytes 0-19 and byte 32 are valid. All other fields (including fields marked Unused) contain ASCII spaces (20h) with the exception of the Unit Serial Number which contains ASCII zeros (30h) and all Reserved fields which contain zero (00h).

The Inquiry command is different from other commands in that it responds to a invalid LUN specification by returning a Qualifier of 3h and a Peripheral Device Type of 1Fh rather than a Check Condition. This indicates “Logical Unit Not Present” and is returned by this Drive for all logical unit numbers except 0.

The target returns a Qualifier of 0h and a Peripheral Device Type of 00h when the logical unit is 0. The Peripheral Qualifier field of zero (0) indicates that the peripheral device is currently connected to this logical unit. A Peripheral Device Type field of zero (0) indicates that this device is a Direct Access Storage Device (DASD).

The RMB (removable media bit) field of 0 indicates that no removal media exists.

The Device Type Modifier field is not used and is set to 0.

The ISO field of 0 indicates that this product does not claim compliance to the International Organization for Standards (ISO) version of SCSI (ISO DIS 9316).

The ECMA field of 0 indicates that this product does not claim compliance to the European Computer Manufacturers Association (ECMA) version of SCSI (ECMA-111).

The ANSI field of 2 indicates the target supports American National Standards Institute (ANSI) SCSI version 2.

The TrmIOP field of 0 indicates the Target does not support the Terminate I/O Process Message.

A Response Data Format value of 2 indicates that the data is in the format specified in ANSI X3.133-199X (SCSI-2).

Additional Length specifies the length in bytes of the parameters that follow the additional length field. If the Allocation Length of the command descriptor block is too small to transfer all of the parameters, the Additional Length parameter is not adjusted to reflect the truncation.

The RelAdr field of 0 indicates the Target does not support the Relative Addressing Mode.

The WBus32 field of 0 indicates the Target does not support 32-bit wide data transfers.

The WBus16 field of 0 indicates the Target does not support 16-bit wide data transfers. A value of 1 indicates the Target supports 16-bit wide data transfers.

Sync set to 1 indicates the Target supports synchronous data transfer.

Linked set to 1 indicates the Target supports linked commands for this logical unit.

A transfer disable (TranDis) bit of 1 indicates that the device supports the CONTINUE I/O PROCESS and TARGET TRANSFER DISABLE messages for this logical unit.

The CmdQue field of 1 indicates that the Target supports tagged command queuing.

The SftRe field of 0 indicates the Target does not respond to the Reset condition with the soft Reset alternative.

The Vendor ID is 'IBM OEM' or in hex (49424D204F454D20h). The field is left aligned and the unused bytes are filled with space characters (20h). This field can be customized to meet the requirements of a user.

The Product Type is a 4-character ASCII field. This field is '0664'.

The Model Number field contains an ASCII 3-character model number for the device. For model information See (Product Hardware Specification).

The ROM Code Revision Level field indicates the level of the ROM-based microcode. This is an ASCII 2-character field. The field is left aligned and the unused bytes are filled with space characters (20h).

The RAM Code Revision Level field indicates the level of the RAM-based microcode. This is an ASCII 2-character field. The field is left aligned and the unused bytes are filled with space characters (20h).

The Unit Serial Number field contains the Drive serial number. The field is numeric, right aligned and the unused bytes are ASCII zero (30h). The most significant byte, byte 36, will always be an ASCII zero.

The RAM uCode Load P/N field contains the IBM part number of the RAM microcode loaded on the media. The field is left aligned and the unused bytes are filled with space characters (20h).

The reserved bytes 56-97 contain 00h.

The Plant of Manufacture field indicates the location of the manufacturing facility that made the Drive. A Manufacturing Plant code of '0980' is for Rochester, Minnesota, '0902' is Havant, UK, and '0983' is for San Jose, California.

The Date of Manufacture field is a 5-byte field containing the Julian date (two-digit year followed by the three-digit day of the year).

The FRU Count field indicates the number of field replaceable units (FRU) identification fields in inquiry.

The FRU Field Length indicates the number of following bytes used to identify each FRU.

The Assembly P/N field is a 12-byte ASCII field that contains the part number for the entire Drive assembly. The field is left aligned and the unused bytes are filled with space characters (20h).

The Assembly EC Level field is a 10-byte ASCII field that contains the Engineering Change level of the Drive assembly. The field is left aligned and the unused bytes are filled with space characters (20h).

The Card Assembly P/N field is a 12-byte ASCII field that contains the part number of the electronics package for the Drive. The field is left aligned and the unused bytes are filled with space characters (20h).

The Card Assembly EC field is a 10-byte ASCII field that contains the Engineering Change level of the electronics package for the Drive. The field is left aligned and the unused bytes are filled with space characters (20h).

### 1.3.2 Inquiry Data Format **EVPD = 1 - Page Code = 00**

Table 10. Inquiry data - EVPD = 1 - Page Code = 00							
Byte	BIT						
	7	6	5	4	3	2	1
0	Peripheral Qualifier			Peripheral Device Type			
1	Page Code = 00h						
2	Reserved = 0						
3	Page Length = 5						
4	First Supported Page Code = 01h						
5	Second Supported Page Code = 02h						
6	Third Supported Page Code = 03h						
7	Fourth Supported Page Code = 80h						
8	Fifth Supported Page Code = 82h						

Bytes 0-7 are always available.

See 1.3.1, “Inquiry Data Format - EVPD = 0” on page 30 for a description of the fields in byte 0.

The Page Code field is set to the value of the Page Code field in the CDB.

The Page Length field specifies the length in bytes of the parameters that follow the Page Length field (byte 3). If the Allocation Length of the command descriptor block is too small to transfer all parameters, the Page Length field is not adjusted to reflect the truncation.

The Supported Page Code fields contain the Page Code supported by the target or logical unit. The list is in ascending order.

### 1.3.3 Inquiry Data Format - EVPD = 1 - Page Code = 01

Table 11. Inquiry data - EVPD = 1 - Page Code = 01							
Byte	BIT						
	7	6	5	4	3	2	1
0	Peripheral Qualifier			Peripheral Device Type			
1	Page Code = 01h						
2	Reserved = 0						
3	Page Length = 47						
4	ASCII Length = 24						
5 - 16	ASCII Assembly P/N						
17	0						
18 - 27	ASCII Assembly EC						
28	0						
29 - 40	EBCDIC Assembly P/N						
41 - 50	EBCDIC Assembly EC						

If the media is not available, bytes 0-4 are valid, the ASCII fields contain ASCII spaces (20h) and the EBCDIC fields contain EBCDIC spaces (40h), all other fields are zero (00h).

See 1.3.1, “Inquiry Data Format - EVPD = 0” on page 30 for a description of the fields in byte 0.

The Page Code field is set to the value of the Page Code field in the CDB and is associated with the field replaceable unit code returned by the Request Sense Command.

The Page Length field specifies the length (in bytes) of the vendor unique VPD information (bytes 4 to 50). If the allocation length of the command descriptor block is too small to transfer all the data, the Page Length field is not adjusted to reflect the truncation.

The ASCII Length field specifies the length of the ASCII VPD information (bytes 5 through 28). If the allocation length of the command descriptor block is too small to transfer all of the ASCII VPD data, the ASCII Length field is not adjusted to reflect the truncation.

The ASCII Assembly P/N field is a twelve (12) byte ASCII field that contains the part number for the Drive assembly. The field is left aligned and the unused bytes are filled with space characters (20h).

The ASCII Assembly EC field is a ten (10) byte ASCII field that contains the Engineering Change level of the Drive assembly. The field is left aligned and the unused bytes are filled with space characters (20h).

The EBCDIC Assembly P/N field is a twelve (12) byte EBCDIC field that contains the part number for the Drive assembly. The field is left aligned and the unused bytes are filled with space characters (40h).

The EBCDIC Assembly EC field is a ten (10) byte EBCDIC field that contains the Engineering Change level of the Drive assembly. The field is left aligned and the unused bytes are filled with space characters (40h).

### 1.3.4 Inquiry Data Format - EVPD = 1 - Page Code = 02

Table 12. Inquiry data - EVPD = 1 - Page Code = 02							
Byte	BIT						
	7	6	5	4	3	2	1
0	Peripheral Qualifier			Peripheral Device Type			
1	Page Code = 02h						
2	Reserved = 0						
3	Page Length = 47						
4	ASCII Length = 24						
5 - 16	ASCII Card P/N						
17	0						
18 - 27	ASCII Card EC						
28	0						
29 - 40	EBCDIC Card P/N						
41 - 50	EBCDIC Card EC						

If the media is not available, bytes 0-4 are valid, the ASCII fields contain ASCII spaces (20h) and the EBCDIC fields contain EBCDIC spaces (40h), all other fields are zero (00h).

See 1.3.1, “Inquiry Data Format - EVPD = 0” on page 30 for a description of the fields in byte 0.

The Page Code field is set to the value of the Page Code field in the CDB and is associated with the field replaceable unit code returned by the Request Sense Command.

The Page Length field specifies the length (in bytes) of the vendor unique VPD information (bytes 4 to 50). If the allocation length of the command descriptor block is too small to transfer all the data, the Page Length field is not adjusted to reflect the truncation.

The ASCII Length field specifies the length of the ASCII VPD information that follows (bytes 5 through 27). If the allocation length of the command descriptor block is too small to transfer all of the ASCII VPD data, the ASCII Length field is not adjusted to reflect the truncation.

The ASCII Card P/N field is a twelve (12) byte ASCII field that contains the part number for the Drive card assembly. The field is left aligned and the unused bytes are filled with space characters (20h).

The ASCII Card EC field is a 10-byte ASCII field that contains the Engineering Change level of the electronics package for the Drive. The field is left aligned and the unused bytes are filled with space characters (20h).

The EBCDIC Card P/N field is a twelve (12) byte EBCDIC field that contains the part number for the Drive card assembly.



The EBCDIC Card EC field is a 10-byte EBCDIC field that contains the Engineering Change level of the electronics package for the Drive. The field is left aligned and the unused bytes are filled with space characters (40h).

### 1.3.5 Inquiry Data Format - EVPD = 1 - Page Code = 03

Table 13. Inquiry data - EVPD = 1 - Page Code = 03								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	Page Code = 03h							
2	Reserved = 0							
3	Page Length = 20							
4	ASCII Length = 0							
5 - 7	Reserved = 0							
8 - 11	Load ID							
12 - 15	Release level/modification number							
16 - 19	PTF Number = 0							
20 - 23	Patch Number = 0							

If the media is not available, bytes 0-11 are valid. all other fields contain zeros (00h).

See 1.3.1, “Inquiry Data Format - EVPD = 0” on page 30 for a description of the fields in byte 0.

The Page Code field is set to the value of the Page Code field in the CDB and is associated with the field replaceable unit code returned by the Request Sense Command.

The Page Length field specifies the length (in bytes) of the vendor unique VPD information (bytes 4-20). If the allocation length of the command descriptor block is too small to transfer all the data, the Page Length field is not adjusted to reflect the truncation.

The Load ID is used as a mechanism to determine whether compatibility exists between ROM and RAM microcode levels.

The Release level/modification number is an indication of the generation level within the Load ID.

Both the PTF Number and the Patch Number are not used and should be reserved as zero.

Bytes 8-23 of this page are used to control the level of microcode that is installed on the Drive. See 1.34, “Write Buffer” on page 142.

### 1.3.6 Inquiry Data Format - EVPD = 1 - Page Code = 80

Table 14. Inquiry data - EVPD = 1 - Page Code = 80							
Byte	BIT						
	7	6	5	4	3	2	1
0	Periph Qualifier			Peripheral Device Type			
1	Page Code = 80h						
2	Reserved = 0						
3	Page Length = 16						
4-19	ASCII Unit Serial Number						

If the media is not available, bytes 0-3 are valid. All other fields contain ASCII spaces (20h).

See 1.3.1, “Inquiry Data Format - EVPD = 0” on page 30 for a description of the fields in byte 0.

The Page Code field is set to the value of the Page Code field in the CDB

The Page Length field specifies the length (in bytes) of the vendor unique VPD information (bytes 4-20). If the allocation length of the command descriptor block is too small to transfer all the data, the Page Length field is not adjusted to reflect the truncation.

The Unit Serial Number field contains the Drive serial number. The field is numeric (ASCII), right aligned and the unused bytes are ASCII spaces (20h).

### 1.3.7 Inquiry Data Format - EVPD = 1 - Page Code = 82

Table 15. Inquiry data - EVPD = 1 - Page Code = 82							
Byte	BIT						
	7	6	5	4	3	2	1
0	Peripheral Qualifier			Peripheral Device Type			
1	Page Code = 82h						
2	Reserved = 0						
3	Page Length = 48						
4	ASCII Length = 25						
5 - 8	ASCII Product Type '0664'						
9	0						
10 - 12	ASCII Model Number						
13	0						
14 - 21	ASCII Serial Number						
22	0						
23 - 28	ASCII Vendor ID 'IBM'						
29	0						
30 - 33	EBCDIC Product Type '0664'						
34 - 36	EBCDIC Model Number						
37	Reserved = 0						
38 - 45	EBCDIC Serial Number						
46 - 51	EBCDIC Vendor ID 'IBM'						

If the media is not available, bytes 0-4 are valid, the ASCII fields contain ASCII spaces (20h) and the EBCDIC fields contain EBCDIC spaces (40h), all other fields are zero (00h).

See 1.3.1, "Inquiry Data Format - EVPD = 0" on page 30 for a description of the fields in byte 0.

The Page Code field is set to the value of the Page Code field in the CDB.

The Page Length field specifies the length (in bytes) of the vendor unique VPD information (bytes 4 to 51). If the allocation length of the command descriptor block is too small to transfer all the data, the Page Length field is not adjusted to reflect the truncation.

The ASCII Length field specifies the length of the ASCII VPD information that follows (bytes 5 through 29). If the allocation length of the command descriptor block is too small to transfer all of the ASCII VPD data, the ASCII VPD length field is not adjusted to reflect the truncation.

The ASCII Product Type is the four (4) character ASCII field. This field is '0664'.

The ASCII Model Number field contains a three (3) character ASCII model number for the device. See 1.3.1, "Inquiry Data Format - EVPD = 0" on page 30 for the model numbers.

The ASCII Serial Number field contains the unit serial number. The field is right aligned and the unused bytes are filled with ASCII zeros (30h).

The ASCII Vendor ID field contains the vendor ID (IBM). The field is left aligned and the unused bytes are filled with ASCII space characters (20h). This field can be customized to meet the requirements of a user.

The EBCDIC Product Type is the four (4) character EBCDIC field. This field is **'0664'**.

The EBCDIC Model Number field contains a three (3) character EBCDIC model number for the device. See 1.3.1, "Inquiry Data Format - EVPD = 0" on page 30 for the model numbers.

The EBCDIC Serial Number field contains the unit serial number. The field is right aligned and the unused bytes are filled with EBCDIC zeroes (F0h).

The EBCDIC Vendor ID field contains the vendor ID (IBM). The field is left aligned and the unused bytes are filled with EBCDIC space characters (40h). This field can be customized to meet the requirements of a user.

## 1.4 Log Select

Table 16. Log Select Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 4Ch							
1	Logical Unit Number			Reserved = 0			PCR	SP = 0
2	PC = 11b		Reserved = 0					
3	Reserved = 0							
4	Reserved = 0							
5	Reserved = 0							
6	Reserved = 0							
7	(MSB) Parameter List Length = 0 (LSB)							
8								
9	VU = 0		Reserved = 0			Flag	Link	

The *Log Select* command provides a means for the initiator to clear statistical information maintained by the Drive and reported via the Log Sense Command.

**PCR bit** The Parameter Code Reset (PCR) bit determines whether the Log Sense parameters will be cleared and unit attentions posted for all other initiators. A value of 1 indicates that the parameters should be cleared and unit attentions posted, while a value of 0 indicates that the parameters should not be cleared and no unit attentions should be posted.

**SP bit** The Save Parameters (SP) bit must be set to zero indicating no pages are savable.

**PC field** The Page Control field defines the type of parameters to be selected. The PC field must be set to 11b, indicating that the Current Cumulative values are set to their default values of 00h.

**Parameter List Length** The Parameter List Length must be zero to indicate that no data is transferred from the initiator to the target during the DATA OUT phase.

If one or more fields in the CDB is not correctly set, the command shall be terminated with a Check Condition Status. The Sense Key shall be set to Illegal Request, and the additional sense code set to Invalid Field in CDB.

The following list contains all individual page parameters that are set to their default value of zero by the Log Select command.

- Page 30h parameters:
  - Zero Seeks counter.
  - Seeks > = to 2/3 counter.
  - Seeks > = 1/3 and < 2/3 counter.
  - Seeks > = 1/6 and < 1/3 counter.
  - Seeks > = 1/12 and < 1/6 counter.
  - Seeks > 0 and < 1/12 counter.
  - Overrun Counter.
  - Underrun Counter.
  - Device Cache Read Hits
  - Device Cache Partial Read Hits
  - Device Cache Write Hits
  - Device Cache Fast Writes
- Page 35h parameters:
  - Cumulative Cache Hits on Reads.
  - Cumulative Cache Partial Hits on Reads.
  - Cumulative Cache misses on Reads.

The target generates a unit attention condition (to indicate that parameters have changed) for all initiators except the one that issued the Log Select command; this is only done when the PCR bit is set to 1.

## 1.5 Log Sense

Table 17. Log Sense Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 4Dh							
1	Logical Unit Number			RSVD = 0			PPC = 0	SP = 0
2	PC = 01		Page Code					
3	Reserved = 0							
4	Reserved = 0							
5	(MSB) Parameter Pointer = 0 (LSB)							
6								
7	(MSB) Allocation Length (LSB)							
8								
9	VU = 0		Reserved = 0			Flag	Link	

The *Log Sense* command allows the Initiator to retrieve statistical data about the Drive.

- PPC bit** The Parameter Pointer Control (PPC) bit must be set to zero. This specifies that the Drive start transferring data starting from the field specified in the parameter pointer field for the number of bytes specified by the allocation length. If PPC bit is set to 1, Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.
- SP bit** The Save Parameters (SP) bit must be set to zero for all pages. This specifies that the Drive not save any log parameters. If SP bit is set to 1, Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.
- PC field** The Page Control field defines the type of parameters to be selected. This field must be set to 01b to specify Current Cumulative values. If PC field is not set to 01b, Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.
- Page Code** The Page Code field identifies which page is being requested. This field must be set to the values indicated in Page 00h. If the Page Code value is invalid, a Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.
- Parameter Pointer Field** The parameter pointer field specifies the beginning field for the transfer. This field must be set to 0000h. If the Parameter Pointer Field is not zero, a Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.



**Allocation Length**

The Allocation Length field specifies the maximum number of bytes the Initiator has allocated for returned Log Sense Data. No bytes are transferred if the allocation length is zero. This condition is not considered an error. The Target terminates the Data In phase when all available Log Sense Data has been transferred or when the number of bytes transferred equals the allocation length, whichever is less.

## 1.5.1 Log Page Parameters

Each log page begins with a four-byte page header followed by zero or more variable-length log parameter.

**Page Header** The Page Code field identifies which log page is being transferred.

The Page Length field specifies the length in bytes of the following log parameters.

**Log Parameters** Each log parameter begins with a four-byte parameter header followed by one or more bytes of parameter value data.

The Parameter Code field identifies which log parameter is being transferred for that log page.

## 1.5.2 Log Sense Page 00h

Table 18. Log Sense Page 00h							
Byte	BIT						
	7	6	5	4	3	2	1
0	Reserved = 0		Page Code = 0				
1	Reserved = 0						
2 - 3	Page Length = 0005h (Number of Pages Supported)						
4	First supported page code = 00h						
5	Second supported page code = 30h						
6	Third supported page code = 32h						
7	Fourth supported page code = 33h						
8	Fifth supported page code = 35h						

Page 00h indicates the supported log sense pages. This page is used to determine which additional pages can be requested by the Initiator.

### 1.5.3 Log Sense Page 30h

Table 19. Log Sense Page 30h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	Reserved = 0		Page Code = 30h						
1	Reserved = 0								
2 - 3	Page Length = 0030h								
4 - 5	Parameter Code = 0								
6	Reserved = 0								
7	Parameter Length = 2Ch								
8 - 9	Zero Seeks								
10 - 11	Seeks > = to 2/3 disk								
12 - 13	Seeks > = 1/3 and < 2/3								
14 - 15	Seeks > = 1/6 and < 1/3								
16 - 17	Seeks > = 1/12 and < 1/6								
18 - 19	Seeks > 0 and < 1/12								
20 - 21	Reserved = 0								
22 - 23	Reserved = 0								
24 - 25	Overrun Counter								
26 - 27	Underrun Counter								
28 - 31	Device Cache Read Hits								
32 - 35	Device Cache Partial Read Hits								
36 - 39	Device Cache Write Hits = 0								
40 - 43	Device Cache Fast Writes = 0								
44 - 51	Reserved = 00h								

Page 30h returns performance counter information. This includes seek counters and buffer overrun/underrun counters.

The appropriate seek counter is incremented once during the execution of each of the following SCSI commands or pair of linked commands:

- Pre-Fetch
- Read (6)
- Read (10)
- Seek (6)
- Seek (10)
- Verify
- Write (6)
- Write (10)
- Write and Verify
- Write Same

Only one seek counter is incremented for each of these commands and the counter is incremented only once per command. The length of the initial seek that is required to access the first Logical Block specified for the SCSI command, determines which seek counter is incremented. The Zero Seek counter is incremented if a seek is not required or if only a head switch is required to access the first Logical

Block. After the initial seek, no further counter incrementing is performed for that command.

**Implementer's Note** - The length of a seek as reported in page 30 may differ from expected results. The reason for this is that the drive executes Idle Time Functions as explained in the product Functional Specification in between operations of the drive. The seek operations which occur in Idle Time Functions are not directly entered into page 30 seek counters but they change the length of the following seek. This is because after the Idle Time Function is completed, the heads will not necessarily be in the same position as they were at the completion of the previous command.

A buffer overrun or underrun condition occurs when the Initiator does not transfer data to or from the Target data buffer fast enough to keep up with reading or writing the media. The buffer overrun counter is incremented during operations that require a Data In phase when a buffer full condition prevents the continued transfer of data from the media to the data buffer. The buffer underrun counter is incremented during operations that require a Data Out phase when a buffer empty condition prevents the start or continuation of a data transfer from the data buffer to the media (or a data transfer from the media for a Verify command with BytChk=1).

Buffer Overrun conditions are detected during the following SCSI commands or pair of linked commands:

- Read (6)
- Read (10)

Buffer Underrun conditions are detected during the following SCSI commands or pair of linked commands:

- Verify with BytChk=1
- Write (6)
- Write (10)
- Write and Verify
- Write Same

**ZERO SEEKS** The number of times no seek was required. The operation may have resulted in a head switch.

**SEEKS > = 2/3 DISK** The number of seeks equal to or greater than 2/3 of the disk.

**SEEKS > = 1/3 AND < 2/3 DISK** The number of seeks equal to or greater than 1/3 and less than 2/3 of the disk.

**SEEKS > = 1/6 AND < 1/3 DISK** The number of seeks equal to or greater than 1/6 and less than 1/3 of the disk.

**SEEKS > = 1/12 AND < 1/6 DISK** The number of seeks equal to or greater than 1/12 and less than 1/6 of the disk.

**SEEKS > 0 AND < 1/12 DISK** The number of seeks less than 1/12 of the disk.

**OVERRUN COUNTER** The number of times that data was available to be transferred from the arm but the device buffer still contained data that had not been retrieved by the Initiator. Consequently, the disk had to take an additional revolution until the buffer was available to accept data.

**UNDERRUN COUNTER** The number of times that the DASD was ready to transfer data to its disk (on a write), but its buffer was empty (ie. had not been filled by the Initiator) thus the disk was forced to take an extra revolution.

**DEVICE CACHE READ HITS** The number of times that all of the data requested by the read operation was obtained from the device read or write cache. This is a duplicate of the Cumulative Cache Hits on Reads counter on page 35.

**DEVICE CACHE PARTIAL READ HITS** The number of times that a portion, but not all, of the data requested by the read operation was obtained from the device read or write cache. A physical operation to the device media was required to obtain the remaining data. This is a duplicate of the Cumulative Cache Partial Hits on Reads counter on page 35.

**DEVICE CACHE WRITE HITS** The number of times that the data associated with a write operation replaces, or is combined with, existing data in the device write cache, thereby eliminating a write operation. This counter will always be zero.

**DEVICE CACHE FAST WRITES** The number of times that space was available in the device write cache for the data associated with a write operation and a response was returned immediately. This counter will always be zero.

The statistics reported by this page are lost on a self-initiated reset or when the Drive is powered off.

The statistics reported by this page are NOT lost on a SCSI reset or Bus Device Reset message.

## 1.5.4 Log Sense Page 32h

Table 20. Log Sense Page 32h							
Byte	BIT						
	7	6	5	4	3	2	1
0	Reserved = 0		Page Code = 32h				
1	Reserved = 0						
2	(MSB) Page Length = $4 + \ln_1 + \dots + \ln_n$ (LSB)						
3							
<b>Log Parameter(s)</b>							
4 $\ln_1 +$ 3	Log Parameter 0 (Length $\ln_1$ )						
-----	-----						
$x - \ln_n$ x	Log Parameter n (Length $\ln_n$ )						

### 1.5.4.1 Log Parameter Format - Page 32h

Table 21. Log Parameter Format - Page 32h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0 1	(MSB) Parameter Code = 0, 1, ..., n-1, or n (LSB)								
2	Reserved = 0						LBIN = 1	LP = 1	
3	Parameter Length = 8m								
4 5 6	(MSB) Cylinder Number of Site 0 (LSB)								
7	Head Number of Site 0								
8 9 10 11	(MSB) Sector Number of Site 0 (LSB)								
-----	-----								
8m + 4 8m + 5 8m + 6	(MSB) Cylinder Number of Site m (LSB)								
8m + 7	Head Number of Site m								
8m + 8 8m + 9 8m + 10 8m + 11	(MSB) Sector Number of Site m (LSB)								

**LBIN bit** The LBIN bit is set to one, indicating that the following list parameters are in binary format.

**LP bit** The LP bit is set to one, indicating that the following parameters are list parameters.

Page 32h is a listing of physical error sites that should be reallocated. (See 4.7, “Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign” on page 211 for more information.) When an error site is reallocated it is removed from this list. When a format is performed, this list is cleared. If no sites require reallocation, 8 bytes are returned, the page length is 4 and the parameter length is 0.

The statistics reported to the Initiator by this page are part of error log information maintained by the Target. The logs are periodically saved to the disk and are restored from the disk after a power cycle or self-initiated reset.

**Implementer's Note** - The Initiator may request this page and return the 8 byte site descriptions to the Target as a Dlist when issuing a Format command with Dlist option.

## 1.5.5 Log Sense Page 33h

Table 22. Log Sense Page 33h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	Reserved = 0		Page Code = 33h						
1	Reserved = 0								
2	(MSB)								
3	Page Length = 4 + ln <sub>1</sub> + ... + ln <sub>n</sub> (LSB)								
Log Parameter(s)									
4	Log Parameter 0								
ln <sub>1</sub> + 3	(Length ln <sub>1</sub> )								
-----	-----								
x - ln <sub>n</sub>	Log Parameter n								
x	(Length ln <sub>n</sub> )								

### 1.5.5.1 Log Parameter Format - Page 33h

Table 23. Log Parameter Format - Page 33h								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB)							
1	Parameter Code = 0, 1, ..., n-1, or n (LSB)							
2	Reserved = 0					LBIN = 1	LP = 1	
3	Parameter Length = 4m							
4	(MSB)							
5	Logical Block Address 0							
6								
7								(LSB)
-----	-----							
4m + 4	(MSB)							
4m + 5	Logical Block Address m							
4m + 6								
4m + 7								(LSB)

**LBIN bit** The LBIN bit is set to one, indicating that the following list parameters are in binary format.

**LP bit** The LP bit is set to one, indicating that the following parameters are list parameters.

Page 33h is a listing of LBAs that should be reallocated. (See 4.7, “Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign” on page 211 for more information.) The data returned by the Target is sorted in descending LBA order. When an error site is reallocated it is removed from this list. When a format is performed, this list is cleared. If no sites require reallocation, 8 bytes are returned, the page length is 4 and the parameter length is 0.



The statistics reported by this page are part of error log information maintained by the Target. The logs are periodically saved to the disk and are restored from the disk after a power cycle or self-initiated reset.

**Implementer's Note** - The Initiator may request this page when ready to reassign LBAs that the Target has recommended for reallocation. The Initiator should issue a reassign command for each LBA in the order received. This assures that the reassigns occur in the correct descending order. (Otherwise, incorrect LBAs may be reassigned.)

## 1.5.6 Log Sense Page 35h

Table 24. Log Sense Page 35h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	Reserved = 0		Page Code = 35h						
1	Reserved = 0								
2 - 3	Page Length = 0028h (40)								
4 - 5	Parameter Code = 0								
6	Reserved = 0								
7	Parameter Length = 24h (36)								
8 - 11	Cumulative Cache Hits on Reads.								
12 - 15	Cumulative Cache Partial Hits on Reads.								
16 - 19	Cumulative Cache misses on Reads.								
20 - 23	Reserved = 0								
24 - 27	Reserved = 0								
28 - 31	Reserved = 0								
32 - 35	Reserved = 0.								
36 - 39	Reserved = 0.								
40 - 43	Reserved = 0.								

Page 35h contains information about cache utilization.

The Cache Hit parameter is incremented when all of the Requested Data is in the cache. The Cache Miss parameter is incremented when the start of the Requested Data is not in the cache. The Cache Partial Hit parameter is incremented when the start of the Requested Data is in the cache and the end is not.

**Implementer's Note** - The Initiator may use this information to "fine tune" the caching parameters for the particular application it is using. (See 4.9, "Segmented Caching" on page 214 for more information.)

The statistics reported by this page are lost when the Drive is powered off or a self-initiated reset occurs.

The statistics reported by this page are NOT lost on a SCSI reset or Bus Device Reset message.

## 1.6 Mode Select (6)

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 15h							
1	LUN			PF	Reserved = 0			SP
2	Reserved = 0							
3	Reserved = 0							
4	Parameter List Length							
5	VU = 0		Reserved = 0			Flag	Link	

The Mode Select command provides a means for the Initiator to specify LUN or device parameters to the Target. It also allows an Initiator to specify options the Target uses in error recovery and formatting.

There is a single set of Mode Page parameters shared by all initiators.

A PF (Page Format) bit value of 1 indicates the data sent by the Initiator after the Mode Select Header and the Block Descriptor, if any, complies to the Page Format. The Target ignores this field since it only accepts mode parameters in the Page Format.

The SP (Save Parameter) bit set to 0 indicates that the Target does not save the saveable pages sent during the Data Out phase to disk. The SP bit value of 1, indicates that the Target saves the saveable pages to the Reserved Area. Saveable Pages are Pages for which a preceding Mode Sense command returned the PS bit of the Page Header set to 1 (see 1.7, “Mode Sense (6)” on page 90). Unlike the page data, which is saved as a function of the SP bit, the Block Descriptor parameters are saved only at the successful completion of a Format Unit command.

The Parameter List Length field specifies the number of bytes of data the Initiator has available to send to the LUN. A parameter list length of zero suppresses data transfer and is not considered an error.

The Mode Select parameter list contains a 4-byte header, followed by zero or one Block Descriptor followed by zero or more pages, as shown in Table 26 on page 57.

The Initiator should issue a Mode Sense command requesting all changeable values (see PC field in byte two of the CDB for 1.7, “Mode Sense (6)” on page 90) prior to issuing a Mode Select command. This is necessary to find out which pages are implemented by the Target and the length of those pages. The Target returns in the pages of the Mode Sense command the number of bytes supported for each page. The Page Length set by the Initiator in the Mode Select command must be the same value as that returned by the Target in Mode Sense page length. If this is not true, the Target sets *Check Condition Status* with the sense key of Illegal Request and additional sense code of Parameter List Length Error.

If any of the parameters sent by a Mode Select command are invalid, *Check Condition Status* is returned with a Sense Key of Illegal Request and Additional Sense Code of Invalid Field in Parameter List. The SKSV bit is on and the invalid byte is

indicated in the Field Pointer field. The BPV bit is on and the invalid bit is indicated in the Bit Pointer field (see Appendix A, “SCSI Sense Data Format” on page 229).

Each time the Initiator receives a Unit Attention (see 4.1.5, “Unit Attention Condition” on page 184) sense key from the Target, the Initiator must assume the Target’s mode of operation has been changed. The Initiator may need to reconfigure the Target to its requirements. See 1.7, “Mode Sense (6)” on page 90 to determine the Target’s parameters.

A Mode Select command also causes a Unit Attention condition (see 4.1.5, “Unit Attention Condition” on page 184) for all initiators except the one that issued the Mode Select command.

Mode page parameter Current values are equal to the Default values before the Saved values are read from the Reserved Area of the media. The non-zero Mode page parameter Default values are always:

PAGE 00h

- *UQE* bit is set to 1.
- *ASDPE* bit is set to 1.

PAGE 01h

- *Read Retry Count* field is set to 01h.
- *Correction Span* field is set to 30h (Decimal 48).
- *Write Retry Count* field is set to 01h.

PAGE 03h

- *Data Bytes per Physical Sector* is set to 0200h
- The exact Default values of the following parameters are Model Number dependant and are defined in the *product Functional Specification*. Use the assumed Default logical Block Length of 512 bytes per block when calculating them.
  - *Tracks per Zone*
  - *Sectors per Track*
  - *Track Skew Factor*
  - *Cylinder Skew Factor*
- The *Interleave* field is set to 0001h (Decimal 1),
- The *HSEC* bit is set to 1.

PAGE 04h

- *Number of Cylinders* is set to 0B29h (Decimal 2857),
- *Number of Heads* is set to 0Fh (Decimal 15).
- *Landing Zone* is based on the Device Model Number. The Landing Zone can be calculated by adding 200 to the maximum cylinder for the device. Refer to the product Functional Specification for the maximum cylinder value.
- *Medium Rotation Rate* is set to 1518h (Decimal 5400).

PAGE 07h

- *Verify Retry Count* is set to 01h.

PAGE 08h

- *Disable Pre-fetch Transfer Length* is set to FFFFh,
- *Maximum Pre-fetch* is set to FFFFh,
- *Maximum Pre-fetch Ceiling* is set to FFFFh.
- *Number of Cache Segments* field is set to 8.

PAGE 0Ah All fields are set to zero.

Not all of the Mode Select options on all pages are both changeable and saveable (also see 1.7, “Mode Sense (6)” on page 90).

All Saved values that are changeable, are stored on the disk and can be initialized with a Mode Select command of the Saved parameters. This allows customers to individualize the value of the Current parameters after a start-up sequence without having to issue a Mode Select command to change the Current parameters every time the Drive is spun up.

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0							
1	Medium Type = 0							
2	WP	Reserved = 0	DPOFUA	Reserved = 0				
3	Block Descriptor Length = 0 or 8							

### 1.6.1.1 Header

The only portion of the Mode Select Header that the Initiator can specify is the Block Descriptor Length field. A maximum of one block Descriptor may be specified. Therefore, the Block Descriptor Length field may only have values of 0 or 8.

A Medium Type value of 00h indicates that the Drive is using its only recording density.

The WP (Write Protect) bit of zero indicates the target is write enabled. A WP bit of one indicates the target is write protected. The Target ignores the value in the WP field on a Mode select command.

The DPOFUA bit, is considered reserved on the Mode Select Command by SCSI-2. The Target ignores the value in the DPOFUA field on a Mode Select command.

Table 27. Mode Select Parameter List - Block Descriptor								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0							
1	(MSB) Number of Blocks (LSB)							
2								
3								
4	Reserved = 0							
5	(MSB) Block Length (LSB)							
6								
7								

### 1.6.1.2 Block Descriptor

The Block Descriptor provides formatting information about the Number of Blocks (user addressable) to format at the specified Block Length in bytes. The Initiator must specify either the Number of Blocks field as 000000h (indicating all blocks) or supply the exact number of blocks that this device supports for the specified Block Length. The exact number of blocks the Drive supports can be obtained with the Mode Sense command only if the media is currently formatted to the same Block Length as given in the Block Length field of the Block Descriptor. The exact number of blocks can also be calculated with the 'logical blocks/Drive' equation as defined in the *product Functional Specification*. If any other value is specified for the Number of Blocks field, the command fails with *Check Condition Status*. The Block Length field in the Block Descriptor parameter is not saved and is not considered the currently formatted Block Length (current Block Length) until the Format Unit command is issued and completes successfully.

**Note:** Please refer to the product Functional Specification for information regarding supported block lengths for this product.

There is an implicit association between parameters defined in the Format Device Page (03h) and the Block Descriptor. If the values in the Block Descriptor are different than the Current values, then the Current and Saved values of some or all of the Format Device page parameters may change. See 1.6.5, “Page 3 - Format Device Parameters” on page 77 for details on those values.

Table 28. Mode Select Parameter List - Page Descriptor(s)								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	RSVD = 0	RSVD = 0	Page Code					
1 2-n	Page Length (in bytes) Refer to each page							

### 1.6.1.3 Page Descriptor

Byte zero contains the code of the desired page. Bits 7 and 6 of Byte zero are reserved and must be set to 0. The Page Length field must be set equal to the supported length of the entire page minus 2 bytes. The remaining bytes contain the page parameters.

If an Initiator sends multiple copies of a page to the Target, the last one received is used. This is not considered an error.

## 1.6.2 Page 0 - Vendor Unique Parameters

Table 29. Mode Select Data Format Page 0								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	RSVD = 0	RSVD = 0	Page Code = 00h					
1	Page Length = 0Ah							
2	QPE	UQE	DWD	Reserved = 0				
3	ASDPE	RSVD = 0	CMDAC	RPFAE	RSVD = 0			CPE
4	Ignored							
5	TCHD	TCC	DSN	FRDD	DPSDP	WPEN	CAEN	LITF
6	Reserved = 0							
7	Reserved = 0							
8	RSVD = 0	ADC	QEMC	DRD	LED Mode			
9	Reserved = 0							
10	Command Aging Time Limit (HI)							
11	Command Aging Time Limit (LO)							

Following are parameter options for Page 0 of Mode Select. Refer to Table 29.

**Byte 2** contains the following fields -

**QPE** The QPE (Qualify Post Error) bit applies to read data operation errors (No Sector Found, Sync Errors and ECC Errors) and the read ID field portion of a write operation that are recovered when the PER bit is one. The QPE bit has no effect on non-data errors or errors detected during verify operations.

**Note:** (See 1.6.7, “Page 7h - Error Recovery Parameters for Verify” on page 83 for information on control of reporting verify errors via the Verify PER bit.)

The QPE bit allows the Initiator to inhibit the reporting of soft read data errors occurring below a set DRP level. This level is 10 for read operations and 2 for write operations. A QPE bit of 0 enables the Target to report all recovered read data errors as defined by Page 01. A QPE bit of 1 indicates that the Target only report recovered read data errors which exceed DRP level 9 and recovered write errors which exceed DRP level 1. (DRP levels are described in the Appendix section B.1, “Data Recovery Procedure for Data Field Errors” on page 253.) The QPE bit setting is used by the Target when reporting errors associated with the transfer of the Initiator's data for the following commands:

- Read(6)
- Read(10)
- Write(6)
- Write(10)
- Write Same
- Write portion of Write and Verify

For all other commands, the QPE bit setting is unused and treated as 0.

- UQE The UQE (untagged queuing enable) bit has no effect. Untagged queuing is always enabled. 4.3.1, “Untagged Queuing” on page 204 .
- DWD The DWD (disable write disconnect) bit allows the Initiator to control whether the Target is allowed to disconnect following the receipt of a Write (6), Write (10), Write and Verify or Write Same command (opcodes 0Ah, 2Ah, 2Eh and 41h). A DWD bit of 0 allows the Target to disconnect from the SCSI Bus after receiving a Write command and prior to starting the Data Out phase. (The previous Identify message must also grant the Target the privilege of disconnection or the Target does not disconnect.) A DWD bit of 1 indicates that the Target is not allowed to disconnect from the SCSI bus after receiving a Write command and prior to starting the Data Out phase unless the write command must be queued. The Target remains connected to the SCSI bus until the Data Out phase is started. After the Data Out phase is started, the Target may disconnect to free the SCSI bus for use by other devices. This occurs if the Target's internal control algorithms and other disconnect/reconnect control parameters indicate that this is appropriate and permissible.

**Byte 3** contains the following fields -

ASDPE The ASDPE (Additional Save Data Pointer Enable) bit determines whether or not the Save Data Pointer message is sent to the Initiator prior to disconnection. This bit is only used by the Target after the Default Mode parameter values are overridden with the Saved values which are read from the Reserved Area of the media as a part of the motor start-up sequence. Before the Saved values are read from the Reserved Area of the media, the Save Data Pointer message is always sent to the Initiator prior to disconnection.

When ASDPE is set to 1, the Save Data Pointer message is sent prior to every disconnect. When ASDPE is set to 0, the Save Data Pointer message is only sent prior to disconnection if the following conditions are true:

- A Data phase has occurred since the connection for the current command was established, and
- Another Data phase is required to successfully complete the command.

**Note:** A Save Data Pointer message is not sent prior to the Disconnect message of the TTD data ready sequence regardless of the ASDPE bit. (See 3.1.16, “Target Transfer Disable (13h)” on page 154 for more information.)

CMDAC The CMDAC bit controls how the Target defines a signal on the LED pin that indicates when the Drive is "active". See the *product Functional Specification* for details on the electrical parameters of this LED Pin signal.

The signal is provided on the LED pin only when the LED pin is initially detected to be at a TTL high level (HIGH) when the Drive is powered on. If the LED Pin is detected to be HIGH at that time, then

- When the CMDAC bit = 1
  - The LED pin goes to a TTL low level (LOW) for a "Command Active" condition when -



1. A Command is received by the Target.
  2. An "Active LUN" condition is in effect.
- The LED pin goes HIGH for a "Command Inactive" condition when a -
    1. LUN becomes inactive following the successful completion of a host initiated command or termination due to an error or exception condition
    2. LUN becomes inactive following the completion of Active LUN condition or termination due to an error or exception condition
    3. SCSI Bus Reset or a SCSI Bus Reset Message is received
    4. Self-Initiated reset is performed
    5. previously received command is terminated via an Abort message

**Note:** In the case of Immediate commands such as Format and Start Unit, the LED pin does not become HIGH until the corresponding Active Lun condition for this command has been completed.

**Note:** The LED pin does not become LOW during idle activity such as Thermal Update or Disk Sweep.

    - When the CMDAC bit = 0
      - The LED pin goes LOW for a "Motor Active" condition which means that the Spindle Motor is spinning.
      - The LED pin goes HIGH for a "Motor Inactive" condition which means that the Spindle Motor is not spinning.

**RPF**AE The RPFAE (Report Predictive Failure Analysis Error) bit allows certain recovered errors to be reported at the completion of the command that handled the data when PER=0. (PER=0 normally inhibits the reporting of all recovered errors. See 1.6.3, "Page 1 - Error Recovery Parameters" on page 66 for more details on reporting recovered errors.)

An RPFAE bit of 1 indicates that recovered errors that are associated with the drive's Predictive Failure Analysis (PFA) functions are reported. When RPFAE is set to 0, whether or not recovered errors are reported is based upon the state of the PER bit. The PFA functions include the Auto/Recommend Rewrite/Reassign functions.

The reporting of recovered errors are controlled by the setting of the PER, QPE, and RPFAE bits as follows:

PER	QPE	RPFAE	Reporting of Recovered Errors
0	x	0	Recovered errors are not reported.
0	x	1	Only recovered errors associated with the PFA functions are reported.
1	0	x	All recovered errors are reported.
1	1	x	Recovered errors that are above the threshold defined by the QPE bit are reported, plus recovered errors that are associated with the PFA functions.
<b>Note:</b> See 4.8, “Predictive Failure Analysis” on page 213 for details concerning PFA.			

**CPE** The CPE (concurrent processing enable) bit controls whether or not I/O process are allowed to execute concurrently. Concurrent processing is when multiple I/O processes are active (not queued). This implies the Data phase of I/O processes are allowed to overlap. The Target utilizes concurrent processing to perform back to back writes. (See 4.4.1, “Back to Back Writes” on page 208.)

When concurrent processing is disabled, only one non-priority I/O process will be active at a time. Priority commands are never queued and are always allowed to execute concurrently. (See 4.2, “Priority Commands” on page 203.)

Concurrent Processing is enabled if CPE = 1.

Concurrent Processing is disabled if CPE = 0.

**Byte 4 is Ignored** which is treated the same as a Reserved field except that it is allowed to be set to any value.

**Byte 5** contains the following fields:

**TCHD** The Thermal Compensation Head Control bit is used to control whether thermal compensation is done for all heads at once, or one head at a time. A TCHD bit of 1 indicates that all heads are thermally compensated at one time. A TCHD bit of 0 indicates that heads are thermally compensated one head at a time.

**TCC** The Thermal Compensation Control bit allows the Initiator to force a thermal compensation early by issuing a Rezero Unit Command. A TCC bit of 1 indicates that a Rezero Unit Command will force thermal compensation while a TCC bit of 0 indicates that a Rezero Unit Command will NOT force a thermal compensation to take place.

**Note:** If the TCC bit is 1 and thermal compensation is due based on the predetermined file algorithm, thermal compensation will still be done. This bit cannot be used to delay thermal compensation beyond the regularly scheduled times.

Table 30. TCHD and TCC bit combinations.

TCHD	TCC	Description
0	0	Thermal Compensation is not forced early, it is done according to the normal predetermined file algorithm.
1	0	Thermal Compensation is done on the regular schedule, based on the predetermined file algorithm, but all heads are updated at once.
0	1	Thermal Compensation is done one head at a time but can be forced early via a Rezero Unit Command.
1	1	Thermal Compensation is done to all heads at once and can be forced early via a Rezero Unit Command.

**DSN** The Disable Synchronous Negotiations bit is used to control whether the Drive initiates synchronous and wide negotiations after initial spin up. A DSN bit of 0 indicates that the Drive does initiate synchronous/wide negotiations, while a DSN bit of 1 indicates that the Drive does not.

**FRDD** The Format and Reassign Degraded Disable bit prevents the file from reporting Format or Reassign degraded on a Test Unit Ready command or a Start Unit command and causes media access commands (ex write, read) to report a media error if degraded. A FRDD bit of 1 indicates that Format or Reassign degraded are not reported on Test Unit Ready or Start Unit commands and media access commands report media errors. A FRDD bit of 0 indicates that Format or Reassign degraded is reported on Test Unit Ready and Start Unit commands.

**DPSDP** The Data Phase Save Data Pointer bit controls whether the file sends a Save Data Pointer message at the end of the data phase. A DPSDP bit of 0 indicates that the Drive sends a Save Data Pointer message prior to disconnection only if the following conditions are true:

A data phase has occurred since the connection for the current command was established, and

Another data phase is required to successfully complete the command.

A DPSDP bit of 1 indicates that the Drive will send a Save Data Pointer message prior to every disconnection once a data phase has occurred for the current command. If the ASDPE bit is set to 1, the Save Data Pointer message is sent prior to every disconnect regardless of the value of the DPSDP bit.

**Note:** A Save Data Pointer message is not sent prior to the Disconnect message of the TTD data ready sequence regardless of the ASDPE and DPSDP bits. (See 3.1.16, “Target Transfer Disable (13h)” on page 154 for more information.)

**WPEN** The Write Protect Enable bit allows the Drive to enter a Write Protect mode via the Drive’s Option Pin. Refer to 4.15, “Option Pin” on page 223 for information regarding the Option Pin. When WPEN is 1 and the Option Pin is grounded, the Drive enters a Write Protect mode, disallowing the following commands:

Format Unit

Synchronize Cache

Write  
Write Extended  
Write and Verify  
Write Same  
Write Long  
Reassign Blocks

In addition, automatic rewrites and reallocates are prohibited. If the Option Pin is not grounded this bit will have no effect on the Drive's operation. WPEN set to 1 inhibits Automatic Rewrite/Reallocate. Recommend Rewrite/Reallocate is not affected by the setting of WPEN.

A WPEN of 0 has no effect on the Drive's operation regarding reads and writes.

- CAEN** The Command Aging Enable bit used in conjunction with the Command Aging Time Limit bytes, allows the Initiator to limit the amount of time a command can be queued. The CAEN bit set to 1, causes the target to reorder a command to the earliest possible execution time after the command has been in the queue for the time duration specified in the Command Aging Time Limit bytes. Restrictions on the reordering algorithm as specified by the Queue Algorithm Modified (see 1.6.9, "Page 0Ah - Control Mode Parameters" on page 88) still apply in this case. Data integrity is maintained when Restricted Reordering is in effect. A CAEN bit equal to 0 indicates that Command Aging is not enabled and the Command Aging Time Limit bytes are ignored.
- LITF** The Limit Idle Time Functions bit controls whether the functions, Save Error Logs and Predictive Failure Analysis, are performed during Idle Time. For an explanation of "Idle Time Function" refer to the product Functional Specification. A bit of 1 indicates that Save Error Logs and Predictive Failure Analysis functions will not be performed during Idle periods. This may help prevent delays as mentioned in the product Functional Specification. A bit of 0 indicates that the Drive will perform all Idle Time Functions.

**Bytes 6 and 7** are reserved and must be equal to 0.

**Byte 8** contains the QEMC bit, ADC bit, DRD bit, and the LED Pin Indicator Modes for the LED.

**Byte 8** contains the QEMC bit, DRD bit, and the LED Pin Indicator Modes for the LED.

- QEMC** The QEMC (Queue Error Management Control) affects the Target's operation when QErr is set to a one. QEMC has no effect if QErr = 0. See QErr on 1.6.9, "Page 0Ah - Control Mode Parameters" on page 88 for a description on the how QEMC effects QErr.
- ADC** The ADC (Adaptive Caching) affects the Target's caching algorithm. If the ADC bit is set to one, the Target will set the Number of Cache Segments equal to 8, and may alter this number to optimize performance. When the initiator clears ADC after having ADC set, the drive will default the number of cache segments to 8. This can be modified by changing the Number of Cache Segments in Page 8, Byte 13 to the

number desired. With ADC set, the Target may also alter the prefetch based on perceived random or sequential operations. If the ADC bit is zero the Target will use the cache as defined by Page 8h, Read Caching Parameters.

**DRD** The DRD (disable read disconnect) bit when set to 1 prevents the target from disconnecting from the SCSI Bus following the receipt of a Read (6) or a Read (10) command until it has determined that the requested data is not available in the data buffer. If the data is available, the target will not disconnect prior to starting the data transfer unless the read command must be queued. If the data is not available, the target will disconnect until enough data has been read from the disk to satisfy the reconnection criteria as determined by the Read Buffer Full Ratio. A DRD bit of 0 allows the target to disconnect immediately after the receipt of a read command without determining if the requested data is available.

LED Mode = 0h (Compatibility) The CMDAC bit controls the LED mode.

CMDAC = 1 (Command Active)

CMDAC = 0 (Motor Active)

LED Mode = 1h (Motor Active). When the motor is spinning, the LED is high.

LED Mode = 2h (Command Active). When there is a command active, the LED is high.

LED Mode = 3h (Degraded). When the Target is in any degraded mode, the LED is high.

LED Mode = 4h (Command Active | Degraded). When there is a command active OR when the Target is in any degraded mode, the LED is high.

LED Mode = 5h-Fh (Reserved for future use).

**Byte 9** is reserved, and must be equal to 0.

**Byte 10 and Byte 11** are used as a timer for the Drive's Command Aging Feature when enabled by the CAEN bit in byte 5. The Command Aging Time Limit Timer is in 50 ms increments.

### 1.6.3 Page 1 - Error Recovery Parameters

Table 31. Mode Select Data Format Page 1								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	RSVD = 0	RSVD = 0	Page Code = 01h					
1	Page Length = 0Ah							
2	AWRE	ARRE	TB	RC	EER = 0	PER	DTE	DCR
3	Read Retry Count							
4	Correction Span = 30h (or 0)							
5	Head Offset Count = 0							
6	Data Strobe Offset Count = 0							
7	Reserved = 0							
8	Write Retry Count = 01h							
9	Reserved = 0							
10 11	(MSB)	Recovery Time Limit = 0						(LSB)

Since bytes 4-11 are not changeable, the Mode Select Command accepts only the values indicated in Table 31 for bytes 4 - 11, with the exception of Correction Span which is noted below.

Following are parameter options for Page 1 of Mode Select. Refer to Table 31.

**Byte 2** contains the following fields -

**AWRE** (Automatic Write Reallocation Enable) bit of 1 enables automatic reallocation to be performed during write operations. The automatic reallocation is performed only if the Target has the valid data (e.g., original data in the buffer or recovered from the medium). The valid data is placed in the reallocated block. All error recovery actions required by the error recovery bits (TB, EER, PER, DTE and DCR) are executed. Error reporting as required by the error recovery bits (EER, PER, DTE, and DCR) is performed only after completion of the reallocation. Therefore any failures that occur during the reallocation are reported. See 4.7, “Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign” on page 211 for execution details and error procedures.

**Note:** Only write errors detected while reading an ID can cause an Automatic Write Reallocation to occur.

When AWRE bit = 0, the Target does not perform automatic reallocation of defective data blocks during write operations. The AWRE bit setting is used by the Target when an error occurs during the transfer of the Initiator's data for the following commands:

- Write(6)
- Write(10)
- Write Same
- Write portion of Write and Verify

For all other commands, the AWRE bit setting is unused and treated as 0.

**ARRE** (Automatic Read Reallocation Enable) = 1, enables automatic reallocation of defective blocks found during read operations. The automatic reallocation is performed only if the Target successfully recovers the data. All error recovery actions required by the error recovery bits (TB, EER, PER, DTE and DCR) are executed. Error reporting as required by the error recovery bits (EER, PER, DTE, and DCR) is performed only after completion of the reallocation. Therefore any failures that occur during the reallocation are reported. See 4.7, “Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign” on page 211 for execution details and error procedures.

When ARRE = 0, the Target does not perform automatic reallocation of defective blocks found during read operations. The ARRE bit setting is used by the Target when an error occurs during the transfer of the Initiator's data for the following commands:

- Read(6)
- Read(10)

For all other commands, the ARRE bit setting is unused and treated as 0.

**TB** (Transfer Block) Bit set to 1 indicates that the failing block be transferred to the Initiator.

TB set to 0 requests that a failing block not be transferred to the Initiator.

If an unrecoverable error occurs and TB is set to a value of 1, the Target transfers the failing block of uncorrected data if the unrecoverable error is a data error (that is, there is data available to transfer). An Initiator may make this distinction by issuing a Request Sense command and examining the Additional Sense code. An Additional Sense code of 11h indicates an unrecoverable read error in the data block. The failing block is transferred only for this case. The TB bit setting is used by the Target when reporting errors associated with the transfer of the Initiator's data for the following commands:

- Read(6)
- Read(10)

For all other commands, the TB bit setting is unused and treated as 0.

**RC** (Read Continuous) bit set to 1 requests the Target to transfer the entire requested length of data without adding delays which would increase or ensure data integrity. This implies that the Target may send erroneous data. This bit has priority over all other error control bits (PER, DTE, DCR, TB).

**Note:** The Target implementation of the RC option is to disable error detection of the data fields but continue normal error detection and recovery for errors occurring in the ID field. If an ID cannot be found, then normal DRP could result in considerable recovery action, including proceeding through all levels of DRP.

RC set to 0 indicates normal interpretation of PER, DTE, DCR, and TB values. The RC bit setting is used by the Target when reporting errors associated with the transfer of the Initiator's data for the following commands:

- Read(6)
- Read(10)
- Prefetch

For all other commands, the RC bit setting is unused and treated as 0.

**EER** (Enable Early Recovery) The EER bit must be 0. The Target does not support early recovery.

**PER** (Post Error) Bit set to 1 requests that the Target report *Check Condition Status* for all recovered data and non-data errors, with the appropriate Sense key. The *Check Condition Status* occurs during data transfer depending either on the DTE setting or occurrence of an unrecoverable error. If multiple errors occur, the Request Sense data reports the block address of either the last block on which the recovered error occurred or of the first unrecoverable error.

PER set to 0 requests that the Target not create *Check Condition Status* for recovered errors. The PER bit setting is used by the Target when reporting errors associated with the transfer of the Initiator's data for the following commands:

- Read(6)
- Read(10)
- Write(6)
- Write(10)
- Write Same
- Write and Verify - the write portion of the command only.

**Note:** (See the description of the RPF AE bit in the section describing 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59 for information regarding the reporting of soft errors associated with Predictive Failure Analysis functions.)

For all other commands, the PER bit setting is unused and treated as 0.

**DTE** (Disable Transfer on Error) Bit set to 1 indicates that the Target creates the *Check Condition Status* and terminates the data transfer to the Initiator upon transferring the last byte of the recovered block when a data error is recovered. The Transfer Length may not be exhausted. DTE may only be set to 1 if PER is also set to 1. The Target creates *Check Condition Status* with Illegal Request sense key if this is not so. DTE set to 1 inhibits Automatic/Recommend Rewrite/Reallocate.

DTE set to 0 continues data transfer through recovered errors. The DTE bit setting is used by the Target when reporting errors associated with the transfer of the Initiator's data for the following commands:

- Read(6)
- Read(10)
- Write(6)
- Write(10)
- Write Same
- Write and Verify - the write portion of the command only.

For all other commands, the DTE bit setting is unused and treated as 0.

**DCR** (Disable Correction) set to 1 indicates that ECC (error correction codes) are not applied in the course of error recovery. DCR set to 1 inhibits Automatic/Recommend Rewrite/Reallocate.

DCR set to 0 enables error correction. The DCR bit setting is used by the Target when reporting errors associated with the transfer of the Initiator's data for the following commands:



- Read(6)
- Read(10)

For all other commands, the DCR bit setting is unused and treated as 0.

The following summarizes valid modes of operation.

PER	DTE	DCR	TB	Description	
0	0	0	0	Retries and error correction are attempted. Recovered and/or corrected data (if any) is transferred with no <i>Check Condition Status</i> at the end of the transfer.	
				soft error	The transfer length is exhausted. Transferred data includes blocks containing recovered errors.
				hard error	Data transfer stops when the unrecoverable error is encountered. The unrecoverable block is not transferred to the Initiator.
0	0	0	1	Retries and error correction are attempted. Recovered and/or corrected data (if any) is transferred with no <i>Check Condition Status</i> at the end of the transfer.	
				soft error	The transfer length is exhausted. Transferred data includes blocks containing recovered errors.
				hard error	Data transfer stops when the unrecoverable error is encountered. The unrecoverable block is transferred to the Initiator (if data error).
0	0	1	0	Retries are attempted but no error correction (ECC) is applied. Recovered data (if any) is transferred with no <i>Check Condition Status</i> at the end of the transfer.	
				soft error	The transfer length is exhausted. Transferred data includes blocks containing recovered errors.
				hard error	Data transfer stops when the unrecoverable error is encountered. The unrecoverable block is not transferred to the Initiator.
PER	DTE	DCR	TB	Description	

PER	DTE	DCR	TB	Description	
0	0	1	1	Retries are attempted but no error correction (ECC) is applied. Recovered data (if any) is transferred with no <i>Check Condition Status</i> at the end of the transfer.	
				soft error	The transfer length is exhausted. Transferred data includes blocks containing recovered errors.
				hard error	Data transfer stops when the unrecoverable error is encountered. The unrecoverable block is transferred to the Initiator (if data error).
0	1	0	0	Illegal request - DTE cannot be 1 when PER is 0.	
0	1	0	1	Illegal request - DTE cannot be 1 when PER is 0.	
0	1	1	0	Illegal request - DTE cannot be 1 when PER is 0.	
0	1	1	1	Illegal request - DTE cannot be 1 when PER is 0.	
1	0	0	0	The highest error level is reported at the end of transfer (see B.3, "Priority of Error Reporting" on page 263). Retries and error correction are attempted. Recovered and/or corrected data (if any) is transferred with <i>Check Condition Status</i> and Recovered Error sense key set at the end of the transfer.	
				soft error	The transfer length is exhausted. Transferred data includes blocks containing recovered errors.
				hard error	Data transfer stops when the unrecoverable error is encountered. The unrecoverable block is not transferred to the Initiator.
1	0	0	1	The highest error level is reported at the end of transfer (see B.3, "Priority of Error Reporting" on page 263). Retries and error correction are attempted. Recovered and/or corrected data (if any) is transferred with <i>Check Condition Status</i> and Recovered Error sense key set at the end of the transfer.	
				soft error	The transfer length is exhausted. Transferred data includes blocks containing recovered errors.
				hard error	Data transfer stops when the unrecoverable error is encountered. The unrecoverable block is transferred to the Initiator (if data error).
PER	DTE	DCR	TB	Description	

PER	DTE	DCR	TB	Description	
1	0	1	0	The highest error level is reported at the end of transfer (see B.3, “Priority of Error Reporting” on page 263). Retries are attempted but ECC is not applied. Recovered data (if any) is transferred with <i>Check Condition Status</i> and Recovered Error sense key set at the end of the transfer.	
				soft error	The transfer length is exhausted. Transferred data includes blocks containing recovered errors.
				hard error	Data transfer stops when the unrecoverable error is encountered. The unrecoverable block is not transferred to the Initiator.
1	0	1	1	The highest error level is reported at the end of transfer (see B.3, “Priority of Error Reporting” on page 263). Retries are attempted but ECC is not applied. Recovered data (if any) is transferred with <i>Check Condition Status</i> and Recovered Error sense key set at the end of the transfer.	
				soft error	The transfer length is exhausted. Transferred data includes blocks containing recovered errors.
				hard error	Data transfer stops when the unrecoverable error is encountered. The unrecoverable block is transferred to the Initiator (if data error).
1	1	0	0	The highest error level is reported at the end of transfer (see B.3, “Priority of Error Reporting” on page 263). Retries and error correction are attempted. Recovered and/or corrected data (if any) is transferred with <i>Check Condition Status</i> and Recovered Error sense key set at the end of the transfer.	
				soft error	Data transfer stops on the first soft error detected. The recovered error block is returned to the Initiator.
				hard error	Data transfer stops on the unrecovered error. The error block is not returned to the Initiator.
PER	DTE	DCR	TB	Description	

PER	DTE	DCR	TB	Description	
1	1	0	1	The highest error level is reported at the end of transfer (see B.3, “Priority of Error Reporting” on page 263). Retries and error correction are attempted. Recovered and/or corrected data (if any) is transferred with <i>Check Condition Status</i> and Recovered Error sense key set at the end of the transfer.	
				soft error	Data transfer stops on the first soft error detected. The recovered error block is returned to the Initiator.
				hard error	Data transfer stops on the unrecovered error. The unrecovered error block is returned to the Initiator (if data error).
1	1	1	0	The highest error level is reported at the end of transfer (see B.3, “Priority of Error Reporting” on page 263). Retries are attempted but ECC is not applied. Recovered data (if any) is transferred with <i>Check Condition Status</i> and Recovered Error sense key set at the end of the transfer.	
				soft error	Data transfer stops on the first soft error detected. The recovered error block is returned to the Initiator.
				hard error	Data transfer stops on the unrecovered error. The error block is not returned to the Initiator.
1	1	1	1	The highest error level is reported at the end of transfer (see B.3, “Priority of Error Reporting” on page 2630. Retries are attempted but ECC is not applied. Recovered data (if any) is transferred with <i>Check Condition Status</i> and Recovered Error sense key set at the end of the transfer.	
				soft error	Data transfer stops on the first soft error detected. The recovered error block is returned to the Initiator.
				hard error	Data transfer stops on the unrecovered error. The unrecovered error block is returned to the Initiator (if data error).
PER	DTE	DCR	TB	Description	

The **Read Retry Count** sets a limit on the amount of data recovery procedure (DRP) passes the Target attempts when recovering read errors. One pass through DRP involves executing all steps of DRP. (See B.1, “Data Recovery Procedure for Data Field Errors” on page 253 for a description of one DRP pass.) A Read Retry Count of 00h disables all recovery. Read Retry Count set to 0 inhibits Automatic/Recommend Rewrite/Reallocate. A nonzero value for Read Retry Count causes the Target to attempt up to one DRP pass when a medium error occurs during a read operation. The Read Retry Count is used by the Target for errors associated with the transfer of the Initiator's data for the following commands:

- Read(6)
- Read(10)

For all other commands, the Read Retry Count is unused and treated as 1.

The **Correction Span** is the size, in bits, of the largest data error burst for which data error correction may be attempted. A value of 00h indicates the Target uses the default value.

**Note:** The Mode Select command accepts a value of 30h or 00h in this field.

The **Head Offset Count** specifies in two's-compliment notation an incremental offset position from the track center the heads are moved. A positive value indicates moving in the direction of increasing logical block addresses. A negative value indicates moving in the direction of decreasing logical block addresses. The Head Offset Count of 00h indicates no offset is supported.

The **Data Strobe Offset Count** specifies in two's-compliment notation an incremental offset position to which the recovered data strobe is adjusted. The value of 00h indicates no offset is supported.

The **Write Retry Count** is the number of times the Target attempts its retry algorithm during write operations. The value of 01h is the only valid value for this field.

The **Recovery Time Limit** specifies in increments of one millisecond the maximum time duration the Target uses for data error recovery procedures. A recovery time limit of 0000h specifies the Target uses the default value of no time limit.

## 1.6.4 Page 2 - Disconnect/Reconnect Parameters

Table 32. Mode Select Data Format Page 2									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	RSVD = 0	RSVD = 0	Page Code = 02h						
1	Page Length = 0Eh								
2	Read Buffer Full Ratio								
3	Write Buffer Empty Ratio								
4	(MSB)	Bus Inactivity Limit = 0							
5		(LSB)							
6	(MSB)	Disconnect Time Limit = 0							
7		(LSB)							
8	(MSB)	Connect Time Limit = 0							
9		(LSB)							
10	(MSB)	Maximum Burst Size							
11		(LSB)							
12	Reserved = 0					DTDC			
13	Reserved = 0								
14	Reserved = 0								
15	Reserved = 0								

Since bytes 4-9 and 12-15 are not changeable, the Mode Select Command accepts only the values indicated in Table 32 for bytes 4-9 and 12-15.

Following are parameter options for Page 2 of Mode Select. Refer to Table 32.

An Initiator may use the Identify message to grant the Target the general privilege of disconnecting. (Disconnect requests from the target via Disconnect messages may still be selectively rejected by the Initiator by issuing Message Reject).

The Target uses Mode Select Page 2 parameters to control reconnection during Read and Write operations.

**Read Buffer Full Ratio** and **Write Buffer Empty Ratio** are the numerators of a fraction whose denominator is 256. This fraction indicates how full (or empty) the Target's data buffer segment should be prior to attempting to reconnect to the SCSI bus. When applying these ratios, the Target rounds down to a whole buffer block. If the ratio is set to 0h (the default), the Target will calculate and use an optimal ratio based on the negotiated transfer rate (width and period), and the formatted block size. If the ratio is set to FFh, the Target does not attempt to reconnect until the buffer segment is completely full (or empty).

**Reconnecting to a Read Command** For a Read command, the reconnect is delayed relative to the availability of the first block in the Target's data buffer segment by the fraction of the Target's data buffer segment size or the data transfer length, whichever is less.

Example: If Read Buffer Full Ratio is C0h (decimal 192) for a ratio of 0.75, the reconnect

is delayed until the Target's data buffer segment is 3/4 full or until approximately 3/4 of the remaining transfer length is in the Target's data buffer segment (whichever is smaller).

#### Reconnecting to Write Command

For a Write command, the Write Buffer Empty Ratio is significant only if the total data transfer length is greater than the size of the Target's data buffer segment. The fraction determines how empty the Target's data buffer segment should be before reconnecting to begin filling the buffer segment again.

Example: If Write Buffer Empty Ratio is C0h (decimal 192) for a ratio of 0.75, the reconnect is delayed until the Target's data buffer segment is 3/4 empty or until approximately 3/4 of the remaining transfer length is empty in the Target's data buffer segment (whichever is smaller).

#### Choosing Buffer Ratios

For single Initiator/single Target systems, the Initiator should set the buffer ratios to match the SCSI bus instantaneous data transfer rate to the Target's internal data sector transfer rate. (See the *product Functional Specification* for the data rate values.)

$$\text{Buffer Ratio} = 256 \times (\text{Instantaneous SCSI Data Transfer Rate} - \text{Data Sector Transfer Rate}) / \text{Instantaneous SCSI Data Transfer Rate}$$

For systems with SCSI configurations other than single Initiator/single Target, a different ratio may provide better performance.

The **Bus Inactivity Limit** is the maximum time in 100 us increments that the Target is permitted to assert the BSY signal without a REQ/ACK handshake. The value of 00h indicates that there is no limit.

The **Disconnect Time Limit** is the minimum time in 100 us increments that the Target waits after releasing the SCSI bus before attempting reselection. The value of 00h indicates that there is no limit.

The **Connect Time Limit** is the maximum time in 100 us increments that the Target is allowed to use the SCSI bus before disconnecting, if the Initiator has granted the disconnect privilege and it is not restricted by DTDC. The value of 00h indicates that there is no limit.

The **Maximum Burst Size** is the maximum amount of data that the Target transfers during a data phase before disconnecting if the Initiator has granted the disconnect privilege. This value is expressed in increments of 512 bytes (e.g., a value of 0001h means 512 bytes, 0002h means 1024 bytes, etc.). Disconnections attempted by the Target are on block boundaries only. Therefore, the largest number of blocks of data that the Target transfers before disconnecting is the integer portion of the value

of the  $(\text{Maximum Burst Size} \times 512) / (\text{Block Length})$  equation. For the case when  $(\text{Maximum Burst Size} \times 512)$  is less than the Block Length, the Target will transfer 1 block of data before attempting to disconnect.

A value of 0000h indicates there is no limit on the amount of data transferred per connection.

When a nonzero Maximum Burst Size is in effect, the Maximum Burst Size is the basis of the buffer full/empty ratios. For example, if the Maximum Burst Size field is set to 0010h (15 blocks, if Block Length = 520 bytes) and the Read Buffer Full Ratio is set to C0h (ratio of .75) then the reconnect is delayed until approximately 3/4 of the maximum burst (11 blocks) or approximately 3/4 of the remaining transfer length is in the Target's data buffer segment (whichever is smaller).

Regardless of the value in Maximum Burst Size, the Target disconnects prior to completion of the data phase if the internal data buffer segment becomes empty during a Read command or full during a Write command.

The **Data Transfer Disconnect Control (DTDC)** field defines further restrictions on when a disconnect is permitted.

A value of 00b indicates that DTDC is not used by the Target and the disconnect is controlled by the other fields in this page.

A value of 01b indicates that the target shall not attempt to disconnect once the data transfer of a command has started until all data the command is to transfer has been transferred. The connect time limit and bus inactivity limit are ignored during the data transfer.

The value 10b is reserved.

A value of 11b indicates that the target shall not attempt to disconnect once the data transfer of a command has started until the command is complete. The connect time limit and bus inactivity limit are ignored once data transfer has started.

**Note:** If DTDC is nonzero and the maximum burst size is nonzero, a CHECK CONDITION status be returned. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN PARAMETER LIST.



## 1.6.5 Page 3 - Format Device Parameters

Table 33. Mode Select Data Format Page 3									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	RSVD = 0	RSVD = 0	Page Code = 03h						
1	Page Length = 16h (22)								
2	(MSB)	Tracks per Zone							
3		(LSB)							
4	(MSB)	Alternate Sectors per Zone							
5		(LSB)							
6	(MSB)	Alternate Tracks per Zone = 0							
7		(LSB)							
8	(MSB)	Alternate Tracks per Logical Unit = 0							
9		(LSB)							
10	(MSB)	Sectors per Track							
11		(LSB)							
12	(MSB)	Data Bytes per Physical Sector (see note below)							
13		(LSB)							
14	(MSB)	Interleave = 0001h (or 0000h)							
15		(LSB)							
16	(MSB)	Track Skew Factor							
17		(LSB)							
18	(MSB)	Cylinder skew Factor							
19		(LSB)							
20	SSEC = 0	HSEC = 1	RMB = 0	SURF = 0	Reserved = 0				
21	Reserved = 0								
22	Reserved = 0								
23	Reserved = 0								

The format device page contains parameters which specify the medium format.

**Note:** Refer to 1.2, “Format Unit” on page 21 for information on changing parameters prior to issuing the Format Unit Command.

This page contains no changeable parameters. If this page is sent during a Mode Select command with parameter values other than those equal to the Default values for the fields that are not Block Length dependant or Current values of fields that are dependant on Block Length, then the Target reports a *Check Condition Status* with a Sense Key of Illegal Request and an additional Sense Code of Invalid Field in Parameter List. Exceptions to this are for the Sectors per Track, Data Bytes per Physical Sector, Interleave, Track Skew Factor and Cylinder skew Factor fields which may have values of zero in them to indicate that the values used are defined by the Target.

This page contains saved parameters but none of the fields are saveable by a Mode Select command with the SP bit set to 1. (The PS bit of Page 3 is set to a 0 on a Mode Sense command.) Even though the data in this page is not saveable, it is valid to send this page on a Mode Select command with SP set to 1.

The **Tracks per Zone** value is the number of tracks in a cylinder. The Target uses cylinder sparing. This field is a function of the device Model Number. The exact values are defined in the *product Functional Specification*.

The **Alternate Sectors per Zone** a value of 0000h or the value returned by the Mode Sense command may be used. (This is really the average number of alternate sectors per zone.)

The **Alternate Tracks per Zone** value of 0000h indicates that this is Target specific. (Different zones can contain different amounts of alternate tracks.)

The **Alternate Tracks per Logical Unit** value of 0000h indicates that this is Target specific.

**Sectors per Track** specifies the number of physical sectors included within each track. This number includes any alternate sectors that may have been allocated and any defective sectors that may have been found and marked defective. This field is a function of the current Block Length. The exact values can be calculated using the formula for "sectors/track" as defined in the *product Functional Specification*. A value of zero is also be accepted in this field.

**Data Bytes per Physical Sector** specifies the number of user data bytes per physical sector. The value depends upon the current formatted Block Length. The exact values can be calculated using the formula for "user bytes/sector" as defined in the *product Functional Specification*. A value of zero is also be accepted in this field.

**Interleave** values of 1 or 0 are valid. No other interleave is provided.

**Track Skew Factor** indicates 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. This field is a function of the current Block Length. The exact values can be calculated using the formula for "track skew" as defined in the *product Functional Specification*. A value of zero is also accepted in this field.

**Cylinder Skew Factor** can be calculated using the formula for "cylinder skew @OD" as defined in the *product Functional Specification*. The value is a function of current Block Length.

**Note:** Actual cylinder skew values are not the same across entire disk. The value returned is for outermost cylinders only. A value of zero is also accepted in this field. The value for unsynchronized spindle motors is always returned. For more information about the actual formatted cylinder skew see 4.5, "Motor Synchronization" on page 208 since the definition of cylinder skew depends upon which of two skewing methods was selected via the Mode Parameter Page 4h RPL field when the last Format Unit command was executed.

**Byte 20** contains the following fields -

- SSEC The SSEC bit is set to 0 indicating that the Target does not support soft sector formatting.
- HSEC The HSEC bit is set to 1 indicating that the Target uses hard sector formatting.
- RMB The RMB (Removable Media Bit) is set to 0 indicating that the Target does not support removable media.

**SURF** The SURF (Surface) bit set to 0 indicates that the Target allocates progressive addresses to all sectors within a cylinder prior to allocating sector addresses to the next cylinder.

## 1.6.6 Page 4h - Rigid Disk Drive Geometry Parameters

Table 34. Mode Select Data Format Page 4h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	RSVD = 0	RSVD = 0	Page Code = 04h						
1	Page Length = 16h								
2	(MSB)	Number of Cylinders (LSB)							
3									
4									
5	Number of Heads								
6	(MSB)	Starting Cylinder-Write Precompensation = 0 (LSB)							
7									
8									
9	(MSB)	Starting Cylinder-Reduced Write Current = 0 (LSB)							
10									
11									
12	(MSB)	Drive Step Rate = 0 (LSB)							
13									
14	(MSB)	Landing Zone Cylinder = 0C08 h (Decimal 3080) (LSB)							
15									
16									
17	Reserved = 0					RPL			
18	Rotational Offset								
19	Reserved = 0								
20	(MSB)	Medium Rotation Rate = 1518 h (5400) (LSB)							
21									
22	Reserved = 0								
23	Reserved = 0								

This page describes some physical attributes of the Drive. The RPL field in byte 17 and Rotational Offset field in byte 18 are changeable.

**Number of Cylinders** defines the number of physical cylinders used for customer data storage. This field is a function of the device Model Number. The exact values are defined in the *product Functional Specification*. A value of zero is also accepted in this field.

**Number of Heads** defines the number of physical heads used for customer data storage. This field is a function of the device Model Number. The exact values are defined in the *product Functional Specification*. A value of zero is also accepted in this field.

**Starting Cylinder-Write Precompensation** indicates the physical cylinder at which write precompensation begins. The value is set to 000000h indicating that the field is vendor specific.

**Starting Cylinder-Reduced Write Current** indicates the outer-most physical cylinder to be written with reduced write current. The value is set to 000000h indicating that the field is vendor specific.

**Drive Step Rate** is set to 0000h to indicate there is no stepper motor.

The **Landing Zone Cylinder** value indicates that the landing zone is located hundreds of tracks closer to the center of the disk than the inner most data track. A value of zero is also accepted in this field.

The **RPL (Rotational Position Locking)** bits are used to select spindle synchronization modes as defined in Table 35. When the RPL field is modified as a result of a Mode Select command, the physical spindle synchronization state does not change. In order for the new RPL value to take affect (change spindle synchronization state) a start-up sequence initiated via a Start/Stop Unit SCSI command or Auto Start pin being grounded must be executed. See the command description of 1.27, “Start/Stop Unit” on page 132 and section describing 4.5, “Motor Synchronization” on page 208 for more details of how and when spindle synchronization states are altered.

Mode	RPL Bits (bit1/bit0)	Drive/Release Master Sync	Drive/Receive Slave Sync	Is Rotational Offset allowed ?
No Sync	00	release	(see note 1)	no
Slave Sync	01	release	receive	yes
Master Sync	10	Drive	Drive	no
Master Sync Control	11	Drive	receive	no

**Note:** 1) The hardware is placed in a 'receive' state but the signal is not used by the Drive.

Following are definitions of the various modes.

- | Mode -                     | Indicates -  |
|----------------------------|--|
| <b>No Sync</b>             | spindle synchronization is disabled.   |
| <b>Slave Sync</b>          | spindle synchronization is attempted by receiving the Slave Sync signal as the source signal driven by another device. The Master Sync signal is not driven.   |
| <b>Master Sync</b>         | spindle synchronization is not attempted by this device. This device drives the Slave Sync and Master Sync signals with a pulse once per revolution (Index pulse).   |
| <b>Master Sync Control</b> | spindle synchronization is attempted by receiving the Slave Sync signal as the source signal driven by another device. The Master Sync signal is driven with a pulse once per revolution (Clock generated pulse, NOT the Index pulse). |

See the *product Functional Specification* for descriptions of the electrical and mechanical aspects of the two control signals, Slave Sync and Master Sync, and the connectors used to perform spindle synchronization.

The **Rotational Offset** value is the amount of rotational skew that the Target uses when synchronized. The rotational skew is applied in the retarded direction (lagging the synchronized spindle master control). The value in the field is the numerator of

a fractional multiplier that has 256 as its denominator (e.g., a value of 128 indicates a one-half revolution skew). A value of 00h indicates that rotational offset is not used. The rotational offset is only used when the Drive is running in the Slave Sync RPL mode.

The **Medium Rotation Rate** value of 5400 indicates a spindle speed of approximately 5400 rpm. A value of zero is also accepted in this field.

## 1.6.7 Page 7h - Error Recovery Parameters for Verify

Table 36. Mode Select Data Format Page 7h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	RSVD = 0	RSVD = 0	Page Code = 07h						
1	Page Length = 0Ah								
2	Reserved = 0			EER=0	PER	DTE=0	DCR		
3	Verify Retry Count = 01h								
4	Verify Correction Span = 00h								
5	Reserved = 0								
6	Reserved = 0								
7	Reserved = 0								
8	Reserved = 0								
9	Reserved = 0								
10 11	(MSB)	Verify Recovery Time Limit = 0						(LSB)	

Following are parameter options for Page 7 of Mode Select. Refer to Table 36.

Page 7 parameters are used by the Target when recovering from and reporting errors associated with the verification of the Initiator's Data for the following commands:

- *Verify*
- *Write and Verify* - the verify portion of the command only.

Since bytes 3-11 are not changeable, the Mode Select Command accepts only the values indicated for bytes 3 - 11.

**Byte 2** contains the following fields -

**EER** This bit is 0 since the Target does not support early recovery.

**PER** See below for description of bit values.

**DTE** This bit is 0 since the Target always continues on recovered verify operation errors.

**DCR** See below for description of bit values.

PER, DTE, and DCR bit settings in page 7 override those of page 1 during *Verify* and the Verify portion of *Write and Verify*. There are only four valid conditions for the PER, DTE, and DCR bits. All other combinations return *Check Condition Status*.

PER	DTE	DCR	Description
0	0	0	Soft errors are not reported. DCR is ignored.
1	0	0	Soft errors are reported. DCR is ignored.
0	0	1	Soft errors are not reported. DCR is ignored.
1	0	1	Soft errors are reported. DCR is ignored.

Soft errors occur when the data is successfully read by one of the appropriate retry steps described in B.1.2, “Verify Commands” on page 260. Also see the description of 1.30, “Verify” on page 136 and 1.33, “Write and Verify” on page 140 for more details on how soft and hard errors are defined for the *Verify* and *Write and Verify* commands.

**Verify Retry Count** sets a limit on the amount of verify recovery procedure (VRP) passes the Target attempts when recovering verify errors. One pass through VRP involves executing all steps of VRP. (See B.1.2, “Verify Commands” on page 260 for a description of one VRP pass.) The Verify Retry Count of one causes the Target to attempt up to one VRP pass when a medium error occurs during a verify operation.

**Verify Correction Span** specifies the size, in bits, of the largest data error burst for which data error correction may be attempted. A value of 00h indicates the Target does not attempt error correction for verify operations.

**Verify Recovery Time Limit** specifies in increments of one millisecond the maximum time duration the Target uses for data error recovery procedures to recover data for an individual block. A recovery time limit of 0000h specifies that the Target uses its default value.



## 1.6.8 Page 8h - Read Caching Parameters

Table 37. Mode Select Data Format Page 8h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	RSVD = 0	RSVD = 0	Page Code = 08h						
1	Page Length = 12h								
2	Reserved = 0				WCE	MF	RCD		
3	Demand Read Retention Priority			Write Retention Priority					
4	(MSB)	Disable Pre-fetch Transfer Length							
5		(LSB)							
6	(MSB)	Minimum Pre-fetch = 0							
7		(LSB)							
8	(MSB)	Maximum Pre-fetch							
9		(LSB)							
10	(MSB)	Maximum Pre-fetch Ceiling							
11		(LSB)							
12	RSVD = 0	RSVD = 0	RSVD = 0	Reserved = 0					
13	Number of Cache Segments								
14	(MSB)	Reserved = 0							
15		(LSB)							
16	Reserved = 0								
17	(MSB)	Reserved = 0							
18		Reserved = 0							
19		(LSB)							

Page 8 parameters are used for defining the use of the cache. See 4.9, “Segmented Caching” on page 214 for a more detailed description of cache support. Since bytes 6 and 7 are not changeable, the Mode Select Command accepts only the values indicated in Table 37 for bytes 6 and 7.

The use of the cache is also altered by the setting of Mode Select Page 0, ADC (Adaptive Caching), bit.

**Byte 2** contains the following fields -

**WCE** The WCE (Write Cache Enable) bit of 0 specifies that the Target returns *Good Status* for a Write command only after successfully writing all of the data to the medium. The WCE bit of 1 specifies that the Target may return *Good Status* for a Write command after successfully receiving the data but before writing the data to the medium.

**Note:** When Write Cache is enabled (WCE = 1), a Synchronize Cache command must be done to assume data is written to the media before powering down the the Target.

**MF** The MF (Multiplication Factor) bit determines how the Maximum Pre-fetch parameter is interpreted. If this bit is equal to 0, the parameter is used as is. If the bit is equal to 1, the parameter is multiplied by the number of blocks requested in the Read Command.

**RCD** An RCD (read cache disable) bit of 0 indicates that the Target may return some or all of the data requested by a Read (6) or Read (10) command by accessing the data buffer, not the media (see 4.9, “Segmented Caching” on page 214). An RCD bit of 1 indicates that the Target does not return any of the data requested by a Read (6) or Read (10) command by accessing the data buffer. Rather, all of the data requested is read from the media.

**Demand Read Retention Priority** sets the Retention Priority of data requested on a Read Command. It may be set to 0h, 1h, or 0Fh as defined below.

<b>Value</b>	<b>Definition</b>
<b>0h</b>	Do not distinguish between Requested Data and Other Data.
<b>1h</b>	Replace Requested Data before Other Data.
<b>0Fh</b>	Replace Other Data before Requested Data.

Where the Value is the Demand Read Retention Priority or Write Retention Priority. Requested Data is the blocks specified in the Read or Write Command. Other Data is data in the cache from any other source (Pre-fetch, Read-Ahead, ...).

If the Read Retention Priority is not set to Fh or if the DPO bit on the Read command is 1, the Requested Data is overwritten by Read-Ahead data.

If the DPO bit is 0 and the Read Retention Priority is set to Fh, the Requested Data is not overwritten with Read-Ahead Data. If the requested transfer is larger than the segment, the Requested Data is overwritten with more Requested Data and there is no Read-Ahead.

**Write Retention Priority** sets the Retention Priority of data provided on a Write Command. It may be set to 0h, 1h, or 0Fh. See definition of Demand Read Retention Priority above for more details.

**Disable Pre-fetch Transfer Length** is used to prevent Read-Ahead after Read Commands that are longer than the specified number of blocks. If this parameter is set to 0, a Read-Ahead is not performed.

**Minimum Pre-fetch** is used to set a lower limit on the number of blocks to Read-Ahead after a Read Command. The value of 0000h indicates that pre-fetching is terminated whenever another command is ready for executing.

**Maximum Pre-fetch** is used to set an upper limit on the number of blocks to Read-Ahead after a Read Command. Other factors, such as segment size, Drive size, retention priorities, commands in the queue, the value of Page 0, ADC (Adaptive Caching) bit, and new commands may also limit the Read-Ahead.

**Maximum Pre-fetch Ceiling** limits the product of the Maximum Pre-fetch and the number of blocks requested in the Read Command.

The **Number of Cache Segments** field is used to select the number of data buffer cache segments. The value entered in this field is rounded down to the nearest value of each segments supported by this Drive. The number of each segments supported by this Drive are 1, 2, 4, 8, 16, and 32.

All segments are of equal size. The buffer space is divided among the segments. (See the *product Functional Specification* for a description of the total buffer and individual buffer segment sizes.)

If Mode Select, Page 0h, ADC bit is on, the Target will default to using 8 cache segments, and may alter the number of segments to optimize drive performance.

**Note:** Mode Select commands that transfer Page 8h cause the entire cache to be emptied (see 4.9, “Segmented Caching” on page 214).

## 1.6.9 Page 0Ah - Control Mode Parameters

Table 38. Mode Select Data Format Page 0Ah									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	RSVD = 0	RSVD = 0	Page Code = 0Ah						
1	Page Length = 06h								
2	Reserved = 0						RLEC = 0		
3	Queue Algorithm Modifier			Reserved = 0		QErr	DQue		
4	EECA = 0	Reserved = 0			RAENP - 0	UAAENP - 0	EAENP - 0		
5	Reserved = 0								
6	(MSB)	Ready AEN Holdoff Period = 0							
7		(LSB)							

The Control Mode page (Table 38) provides controls over several features. The features are tagged queuing (see 4.3.2, “Tagged Queuing” on page 205), extended contingent allegiance, asynchronous event notification, and error logging (see 4.6, “Error Logs” on page 211).

Since bytes 2,4-7 are not changeable, the Mode Select Command accepts only the values indicated in Table 38 for bytes 2,4-7.

**RLEC (Report Log Exception Condition)** bit of 0 specifies that the Target does not report log exception conditions.

**Queue Algorithm Modifier** specifies restrictions on the algorithm used for re-ordering commands that are tagged with the SIMPLE QUEUE TAG message. The value of zero in this field indicates that the file will reorder the execution sequence of queued commands from each initiator such that data integrity is maintained for that initiator. This means that if the transmission of new commands was halted at any time, the final value of all data observable on the medium shall have exactly the same value as it would have if the commands had been executed in the same received sequence without tagged queuing. The restricted reordering value is the default value.

A value of one in the Queue Algorithm Modifier field specifies that the target may reorder the actual execution sequence of the queue in any manner it selects. Any data integrity exposures related to command sequence order are explicitly handled by the initiator through the selection of appropriate commands and queue tag messages.

A value of eight (8) specifies that no reordering be done.

**QErr (Queue Error management)** bit of 0 specifies that the Target suspends execution of queued and active commands from any Initiator which receives a Check Condition Status until the pending sense data is cleared. Those commands still queued or suspended after the after the Target has returned Check Condition Status, continue execution in a normal manner when the pending sense has been cleared.

A QErr bit of 1 specifies that the Target aborts queued and active commands when the Target returns Check Condition Status. An active Start Unit command or an active Format Unit command are not aborted and continue to execute when an Initiator returns Check Condition Status. The setting of the QEMC bit (see 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59) determines if the commands from all Initiators or just the Initiator receiving the Check Condition Status are aborted.

When the QErr bit is 1 and the QEMC bit is 0, all active commands and all queued commands from the Initiator receiving the Check Condition Status are aborted. Commands from other Initiators are unaffected.

When the QErr bit is 1 and the QEMC bit is 1, all active commands (except Start Unit and Format Unit) and all queued commands from all Initiators are aborted when the Target returns Check Condition Status. When this condition occurs a Unit Attention will be generated for each Initiator that had a I/O process aborted except for the Initiator that received the Check Condition Status. The sense Key will be set to UNIT ATTENTION and the additional sense code will be set to COMMANDS CLEARED BY ANOTHER INITIATOR.

**DQue (Disable Queuing)** bit of 0 specifies that tagged queuing is enabled. A DQue bit of 1 specifies that tagged queuing is disabled. Any queued commands for that I\_T\_L nexus are aborted. No status is sent for aborted commands. Any subsequent queue tag message received are rejected with a MESSAGE REJECT message and the I/O process is executed as an untagged command.

**EECA (Enable Extended Contingent Allegiance)** bit of 0 specifies that extended contingent allegiance is disabled.

The **RAENP**, **UAAENP**, and **EAENP** bits enable specific events to be reported via the asynchronous event notification protocol. Since the Target does not create asynchronous event notifications, all three bits are always 0.

**Ready AEN Holdoff Period** specifies the minimum time in milliseconds after the Target starts its initialization sequence that it delays before attempting to issue an asynchronous event notification. The value of 0000h indicates that this field is not supported (because asynchronous event notification is not supported).

## 1.7 Mode Sense (6)

Table 39. Mode Sense Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 1Ah							
1	LUN			RSVD = 0	DBD	Reserved = 0		
2	PC		Page Code					
3	Reserved = 0							
4	Allocation Length							
5	VU = 0		Reserved = 0			Flag	Link	

The Mode Sense command provides a means for the Target to report various device parameters to Initiators. It is the complement to the Mode Select command.

A DBD (Disable Block Descriptor) bit of zero indicates that the target may return zero or more block descriptors in the returned MODE SENSE data at the target's discretion. A DBD bit of one specifies that the target shall not return any block descriptors in the returned MODE SENSE data.

PC (Page Control) defines the type of Page Parameter values to be returned.

Page Code indicates which page(s) to return.

Page Code of 3Fh is a request for the Target to return all supported pages. Otherwise, a single page may be selected. If an unsupported page is selected, the command is terminated with a *Check Condition Status* and available sense of ***Illegal Request/Invalid Field in CDB***.

If a Page Code of 3Fh is used, Mode Sense returns the pages in ascending order with one exception. Page 0 is always returned last in response to a *Mode Sense* command.

The Allocation Length field indicates the maximum number of bytes the Initiator has set aside for the Data In phase. A value of zero is not considered an error. If the allocation length is smaller than the number of bytes of data the target has available to send to the initiator, then that portion of the data up to the allocation length is sent. This may result in only a portion of a multiple byte field being sent.

**Note:** An Initiator may issue the Mode Sense command to determine Mode Select parameters that are:

- Current values
- Changeable values
- Default values
- Saved values

Fields not supported by the Target have values set to zero.

The types of parameters returned for the different setting of the PC bits are as follows:

## PC Meaning

### 0 0 Report current values.

The definition of current values changes as the Drive progresses from power-up through command execution. The following list defines the various sources of current values:

- Initially following power-up but before the media is accessed, the default values become current.
- Once the media can be accessed, the saved values are read from the Reserved Area and become current.

**Note:** The Target does not process the Mode Select command until the completion of start-up sequence. Therefore, an Initiator cannot modify the current values prior to the saved values being read from the Reserved Area, unless a problem prohibited the saved parameters from being read successfully.

- Following the completion of start-up, execution of the Mode Select command can modify the current values.

**Note:** Those parameters associated with format are not considered current and are not saved until the successful completion of a "Format Unit Command".

- In addition, the current values take on the saved values after a reset if the parameters were saved.

If the Page Code is 3Fh, then all pages implemented by the Target are returned to the Initiator with fields and bit values set to current values.

If the Page Code is not 3Fh, the page defined by the Page Code, if supported by the Target, is returned with fields and bits set to current values.

### 0 1 Report Changeable Values.

Reporting changeable values presents a bit mask for Mode parameter bytes (i.e. bytes following the page length field) indicating which fields are changeable. For a value field such as the buffer ratios in page 2, the bit field does not indicate the range of supported values but rather that the field is simply supported.

If the Page Code is 3Fh, all pages implemented by the Target are returned with bits and fields that may be changed by the Initiator set to one.

If the Page Code is not 3Fh, the page defined by Page Code, if supported, is returned with bits and fields that may be changed by the Initiator set to one.

**Note:** No field in Page 3 is changeable, but certain fields implicitly have varying values depending on the current formatted Block Length.

### 1 0 Report Default Values

The default values do not require a media access. See 1.6, "Mode Select (6)" on page 55 for the Target's default values.

If the Page Code is 3Fh, all pages implemented by the Target are returned with bits and fields set to default values.

If the Page Code is not 3Fh, the page defined by Page Code (if supported) is returned with bits and fields set to default values.

## 1.1 Report Saved Values

The saved values reside in the Reserved Area of the Drive.

Returned values can vary depending upon whether or not those parameters can be successfully read from the Reserved Area or if they have ever been saved. The possible saved values returned are either

- Values saved in the Reserved Area during the last Mode Select command that had the SP bit on, or
- Values identical to the default values if an error prohibits the saved parameters from being read from the Reserved Area of the disk during a start-up sequence, or
- Values identical to the default values if never saved.

If the Page Code is 3Fh, all pages implemented by the Target are returned with bits and fields set to their saved values.

If the Page Code is not 3Fh, the page defined by Page Code, if supported, is returned with bits and fields set to their saved values.

**Note:** When Page 3 saved values are requested, the Target will return parameter values equivalent to the current values.

### 1.7.1 Mode Sense Data Format (of Header & Block Descriptor)

The following diagram depicts the format of the Mode Sense data available from the Target.

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Sense Data Length							
1	Medium Type = 0							
2	WP	Reserved = 0		DPOFUA - 1	Reserved = 0			
3	Block Descriptor Length = 08h							

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Density Code = 0							
1	(MSB) Number of Blocks (LSB)							
2								
3								
4	Reserved = 0							
5	(MSB) Block Length (LSB)							
6								
7								

See the text describing 1.6.1.1, “Header” on page 57 for a description of the Medium Type and WP fields.



The DPOFUA bit value of 1 indicates that the Target supports the FUA and DPO bits in the Read and Write Commands. See 4.9, “Segmented Caching” on page 214 for detailed information on this support.

The Block Descriptor Length field value of 8 indicates that only a single block descriptor follows.

The Number of Blocks field returns a different value depending on what Block Length the media is currently formatted to. Refer to the product Functional Specification for the exact value.

The Block Length field is dependant on how the media is currently formatted.

## 1.7.2 Mode Sense Data Format (of All Pages)

Table 42. Mode Sense Data Page 00h								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	PS = 1	RSVD = 0	Page Code = 0					
1	Page Length = 0Ah							
2	QPE	UQE	DWD	Reserved = 0				
3	ASDPE	RSVD = 0	CMDAC	RPFAE	RSVD = 0		CPE	
4	Reserved = 0							
5	TCHD	TCC	DSN	FRDD	DPSDP	WPEN	CAEN	LITF
6	Reserved = 0							
7	Reserved = 0							
8	Reserved = 0		QEMC	DRD	LED Mode			
8	RSVD = 0	ADC	QEMC	DRD	LED Mode			
9	Reserved = 0							
10	Command Aging Time Limit (HI)							
11	Command Aging Time Limit (LO)							

Table 43. Mode Sense Data Page 01h								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	PS = 1	RSVD = 0	Page Code = 01h					
1	Page Length = 0Ah							
2	AWRE	ARRE	TB	RC	EER = 0	PER	DTE	DCR
3	Read Retry Count							
4	Correction Span = 30h							
5	Head Offset Count = 0							
6	Data Strobe Offset Count = 0							
7	Reserved = 0							
8	Write Retry Count = 01h							
9	Reserved = 0							
10	(MSB)	Recovery Time Limit = 0						
11		(LSB)						

Table 44. Mode Sense Data Page 02h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	PS = 1	RSVD = 0	Page Code = 02h						
1	Page Length = 0Eh								
2	Read Buffer Full Ratio								
3	Write Buffer Empty Ratio								
4	(MSB)	Bus Inactivity Limit = 0							
5		(LSB)							
6	(MSB)	Disconnect Time Limit = 0							
7		(LSB)							
8	(MSB)	Connect Time Limit = 0							
9		(LSB)							
10	(MSB)	Maximum Burst Size							
11		(LSB)							
12	Reserved = 0						DTDC		
13	Reserved = 0								
14	Reserved = 0								
15	Reserved = 0								

Table 45. Mode Sense Data Page 03h								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	PS = 0	RSVD = 0	Page Code = 03h					
1	Page Length = 16h							
2 3	(MSB)	Tracks per Zone (LSB)						
4 5	(MSB)	Alternate Sectors per Zone (LSB)						
6 7	(MSB)	Alternate Tracks per Zone = 0 (LSB)						
8 9	(MSB)	Alternate Tracks per Logical Unit = 0 (LSB)						
10 11	(MSB)	Sectors per Track (LSB)						
12 13	(MSB)	Data Bytes per Physical Sector (LSB)						
14 15	(MSB)	Interleave = 0001h (LSB)						
16 17	(MSB)	Track Skew Factor (LSB)						
18 19	(MSB)	Cylinder skew Factor (LSB)						
20	SSEC = 0	HSEC = 1	RMB = 0	SURF = 0	Reserved = 0			
21	Reserved = 0							
22	Reserved = 0							
23	Reserved = 0							

Table 46. Mode Sense Data Page 04h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	PS = 1	RSVD = 0	Page Code = 04h						
1	Page Length = 16h								
2	(MSB)								
3	Number of Cylinders								
4	(LSB)								
5	Number of Heads								
6	(MSB)								
7	Starting Cylinder-Write Precompensation = 0								
8	(LSB)								
9	(MSB)								
10	Starting Cylinder-Reduced Write Current = 0								
11	(LSB)								
12	(MSB)								
13	Drive Step Rate = 0								
14	(MSB)								
15	Landing Zone Cylinder = 0C08 h (Decimal 3080)								
16	(LSB)								
17	Reserved = 0						RPL		
18	Rotational Offset								
19	Reserved = 0								
20	(MSB)								
21	Medium Rotation Rate = 1518 h (5400)								
22	(LSB)								
22	Reserved = 0								
23	Reserved = 0								

Table 47. Mode Sense Data Page 07h									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	PS = 1	RSVD = 0	Page Code = 07h						
1	Page Length = 0Ah								
2	Reserved = 0				EER=0	PER	DTE=0	DCR	
3	Verify Retry Count = 01h								
4	Verify Correction Span = 0								
5	Reserved = 0								
6	Reserved = 0								
7	Reserved = 0								
8	Reserved = 0								
9	Reserved = 0								
10	(MSB)								
11	Verify Recovery Time Limit = 0								
11	(LSB)								

Table 48. Mode Sense Data Page 08h								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	PS = 1	RSVD = 0	Page Code = 08h					
1	Page Length = 12h							
2	Reserved = 0					WCE	MF	RCD
3	Demand Read Retention Priority				Write Retention Priority			
4	(MSB)	Disable Pre-fetch Transfer Length						
5		(LSB)						
6	(MSB)	Minimum Pre-fetch = 0						
7		(LSB)						
8	(MSB)	Maximum Pre-fetch						
9		(LSB)						
10	(MSB)	Maximum Pre-fetch Ceiling						
11		(LSB)						
12	RSVD = 0	RSVD = 0	RSVD = 0	Reserved = 0				
13	Number of Cache Segments							
14	(MSB)	Reserved = 0						
15		(LSB)						
16	Reserved = 0							
17	(MSB)	Reserved = 0						
18		Reserved = 0						
19		(LSB)						

Table 49. Mode Sense Data Page 0Ah								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	PS = 1	RSVD = 0	Page Code = 0Ah					
1	Page Length = 06h							
2	Reserved = 0							RLEC = 0
3	Queue Algorithm Modifier				Reserved = 0		QErr	DQue
4	ECCA = 0	Reserved = 0				RAENP = 0	UAAENP = 0	EAENP = 0
5	Reserved = 0							
6	(MSB)	Ready AEN Holdoff Period = 0						
7		(LSB)						

The Parameters Saveable (PS) bit is set to 1 for pages 0, 1, 2, 4, 7, 8, and A to indicate that some of the parameters of those pages can be saved.

The Parameters Saveable (PS) bit is set to 0 for page 3 to indicate that none of the parameters of that page can be saved with a Mode Select command. They can, however, change as a result of a Format command.

The Page Length byte value of each page returned by the Target indicates the number of Mode parameter bytes returned. Bytes 1 and 0 are not included.

The bit descriptions for all of the fields of all of the pages are described in the Mode Select section (see 1.6, “Mode Select (6)” on page 55).

## 1.8 Pre-Fetch

Table 50. Pre-Fetch Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 34h							
1	LUN			Reserved = 0			Immed	RelAdr = 0
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved = 0							
7	(MSB) Transfer Length (LSB)							
8								
9	VU = 0			Reserved = 0			Flag	Link

The Pre-Fetch command requests the target to transfer the specified logical blocks to the cache. No data is transferred to the initiator.

The transfer length field specifies the number of contiguous blocks of data that are to be transferred into the cache. A transfer length of zero indicates that blocks are to be transferred into the cache until the segment is filled or there are no more blocks on the media.

If the Immediate (Immed) bit of the CDB is zero:

- If an error occurs while reading, error recovery procedures are attempted. The Drive returns *Good Status* or *Check Condition Status* based on the setting of the MODE SELECT Page 1 parameters.
- If there is enough room in the segment for all of the Requested Data or if the Transfer Length is zero and no error occurs while reading, the Drive returns *Condition Met Status* when the command completes.
- If there is not enough room in the segment, the Transfer Length is not zero, and no error occurred while reading, the Drive returns *Good Status* when the command completes.

If the Immediate (Immed) bit of the CDB is one:

- If there is enough room in the segment for all of the Requested Data or if the Transfer Length is zero, the Drive returns *Condition Met Status* as soon as the CDB is verified.
- If there is not enough room in the segment and the Transfer Length is not zero, the Drive returns *Good Status* as soon as the CDB is verified.
- The reading of data is handled the same as Read-Ahead operation. This implies the prefetch may be terminated upon receipt of another command, (see 4.9.3, “Read-Ahead” on page 214) for more details.
- If an error is encountered:
  1. The Target terminates the Pre-Fetch operation and does not attempt to recover the data.



2. The error is not reported to the Initiator for the current command. (The error will be reported during the next command if the next command is a Read command that requests the block which encountered the error.)
3. The blocks which were successfully read prior to the block in error are retained in the cache.

The RelAdr (relative address) bit must be zero. The Drive does not support relative addressing. If RelAdr is set to 1, *Check Condition Status* is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.

## 1.9 Read (6)

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 08h							
1	LUN			(MSB)	LBA			
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	VU = 0		Reserved = 0				Flag	Link

The Read (6) command requests that the Target transfer the specified number of blocks of data to the Initiator starting at the specified logical block address.

The logical block address specifies where the read operation is to begin.

Transfer length may be a value from 0 to 255 where 0 implies a length of 256 blocks.

Errors are handled by ERP (error recovery procedures) (see B.1, “Data Recovery Procedure for Data Field Errors” on page 253). ERPs are controlled by the error recovery parameters (see 1.6.3, “Page 1 - Error Recovery Parameters” on page 66).

**Note:** The 6 byte command has only 21 bits for address for LBAs. The 21 bits allows for  $2 \times 10^{(21)}$  lbas (2,097,152). The 10 byte command should be used to read LBAs which require more bits. At 512 byte blocks, this only allows access to 1.048 Gigabytes. The 10 byte command must be used to read lbas which require more bits.

## 1.10 Read (10)

Table 52. Read (10) Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 28h							
1	LUN		DPO	FUA	Reserved = 0		RelAdr = 0	
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved = 0							
7	(MSB) Transfer Length (LSB)							
8								
9	VU = 0		Reserved = 0			Flag	Link	

The Read (10) command requests that the Target transfer the specified number of blocks of data to the Initiator starting at the specified logical block address.

The logical block address specifies where the read operation is to begin.

A DPO (Disable Page Out) bit of 1 indicates that the data accessed by this command is to be assigned the lowest priority for being written into or retained by the cache. A DPO bit of 1 overrides any retention priority specified in the Mode Select Page 8 Caching Parameters. A DPO bit of 0 indicates the priority is determined by the retention priority. See 1.6.8, “Page 8h - Read Caching Parameters” on page 85 and 4.9, “Segmented Caching” on page 214.

The initiator should set the DPO bit when the blocks read by this command are not likely to be read again in the near future.

A FUA (Force Unit Access) bit of 1 indicates that the data is read from the media and not from the cache. A FUA bit of 0 allows the data to be read from either the media or the cache.

The RelAdr (Relative Address) bit must be 0. Relative addressing is not supported. If RelAdr is set to 1, Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.

If the transfer length is zero, the seek occurs, but no data is transferred. This condition is not considered an error. If read ahead is enabled, a read ahead is started after the seek completes (see 4.9.3, “Read-Ahead” on page 214).

## 1.11 Read Buffer

Table 53. Read Buffer Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 3Ch							
1	LUN			Reserved = 0		Mode		
2	Buffer ID = 0							
3	(MSB) Buffer Offset (LSB)							
4								
5								
6	(MSB) Allocation Length (LSB)							
7								
8								
9	VU = 0		Reserved = 0			Flag	Link	

The Read Buffer command is used in conjunction with the Write Buffer command to test the SCSI bus and target memory. The only valid Modes are:

- 000b Read combined header and data
- 010b Data mode
- 011b Descriptor Mode

If any modes other than shown above are specified, then *Check Condition Status* is returned with a Sense Key of **Illegal Request** and additional sense code of **Invalid Field in CDB**.

- Mode = 000b Read combined header and data

In this mode Read Buffer transfers a four-byte header and the specified amount of data from the data buffer.

Byte 0 of the header is reserved. Bytes 1 through 3 contain the buffer capacity.

The Buffer ID and Buffer Offset fields must be zero. Receipt of a non-zero Buffer ID or a non-zero Buffer Offset results in *Check Condition Status* with sense key of **Illegal Request** and additional Sense Code of **Invalid Field in CDB**.

The Allocation Length includes the length of the header. The target terminates the Data-In phase when the number of bytes specified in the Allocation Length (header plus the data) have been transferred or when the header and all available data have been transferred to the Initiator, whichever is less.

- Mode = 010b Data mode

In this mode Read Buffer transfers the specified amount of data from the data buffer.

The Buffer ID field must be zero. Receipt of a non-zero Buffer ID results in *Check Condition Status* with sense key of **Illegal Request** and additional Sense Code of **Invalid Field in CDB**.

The Buffer Offset field specifies the byte offset within the buffer from which the data will be transferred. The Buffer Offset is required to be less than the buffer capacity. An offset greater than or equal to the buffer capacity results in *Check*

*Condition Status* with Sense key of ***Illegal Request*** and additional Sense Code of ***Invalid Field in CDB***.

**Note:** The buffer capacity can be determined using Read Buffer mode 000b or mode 011b.

The target terminates the Data-In phase when the number of bytes specified in the Allocation Length have been transferred or when all available data (from the specified offset to the end of the buffer) have been transferred to the Initiator, whichever is less.

- Mode = 011b Descriptor mode

In this mode, Read Buffer transfers a four-byte header. Header information is returned for the buffer specified by the buffer ID. If there is no buffer associated with the specified buffer ID, Read Buffer transfers all zeros in the header.

Byte 0 of the header is reserved. Bytes 1 through 3 contain the buffer capacity.

The Buffer Offset fields must be zero. Receipt of a non-zero Buffer Offset results in Check Condition Status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

The allocation length should be set to four or greater. An allocation length of less than four indicates no data is transferred.

## 1.12 Read Capacity

Table 54. Read Capacity Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 25h							
1	LUN				Reserved = 0			
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved = 0							
7	Reserved = 0							
8	Reserved = 0							PMI
9	VU = 0			Reserved = 0			Flag	Link

The Read Capacity command returns information to the Initiator regarding the capacity of the LUN.

A Partial Medium Indicator (PMI) bit of 0 indicates that the information returned is the logical block address (LBA) and block length (in bytes) of the last logical block on the LUN. The LBA bytes of the command descriptor block must be set to zero for this option.

A PMI bit of 1 indicates that the information returned is the LBA and block length (in bytes) of the last logical block on the same cylinder as the LBA specified in the command descriptor block. This option provides an indication of the amount of contiguous space beyond or equal to the LBA specified without the need for a seek.

Following is the format of the data returned to the Initiator in the Data In phase.

Table 55. Format of Read Capacity Data Returned								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB) Logical Block Address (LSB)							
1								
2								
3								
4	(MSB) Block Length (LSB)							
5								
6								
7								

## 1.13 Read Defect Data (10)

Table 56. Read Defect Data Command Descriptor Block							
Byte	BIT						
	7	6	5	4	3	2	1
0	Operation Code = 37h						
1	LUN			Reserved = 0			
2	Reserved = 0			PList	GList	Defect List Format	
3-6	Reserved = 0						
7	(MSB) Allocation Length (LSB)						
8							
9	VU = 0		Reserved = 0			Flag	Link

The Read Defect Data command requests that the Target transfer the medium defect data to the Initiator. The Target does not return defects that exist in the Reserved area.

A PList bit of one requests that the Target return the Primary defect list. A PList bit of zero requests that the Target not return the Primary defect list.

A GList bit of one requests that the Target return the Grown defect list. A GList bit of zero requests that the Target not return the Grown defect list.

A PList bit of one and a GList bit of one requests that the Target return both the Primary and Grown defect lists. The Target returns the Primary list first with defects in ascending order. The Target returns the Grown list second with defects in the order that the Reassign Block command, the Auto-Reassign function, and the Format Unit command (with Defect List) added them to the list.

The Defect List Format field indicates the preferred format of defect data to be returned to the Initiator.

Preferred Defect List Format	Returned Defect List Format
Block (000b)	Physical Sector
Bytes from Index (100b)	Bytes from Index
Physical Sector (101b)	Physical Sector
Vendor Unique (110b)	Physical Sector
Reserved (001b)	Physical Sector
Reserved (010b)	Physical Sector
Reserved (011b)	Physical Sector
Reserved (111b)	Physical Sector

If the Preferred Defect List Format is not the Returned Defect List Format, the Target transfers the defect data then terminates the command with Check Condition status. The sense key is set to Recovered Error and the additional sense code is set to Defect List Not Found.

The Allocation Length field specifies the maximum number of bytes that the Initiator has allocated for receiving the defect data. An Allocation length of zero is not an error and in this case no defect data will be transferred.

The Read Defect Data defect list contains a four byte header, followed by zero or more defect descriptors.

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0							
1	Reserved = 0			PList	GList	Defect List Format		
2	(MSB) Defect List Length (LSB)							
3								

Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 7	Defect Descriptor 0							
8n - 8n+7	Defect Descriptor n							

If a Requested List is empty then the Target will return Good status. A PList bit of one indicates that the data returned contains the primary defect list. A PList bit of zero indicates that the data returned does not contain the primary defect list.

A GList bit of one indicates that the data returned contains the Grown defect list. A GList bit of zero indicates that the data returned does not contain the Grown defect list.

Defect List Format indicates the format of the defect descriptors returned to the Initiator.

Defect list length specifies the length in bytes of the defect descriptors that follow. The defect list length is equal to eight times the number of defect descriptors.

If the allocation length is insufficient to transfer the four byte header and all of the defect descriptors, the defect list length is not adjusted to reflect the truncation. The Target does not create Check Condition status. The Initiator should compare the defect list length with the allocation length to determine if a partial list was received.

**Note:** An Initiator can determine the length of the defect list by issuing the Read Defect Data command with an allocation length of four. The Target returns the defect list header which contains the length of the defect list.

Normally the target will set the defect list length field to the amount of space needed to contain the entire defect list. However, the Target is capable of building a defect list with a length such that the entire list cannot be transferred using the maximum allocation length. If the defect list grows beyond 8191 entries the defect data cannot be transferred with an allocation length of FFFFh. The Target will transfer a partial defect list and return Check Condition status with the sense key set to Recovered Error and the Additional Sense Code is set to partial defect list transferred. The



defect list length will be set to FFF8h, indicating the maximum number of defect descriptors which can be transferred. Defects beyond this number can not be read by an initiator.

The formats for the defect descriptor are shown in the following tables.

Table 59. Defect Descriptor - Bytes From Index Format								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB) Cylinder Number of Defect (LSB)							
1								
2								
3	Head Number of Defect							
4	(MSB) Defect Bytes From Index (LSB)							
5								
6								
7								

Each defect descriptor for the Bytes From Index format specifies the middle of a sector-sized defect location on the medium. Each defect descriptor is comprised of the cylinder number of defect, the head number of defect, and the defect bytes from index.

The middle byte of a physical sector is calculated as follows:

$$(\text{sector number} * \text{gross bytes/sector}) + (1/2 * \text{gross bytes/sector})$$

Table 60. Defect Descriptor - Physical Sector Format								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB) Cylinder Number of Defect (LSB)							
1								
2								
3	Head Number of Defect							
4	(MSB) Defective Sector Number (LSB)							
5								
6								
7								

Each defect descriptor for the Physical Sector format specifies a sector-size defect location comprised of the cylinder number of defect, the head number of defect, and the defect sector number.

## 1.14 Read Defect Data (12)

Table 61. Read Defect Data Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = B7h							
1	LUN			PList	GList	Defect List Format		
2	Reserved = 0							
3	Reserved = 0							
4	Reserved = 0							
5	Reserved = 0							
6-9	(MSB)	Allocation Length						(LSB)
10	Reserved = 0							
11	VU = 0		Reserved = 0			Flag	Link	

The Read Defect Data command requests that the Target transfer the medium defect data to the Initiator. The Target does not return defects that exist in the Reserved area.

A PList bit of one requests that the Target return the Primary defect list. A PList bit of zero requests that the Target not return the Primary defect list.

A GList bit of one requests that the Target return the Grown defect list. A GList bit of zero requests that the Target not return the Grown defect list.

A PList bit of one and a GList bit of one requests that the Target return both the Primary and Grown defect lists. The Target returns the Primary list first and defects in ascending order. The Target returns the Grown list second and defects in the order that the Reassign Block command, the Auto-Reassign function, and the Format Unit command (with Defect List) added them to the list.

The Defect List Format field indicates the preferred format of defect data to be returned to the Initiator.

Preferred Defect List Format	Returned Defect List Format
Block (000b)	Physical Sector
Bytes from Index (100b)	Bytes from Index
Physical Sector (101b)	Physical Sector
Vendor Unique (110b)	Physical Sector
Reserved (001b)	Physical Sector
Reserved (010b)	Physical Sector
Reserved (011b)	Physical Sector
Reserved (111b)	Physical Sector

If the Preferred Defect List Format is not the Returned Defect List Format, the Target transfers the defect data then terminates the command with Check Condition status. The sense key is set to Recovered Error and the additional sense code is set to Defect List Not Found.

The Allocation Length field specifies the maximum number of bytes that the Initiator has allocated for receiving the defect data. An Allocation length of zero is not an error and in this case no defect data will be transferred.

The Read Defect Data defect list contains a four byte header, followed by zero or more defect descriptors.

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0							
1	Reserved = 0			PList	GList	Defect List Format		
2	Reserved = 0							
3	Reserved = 0							
4-7	(MSB)	Defect List Length						(LSB)

Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 7	Defect Descriptor 0							
8n - 8n+7	Defect Descriptor n							

If a Requested List is empty then the Target will return Good status. A PList bit of one indicates that the data returned contains the primary defect list. A PList bit of zero indicates that the data returned does not contain the primary defect list.

A GList bit of one indicates that the data returned contains the Grown defect list. A GList bit of zero indicates that the data returned does not contain the Grown defect list.

Defect List Format indicates the format of the defect descriptors returned to the Initiator.

Defect list length specifies the length in bytes of the defect descriptors that follow. The defect list length is equal to eight times the number of defect descriptors.

If the allocation length is insufficient to transfer the four byte header and all of the defect descriptors, the defect list length is not adjusted to reflect the truncation. The Target does not create Check Condition status. The Initiator should compare the defect list length with the allocation length to determine if a partial list was received.

**Note:** An Initiator can determine the length of the defect list by issuing the Read Defect Data command with an allocation length of four. The Target returns the defect list header which contains the length of the defect list.

The formats for the defect descriptor are shown in the following tables.

Table 64. Defect Descriptor - Bytes From Index Format								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB) Cylinder Number of Defect (LSB)							
1								
2								
3	Head Number of Defect							
4	(MSB) Defect Bytes From Index (LSB)							
5								
6								
7								

Each defect descriptor for the Bytes From Index format specifies the middle of a sector-sized defect location on the medium. Each defect descriptor is comprised of the cylinder number of defect, the head number of defect, and the defect bytes from index.

The middle byte of a physical sector is calculated as follows:

$$(\text{sector number} * \text{gross bytes/sector}) + (1/2 * \text{gross bytes/sector})$$

Table 65. Defect Descriptor - Physical Sector Format								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB) Cylinder Number of Defect (LSB)							
1								
2								
3	Head Number of Defect							
4	(MSB) Defective Sector Number (LSB)							
5								
6								
7								

Each defect descriptor for the Physical Sector format specifies a sector-size defect location comprised of the cylinder number of defect, the head number of defect, and the defect sector number.

## 1.15 Read Long

Table 66. Read Long Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 3Eh							
1	LUN			Reserved = 0			CORRCT = 0	RelAdr = 0
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved = 0							
7	(MSB) Byte Transfer Length (LSB)							
8								
9	VU = 0		Reserved = 0				Flag	Link

The Read Long command requests that the target transfer data to the Initiator. The data returned by the Read Long command consists of the data bytes followed by the LRC and ECC bytes recorded on the medium.

The CORRCT (corrected) bit must be zero. If CORRCT is set to 1, Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB. The Drive does not attempt data correction. The Drive will however, attempt to retry the command an internally specified number of times.

The RelAdr (relative address) bit must be zero. The Drive does not support relative addressing. If RelAdr is set to 1, Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.

The logical block address field specifies the logical block at which the read operation occurs.

The byte transfer length field must exactly specify the number of bytes of data that are available for transfer. If the value does not exactly match the available data length, the command terminates with Check Condition status, a sense key set to Illegal Request, and an additional sense code set to Invalid Field in CDB. The valid and ILI bits are set to one and the information field is set to the difference (residue) of the requested length minus the actual length in bytes. Negative values are indicated by two's complement notation.

The transfer length is calculated as follows:

$$\text{transfer length} = \text{logical block size} + (20 * n),$$

where n is the number of physical sectors per LBA.

If the byte transfer length is zero, the seek occurs, but no data is transferred. This condition is not considered an error.

The data read by this command is not read from nor retained in the cache.

## 1.16 Reassign Blocks

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 07h							
1	LUN				Reserved = 0			
2	Reserved = 0							
3	Reserved = 0							
4	Reserved = 0							
5	VU = 0		Reserved = 0				Flag	Link

The Reassign Blocks command requests the Target to reassign a Logical Block to an available spare location. The Logical Block Address is transferred to the Target during the Data Out phase.

The physical sector of the LBA in the Reassign Blocks Data Descriptor is reassigned. Additional physical sectors residing in the affected data area may also be reassigned.

All spare physical sectors on each cylinder are located on the last track of that cylinder. As new spares are needed, the next spare in order is used. If all spares on that cylinder are filled, the first available spare of the next cylinder is used. This continues for a maximum of three cylinders beyond the location of the Defective Logical Block Address in the Reassign Blocks Data Descriptor. If there are not enough available spares within three cylinders to reassign the initial and all additional physical sectors, the Drive returns Check Condition and builds sense data for No Defect Spare Location Available.

Execution of this command causes movement of data adjacent and including the LBA specified in the Reassign Block Data Descriptor. The data moved consists of sectors that reside on the track containing the LBA reassigned and all sectors on subsequent tracks required to perform the reassign. A maximum of three consecutive cylinders of data is allowed to be moved. The sequential order of all logical blocks is maintained during this movement.

During the reading of the affected data area, an internal reallocate log is used to determine if additional physical sectors need reassignment. Additional physical sectors determined to need reassignment are reassigned.

Upon successful completion of this command, the location of all physical sectors reassigned during the command is added to the Glist.

If the command is interrupted by a reset or power outage, the command automatically resumes the reassignment when the Drive's Bring-up Sequence is executed or restarted. Refer to 4.18, "Bring-Up Sequence" on page 225 for more information.

If the reassignment fails to complete successfully, the Target enters a degraded mode of operation. Writing to the data area affected is prohibited. For more information on the degraded mode and when reassign operations can get restarted see 4.1.9, "Degraded Mode" on page 190.

Data adjacent to defective logical block is preserved. The target also attempts to preserve the data residing at the Defective Logical Block Address described in the Reassign Blocks Data Descriptor. If the target is unsuccessful in its attempt to preserve the LBA being reassigned, this LBA will be filled with FFh values.

The ID of sectors reassigned is rewritten to flag the physical sector defective and cannot be accessed.

Following is the format of the data sent by the Initiator during the Data Out phase. SCSI bus data phase retries are not supported.

Table 68. Format of Reassign Blocks Data - Header								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0							
1	Reserved = 0							
2	(MSB)	Defect List length (0004h)						
3		(LSB)						

Table 69. Format of Reassign Blocks Data - Descriptor								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB)	Defective Logical Block Address						
1								
2								
3	(LSB)							

The Reassign Blocks defect list contains a 4-byte header followed by one defect descriptor. The Reassign Blocks command requests the reassignment of one logical block per command. Therefore only one defect descriptor is allowed. The length of the defect descriptor is 4 bytes.

## 1.17 Receive Diagnostic

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 1Ch							
1	LUN				Reserved = 0			
2	Reserved = 0							
3	(MSB) Allocation Length (LSB)							
4								
5	VU = 0			Reserved = 0			Flag	Link

The Receive Diagnostic Results command requests that analysis data requested by a Send Diagnostics command be sent to the initiator.

Allocation Length specifies the amount of data to be returned to the initiator. No bytes are transferred if the allocation length is zero. This condition is not considered an error. The target terminates the Data In phase when all available diagnostic data has been transferred or when the number of bytes transferred equals the Allocation length.



## 1.17.1 Receive Diagnostics Page 40h

The translate address page allows the initiator to translate a logical block address or physical sector address to the other format. The address to be translated is passed to the target with the Send Diagnostic command and the results are returned to the initiator by the Receive Diagnostics commands' translate address page.

## 1.17.2 Translate Address Page - Receive Diagnostics

Table 71. Translate Address Page - Receive Diagnostics								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Page Code = 40h							
1	Reserved = 0							
2	(MSB) Page Length (LSB)							
3								
4	Reserved = 0					Supplied Format		
5	Rarea	Altsec	Altrk	Reserved = 0		Translated Format		
6 - 13	Translated Address 1							
n - n + 7	Translated Address n (if required)							

The page begins with a four-byte page header which specifies the page code and length, followed by two bytes which describe the translated address followed by zero or more translated addresses.

The Page Code field is equal to 40h.

The Page Length field specifies the length of the parameter bytes which follow.

The Supplied format field contains the value from the Send Diagnostics command supplied format field. The target supports 000b (Block Format) and 101b (Physical Sector Format).

The Translated format field contains the value from the Send Diagnostics command translate format field. Valid values are 000b (Block Format) and 101b (Physical Sector Format).

A reserved area (Rarea) bit of one indicates that all or part of the translated address falls within a reserved area of the medium. If the Rarea bit is one a translated address will not be returned. An Rarea bit of zero indicates that no part of the translated address falls within a reserved area of the medium.

An alternate sector (Altsec) bit of one indicates that the translated address is physically located in an alternate sector of the medium. An Altsec bit of zero indicates that no part of the translated address is located in an alternate sector of the medium.

An alternate track (Altrk) bit of one indicates that all or part of the translated address is located on an alternate track of the medium. An Altrk bit of zero indicates that no of the translated address is on an alternate track of the medium.

The translated address field contains the result the target obtained by translating the address supplied by the initiator in the previous Send Diagnostic command. This field shall be in the format specified in the translate format field. The Physical

Sector format is described in Table 7 on page 26. If the Logical Block format is specified the block address shall be in the first four bytes of the field and the remaining bytes shall be zero.

If the address to be translated covers more than one address after translation (multiple physical sectors per logical block) the target will return all possible addresses which are contained for the address to be translated.

## 1.18 Release (6)

Table 72. Release Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 17h							
1	LUN			3rdPty	3rd Party ID			Ext = 0
2	Reservation Identification							
3	Reserved = 0							
4	Reserved = 0							
5	VU = 0			Reserved = 0			Flag	Link

The Release command is used to release a LUN previously reserved. It is not an error for an Initiator to attempt to release a reservation that is not currently active. In this case, the Drive returns Good status without altering the reservation.

Extents are not supported by the Drive. The Ext (Extent) bit must be zero and the Reservation Identification field is ignored. If the Ext bit is not zero, *Check Condition Status* is returned with a Sense Key of ***Illegal Request*** and additional sense code of ***Invalid Field in CDB***

If the 3rdPty bit is zero, then the third-party release option is not requested. If the 3rdPty bit is one, then the Target releases the LUN, but only if the reservation was made using the third-party reservation option and the 3rd Party ID is the ID of the initiator that made the reservation. (see 1.21, “Reserve (6)” on page 123) (see 4.1.10, “Command Processing While Reserved” on page 202)

## 1.19 Release (10)

Table 73. Release Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 57h							
1	LUN		3rdPty	Reserved			Ext = 0	
2	Reservation Identification							
3	Third Party Device ID							
4	Reserved = 0							
5	Reserved = 0							
6	Reserved = 0							
7	Reserved = 0							
8	Reserved = 0							
9	VU = 0		Reserved = 0			Flag	Link	

The Release command is used to release a LUN previously reserved. It is not an error for an Initiator to attempt to release a reservation that is not currently active. In this case, the Drive returns Good status without altering the reservation.

Extents are not supported by the Drive. The Ext (Extent) bit must be zero and the Reservation Identification field is ignored. If the Ext bit is not zero, *Check Condition Status* is returned with a Sense Key of ***Illegal Request*** and additional sense code of ***Invalid Field in CDB***

If the 3rdPty bit is zero, then the third-party release option is not requested. If the 3rdPty bit is one, then the Target releases the LUN, but only if the reservation was made using the third-party reservation option and the 3rd Party ID is the ID of the initiator that made the reservation. (see 1.21, “Reserve (6)” on page 123) (see 4.1.10, “Command Processing While Reserved” on page 202)

## 1.20 Request Sense

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 03h							
1	LUN				Reserved = 0			
2	Reserved = 0							
3	Reserved = 0							
4	Allocation Length							
5	VU = 0		Reserved = 0				Flag	Link

The Request Sense command requests that the Target transfer sense data to the Initiator.

Sense data is valid for a Check Condition status returned on the prior command and for most unexpected bus free conditions. The sense data is preserved by the Target for the Initiator until retrieved by the Request Sense command or until any other command is received for the same LUN from the same Initiator that issued the command resulting in the Check Condition status. In the case of the single Initiator option, the Target assumes that the Request Sense command is from the same Initiator.

If a Request Sense command is received and Unit Attention is active, the Drive reports any pending sense data and preserves the Unit Attention Condition (see 4.1.5, “Unit Attention Condition” on page 184). If there is no pending sense data for conditions other than a Unit Attention, the sense data for the Unit Attention is returned and the Unit Attention condition is cleared.

The Target transfers all of the sense data up to the number of bytes allocated by the Initiator. No bytes are transferred if the allocation length is 0; this is not an error condition.

**Note:** For support of the Drive, a minimum allocation length of 32 bytes is required. However, for a generalized SCSI Initiator design, only an allocation of 255 bytes (FFh) ensures that sense data is not lost.

For a description of the sense data returned, see Appendix A, “SCSI Sense Data Format” on page 229.

Separate sense data is maintained for each Initiator.

If a Request Sense command is issued with an invalid LUN, sense data is returned indicating an illegal request due to the invalid LUN.

The Request Sense command returns the Check Condition status only to report fatal errors for the Request Sense command. Possible fatal errors include:

- Nonzero reserved bit in the command descriptor block
- Unrecovered SCSI bus parity error
- Incorrect Initiator Connection

**Note:** The Request Sense Command is not queued. For more information see 4.3, “Command Queuing” on page 204 .

## 1.21 Reserve (6)

Table 75. Reserve Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 16h							
1	LUN			3rdPty	3rd Party ID			Ext = 0
2	Reservation Identification							
3	(MSB) Extent List Length = 0 (LSB)							
4								
5	VU = 0		Reserved = 0			Flag	Link	

The Reserve command is used to reserve a LUN for an Initiator. This reservation can be either for the Initiator sending the command or for a third party as specified by the Initiator.

Extents are not supported by the Drive. The Ext bit must be 0. If Ext bit is set to 1, *Check Condition Status* is returned with a Sense Key of **Illegal Request** and additional sense code of **Invalid Field in CDB**. The Reservation Identification and Extent List Length fields are ignored.

The Reserve command requests that the entire Lun be reserved for the Initiator until;

- The reservation is superseded by another valid Reserve command from the Initiator that made the reservation.
- The reservation is released by a Release command from the same Initiator.
- A hard Reset condition occurs.
- A Bus Device Reset message is received from any Initiator.
- A power off/on cycle occurs.

The 3rdPty bit of 0 indicates that the Initiator that issued the the Reserve command is the Initiator for which the LUN is reserved.

The 3rdPty bit of 1 indicates that this is a third-party reservation . The 3rd Party ID field specifies the ID of the third party for which the LUN is reserved. A reservation made with the 3rdPty bit of 1 and the 3rd Party ID field set to the Initiator that issued this Reserve command is considered equivalent to a reservation made with the 3rdPty bit set to 0.

Only the Initiator that issued the Reserve command for a LUN may release the LUN, regardless of the 3rdPty option. This Initiator may also release the LUN by issuing another Reserve command. This superseding Reserve command releases the previous reservation when the new reservation is granted (see 4.1.10, “Command Processing While Reserved” on page 202).

Reservation queuing is not supported by the Drive. If a LUN is reserved and a Reserve command is issued from a different initiator, the Target responds with a reservation Conflict.

## 1.22 Reserve (10)

Table 76. Reserve Command Descriptor Block									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	Operation Code = 56h								
1	LUN			3rdPty	Reserved			Ext = 0	
2	Reservation Identification								
3	Third Party Device ID								
4	Reserved = 0								
5	Reserved = 0								
6	Reserved = 0								
7	(MSB)	Extent List Length = 0						(LSB)	
8									
9	VU = 0			Reserved = 0			Flag	Link	

The Reserve command is used to reserve a LUN for an Initiator. This reservation can be either for the Initiator sending the command or for a third party as specified by the Initiator.

Extents are not supported by the Drive. The Ext bit must be 0. If Ext bit is set to 1, *Check Condition Status* is returned with a Sense Key of **Illegal Request** and additional sense code of **Invalid Field in CDB**. The Reservation Identification and Extent List Length fields are ignored.

The Reserve command requests that the entire Lun be reserved for the Initiator until;

- The reservation is superseded by another valid Reserve command from the Initiator that made the reservation.
- The reservation is released by a Release command from the same Initiator.
- A hard Reset condition occurs.
- A Bus Device Reset message is received from any Initiator.
- A power off/on cycle occurs.

The 3rdPty bit of 0 indicates that the Initiator that issued the the Reserve command is the Initiator for which the LUN is reserved.

The 3rdPty bit of 1 indicates that this is a third-party reservation . The 3rd Party ID byte specifies the ID of the third party for which the LUN is reserved. A reservation made with the 3rdPty bit of 1 and the 3rd Party ID byte set to the Initiator that issued this Reserve command is considered equivalent to a reservation made with the 3rdPty bit set to 0.

Only the Initiator that issued the Reserve command for a LUN may release the LUN, regardless of the 3rdPty option. This Initiator may also release the LUN by issuing another Reserve command. This superseding Reserve command releases the previous reservation when the new reservation is granted (see 4.1.10, “Command Processing While Reserved” on page 202).



Reservation queuing is not supported by the Drive. If a LUN is reserved and a Reserve command is issued from a different initiator, the Target responds with a reservation Conflict.

## 1.23 Rezero Unit

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 01h							
1	LUN			Reserved = 0				
2	Reserved = 0							
3	Reserved = 0							
4	Reserved = 0							
5	VU = 0		Reserved = 0				Flag	Link

The Rezero Unit command causes the Drive to reposition heads to cylinder zero.

The Rezero Unit command will perform thermal compensation based on the TCC and TCHD bits in byte 5, Page 0 of the mode select parameters. Please refer to 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59 for information regarding these bits. When TCC is set, the Rezero command performs a thermal update on one or all heads depending on the setting of the TCHD bit, and it resets the timer based on an internal algorithm, that determines when the next update will occur. If another Rezero Unit command is not issued prior to the expiration of this timer, the required update will be performed by the Drive.

## 1.24 Seek (6)

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 0Bh							
1	LUN			(MSB) LBA.				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Reserved = 0							
5	VU = 0		Reserved = 0				Flag	Link

The Seek Command causes the Drive to seek to the track that contains the specified LBA. If the LBA is not less than the value returned by the Read Capacity command the Drive returns a Check Condition status, a sense key set to Illegal Request, and an additional sense code set to Invalid Field in CDB.

**Note:** The 6 byte command has only 21 bits of address for LBAs. The 21 bits allows for  $2 \times 10^{21}$  LBAs. (2,097,152)

## 1.25 Seek (10)

Table 79. Seek (10) Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 2Bh							
1	LUN			Reserved = 0				
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6-8	Reserved = 0							
9	VU = 0		Reserved = 0			Flag		Link

The Seek Command causes the Drive to seek to the track that contains the specified LBA. If the LBA is not less than the value returned by the Read Capacity command the Drive returns a Check Condition status, a sense key set to Illegal Request, and an additional sense code set to Invalid Field in CDB.

## 1.26 Send Diagnostic

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 1Dh							
1	LUN			PF	RSVD = 0	Self Test	Dev Ofl	Unit Ofl
2	Reserved = 0							
3	(MSB) Parameter List Length (LSB)							
4								
5	VU = 0			Reserved = 0			Flag	Link

The Send Diagnostic command requests that the Target perform self diagnostic tests.

If the SelfTest bit is set to one the Parameter List Length must be zero.

If the SelfTest bit is one the following self-tests will be performed if the spindle motor is running and up to speed.

- RAM checksum Test
- ROM checksum Test
- Data Buffer Test
- Electronics/Disk Enclosure(DE) Compatibility Test
- Seek Test
- Head Offset Test
- Read Test(all heads)
- Write Test(all heads)
- Cyclic Redundancy Code (CRC) Test
- Error Correction Code (ECC) Test
- Longitudinal Redundancy Code (LRC) Test
- Predictive Failure Analysis (PFA) Test

In addition to the above tests the Target will resume an interrupted Reassign Blocks command or Auto Reallocate.

All of the above must complete successfully for the Target to return Good status. If a test fails, the command terminates, an interrupted Reassign Blocks command or Auto Reallocate is not resumed, Check Condition status will be returned and the Target will be in Degraded Mode (see 4.1.9, “Degraded Mode” on page 190).

If the SelfTest bit is zero the target will perform the diagnostic operation specified in the passed parameter list.

If the SelfTest bit is zero, the parameter list length specifies the length in bytes of the parameter list to be transferred from the initiator to the target. If the length is zero, no data will be transferred. This is not an error. If the specified length results in truncation of the supported page, the target will return *Check Condition Status* with a Sense Key of **Illegal Request/Invalid Field in CDB**. The target supports the Translate Address Page, Page Code 40h.

The PF bit may be 0 or 1 if the SelfTest bit is zero.

| The DevOfl and UnitOfl bits are ignored because they are not applicable for this  
| device.

## 1.26.1 Send Diagnostics Page 40h

The translate address page allows the initiator to translate a logical block address or physical sector address to the other format. The address to be translated is passed to the target with the Send Diagnostic command and the results are returned to the initiator by the Receive Diagnostics command.

The target will read the parameter list from the initiator, and if no errors are detected in the parameter list, Good Status will be returned. The data translation will be performed upon receipt of the Receive Diagnostic Results Command.

## 1.26.2 Translate Address Page - Send Diagnostics

Table 81. Translate Address Page - Send Diagnostics								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Page Code = 40h							
1	Reserved = 0							
2	(MSB) Page Length = 000Ah (LSB)							
3								
4	Reserved = 0				Supplied Format			
5	Reserved = 0				Translate Format			
6 - 13	Address to Translate							

The page begins with a four-byte page header which specifies the page code and length, followed by two bytes which describe the translated address followed by the address to be translated.

The Page Code field is 40h.

The Page Length field specifies the length of the parameter bytes which follow, equal to ten.

The Supplied format field specifies the format of the address to be translated. The target supports 000b (Block Format) and 101b (Physical Sector Format).

The Translated format field specifies which format the initiator would like the address to be translated to. Valid values are the same as for the Supplied format.

If an invalid Supplied or Translated format field is passed, or if the Supplied format is the same as the Translated format field, the target will terminate the command with *Check Condition Status*. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN PARAMETER LIST.

The address to translate field contains a single address for the target to translate. The format of this field depends on the value in the supplied format field. The Physical Sector format is described in Table 7 on page 26. If the logical block format is the supplied format, the block address must be in the first four bytes of this field with the remaining bytes set to zero.

## 1.27 Start/Stop Unit

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 1Bh							
1	LUN			Reserved = 0				Immed
2	Reserved = 0							
3	Reserved = 0							
4	Reserved = 0						LoEj = 0	Start
5	VU = 0		Reserved = 0				Flag	Link

The Start/Stop Unit command requests that the Target enable/disable the LUN for further media access operations. Specifically, the Start/Stop Unit command spins up/stops the spindle motor.

Request Sense may be used to determine when the LUN is ready after the start-up sequence has completed.

- See 4.1.6, “Command Processing During Start-up and Format Operations” on page 187 for a description of how the Target reacts while the motor is performing the start-up sequence.
- Once the start-up has successfully completed, Request Sense returns sense data with a Sense key of *No Sense*.

When Immed is set to 1:

- Status is returned immediately.
- The Link bit must equal zero.
- *Good Status* is returned unless a higher priority status is pending (see 4.1.1, “Priority of SCSI Status Byte Reporting” on page 179).
- If there is an error during start-up, the command issued after the error occurred returns a *Check Condition Status*. The Request Sense command then returns a deferred error sense data combination (see 4.1.8, “Deferred Error Condition” on page 189).

When Immed is set to 0:

- Status is returned at the end of the operation (after start-up is complete).
- If an error occurs, Start Unit returns a *Check Condition Status*. The Request Sense command then returns a *Medium Error* or *Hardware Error* sense key (see Appendix A, “SCSI Sense Data Format” on page 229).

A Start bit of 1 requests the LUN be made ready for use. Being made 'ready for use' involves making sure that all steps of the start-up sequence are performed (see 4.18, “Bring-Up Sequence” on page 225). After the spindle motor reaches nominal operating speed, the Target performs the remainder of the start-up sequence if necessary. The individual elements of the start-up sequence are only executed if they have not been executed previously.



**Note:** When Start = 1 the Target will check the motor synchronization mode in the RPL field in Page 4 - Rigid Disk Drive Geometry Page and change the spindle state of operation if the RPL mode has been changed since the motor was last started. See 4.5, “Motor Synchronization” on page 208 for details on error reporting and Unit Attention condition generation.

Start set to 0 requests that the logical unit be stopped (media cannot be accessed by the Initiator). The Target performs an implicit Synchronize Cache command for the entire media prior to executing the Start/Stop Unit command.

The LoEj (Load or Eject) bit must be zero. Removable media is not supported. If LoEj is set to 1, *Check Condition Status* is returned with a Sense Key of ***Illegal Request*** and additional sense code of ***Invalid Field in CDB***

## 1.28 Synchronize Cache

Table 83. Synchronize Cache Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 35h							
1	LUN			Reserved = 0			Immed	RelAdr = 0
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved = 0							
7	(MSB) Number of Blocks (LSB)							
8								
9	VU = 0			Reserved = 0			Flag	Link

The Synchronize Cache command ensures that logical blocks in the cache, have their most recent data value recorded on the media. If more recent data in the specified logical block range exists in the cache than on the media, then the data from the cache is written to the media.

The Logical Block Address specifies where the operation is to begin.

The number of blocks field specifies the total number of contiguous logical blocks within the range. A number of blocks of zero indicates that all remaining logical blocks on the logical unit shall be within the range.

The RelAdr (relative address) bit must be zero. The target does not support relative addressing. If RelAdr is set to 1, *Check Condition Status* is returned with a Sense Key of **Illegal Request** and additional sense code of **Invalid Field in CDB**

An Immed (Immediate) bit of 0 indicates that the Target completes the operation before returning *Good Status*. An Immed bit of 1 indicates that the Target return status as soon as the command descriptor block has been verified.

## 1.29 Test Unit Ready

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 00h							
1	LUN			Reserved = 0				
2	Reserved = 0							
3	Reserved = 0							
4	Reserved = 0							
5	VU = 0		Reserved = 0				Flag	Link

The Test Unit Ready command allows the Initiator to check if the logical unit is ready.

If the logical unit can accept any supported medium-access command without returning Check Condition status with a Sense key of Not Ready then this command returns Good status (provided the LUN is not Busy or Reserved). If a Degraded Mode condition currently exists for the logical unit, then this command returns Check Condition status with a sense key of Not Ready, unless a higher priority response is required. (see B.3, “Priority of Error Reporting” on page 263).

The Test Unit Ready command is not intended as a diagnostic. No self-analysis is performed by the logical unit as a result of this command.

## 1.30 Verify

Table 85. Verify Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 2Fh							
1	LUN			DPO	Reserved = 0		BytChk	RelAdr = 0
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved = 0							
7	(MSB) Verification Length (LSB)							
8								
9	VU = 0			Reserved = 0			Flag	Link

The Verify command requests that the Target verify the data written on the medium. With a verification length of 0, the seek occurs but no data is verified. This condition is not considered an error.

The Logical Block Address specifies where the operation is to begin.

A BytChk bit of zero causes the Drive to do an ECC check on the specified range. If an ECC check is detected on all re-reads and the data was not corrected (either because it was uncorrectable or the correction was not attempted), a Check Condition status is returned with a Medium Error sense key.

A BytChk bit of one causes the Drive to do a byte-by-byte compare of data on the Drive to data sent by the initiator during the data-out phase. If an ECC check persists then Check Condition status is returned with Medium Error sense key. If there is no ECC check and a compare error is detected then Check Condition status is returned with a Miscompare sense key.

DRP actions for this command are controlled by Mode parameters in 1.6.7, “Page 7h - Error Recovery Parameters for Verify” on page 83. The actual DRP steps are defined in B.1.2, “Verify Commands” on page 260. Miscompare errors are not retried.

If caching is enabled, the command performs an implied Force Unit Access (FUA) and an implied Synchronize Cache before starting the verify. This insures that the medium, not the cache, is being verified.

The command stops on Check Condition and reports the LBA in error. The command must be reissued, starting with the next LBA, to verify the remainder of the Drive.

The Verification Length is the number of blocks to check.

The data (if any) from the data-out phase and the data from the media are not retained in the cache. Therefore, the DPO bit has no effect on this command and is ignored.

The RelAdr (Relative Address) bit must be 0. Relative addressing is not supported. If RelAdr is set to 1, Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.

## 1.31 Write (6)

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 0Ah							
1	LUN			(MSB)	LBA			
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	VU = 0		Reserved = 0				Flag	Link

The Write (6) command requests that the Target write the specified number of blocks of data from the Initiator to the LUN starting at the specified logical block address.

Transfer length may be a value from 0 to 255 where 0 implies a length of 256 blocks.

Errors are handled by ERP (error recovery procedures) (see B.1, “Data Recovery Procedure for Data Field Errors” on page 253). ERPs are controlled by the error recovery parameters (see 1.6.3, “Page 1 - Error Recovery Parameters” on page 66).

**Note:** The 6 byte command has only 21 bits of address for LBAs. The 21 bits allows for  $2 \times 10^{(21)}$  LBAs. (2,097,152) The 10 byte command should be used to write LBAs which require more bits.

## 1.32 Write (10)

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 2Ah							
1	LUN			DPO	FUA	Reserved = 0		RelAdr = 0
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved = 0							
7	(MSB) Transfer Length (LSB)							
8								
9	VU = 0			Reserved = 0			Flag	Link

The Write (10) command requests that the Target write the specified number of blocks of data from the Initiator to the LUN starting at the specified logical block address.

A FUA (Force Unit Access) bit of 1 indicates the Target must write the logical blocks of data to media before returning *Good Status*. A FUA bit of 0 indicates the Target may return *Good Status* prior to writing the logical blocks of data to the media.

A DPO (Disable Page Out) bit of 1 indicates that the data accessed by this command is to be assigned the lowest priority for being written into or retained by the cache. A DPO bit of 1 overrides any retention priority specified in the Mode Select Page 8 Caching Parameters. A DPO bit of 0 indicates the priority is determined by the retention priority. See 1.6.8, “Page 8h - Read Caching Parameters” on page 85 and 4.9, “Segmented Caching” on page 214.

The initiator should set the DPO bit when the blocks written by this command are not likely to be read in the near future.

The RelAdr (Relative Address) bit must be 0. Relative addressing is not supported. If RelAdr is set to 1, *Check Condition Status* is returned with a Sense Key of *Illegal Request* and additional sense code of *Invalid Field in CDB*.

If the transfer length is 0, the seek occurs, but no data is transferred. Status is returned after the seek completes. This condition is not considered an error.

## 1.33 Write and Verify

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 2Eh							
1	LUN			DPO	Reserved = 0		BytChk	RelAdr = 0
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved = 0							
7	(MSB) Transfer Length (LSB)							
8								
9	VU = 0			Reserved = 0			Flag	Link

The Write and Verify command requests that the Target write the data transferred from the Initiator to the medium and then verify that the data is correctly written. A transfer length of zero indicates that no data is transferred.

The Logical Block Address specifies where the operation is to begin.

A BytChk bit of zero causes the Drive to do an ECC check on the specified range after the write completes successfully. If an ECC check is detected on all re-reads and the data was not corrected (either because it was uncorrectable or the correction was not attempted), a Check Condition status is returned and a Medium Error sense key is built.

A BytChk bit of one causes the Drive to do a byte-by-byte compare of data after it is written to the Drive. The Drive uses the data sent for the write as the compare data, thus there is no extra data-out phase or extra save pointer messages. During the verify step if an ECC check is detected on all re-reads then Check Condition status is returned with Medium Error sense key. If there is no ECC check and a compare error is detected then Check Condition status is returned with a Mismatch sense key. When a compare or medium error occurs, the write may not have completed.

DRP actions for the verify portion of this command are controlled by Mode parameters in 1.6.7, “Page 7h - Error Recovery Parameters for Verify” on page 83. The actual DRP steps are defined in B.1.2, “Verify Commands” on page 260. Normal write command DRP is used for the Write portion of the command. A Mismatch error is not retried.

If caching is enabled, the command performs an implied Force Unit Access (FUA) and an implied Synchronize Cache before starting the operation. This insures that the medium, not the cache, is being verified.

A DPO (Disable Page Out) bit of 1 indicates that the data written by this command is to be assigned the lowest priority for being written into or retained by the cache. A DPO bit of 1 overrides any retention priority specified in the Mode Select Page 8 Caching parameters. A DPO bit of 0 indicates the priority is determined by the



retention priority. See 1.6.8, “Page 8h - Read Caching Parameters” on page 85 and 4.9, “Segmented Caching” on page 214.

The initiator should set the DPO bit when the blocks written by this command are not likely to be read again in the near future.

The RelAdr (Relative Address) bit must be 0. Relative addressing is not supported. If RelAdr is set to 1, Check Condition Status is returned with a Sense Key of Illegal Request and additional sense code of Invalid Field in CDB.

## 1.34 Write Buffer

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 3Bh							
1	LUN			Reserved = 0			Mode	
2	Buffer ID							
3	(MSB)							
4	Buffer Offset							
5								
6	(MSB)							
7	Parameter List Length							
8								
9	VU = 0			Reserved = 0			Flag	Link

The Write Buffer command is used to test the Target buffer and download micro-code to the Target. The values allowed in the Mode field are:

000b - Combined Header and Data Mode

010b - Write Data

100b - Download Microcode

101b - Download and Save

If any modes other than shown above are specified, then *Check Condition Status* is returned with a Sense Key of **Illegal Request** and additional sense code of **Invalid Field in CDB**.

- Mode = 000b (Combined Header and Data)

In this mode, the data specified is written to the buffer in sequence.

The Buffer ID must be zero. Receipt of a non-zero Buffer ID results in *Check Condition Status* with Sense Key of **Illegal Request** and additional Sense Code of **Invalid Field in CDB**.

The Buffer Offset must be zero. Receipt of a non-zero Buffer Offset results in *Check Condition Status* with Sense Key of **Illegal Request** and additional Sense Code of **Invalid Field in CDB**.

The Parameter List Length includes the length of the header. Therefore, the data written is 4 bytes less than the Parameter List Length. A Parameter List Length of zero indicates no data is transferred. The Parameter List Length must not be greater than the buffer capacity plus four bytes. Receipt of an invalid Parameter List Length results in *Check Condition Status* with Sense Key of a **Illegal Request** and additional Sense Code of **Parameter List Length Error**.

**Note:** The Buffer Capacity can be determined using Read Buffer mode 000b.

- Mode = 010b (Write Data)

In this mode, the data specified is written to the buffer in sequence starting at the specified offset.

The Buffer ID field must be 0. Receipt of a non-zero Buffer ID results in *Check Condition Status* with Sense Key of **Illegal Request** and additional Sense Code of **Invalid Field in CDB**.

The Buffer Offset field contains the starting byte offset within the data buffer to which the data is transferred.

The Parameter List Length specifies the number of bytes to transfer. A Parameter List Length of zero indicates no data is transferred. If the sum of the Buffer Offset and the Parameter List Length exceeds the buffer capacity, then the Target returns *Check Condition Status* with Sense Key of **Illegal Request** and additional Sense Code of **Parameter List Length Error**.

**Note:** The Buffer Capacity can be determined using Read Buffer mode 000b.

- Mode = 100b (Download Microcode) and Mode = 101b (Download and Save)

The microcode is downloaded to the control storage of the Target.

When the mode = 100b (Download Microcode), the downloaded microcode is not saved on the media. The original microcode which resides in the Reserved Area of the media will be reloaded at the next power-on.

When the mode = 101b (Download and Save), the downloaded microcode is saved in the Reserved Area of the media. The original microcode is lost.

The microcode to be downloaded is supplied to the using system in a single data set whose length is divisible by 8000h. The using system can download the entire microcode with a single Write Buffer command or with multiple Write Buffer commands.

When multiple Write Buffer commands are used, the system must divide the microcode into 8000h byte pieces and send each piece with consecutive Write Buffer commands. The Buffer ID of the first piece must be set to zero and the Buffer ID must be incremented by one for each subsequent Write Buffer command until the download is complete.

The Buffer ID is used to determine which portion of the microcode is being downloaded. When downloading the entire microcode, the Buffer ID must be zero. When downloading the microcode in pieces, the Buffer ID is zero for the first piece, and the Buffer ID must be incremented by one for each subsequent Write Buffer command until the download is complete. When downloading the microcode, if a value of the Buffer ID is skipped (i.e. the Buffer ID incremented by two or more), the Target returns *Check Condition Status* with Sense Key of **Illegal Request** and additional Sense Code of **Invalid Field in CDB**.

Buffer Offset is ignored.

When the Buffer ID is zero, valid values for the Parameter List Length are zero, 8000h and a Parameter List Length greater than or equal to the microcode length. When the Buffer ID is non-zero, the Parameter List Length must be equal to 8000h. Receipt of an invalid Parameter List Length results in *Check Condition Status* with Sense Key of a **Illegal Request** and additional Sense Code of **Parameter List Length Error**. A Parameter List Length of zero indicates no data is transferred.

Link must be zero when the Buffer ID is the maximum allowed value.

With Buffer ID is zero and if the Load ID (LID) does not match the LID in the ROM code, the Target returns *Check Condition Status* with Sense Key of

*Illegal Request* and additional Sense Code of *Invalid Field in Parameter List*. The Load ID is not checked when the Buffer ID is non-zero.

The downloaded microcode is tested via a checksum. If the checksum fails, the Target returns *Check Condition Status* with Sense Key of *Hardware Error* and additional Sense Code of *Diagnostic Failure*. The microcode is not saved to the media.

On successful completion, the Target generates a Unit Attention condition for all Initiators except the one that issued the Write Buffer command (see 4.1.5, “Unit Attention Condition” on page 184). The Additional Sense Code *Microcode has been changed*, is reported in the sense data.

The first 19 bytes of the microcode contain the following:

Bytes 0 to 2 = Microcode Length

Bytes 3 to 6 = Load ID

Bytes 7 and 10 = Modification Level

Bytes 11 to 14 = PTF Number

Bytes 15 to 18 = Patch Number

This command will cause the entire cache to be emptied. See 4.9, “Segmented Caching” on page 214.

**Note:** Idle time functions and other initiators may alter the contents of the buffer. The data read by the Read Buffer command may not match the data written by the Write Buffer mode 000b command. To ensure the buffer is not altered, link the two commands together by setting the Link bit when issuing Write Buffer mode 000b command.

## 1.35 Write Long

Table 90. Write Long Command Descriptor Block								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 3Fh							
1	LUN				Reserved = 0			RelAdr = 0
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved = 0							
7	(MSB) Byte Transfer Length (LSB)							
8								
9	VU = 0		Reserved = 0				Flag	Link

The Write Long command requests that the target write the data transferred by the Initiator to the medium. The data passed during the Write Long command consists of the data bytes followed by the LRC and ECC bytes. The ECC field depends solely on the data and LRC bytes and is unaffected by sync, ID, or VCO fields.

The RelAdr (relative address) bit must be zero. The Drive does not support relative addressing. If RelAdr is set to 1, *Check Condition Status* is returned with a Sense Key of **Illegal Request** and additional sense code of **Invalid Field in CDB** in CDB.

The logical block address field specifies the logical block at which the write operation starts.

The byte transfer length field must specify the number of bytes of data that are returned for the Read Long command. If a non-zero byte transfer length does not exactly match a data length that is returned for the Read Long command, the command terminates with *Check Condition Status*, a sense key set to **Illegal Request**, and an additional sense code set to **Invalid Field in CDB**. The ILI and valid bits are set to one and the information field is set to the difference (residue) of the requested length minus the actual length in bytes. Negative values are indicated by two's complement notation.

The transfer length is calculated as follows:

$$\text{transfer length} = \text{logical blocksize} + (20 * n),$$

where n is the number of physical sectors per LBA.

If the byte transfer length is zero, the seek occurs, but no data is transferred. This condition is not considered an error.

The data for this command is not retained in the cache.

**Note:** The Drive will attempt a limited amount of error recovery. This consists of a number of command retries, of which the number of retries is internally specified.

## 1.36 Write Same

Table 91. Write Same Command Descriptor Block									
Byte	BIT								
	7	6	5	4	3	2	1	0	
0	Operation Code = 41h								
1	Logical Unit Number			Reserved = 0		PBdata = 0	LBdata = 0	RelAdr = 0	
2	(MSB) Logical Block Address (LSB)								
3									
4									
5									
6	Reserved								
7	Number of Blocks								
8									
9	VU = 0		Reserved = 0			Flag		Link	

The Write Same command instructs the Target to write a single block of data, transferred to the Target from the Initiator, to a number of sequential logical blocks.

This command is useful to write large data areas without sending all of the data over the SCSI bus.

The Logical Block Address specifies the address at which the write begins.

The Number of Blocks specifies the number of contiguous blocks to be written. If the number is 0, all of the remaining blocks on the specified Logical Unit are written.

The PBdata (Physical Block Data) option is not supported and must be 0.

The LBdata (Logical Block Data) option is not supported and must be 0.

The RelAdr option is not supported and must be 0.

If RelAdr, PBdata or LBdata is set to 1, *Check Condition Status* is returned with a Sense Key of ***Illegal Request*** and additional sense code of ***Invalid Field in CDB***.

The data for this command is not retained in the cache.

## 2.0 SCSI Status Byte

A SCSI Status Byte is sent to the Initiator during the Status phase at the termination of each SCSI command unless the command is cleared by an Abort message, any type of Target Reset, or an unexpected Bus Free error condition. The SCSI Status Byte is defined in Table 92.

Table 92. SCSI Status Byte									
Byte	BIT								
	7	6	5	4	3	2	1	0	
<b>Status</b>	Reserved = 0		Status Code					Rsvd	

Table 93. Status Code Bit Definitions									
Bits of Status Code									
Status	7	6	5	4	3	2	1	0	
Good	R	R	0	0	0	0	0	R	
Check Condition	R	R	0	0	0	0	1	R	
Condition Met	R	R	0	0	0	1	0	R	
Busy	R	R	0	0	1	0	0	R	
Intermediate/ Good	R	R	0	1	0	0	0	R	
Intermediate/ Condition Met	R	R	0	1	0	1	0	R	
Reservation Conflict	R	R	0	1	1	0	0	R	
Queue Full	R	R	1	0	1	0	0	R	
<b>Note:</b> All Reserved fields (R) are set to zero.									

A description of the status represented by each Status Byte is given below:

### 00h Good status

This status indicates that the Target has successfully completed the SCSI command. For this status, sense is not valid and the Sense key and the Sense code are set to zero.

### 02h Check Condition status

This status indicates that an error, exception, or abnormal condition has caused sense data to be set. The Initiator should issue a Request Sense command to obtain the sense data and determine the cause of the Check Condition status.

### 04h Condition Met status

This status indicates that the requested operation is satisfied. (See 1.8, “Pre-Fetch” on page 100) For this status, sense is not valid and the Sense key and the Sense code are set to zero.

### 08h Busy status

This status indicates that the Drive is busy performing another operation for a different Initiator and is unable to execute the command received from the currently connected Initiator. The Initiator should issue the command again at a later time. Also see 4.1.4, “Command Processing During Execution of

Active I/O Process” on page 182 for a discussion of commands that may return the Busy status. For this status, sense is not valid and the Sense key and the Sense code are set to zero.

**10h** Intermediate/Good status

This status indicates that the Target has successfully completed a linked command. This status is returned for every command in a series of linked commands (except the last command) unless an error, exception, or abnormal condition causes a Check Condition, Busy, or Reservation Conflict status to be returned. If this status is not returned, the chain of linked commands is broken. For this status, sense is not valid and the Sense key and the Sense code are set to zero.

**14h** Intermediate/Condition Met

This status is the combination of Condition Met and Intermediate/Good. For this status, sense is not valid and the Sense key and the Sense code are set to zero.

**18h** Reservation Conflict status

This status indicates that the LUN is reserved (refer to 1.21, “Reserve (6)” on page 123 and 1.18, “Release (6)” on page 119) for a different Initiator and is unable to execute the command received from the currently connected Initiator. The Initiator should issue the command again at a later time. Also see 4.1.10, “Command Processing While Reserved” on page 202 for a discussion of commands that may return the Reservation Conflict status. For this status, sense is not valid and the Sense key and the Sense code are set to zero.

**28h** Queue Full status

This status indicates that the targets command queue is full. This status is returned when a Queue Tag message is received and there is no room on the command queue for an I/O process from the issuing initiator. See 4.3.2, “Tagged Queuing” on page 205 for more information concerning tagged queuing. For this status, sense is not valid and the Sense key and the Sense code are set to zero.



## 3.0 SCSI Message System

This chapter details how the message system is implemented on the Drive. Included is a functional description of the supported messages, message timing, message phase error handling and a message state table.

### 3.1 Supported Messages

The following single-byte messages are supported.

Code	Message	Direction	Negate ATN Before Last ACK
00h	Command Complete	In	N/A
02h	Save Data Pointer	In	N/A
03h	Restore Pointers	In	N/A
04h	Disconnect	In	N/A
05h	Initiator Detected Error	Out	Yes
06h	Abort	Out	Yes
07h	Message Reject	In	N/A
07h	Message Reject	Out	Yes
08h	No Operation	Out	Yes
09h	Message Parity Error	Out	Yes
0Ah	Linked Command Complete	In	N/A
0Bh	Linked Command Complete with Flag	In	N/A
0Ch	Bus Device Reset	Out	Yes
0Dh	Abort Tag	Out	Yes
0Eh	Clear Queue	Out	Yes
12h	Continue I/O Process	Out	Yes
13h	Target Transfer Disable	Out	Yes
80h-FFh	Identify	In	N/A
80h-FFh	Identify	Out	No

In: Target to Initiator. Out: Initiator to target.

The following two-byte messages are supported:

Code	Message	Direction	Negate ATN Before Last ACK
20h	Simple Queue Tag	In	N/A
20h	Simple Queue Tag	Out	No
21h	Head of Queue Tag	Out	No
22h	Ordered Queue Tag	Out	No
23h	Ignore Wide Residue	In	N/A

The following multiple-byte extended messages are supported:

Extended Code	Message	Direction	Negate ATN Before Last ACK
01h	Synchronous Data Transfer Request	In	N/A
01h	Synchronous Data Transfer Request	Out	Yes
03h	Wide Data Transfer Request	In	N/A
03h	Wide Data Transfer Request	Out	Yes

### 3.1.1 Command Complete (00h)

The Target sends Command Complete message to the Initiator to indicate that the execution of a command (or series of linked commands) has terminated and that valid status has been sent to the Initiator. After successfully sending this message, the Target goes to the bus Free phase. The Target considers the message transmission to be successful when it detects the negation of ACK for the Message In byte with the ATN signal false.

### 3.1.2 Save Data Pointer (02h)

The Save Data Pointer message is sent from the Target to direct the Initiator to save a copy of the present active data pointer. The value of the current (or active) data pointer should be moved into the corresponding saved data pointer for the currently attached logical unit. The Target sends the Save Data Pointer message to the Initiator prior to sending a Disconnect message to the Initiator if a Data phase has occurred since the physical path was established, and another Data phase is required to successfully complete the command. If the DPSDP bit is set, (see 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59) the Save Data Pointer message will be sent prior to every disconnection once a Data phase has occurred for the current command. If the ASDPE bit (same page) is set, the Save Data Pointer message will precede every Disconnect message regardless of the value of the DPSDP bit. See also 3.1.4, “Disconnect (04h)” on page 151.

### 3.1.3 Restore Pointers (03h)

The Restore Pointers message is sent from the Target to direct the Initiator to restore the most recently saved pointers (for the currently attached logical unit) to the active state. The saved values for the command, data, and status pointers for the currently attached logical unit should be moved into their corresponding current (or active) pointer. The current command and status pointers should be restored to the beginning of the present command and status areas. The current data pointer should be restored to the beginning of the data area in the absence of a Save Data Pointer message, or to the value at the point at which the last Save Data Pointer message occurred. This message is sent when attempting to retry a Command, Data, or Status phase to recover from a SCSI bus related error (see also 3.6, “SCSI Bus Related Error Handling Protocol” on page 174).

### 3.1.4 Disconnect (04h)

The Disconnect message is sent from the Target to inform the Initiator that the present physical path is going to be broken. A later reconnect will be required in order to complete the current command. The Target sends the Disconnect message before disconnecting from the SCSI bus. The disconnection is to free the SCSI bus while the Target performs a relatively long operation that does not require the bus. After successfully sending this message to the Initiator, the Target goes to the Bus Free phase. The Target considers the message transmission to be successful when it detects the negation of ACK for the Message In byte with the ATN signal false. The Target only sends this message if the Initiator previously granted the Target the privilege of disconnecting via the Identify message (see 3.1.19, “Identify (80h or C0h)” on page 156).

### 3.1.5 Initiator Detected Error (05h)

The Initiator Detected Error message is sent from an Initiator to inform the Target that an error has been detected that does not preclude the Target from retrying the previous Command or Status phase. The source of the error either may be related to previous activities on the SCSI bus or may be internal to the Initiator and unrelated to any previous SCSI bus activity.

If the Initiator intends to send this message, the Initiator must assert the ATN signal prior to its release of ACK for the last byte transferred in the Information phase that is to be retried (see also 3.3, “Attention Condition” on page 164). This provides an interlock so the Target can determine which Information phase to retry.

After receiving this message, the Target may retry the previous phase by sending a Restore Pointers message to the Initiator and then repeating the previous Command or Status phase. See also 3.2, “Supported Message Functions” on page 163 and 3.6, “SCSI Bus Related Error Handling Protocol” on page 174.

### 3.1.6 Abort (06h)

The Abort message specifies that the current I/O process be terminated immediately regardless of its state of execution. Additionally, all other I/O processes from the same initiator on the identified LUN are likewise terminated. Queued I/O processes are deleted from the command queue and active I/O processes are aborted. If the current I/O process is also an active I/O process it will be aborted and the next queued I/O process on the command queue will become an active I/O process. If the logical unit is identified, then all pending data and status for the issuing Initiator and the identified logical unit are cleared and the Target goes to the Bus Free phase. Pending data and status for other initiators and other logical units are not cleared. I/O processes associated with other Initiators are likewise not affected. If a logical unit has not been identified, the Target goes to the Bus Free phase without affecting an operation on any logical unit for the issuing Initiator or any other Initiator. No status or message is sent for any of the I/O processes terminated. It is not an error to send the Abort message to a logical unit that is not currently performing an operation for the Initiator. The target goes to BUS FREE following receipt of this message. Pending sense data is preserved for all initiators.

**Note:** It is permissible for an Initiator to select the device after the Target has disconnected from the Initiator, for the purpose of sending an Identify message followed by an Abort message. This aborts all I/O processes associated with the issuing Initiator.

### 3.1.7 Message Reject (07h)

The Message Reject message is sent from either the Initiator or the Target to indicate that the last message received was inappropriate or has not been implemented.

If the Initiator intends to send this message, the Initiator must assert the ATN signal prior to its release of ACK for the REQ/ACK handshake of the message that is to be rejected (see also 3.3, “Attention Condition” on page 164). This provides an interlock so the Target can determine which message is rejected.

If the Target intends to send this message, the Target changes to the Message In phase and sends the Message Reject message to the Initiator prior to transferring any additional message bytes (or any other Information phase bytes) from the Initiator. This implies that the Target may change from the Message Out phase to the Message In phase before all messages are received and ATN is negated. This provides an interlock so the Initiator can determine which message is rejected.

See also 3.2, “Supported Message Functions” on page 163 and 3.6, “SCSI Bus Related Error Handling Protocol” on page 174 for more information on the use of the Message Reject message.

### 3.1.8 No Operation (08h)

The No Operation message is sent from the Initiator to the Target when the Initiator does not currently have any other valid message to send. See 3.5, “Message State/Action Table” on page 168 for the target's response to this message.

### 3.1.9 Message Parity Error (09h)

The Message Parity Error message is sent from the Initiator to inform the Target that one or more bytes in the last message it received had a parity error.

If the Initiator intends to send this message, the Initiator must assert the ATN signal prior to its release of ACK for the REQ/ACK handshake of the message that has the parity error. See also 3.3, “Attention Condition” on page 164. This provides an interlock so the Target can determine which message has the parity error.

If the target receives this message under any other circumstance, the Target signals a catastrophic error condition by releasing the BSY signal without any further information transfer attempted.

After receiving this message, the Target may retry sending the previous message to the Initiator (see 3.5, “Message State/Action Table” on page 168, 3.2, “Supported Message Functions” on page 163, and 3.6, “SCSI Bus Related Error Handling Protocol” on page 174).

### 3.1.10 Linked Command Complete (0Ah)

The Target sends the Linked Command Complete message to the Initiator to indicate that execution of a Linked command (with flag bit equal to zero) has completed and that valid status has been sent to the Initiator. After successfully sending this message, the Target changes to the Command phase to receive the next command.

### 3.1.11 Linked Command Complete with Flag (0Bh)

The Target sends the Linked Command Complete with Flag message to the Initiator to indicate that execution of a Linked command (with flag bit equal to one) has completed and that valid status has been sent to the Initiator. After successfully sending this message, the Target changes to the Command phase to receive the next command.

### 3.1.12 Bus Device Reset (0Ch)

The Bus Device Reset message is sent from the Initiator to direct the Target to clear all I/O processes for all initiators. This message forces a hard reset condition which resets the Target to an initial state with no I/O processes pending for any Initiator. A Unit Attention condition is created for all initiators. After receiving this message, the Target goes to the Bus Free phase.

### 3.1.13 Abort Tag (0Dh)

The Abort Tag message specifies that the current I/O process be terminated immediately regardless of its state of execution. No status or message is sent for the I/O process terminated. I/O processes other than the current I/O process shall not be affected even if they are from the same initiator. If the current I/O process is also an active I/O process then it will be aborted and the next queued I/O process in the command queue will become an active I/O process. If the current I/O process is also a queued I/O process then the queued I/O process will be removed from the command queue and the queue will be adjusted accordingly to preserve the order of execution of the remaining queued I/O processes. It is not an error to establish an I\_T\_L\_Q nexus and send the Abort Tag message to a logical unit that does not currently have an I/O process associated with that I\_T\_L\_Q nexus. The target goes to BUS FREE following receipt of this message. Pending sense data is preserved for all initiators.

**Note:** It is permissible for an Initiator to select the device after the Target has disconnected from the Initiator, for the purpose of sending an Identify message followed by a Queue Tag message followed by an Abort Tag message. This aborts the tagged I/O process, if any, associated with the established I\_T\_L\_Q nexus.

### 3.1.14 Clear Queue (0Eh)

The Clear Queue message specifies that all I/O processes from all initiators for the identified LUN be terminated immediately regardless of their state of execution. This message is equivalent to receiving an Abort message from each initiator. All active I/O processes for the identified LUN are aborted, all queued I/O processes for the identified LUN are deleted from the command queue, and the current I/O process is ended by going to BUS FREE. All pending status and data for the identified LUN shall be cleared for all initiators. No status or message is sent for any of the I/O processes terminated. Unit Attention condition will be generated for all other initiators that had I/O processes that were *either active or queued*. The additional sense code will be set to *COMMANDS CLEARED BY ANOTHER INITIATOR*. See 4.1.5, “Unit Attention Condition” on page 184 The target goes to BUS FREE following receipt of a Clear Queue message. Pending sense data is preserved for all initiators.

### 3.1.15 Continue I/O Process (12h)

The Continue I/O Process message is sent from the initiator to the target to reconnect to an I/O process. This message shall be sent in the same Message Out phase as the Identify message. Thus, if the initiator intends to send the Continue I/O Process message, the Message Out phase following the Selection phase must consist of the Identify, Queue Tag (if any), and Continue I/O Process messages.

The purpose of the Continue I/O Process message is to distinguish a valid initiator reconnection from an incorrect initiator connection.

The initiator may reconnect to an I/O process before the target indicates that it is ready to transfer data (see 3.1.16, “Target Transfer Disable (13h)” for a description of how the target informs the initiator of this condition). In this case, if the target is not ready to continue processing the reconnected I/O process it will attempt to disconnect. If the disconnection attempt is successful and the I/O process is not already active, the target will move the I/O process to the front of the command queue. If the initiator rejects the disconnection or does not grant disconnect privilege in the Identify message, the target will end the connection with a check condition, terminate the I/O process, and generate sense for the initiator/LUN combination. The sense key, code, and qualifier will be set to *ABORTED COMMAND, Message Error*.

It is an error for the initiator to send this message on an initial connection (i.e., there is no I/O process for the nexus) and the target will go to Bus Free phase (see 3.6.1, “Unexpected Bus Free Phase Error Condition” on page 174).

### 3.1.16 Target Transfer Disable (13h)

The Target Transfer Disable (TTD) message is sent from an initiator to the target to request that subsequent reconnections for data transfer on the I/O process be done by the initiator instead of the target. The target may reconnect for other purposes, but will not enter a data phase on a target reconnection.

This message shall be sent as the last message of the first Message Out phase of an initial connection. The target will continue the I/O process, including any Data Out phases on the initial connection, until the target would normally disconnect, but the target will not reconnect to transfer data. The target will not enter a Data In phase on the initial connection (unless disconnection permission is not granted), and the target will not enter any data phase on any subsequent target reconnection for the I/O process.

**Note:** The TTD message does not effect the execution of priority commands. If a TTD message is received with a Request Sense or Inquiry command it is ignored.

When the target is ready to transfer data for a disconnected I/O process for which a TTD message has been sent, the target will reconnect to the initiator for the I/O process (via a Reselection phase, an Identify message, and an optional Queue Tag message), send a Disconnect message, and, if the initiator does not respond with a Message Reject message, go to Bus Free phase. If the initiator rejects the Disconnect message, the target will enter a data phase; otherwise, the initiator may reconnect to the I/O process as described in 3.1.15, “Continue I/O Process (12h)” to do the data transfer.

### 3.1.17 Queue Tag Messages (20h, 21h, 22h)

Byte	Value	Description
0	20h 21h 22h	Queue Tag message
1	XXh	Queue Tag

Queue Tag messages consist of:

- Simple Queue Tag Message
- Ordered Queue Tag Message
- Head of Queue Tag Message

Queue Tag messages are used to specify an identifier, called a *Queue Tag*, for an I/O process which establishes the I\_T\_L\_Q nexus. The queue tag field is an 8-bit unsigned integer assigned by the initiator during an initial connection. The Queue Tag for every I/O process for each I\_T\_L nexus must be unique. If the target receives a Queue Tag that is currently in use for the I\_T\_L nexus it will respond as outlined in 4.1.3, “Incorrect Initiator Connection” on page 181. A Queue Tag becomes available for re-assignment when the I/O process ends. The numeric value of a Queue Tag has no effect on the order of execution.

Whenever an initiator connects to the target, the appropriate Queue Tag message must be sent immediately following the Identify message and within the same MESSAGE OUT phase to establish the I\_T\_L\_Q nexus for the I/O process. Only one I\_T\_L\_Q nexus may be established during a connection. If a Queue Tag message is not sent, then only an I\_T\_L nexus is established for the I/O process (untagged command).

Whenever the target reconnects to an initiator to continue a tagged I/O process, the Simple Queue Tag message is sent immediately following the Identify and within the same MESSAGE IN phase to revive the I\_T\_L\_Q nexus for the I/O process. Only one I\_T\_L\_Q nexus may be revived during a reconnection. If the Simple Queue Tag message is not sent, then only a I\_T\_L nexus is revived for the I/O process (untagged command).

#### 3.1.17.1 Simple Queue Tag (20h)

The Simple Queue Tag Message specifies that the current I/O process be placed in the command queue. The order of execution, with respect to other I/O processes received with Simple Queue Tag Messages, is up to the discretion of the target. The target will send a Simple Queue Tag Message after reselection for I/O processes that were received with either Simple, Ordered, or Head of Queue Tag messages.

#### 3.1.17.2 Head of Queue Tag (21h)

The Head of Queue Tag message specifies that the current I/O process be placed first in the command queue. Active I/O processes (if any) will not be preempted by an I/O process received with a Head of Queue Tag. An I/O process received with a Head of Queue Tag will be executed before all other queued I/O processes regardless of the I\_T\_L\_Q nexus. Successive I/O processes received with Head of Queue Tag messages will be executed in LIFO order.

### 3.1.17.3 Ordered Queue Tag (22h)

The Ordered Queue Tag Message specifies that the current I/O process be placed in the command queue for execution in the order received. All queued I/O processes for the logical unit received prior to the current I/O process are executed before the current I/O process is executed. All queued I/O processes received after the current I/O process are executed after the current I/O process is executed, except for I/O processes received with a HEAD OF QUEUE tag message.

**Note:** In multi-initiator environments where both Tagged and Untagged I/O processes are present together in the command queue, untagged I/O processes are treated as though they were received with a Simple Queue tag for purposes of execution scheduling.

### 3.1.18 Ignore Wide Residue (23h)

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Message Code = 23h							
1	Ignore = 01h							

The Ignore Wide Residue Message is sent from the target to indicate that the number of valid bytes sent during the last REQ/ACK handshake of a DATA IN phase is less than the negotiated transfer width. The ignore field (always = 01h) indicates that one byte (data bits 8-15) should be ignored. This message is sent immediately following the DATA IN phase and prior to any other messages. Even though a byte is invalid, its corresponding parity bit is valid for the value transferred.

### 3.1.19 Identify (80h or C0h)

The Identify message is sent by either the Initiator or the Target to establish the physical path connection between the Initiator and the Target for a particular logical unit.

The Identify message is defined as follows:

- Bit 7** This bit is always set to one to distinguish the Identify message from other messages.
- Bit 6** This bit is only set to one by the Initiator to grant the Target the privilege of disconnecting. If this bit is 0, the Target does not disconnect. This bit is set to 0 when the Target sends an Identify message to the Initiator during reconnection.
- Bit 5** This bit is reserved and must be zero for an Identify message.
- Bits 4-0** These bits specify the LUN. A value other than zero identifies an invalid LUN. For a description of the response to an invalid LUN see 4.1.2, "Invalid LUN Processing" on page 180.

If an Identify message is received with bit 5 not equal to zero during any phase prior to Status phase, the Target returns Check Condition status. If an Identify message is received with bit 5 not equal to zero during or after Status phase, the Target goes to Bus Free phase (see 3.5, "Message State/Action Table" on page 168).



Only one LUN may be identified for any one selection sequence. If the Target receives an Identify message with a new logical unit number after the LUN has previously been identified, the Target goes to the Bus Free phase to signal a catastrophic error. The Initiator may send more than one Identify message during a selection sequence in order to toggle disconnect/reconnect permission if the specified LUN remains the same. (see 3.2, “Supported Message Functions” on page 163).

When the Identify message is sent from the Target to the Initiator during reconnection, an implied Restore Pointers message must be performed by the Initiator.

### 3.1.20 Extended Messages (01h,xxh)

A value of one in the first byte of a message indicates the beginning of a multiple-byte extended message. The minimum number of bytes sent for an extended message is three. The extended message format is shown below.

Byte	Value	Description
0	01h	Extended message
1	n	Extended message length
2	y	Extended message code
3 - (n+1)	x	Extended message arguments

The extended message length specifies the length in bytes of the extended message code plus the extended message arguments to follow. The total length of the message is equal to the extended message length plus two. A value of zero for the extended message length indicates 256 bytes to follow.

The Drive supports two extended messages; Synchronous Data Transfer Request (01h,01h) and Wide Data Transfer Request (01h,03h). The only extended message lengths supported by the Target are 02h and 03h. If an unsupported message code or message length is received, the Target goes to MESSAGE IN phase immediately after the byte in error and sends a Message Reject message to the Initiator.

If attention is negated before all bytes of a multiple-byte extended message are received, the file will go to BUS FREE to signal a catastrophic error.

### 3.1.21 Synchronous Data Transfer Request (01h,01h)

A pair of Synchronous Data Transfer Request messages are exchanged between an Initiator and a Target to establish the synchronous data transfer mode between the two devices. The message exchange establishes the permissible transfer period and REQ/ACK offset for a synchronous data transfer between the two devices. The Initiator may initiate a synchronous data transfer negotiation at any time after the LUN has been identified. The Target initiates a synchronous transfer negotiation if the Target has not negotiated with the Initiator since the last time the Target was Reset (Power-on Reset, SCSI Bus “hard” Reset, Bus Device Reset message, or Self Initiated Reset).

Target-initiated synchronous negotiation normally occurs immediately following the wide data transfer negotiation. If a data transfer width agreement exists, the target-initiated synchronous negotiation occurs either immediately following the Command phase or immediately following the first reconnection. In either case, negotiation occurs before any Data phase between the Target and the Initiator.

The implied synchronous agreement remains in effect until the Target is Reset (Power-on Reset, SCSI Bus “hard” Reset, Bus Device Reset message, or Self Initiated Reset), a new synchronous agreement is negotiated, or the Target receives a Wide Data Transfer Request message. If a Reset occurs, the Target goes to asynchronous mode.

### 3.1.21.1 Synchronous Data Transfer Request Message Definition

Byte	Value	Description
0	01h	Extended message
1	03h	Extended message length
2	01h	Synchronous Data Transfer Request code
3	M	Transfer period (M times 4 nanoseconds)
4	X	REQ/ACK offset.

The transfer period (M above) is the minimum time between leading edges of successive REQ pulses and of successive ACK pulse to meet the device requirements for successful reception of data while using synchronous data transfers. The Drive supports transfer periods in the range 100 nSec to 450 nSec in 25 nSec increments. The Target will use the "fast SCSI data transfer" timings if a synchronous agreement results in a target transfer period of less than 200 nSec.

The REQ/ACK offset (X above) is the maximum number of REQ pulses that can be outstanding before its corresponding ACK pulse is received at the Target. A REQ/ACK offset value of zero indicates asynchronous mode. The Drive supports REQ/ACK offset values in the range 0 through 15 for 16 bit data transfers and offset values in the range 0 through 30 for 8 bit data transfers.

### 3.1.21.2 Synchronous Negotiation Started by the Initiator

If the Initiator recognizes that negotiation is required and sends a Synchronous Data Transfer Request message out, the Target responds by changing to the Message In phase and sending a Synchronous Data Transfer Request message in to the Initiator prior to transferring any additional message bytes (or any other Information phase bytes) from the Initiator. This provides an interlock during the synchronous negotiation.

The Drive responds to each Initiator requested transfer period as shown in the following table:

Initiator Request	Target Response	Target Transfer Period	Maximum Burst Rate
$0 <= M_i <= 25$	$M_t = 25$	100 nSec	10.00 MT/s
$26 <= M_i <= 31$	$M_t = M_i$	125 nSec	8.00 MT/s
$32 <= M_i <= 37$	$M_t = M_i$	150 nSec	6.67 MT/s
$38 <= M_i <= 43$	$M_t = M_i$	175 nSec	5.71 MT/s
$44 <= M_i <= 50$	$M_t = M_i$	200 nSec	5.00 MT/s
$51 <= M_i <= 56$	$M_t = M_i$	225 nSec	4.44 MT/s
$57 <= M_i <= 62$	$M_t = M_i$	250 nSec	4.00 MT/s
$63 <= M_i <= 68$	$M_t = M_i$	275 nSec	3.64 MT/s

Table 101 (Page 2 of 2). Initiator Request/Target Response

Initiator Request	Target Response	Target Transfer Period	Maximum Burst Rate
69 <= Mi <= 75	Mt = Mi	300 nSec	3.33 MT/s
76 <= Mi <= 81	Mt = Mi	325 nSec	3.08 MT/s
82 <= Mi <= 87	Mt = Mi	350 nSec	2.86 MT/s
88 <= Mi <= 93	Mt = Mi	375 nSec	2.67 MT/s
94 <= Mi <= 100	Mt = Mi	400 nSec	2.50 MT/s
101 <= Mi <= 106	Mt = Mi	425 nSec	2.35 MT/s
107 <= Mi <= 112	Mt = Mi	450 nSec	2.22 MT/s
113 <= Mi <= 255	Mt = Mi	(Asynchronous mode)	N/A

**Note:** MT/s means Mega-Transfers per second. If the established data transfer width is 8 bits, the MB/s rate is the same as the MT/s rate. If the established data transfer width is 16 bits, the MB/s is two times the MT/s rate. For example, 10.00 MT/s on a 16 bit transfer width is an effective transfer rate of 20.00 MB/s.

The Drive responds to each Initiator requested REQ/ACK offset as follows:

If the Initiator requests a transfer period that is greater than 448 nSec ( $M > 112$ ), the Target will respond with the same period that the initiator requested but with a REQ/ACK offset of 0.

If the Initiator requests a transfer period in the acceptable range and requests a REQ/ACK offset value that is greater than 15 with an established 16 bit transfer width or greater than 30 with an established 8 bit transfer width, the Target responds with a REQ/ACK offset value of 15 or 30 respectively.

If the Initiator requests a transfer period and an offset value in the acceptable range, the response is equal to the Initiator's requested value.

**Note:** If the REQ/ACK offset value is 0, this indicates asynchronous mode.

If following the Target's response above the Initiator asserts the ATN signal and the first message received is either a Message Parity Error or a Message Reject message, the Target negates the synchronous agreement and goes to asynchronous mode. For the Message Parity Error case, the implied synchronous agreement is reinstated if the Target successfully retransmits the Synchronous Data Transfer Request message to the Initiator (see 3.6.3, "Message In Phase Retry" on page 176). For any other message, the Target completes negotiation and goes to synchronous mode.

### 3.1.21.3 Synchronous Negotiation Started by the Target

If the Target recognizes that negotiation is required, the Target sends a Synchronous Data Transfer Request message to the Initiator with the transfer period equal to 100 nSec ( $M = 25$ ) and the REQ/ACK offset equal to 15 ( $X = 15$ ) for an established 16 bit transfer width or 30 ( $X = 30$ ) for an established 8 bit transfer width. The Initiator must respond by asserting the ATN signal prior to its release of ACK for the REQ/ACK handshake of the last byte of the Synchronous Data Transfer Request message. (see 3.3, "Attention Condition" on page 164). This provides an interlock during the synchronous negotiation. If the Initiator does not assert the ATN signal, the Target goes to asynchronous mode. If the Initiator does assert the ATN signal,

the Target changes to the Message Out phase and receives a message from the Initiator.

If the first message received is a Synchronous Data Transfer Request message, the Target establishes the new data transfer mode. The Drive interprets the Initiator corresponding transfer period as shown in the following table:

Initiator's Response	Target Transfer Period	Maximum Burst Rate
$0 \leq M_i \leq 24$	Send Message Reject (Asynchronous mode)	N/A
$25 \leq M_i \leq 25$	100 nSec	10.00 MT/s
$26 \leq M_i \leq 31$	125 nSec	8.00 MT/s
$32 \leq M_i \leq 37$	150 nSec	6.67 MT/s
$38 \leq M_i \leq 43$	175 nSec	5.71 MT/s
$44 \leq M_i \leq 50$	200 nSec	5.00 MT/s
$51 \leq M_i \leq 56$	225 nSec	4.44 MT/s
$57 \leq M_i \leq 62$	250 nSec	4.00 MT/s
$63 \leq M_i \leq 68$	275 nSec	3.64 MT/s
$69 \leq M_i \leq 75$	300 nSec	3.33 MT/s
$76 \leq M_i \leq 81$	325 nSec	3.08 MT/s
$82 \leq M_i \leq 87$	350 nSec	2.86 MT/s
$88 \leq M_i \leq 93$	375 nSec	2.67 MT/s
$94 \leq M_i \leq 100$	400 nSec	2.50 MT/s
$101 \leq M_i \leq 106$	425 nSec	2.35 MT/s
$107 \leq M_i \leq 112$	450 nSec	2.22 MT/s
$113 \leq M_i \leq 255$	Send Message Reject (Asynchronous mode)	N/A

**Note:** If the corresponding transfer period received from the Initiator indicates a transfer period that is less than 100 nSec ( $M < 25$ ) or greater than 448 nSec ( $M > 112$ ), the Target sends a Message Reject message to the Initiator to indicate asynchronous mode.

The Drive interprets each Initiator requested REQ/ACK offset as follows:

- If the Initiator requests a REQ/ACK offset value that is greater than 15 with an established 16 bit data transfer width or 30 with an established 8 bit data transfer width, the Target changes to the Message In phase, sends a Message Reject message to the Initiator, and goes to asynchronous mode.
- If the value is 0, the Target goes to asynchronous mode. For all other values, the Target interprets the REQ/ACK offset to be equal to the Initiator requested value.

If the first message received from the Initiator is either a Message Parity Error or a Message Reject message, the Target goes to asynchronous mode. For the Message Parity Error case, the synchronous negotiation is restarted if the Target successfully retransmits the Synchronous Data Transfer Request message to the Initiator. (see 3.6.3, "Message In Phase Retry" on page 176).

If the first message received from the Initiator is any other message, the Target goes to asynchronous mode. The Target assumes that the Initiator does not support synchronous mode and does not attempt to renegotiate with this Initiator. (see 3.5,

“Message State/Action Table” on page 168 for more information regarding the Target's response to each message received from the Initiator during synchronous negotiation).

The implied agreement for synchronous operation is not considered to exist by the Target until the Target leaves the Message Out phase, implying that no parity error was detected. If the Target detects a parity error while attempting to receive the message from the Initiator, the Target goes to asynchronous mode. The Target will attempt to resume synchronous negotiation by retrying the Message Out phase (see also 3.6.2, “Message Out Phase Retry” on page 175).

**Note:** If during the Message In phase of negotiations, either Target or Initiator started, ATN is asserted prior to transmission of the last byte of the message and the message is not Message Parity or Message Reject, the Target goes to asynchronous mode. Message Reject and Message Parity Errors are handled as described in 3.1.21.2, “Synchronous Negotiation Started by the Initiator” on page 158 and 3.1.21.3, “Synchronous Negotiation Started by the Target” on page 159.

**Note:** Target initiated synchronous negotiation is selectable via the DSN bit. See 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59 for more information.

### 3.1.22 Wide Data Transfer Request (01h,03h)

A pair of Wide Data Transfer Request messages are exchanged between an Initiator and a Target to establish a data transfer width agreement between the two devices. The Initiator may initiate a wide data transfer negotiation at any time after the LUN has been identified. The Target initiates a wide data transfer negotiation if the Target has not negotiated with the Initiator since the last time the Target was Reset (Power-on Reset, SCSI Bus “hard” Reset, Bus Device Reset message, or Self Initiated Reset).

Target-initiated negotiation occurs either immediately following the Command phase or immediately following the first reconnection. In either case, negotiation occurs before any Data phase between the Target and the Initiator. The Target will negotiate the data transfer width agreement prior to negotiating the synchronous data transfer agreement. If a synchronous data transfer agreement is in effect when a Wide Data Transfer Request message is received, the Target will reset the synchronous agreement to asynchronous mode.

The implied data transfer width agreement remains in effect until the Target is Reset (Power-on Reset, SCSI Bus “hard” Reset, Bus Device Reset message, or Self Initiated Reset) or a new data transfer width agreement is negotiated. If a Reset occurs, the Target goes to eight bit mode.

#### 3.1.22.1 Wide Data Transfer Request Message Definition

Table 103. Wide Data Transfer Request Message Definition		
Byte	Value	Description
0	01h	Extended message
1	02h	Extended message length
2	03h	Wide Data Transfer Request code
3	E	Transfer Width Exponent

The data transfer width is two to the transfer width exponent bytes wide. Valid data transfer widths are 8 bits (E = 00h) and 16 bits (E = 01h). Value of E greater than 01h are reserved.

### 3.1.22.2 Transfer Width Negotiation Started by the Initiator

If the Initiator recognizes that negotiation is required and sends a Wide Data Transfer Request message out, the Target responds by changing to the Message In phase and sending a Wide Data Transfer Request message in to the Initiator prior to transferring any additional message bytes (or any other Information phase bytes) from the Initiator. This provides an interlock during the data transfer width negotiation.

The Drive responds to each Initiator requested transfer width exponent as shown in the following table:

Initiator Request	Target Response	Target Data Transfer Width
Ei = 0	Et = 0	8 Bit Data Transfers
Ei > 0	Et = 1	16 Bit Data Transfers

If following the Target's response above the Initiator asserts the ATN signal and the first message received is either a Message Parity Error or a Message Reject message, the Target negates the data transfer width agreement and goes to 8 bit mode. For the Message Parity Error case, the implied data transfer width agreement is reinstated if the Target successfully retransmits the Wide Data Transfer Request message to the Initiator (see 3.6.3, "Message In Phase Retry" on page 176). For any other message, the Target completes negotiation and goes to the negotiated data transfer width.

### 3.1.22.3 Transfer Width Negotiation Started by the Target

If the Target recognizes that negotiation is required, the Target sends a Wide Data Transfer Request message to the Initiator with the transfer width exponent equal to 1 (E = 1). The Initiator must respond by asserting the ATN signal prior to its release of ACK for the REQ/ACK handshake of the last byte of the Wide Data Transfer Request message. (see 3.3, "Attention Condition" on page 164). This provides an interlock during the wide data transfer negotiation. If the Initiator does not assert the ATN signal, the Target goes to 8 bit mode. If the Initiator does assert the ATN signal, the Target changes to the Message Out phase and receives a message from the Initiator.

If the first message received is a Wide Data Transfer Request message, the Target establishes the new data transfer mode. The Drive interprets the Initiator corresponding transfer width exponent as shown in the following table:

Table 105. Target Response to Initiator's Transfer Period	
Initiator's Response	Target Data Transfer Width
$E_i = 0$	8 Bit Data Transfers
$E_i = 1$	16 Bit Data Transfers
$E_i > 1$	Send Message Reject (8 Bit Data Transfers)

**Note:** If the corresponding transfer width exponent received from the Initiator indicates a data transfer width that is greater than 16 bits ( $E > 1$ ) the Target sends a Message Reject message to the Initiator to indicate 8 bit data transfer mode.

If the first message received from the Initiator is either a Message Parity Error or a Message Reject message, the Target goes to 8 bit data transfer mode. For the Message Parity Error case, the wide data transfer negotiation is restarted if the Target successfully retransmits the Wide Data Transfer Request message to the Initiator. (see 3.6.3, “Message In Phase Retry” on page 176).

If the first message received from the Initiator is any other message, the Target goes to 8 bit data transfer mode. The Target assumes that the Initiator does not support wide data transfers and does not attempt to renegotiate with this Initiator. (see 3.5, “Message State/Action Table” on page 168 for more information regarding the Target's response to each message received from the Initiator during wide data transfer negotiation).

The implied agreement for wide data transfer operation is not considered to exist by the Target until the Target leaves the Message Out phase, implying that no parity error was detected. If the Target detects a parity error while attempting to receive the message from the Initiator, the Target goes to 8 bit data transfer mode. The Target will attempt to resume the wide data transfer negotiation by retrying the Message Out phase (see also 3.6.2, “Message Out Phase Retry” on page 175).

**Note:** If during the Message In phase of negotiations, either Target or Initiator started, ATN is asserted prior to transmission of the last byte of the message and the message is not Message Parity or Message Reject, the Target goes to 8 bit data transfer mode. Message Reject and Message Parity Errors are handled as described in 3.1.22.2, “Transfer Width Negotiation Started by the Initiator” on page 162 and 3.1.22.3, “Transfer Width Negotiation Started by the Target” on page 162.

**Note:** Target initiated negotiation of wide transfers is selectable by the DSN bit. See 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59 for more information.

## 3.2 Supported Message Functions

The implementation of the supported messages also includes the following functions:

- Retry SCSI Command or Status phase

The retry is caused by one of the following error conditions:

- Target detected SCSI bus parity error (Command phase)

- Target receives Initiator Detected Error message during or at the conclusion of an information transfer phase (Command or Status phase)

**Note:** The Initiator may send the Initiator Detected Error message as a result of an Initiator detected SCSI bus parity error or an internal error.

- Retry Message Out phase

The retry is caused by a Target detected SCSI bus parity error during the Message Out phase.

- Retry Message In phase

The retry is caused by the receipt of a Message Parity Error message immediately following a Message In phase.

**Note:** The Initiator may send the Message Parity Error message as a result of an Initiator detected SCSI bus parity error during the Message In phase.

- Receipt of multiple Identify messages

The Initiator is allowed to send multiple Identify messages out in order to toggle the disconnect/reconnect permission bit (bit 6). This may be used to selectively enable or disable disconnect/reconnect permission during portions of a command.

- Message Reject during Target disconnection

If the Initiator rejects the Save Data Pointer message, the Target disables disconnect/reconnect permission. This is equivalent to receiving an Identify message with bit 6 (disconnect/reconnect permission bit) equal to zero. This causes the Target to inhibit the pending disconnection. If the Initiator rejects the Disconnect message, the Target does not disconnect but may attempt to disconnect at a later time. This function may be used to selectively disable disconnection during portions of a command.

---

### 3.3 Attention Condition

The Attention Condition allows an Initiator to inform the Target that a Message Out phase is desired. The Initiator may create the Attention Condition by asserting the ATN signal at any time except during the Arbitration phase or Bus Free phase.

The Initiator must create the Attention Condition by asserting the ATN signal before releasing ACK for the last byte transferred in a bus phase to guarantee that the Attention Condition is honored before transition to a new bus phase. This guarantees a predictable Target response to messages received during the Message Out phase for this Attention Condition. If the ATN signal is asserted later, it may be honored in the current bus phase or the next bus phase and then may not result in the expected action or may result in a Bus Free condition. See Figure 1 on page 165 and Figure 2 on page 166 for an example of the desired time to assert ATN during an information transfer phase.



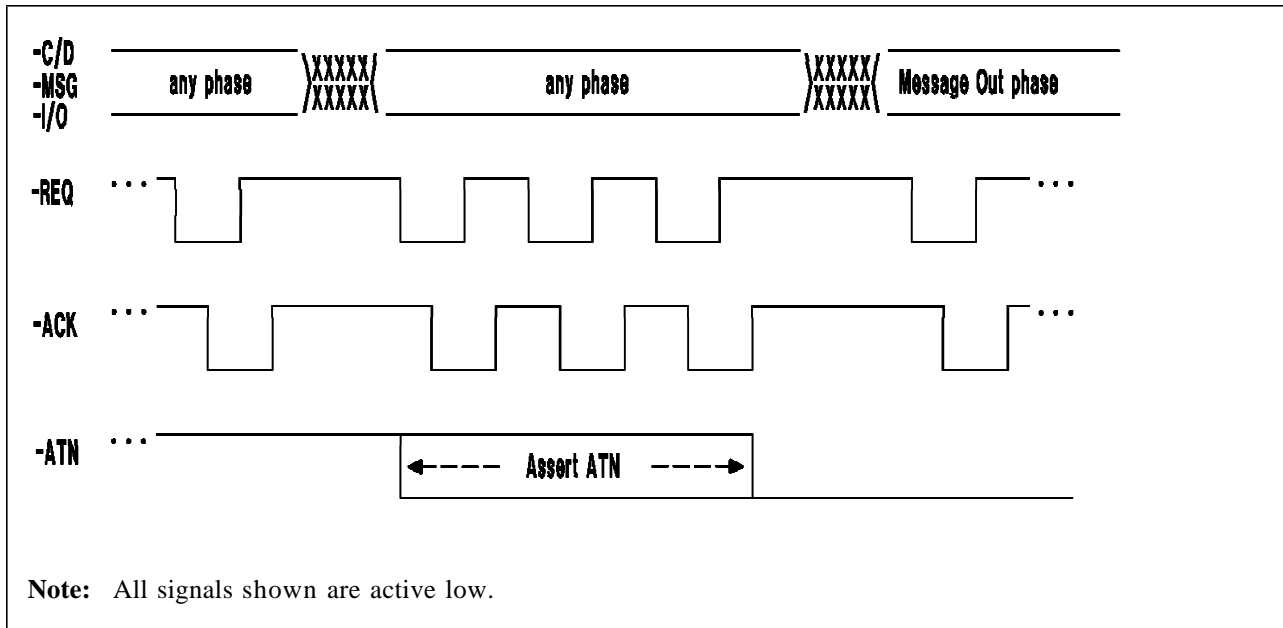


Figure 1. Attention Condition during Asynchronous Information Transfer phase

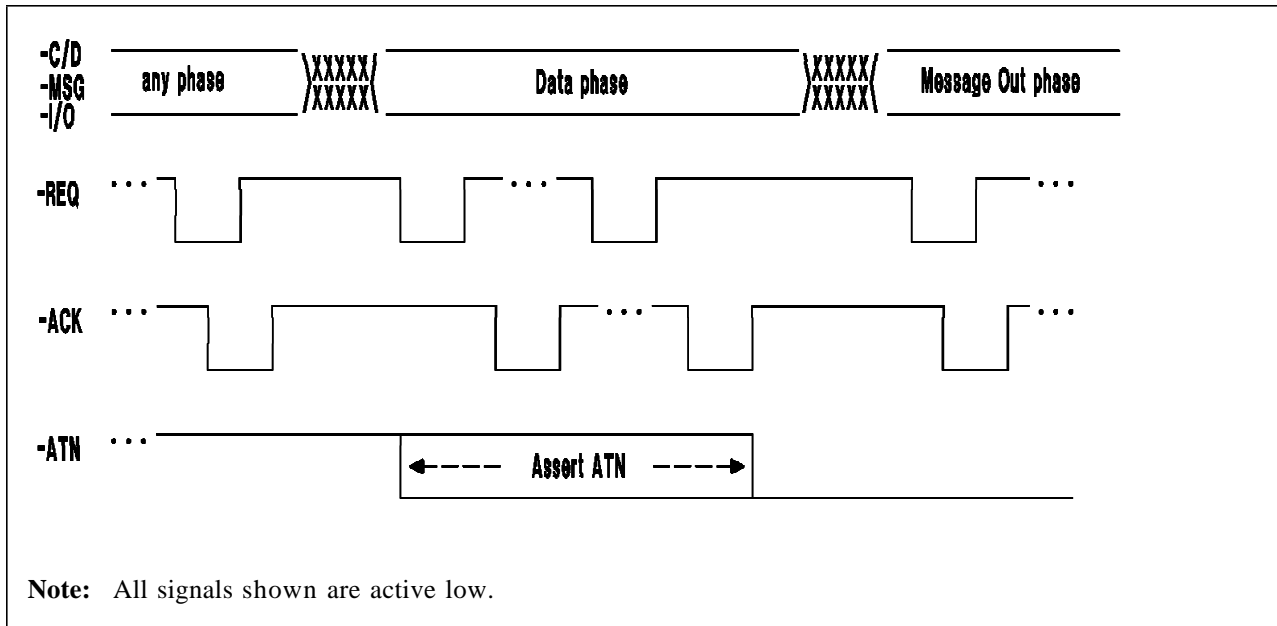


Figure 2. Attention Condition during Synchronous Data Transfer phase

After the Initiator asserts the ATN signal, the Target responds with the Message Out phase as follows:

- If ATN occurs during a Command phase, the Message Out phase occurs after part or all of the Command Descriptor Block has been transferred to the Target. The Initiator must continue REQ/ACK handshakes during the Command phase until the Target enters the Message Out phase.
- If ATN occurs during a Data In or Data Out phase, the Message Out phase occurs after part or all of the data bytes have been transferred and not necessarily on a logical block boundary. The Initiator must continue REQ/ACK handshakes (asynchronous transfer) or ACK pulses to reach an offset of zero (synchronous transfer) until the Target enters the Message Out phase.

**Note:** The Drive SCSI hardware and microcode implementation restricts the Target response to the Attention Condition during a parameter or sense data phase as follows: The Target detects the Attention Condition within 256 bytes after the assertion of the ATN. During a data out phase that is not for parameter data, the target detects the Attention Condition within 2 bytes. During a data in phase that is not for parameter or sense data, the target detects the Attention Condition within 34 bytes. Once the Target detects the Attention Condition, it will wait until all outstanding SCSI ACK pulses have been received before entering the Message Out phase. Any data that is received after attention is asserted will not be written to the disk.

- If ATN occurs during a Status phase, the Message Out phase occurs after the REQ/ACK handshake of the status byte has been completed.
- If ATN occurs during a Message In phase, the target shall enter Message Out phase before it sends another message. This permits a Message Parity Error message from the initiator to be associated with the appropriate message.
- If ATN occurs during a Selection phase and before the Initiator releases the BSY signal, the Message Out phase occurs following the completion of the Selection phase.

- If ATN occurs during a Reselection phase, the Target temporarily ignores the Attention condition and completes the reselection phase and Message In phase to send an Identify message. The Message Out phase occurs after the Target has sent the Identify message to the Initiator. This is indistinguishable from an Attention Condition during the Identify Message In phase.

The Initiator must keep the ATN signal asserted if more than one message byte is to be transferred during the Message Out phase. The Target processes each message byte (multiple-bytes for an extended message) prior to receiving the next message from the Initiator. The Target continues to handshake and process byte(s) in the Message Out phase until ATN goes false, unless one of the following conditions occurs:

- The Target receives an illegal or inappropriate message and goes to the Message In phase to send a Message Reject message.
- The Target detects a catastrophic error condition and goes to the Bus Free phase.

**Note:** If the Target detects a parity error during the Message Out phase, the Target continues to handshake byte(s) in this phase until ATN goes false. The Target does not process any messages received after the parity error is detected. (see 3.6.2, “Message Out Phase Retry” on page 175).

The Initiator must negate the ATN signal while REQ is true and ACK is false during the last REQ/ACK handshake of the Message Out phase to ensure a predictable Target response. If ATN is negated later, the Target may or may not request an additional Message Out byte. If ATN is negated earlier during a multiple-byte Message Out phase, the Target may or may not request additional Message Out byte(s). If ATN is negated prior to the Target entering the Message Out phase, the Target may or may not enter the Message Out phase.

---

## 3.4 SCSI Bus Parity Error Detection

SCSI bus parity errors may be detected by either the Initiator or the Target. The Initiator detects parity errors during the Data In, Status, or Message In phases. If a retry is desired to attempt to recover from one of these parity errors, the Initiator must assert the ATN signal to create an Attention Condition and send either the Initiator Detected Error message (Data In or Status phase) or Message Parity Error message (Message In phase) to the Target. The ATN signal must be asserted within the phase during which the parity error was detected to guarantee proper Target response. See 3.3, “Attention Condition” on page 164 for a description of the proper time to assert the ATN signal and the Target's response to the Attention Condition. See 3.6.7, “Data In Phase Retry” on page 177, 3.6.5, “Status Phase Retry” on page 177, and 3.6.3, “Message In Phase Retry” on page 176 for a description of the retry of each of these phases.

The Target detects parity errors during the Command, Data Out, or Message Out phases. After the parity error is detected, the Target may attempt to retry the phase. The Target parity error detection for each of these phases is defined as follows:

- If a parity error occurs during a Command phase, the retry occurs after part or all of the Command Descriptor Block has been transferred to the Target. The Initiator must continue REQ/ACK handshakes during the Command phase until the Target attempts the retry. See 3.6.4, “Command Phase Retry” on page 176 for a description of the retry of this phase.

- If a parity error occurs during a Data Out phase, the Target responds to the parity error after part or all of the data bytes have been transferred and not necessarily on a logical block boundary. The Initiator must continue REQ/ACK handshakes (asynchronous transfer) or ACK pulses to reach an offset of zero (synchronous transfer) until the Target terminates the Data phase. There is no guarantee that the medium will or will not be altered with the data received prior to the parity error.

**Note:** The Target detects the parity error immediately. The byte with the parity error and any data received after that byte will not be written to the disk.

- If a parity error occurs during a Message Out phase, the retry occurs after all remaining Message Out bytes have been received from the Initiator (indicated by ATN false). The Target does not process any messages received after the parity error is detected. The Initiator must continue REQ/ACK handshakes during the Message Out phase until ATN is false and the Target attempts the retry. See 3.6.2, “Message Out Phase Retry” on page 175 for a description of the retry of this phase.

---

## 3.5 Message State/Action Table

This section describes the Target's action or response to inputs that affect the SCSI message systems during various stages of command execution. The description is in the form of a state/action table, that describes the Target's Action in response to various inputs during command execution states.

The inputs are factors that influence the SCSI Message System. The inputs include:

- Messages received from the Initiator in response to an Attention Condition
- Target detected SCSI bus parity errors

The state represents the current SCSI bus phase interrupted by the input. In addition, some state/input combinations are divided into two States to represent either the first or second error to be retried during this SCSI bus connection.

The Action shows the Target's response to the Input that interrupted the present State. The Action also includes the Target's Next State in response to the input.

The Next State may be examined to show the Target's Action in response to the next input. This process may then be repeated to show the series of Target actions in response to a series of messages (or input) received from the Initiator. The initial State of the Target (prior to the assertion of ATN) and the succeeding actions (internal indicators set by actions) determines how the Target proceeds (for example, continue previous information phase, retry previous information phase, start disconnection, send Check Condition status, and so on).

The entries in the Message State/Action Table should be interpreted as shown below:

Table 106. Action Table Example	
SCSI Phase or State Interrupted	<i>Present SCSI State</i>
SCSI State Number	<i>State Number</i>
<i>Input 1</i>	<i>ACTION 1</i> <i>ACTION 2</i> <i>(NEXT STATE)</i>
<i>Input 2</i>	<i>ACTION 1A</i> <i>ACTION 2A</i> <i>(NEXT STATE)</i>
	<i>ACTION 1B</i> <i>ACTION 2B</i> <i>(NEXT STATE)</i>

- ACTION 1 & ACTION 2 are both done in response to Input 1.
- ACTION 1A & ACTION 2A are both done in response to Input 2 if this is the first error for this connection.
- ACTION 1B & ACTION 2B are both done in response to Input 2 if this is the second error for this connection.

The following Message Action Legend defines the meaning of each Target Action entry in the Message State/Action Table.

compact.

Table 107. Message Action Legend		
AT	=	Return to Asynchronous Transfer mode
BF	=	Go to Bus Free phase
CC	=	Send Command Complete message
CD	=	Clear internal indication of disconnect/reconnect permission
CHK	=	Send Check Condition status
CS	=	Clear Sense for the Initiator
DATA	=	Go to Data Phase
DATA1	=	Go to Data Phase if a TTD message was received for this process, otherwise SS and BF. If data transfer is not ready, DM and BF.
DM	=	Send Disconnect message
DW	=	Return to default Wide Data Transfer mode (8 bit)
ET	=	Establish Synchronous Transfer mode
EW	=	Establish Wide Data Transfer mode
ID	=	Send Identify message
IW	=	Send Ignore Wide Residue message
LCC	=	Send Linked Command Complete message
MR	=	Send Message Reject message
na	=	not applicable
ND	=	Set internal indication to suspend pending disconnection
NP	=	Continue to the next SCSI phase
PS	=	Preserve Sense Data for all Initiators
RMO	=	Flush Message Out bytes until ATN=0 and retry Message Out phase
RP	=	Send Restore Pointers message
RST	=	Perform Hard Reset
SDP	=	Send Save Data Pointer message
SM	=	Send Synchronous Data Transfer Request message
SQ	=	Send Simple Queue Tag message
SS	=	Abort command execution for Identified Initiator/LUN and set appropriate sense data if the LUN is valid
SS1	=	Same as SS unless error on message immediately following ID message and Tagged Queuing is enabled. In this case PS
SVD	=	Save internal indication of disconnect/reconnect permission
SVL	=	Save LUN second LUN different from the first
SVQ	=	Establish ITLQ Nexus and Save TAG if immediately after ID MSG, else message reject
SVT	=	Set internal indication that a TTD message was received for this process
SW	=	Send Wide Data Transfer Request message

Table 108. Message State/Action Table (Part 1)

SCSI Phase or State Interrupted	Selection Phase (LUN not Identified)	Identify Message Out (after Selection)	Queue Tag Message Out (after ID Message)	Command Phase	Message In Restore Pointers (03h) for Command Phase	Message In Save Data Pointer (02h)	Message In Disconnect (04h)	Message In Identify (8Xh)	Message In Simple Queue Tag (20h)
	1	2	3	4	5	6	7	8	9
SCSI State Number									
Valid Identify (80H-87H, COH-C7H)	SVL SVD (2)	SVD (25)	SVD (25)	SVD (25)	SVD (25)	SVD (25)	SVD (25)	SVD (25)	SVD (25)
Invalid Identify with Reserved Bits = 1	SS CHK	SS CHK	SS CHK	SS CHK	SS CHK	SS CHK	SS CHK	SS CHK	SS CHK
Invalid Identify with Different LUN	na	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF
Queue Tag Message (20h, 21h, 22h)	PS BF	SVQ	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)
Target Transfer Disable Message (13h)	PS BF	SVT (4)	SVT (4)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)
Continue I/O Process Message (12h)	PS BF	DATA1 (14-17)	DATA1 (14-17)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)
Sync Data Transfer Request (01h,01h) with Acceptable Transfer Mode	PS BF	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)
Sync Data Transfer Request (01h,01h) with Unacceptable Transfer Mode	PS BF	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)
Wide Data Transfer Request (01h,03h)	PS BF	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)
Initiator Detected Error (05H)	PS BF	SS CHK (19)	SS CHK (19)	RP (5) SS CHK (19)	SS CHK (19)	SS CHK (19)	SS CHK (19)	SS CHK (19)	SS CHK (19)
Abort (06H)	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Message Reject (07H)	PS BF	MR (23)	MR (23)	MR (23)	MR (23)	ND CD NP	ND NP	ID (8) SS BF	SQ (9) SS BF
No Operation (NOP) (08H)	PS BF	NP	NP	NP	NP	NP	NP	NP	NP
Message Parity Error (09H)	PS BF	SS BF	SS BF	SS BF	SS BF	SDP (6) SS BF	DM (7) SS BF	ID (8) SS BF	SQ (9) SS BF
Bus Device Reset (0CH)	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF
Abort Tag Message (0Dh)	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Clear Queue Message (0Eh)	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Any Unsupported or Illegal Message	PS BF	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)
Target Detected SCSI Parity Error During Command	na	na	na	RP (5) SS CHK (19)	na	na	na	na	na
Target Detected SCSI Parity Error During Message Out (response to ATN)	RMO (1) PS BF	RMO (2) SS1 BF	RMO (3) SS BF	RMO (4) SS BF	RMO (6) SS BF	RMO (6) SS BF	RMO (7) SS BF	RMO (8) SS BF	RMO (9) SS BF

Table 109. Message State/Action Table (Part 2)

SCSI Phase or State Interrupted	Message In Sync Data Xfer Request (01h,01h) (Initiator Started)	Message In Sync Data Xfer Request (Target Started)	Message In Sync Data Xfer Request (01h,03h) (Initiator Started)	Message In Sync Data Xfer Request (Target Started)	Message In Wide Data Xfer Request (01h,03h) (Target Started)	Data In Phase for Logical Block (Drive) Data	Data Out Phase for Logical Block (Drive) Data	Data In Phase for Command Parameter/Sense Data	Data Out Phase for Command Parameter/Sense Data	Message In Ignore Wide Residue (23h)
SCSI State Number	10	11	12	13	14	15	16	17	18	
Valid Identify (80H-87H, C0H-C7H)	SVD (25)	AT SVD (25)	SVD (25)	DW SVD SM (11)	SVD SS CHK (19)	SVD SS CHK (19)	SVD (25)	SVD (25)	SVD (25)	SVD (25)
Invalid Identify with Reserved Bits = 1	SS CHK	AT SS CHK	SS CHK	DW SS CHK	SS CHK	SS CHK	SS CHK	SS CHK	SS CHK	SS CHK
Invalid Identify with Different LUN	SS BF	AT SS BF	SS BF	DW SS BF	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF
Queue Tag Message (20h, 21h, 22h)	MR (23)	AT MR (23)	MR (23)	DW MR (23)	MR SS CHK (19)	MR SS CHK (19)	MR (23)	MR (23)	MR (23)	MR (23)
Target Transfer Disable Message (13h)	MR (23)	AT MR (23)	MR (23)	DW MR (23)	MR SS CHK (19)	MR SS CHK (19)	MR (23)	MR (23)	MR (23)	MR (23)
Continue I/O Process Message (12h)	MR (23)	AT MR (23)	MR (23)	DW MR (23)	MR SS CHK (19)	MR SS CHK (19)	MR (23)	MR (23)	MR (23)	MR (23)
Sync Data Transfer Request (01h,01h) with Acceptable Transfer Mode	SM ET (10)	ET NP	SM ET (10)	DW SM ET (10)	SM ET SS CHK (19)	SM ET SS CHK (19)	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)
Sync Data Transfer Request (01h,01h) with Unacceptable Transfer Mode	AT SM (10)	AT MR (23)	AT SM (10)	DW AT SM (10)	AT SM SS CHK (19)	AT SM SS CHK (19)	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)
Wide Data Transfer Request (01h,03h)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SM (11)	AT EW SW SS CHK (19)	AT EW SW SS CHK (19)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)
Initiator Detected Error (05H)	SS CHK (19)	AT SS CHK (19)	SS CHK (19)	DW SS CHK (19)	SS CHK (19)	SS CHK (19)	SS CHK (19)	SS CHK (19)	SS CHK (19)	SS CHK (19)
Abort (06H)	PS BF	AT PS BF	PS BF	DW PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Message Reject (07H)	AT NP	AT NP	DW NP	DW SM (11)	MR SS CHK (19)	MR SS CHK (19)	MR (23)	MR (23)	MR (23)	IW (18) SS BF
No Operation (NOP) (08H)	NP	AT NP	NP	DW SM (11)	SS CHK (19)	SS CHK (19)	NP	NP	NP	NP
Message Parity Error (09H)	SM (10) AT SS BF	SM (11) AT SS BF	SW (12) DW SS BF	SW (13) DW SS BF	SS BF	SS BF	SS BF	SS BF	SS BF	IW (18) SS BF
Bus Device Reset (0CH)	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF
Abort Tag Message (0Dh)	PS BF	AT PS BF	PS BF	DW PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Clear Queue Message (0Eh)	PS BF	AT PS BF	PS BF	DW PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Any Unsupported or Illegal Message	MR (23)	AT MR (23)	MR (23)	DW MR (23)	MR SS CHK (19)	MR SS CHK (19)	MR (23)	MR (23)	MR (23)	MR (23)
Target Detected SCSI Parity Error During Command	na	na	na	na	na	na	na	na	na	na
Target Detected SCSI Parity Error During Message Out (response to ATN)	RMO (10) AT SS BF	RMO (11) AT SS BF	RMO (12) DW SS BF	RMO (13) DW SS BF	RMO (14) SS BF	RMO (15) SS BF	RMO (16) SS BF	RMO (17) SS BF	RMO (18) SS BF	RMO (18) SS BF



Table 110. Message State/Action Table (Part 3)

SCSI Phase or State Interrupted	Status Phase	Message In Restore Pointers (03h) for Status Phase	Message In Command Complete (00h)	Message In Linked Command Complete (0Ah or 0Bh)	Message In Message Reject (07h)	Message In Disconnect (04h) during Target Transfer Disable sequence	ATN Asserted Between Phases
SCSI State Number	19	20	21	22	23	24	25
Valid Identify (80H-87H, C0H-C7H)	SVD (25)	SVD (25)	SVD (25)	SVD (25)	SVD (25)	SVD (25)	SVD (25)
Invalid Identify with Reserved Bits = 1	SS BF	SS BF	PS BF	SS BF	SS CHK	SS CHK	SS CHK
Invalid Identify with Different LUN	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF
Queue Tag Message (20h, 21h, 22h)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)
Target Transfer Disable Message (13h)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)
Continue I/O Process Message (12h)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)
Sync Data Transfer Request (01h,01h) with Acceptable Transfer Mode	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)	SM ET (10)
Sync Data Transfer Request (01h,01h) with Unacceptable Transfer Mode	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)	AT SM (10)
Wide Data Transfer Request (01h,03h)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)	AT EW SW (12)
Initiator Detected Error (05H)	RP (20)	SS BF	PS BF	SS BF	SS CHK (19)	SS CHK (19)	SS CHK (19)
	SS BF	SS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Abort (06H)	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Message Reject (07H)	MR(23)	SS BF	PS BF	SS BF	SS BF	DATA (14-17)	MR(23)
No Operation (NOP) (08H)	NP	NP	NP	NP	NP	NP	NP
Message Parity Error (09H)	SS BF	SS BF	CC (21)	LCC (22)	MR (23)	DM (24)	SS BF
	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF
Bus Device Reset (0CH)	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF	RST BF
Abort Tag Message (0Dh)	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Clear Queue Message (0Eh)	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF	PS BF
Any Unsupported or Illegal Message	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)	MR (23)
Target Detected SCSI Parity Error During Command	na	na	na	na	na	na	na
Target Detected SCSI Parity Error During Message Out (response to ATN)	RMO (19)	SS BF	RMO (21)	RMO (22)	RMO (23)	RMO (24)	RMO (25)
	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF

The following points further describe the Target's response to inputs that affect the SCSI message systems.

- Acceptable messages following Selection

The first message sent from the Initiator to the Target following a Selection phase with the ATN signal active must be the Identify, Abort, or Bus Device Reset message. If any other message is received, the Target changes to the Bus Free phase to signal a catastrophic error condition.

- Messages following Identify

The Identify message establishes the I\_T\_L nexus. The Identify message may be immediately followed by other messages, such as a Queue Tag message, a Target Transfer Disable Message, a Continue I/O Process Message, or the first message of a pair of Synchronous Data Transfer Request messages.

- Queue Tag Messages with Tagged Queuing disabled

Any Queue Tag Message received while Tagged Queuing is disabled is treated like an unsupported message.

- Actions after an Incorrect Initiator Connection

If an Initiator establishes an incorrect connection (see 4.1.3, “Incorrect Initiator Connection” on page 181), the drive will attempt to return check condition status to indicate this condition. The check condition status for this condition will have priority over every BUS FREE action in the message state action table except for the BUS FREE following the Bus Device Reset Message.

- Messages without properly negated ATN

The Initiator is required to end the Message Out phase (by negating ATN) when it sends certain messages (see 3.1, “Supported Messages” on page 149). If the Initiator fails to negate the ATN signal to end the Message Out phase following these messages, the Target changes to the Bus Free phase to signal a catastrophic error condition.

- Retry for unsupported messages

If the ATN signal remains asserted following the receipt of an unsupported message, the Target will Message Reject the unsupported message. If the ATN signal is still asserted, the Target changes to the Message Out phase to receive the next message from the Initiator.

---

## 3.6 SCSI Bus Related Error Handling Protocol

This protocol is used to handle errors that threaten the integrity of a connection between the Target and an Initiator.

### 3.6.1 Unexpected Bus Free Phase Error Condition

There are several error conditions that cause the Target to immediately change to the Bus Free phase, regardless of the state of the ATN signal. The Target does not attempt to reconnect to the Initiator to complete the operation that was in progress when the error condition was detected. The Initiator should interpret this as a catastrophic error condition.

If the LUN was identified by the Target prior to the error condition, then the Target aborts the command and generates sense data for this Initiator/LUN to describe the

cause of the catastrophic error. The Initiator may retrieve this sense data by issuing a Request Sense command to this LUN.

**Note:** The Request Sense command may fail if the catastrophic error condition persists.

If the LUN was not identified by the Target prior to the error condition, then the Target does not affect the sense data or the operation of any currently executing command for this Initiator or any other Initiator.

### 3.6.2 Message Out Phase Retry

A Message Out phase retry may be caused by a Target detected parity error during the Message Out phase.

When an error occurs for the first time, the Target retries the Message Out phase as follows:

- Continue the REQ/ACK handshakes in the Message Out phase until the Initiator negates the ATN signal. The Target ignores all the remaining Message Out phase bytes received after the parity error is detected until the ATN signal is negated.
- Assert the REQ signal prior to changing to any other phase. After detecting this condition, the Initiator must resend all of the previous message byte(s) sent during this continuous Message Out phase. The Target assumes that the message byte(s) are sent in the same order as previously sent during this phase. The Target does not act on any message which it acted on the first time received. When resending more than one message byte, the Initiator must assert the ATN signal prior to asserting ACK for the first message byte and must maintain ATN asserted until the last message byte is sent, as described in 3.3, “Attention Condition” on page 164.
- Repeat the transfer of the Message Out phase byte(s). If the Target receives all of the message byte(s) successfully (without detecting a parity error), it indicates that it does not wish to retry the Message Out phase by changing to any other information transfer phase and transferring at least one byte. The Target may also indicate that it has successfully received an Abort, Abort Tag, Clear Queue or Bus Device Reset message by changing to the Bus Free phase.

**Note:** This situation is indistinguishable from a second Message Out phase parity error, detected while attempting to receive an Abort, Abort Tag, Clear Queue or Bus Device Reset message.

When an error occurs as a result of a retry and the LUN has been previously identified, the Target terminates the current command as follows:

- Change to the Bus Free phase, regardless of the state of the ATN signal. This signals a catastrophic error condition to the Initiator.
- Abort the command and set the sense data to Aborted Command/Interface Parity Error (0Bh/47h).

When an error occurs as a result of a retry and the LUN has not been previously identified, the Target terminates the current command as follows:

- Change to the Bus Free phase, regardless of the state of the ATN signal. This signals a catastrophic error condition to the Initiator.

- The Target does not affect the sense data or the operation of any currently executing command for this Initiator or any other Initiator.

### 3.6.3 Message In Phase Retry

A Message In phase retry may be caused by a Target receipt of a Message Parity Error message immediately following a Message In phase. The Initiator may send the Message Parity Error message as a result of an Initiator detected SCSI bus parity error during the Message In phase.

When an error occurs for the first time, the Target retries the Message In phase as follows:

- Change to the Message In phase and repeat the transfer of the last message sent to the Initiator. If the last message sent to the Initiator was an extended message, then multiple bytes are sent to repeat the transfer of the entire extended message.

When an error occurs as a result of a retry, the Target terminates the current command as follows:

- Change to the Bus Free phase, regardless of the state of the ATN signal. This signals a catastrophic error condition to the Initiator.
- Abort the command and set the sense data to Aborted Command/Interface Parity Error (0Bh/47h).

### 3.6.4 Command Phase Retry

A Command Phase Retry may be caused by a Target detected SCSI bus parity error during the Command phase or a Target receipt of an Initiator Detected Error message immediately following the Command phase. The Initiator may send the Initiator Detected Error message as a result of an Initiator internal error, detected during the Command phase.

When an error occurs for the first time and the LUN has been previously identified by an Identify message, the Target retries the Command phase as follows:

- Change to the Message In phase and send a Restore Pointers message to the Initiator.
- If the Initiator Message Rejects the Restore Pointers message, then the Target terminates the Command phase retry and responds as defined for an error detected during the Command phase retry. If the Initiator accepts the Restore Pointers message, then the Target repeats the transfer of the Command phase bytes.

When an error occurs during a Command phase retry and the LUN has been previously identified by an Identify message, the Target terminates the current command as follows:

- Send Check Condition status followed by Command Complete message to the Initiator.
- Abort the command and set the sense data based on the cause of the error that interrupted the Command phase. The sense data is set to Aborted Command/Interface Parity Error (0Bh/47h) for a Target detected SCSI bus parity error or Aborted Command/Initiator Detected Error (0Bh/48h) for an Initiator Detected Error message.

If this is the first error for retry and the LUN has not been previously identified by an Identify message, the Target terminates the current command as follows:

- Change to the Bus Free phase to signal a catastrophic error condition to the Initiator.
- The Target does not affect the sense data or the operation of any currently executing command for this Initiator or any other Initiator.

### 3.6.5 Status Phase Retry

A Status phase retry may be caused by a Target receipt of an Initiator Detected Error message immediately following the Status phase. The Initiator may send the Initiator Detected Error message as a result of a SCSI bus parity error or internal error, detected by the Initiator during the Status phase.

The first time an error occurs the Target retries the Status phase as follows:

- Change to the Message In phase and send a Restore Pointers message to the Initiator.
- If the Initiator Message Rejects the Restore Pointers message, then the Target terminates the Status phase retry and responds as defined for an error detected during the Status phase retry. If the Initiator accepts the Restore Pointers message, the Target repeats the transfer of the Status phase byte.

When an error occurs during a Status phase retry, the Target terminates the current command as follows:

- Change to the Bus Free phase, regardless of the state of the ATN signal. This signals a catastrophic error condition to the Initiator.
- Abort the command and set the sense data to Aborted Command/Initiator Detected Error (0Bh/48h).

### 3.6.6 Data Out Phase Retry

Data Out phase retries are not supported. If an error occurs during a Data Out phase, the Target terminates the current command as follows:

- Send Check Condition status followed by Command Complete message to the Initiator.
- Abort the command and set the sense data based on the cause of the error which interrupted the Data Out phase. The sense data is set to Aborted Command/Interface Parity Error (0Bh/47h) for a Target detected SCSI bus parity error, Hardware Error/Internal Controller Error (04h/44h) for a Target detected internal error, or Aborted Command/Initiator Detected Error (0Bh/48h) for an Initiator Detected Error message.

### 3.6.7 Data In Phase Retry

Data In phase retries are not supported. If an error occurs during a Data In phase, the Target terminates the current command as follows:

- Send Check Condition status followed by Command Complete message to the Initiator.
- Abort the command and set the sense data based on the cause of the error which interrupted the Data In phase. The sense data is set to Hardware Error/Internal Controller Error (04h/44h) for a Target detected internal error, or

Aborted Command/Initiator Detected Error (0Bh/48h) for an Initiator Detected Error message.

### 3.6.8 Identify Message In Phase Retry

An Identify Message In retry may be caused by a Target receipt of the Message Reject or Message Parity Error message immediately following an Identify Message In phase.

When an error occurs for the first time the Target retries the Identify Message In phase as follows:

- Change to the Message In phase and repeat the transfer of the Identify message to the Initiator.

When an error occurs as a result of a retry, the Target terminates the current command as follows:

- Change to the Bus Free phase, regardless of the state of the ATN signal. This signals a catastrophic error condition to the Initiator.
- Abort the command and set the sense data to Aborted Command/Message Error (0Bh/43h) or Aborted command/parity error (0Bh/47h).

### 3.6.9 Simple Queue Tag Message In Phase Retry

A Simple Queue Tag Message In retry may be caused by a Target receipt of the Message Reject or Message Parity Error message immediately following a Simple Queue Tag Message In phase.

When an error occurs for the first time the Target retries the Simple Queue Tag message In phase as follows:

- Change to the Message In phase and repeat the transfer of the Simple Queue Tag message to the Initiator.

When an error occurs as a result of a retry, the Target terminates the current command as follows:

- Change to the Bus Free phase, regardless of the state of the ATN signal. This signals a catastrophic error condition to the Initiator.
- Abort the command and set the sense data to Aborted Command/Message Error (0Bh/43h) or Aborted command/parity error (0Bh/47h).

---

## 4.0 Additional Information

This chapter provides additional information or descriptions of various functions, features, or operating modes supported by the Target that are not fully described in previous chapters.

---

### 4.1 SCSI Protocol

There are various operating conditions that prevent the Target from executing a SCSI command. This section describes each of these operating conditions and their relative priority.

#### 4.1.1 Priority of SCSI Status Byte Reporting

After establishing the I\_T\_L nexus or I\_T\_L\_Q nexus, the Target must first determine whether command execution is allowed. Execution is deferred until a later time if the command must be added to the command queue. Execution may also be prevented by an internal Target condition that requires the reporting of a *Check Condition Status*, *Busy Status*, *Reservation Conflict Status*, or *Queue Full Status*. There are several different internal conditions that may cause *Check Condition Status* to be reported and Sense data to be generated. It is possible for more than one of these conditions to exist at the same time. The order in which the Target checks for each of these conditions determines their priority (highest priority first) as follows:

1. *Check Condition Status* for Invalid Logical Unit Number (see 4.1.2, “Invalid LUN Processing” on page 180)
2. *Check Condition Status* for Incorrect Initiator Connection (see 4.1.3, “Incorrect Initiator Connection” on page 181)
3. *Busy Status* or *Queue Full Status*, or add command to command queue (see 4.1.4, “Command Processing During Execution of Active I/O Process” on page 182)
4. *Check Condition Status* for Unit Attention condition (see 4.1.5, “Unit Attention Condition” on page 184)
5. *Check Condition Status* during start-up or format operations (see 4.1.6, “Command Processing During Start-up and Format Operations” on page 187)
6. *Check Condition Status* for Internal Error condition (see 4.1.7, “Internal Error Condition” on page 188)
7. *Check Condition Status* for Deferred Error condition (see 4.1.8, “Deferred Error Condition” on page 189)
8. *Check Condition Status* for Degraded Mode condition (see 4.1.9, “Degraded Mode” on page 190)
9. *Reservation Conflict Status* (see 4.1.10, “Command Processing While Reserved” on page 202)
10. *Check Condition Status* for invalid command opcode
11. *Check Condition Status* for invalid command descriptor block

The highest priority internal condition that prevents command execution is reported by the Target, provided there is no bus error.

For all Check Conditions, Sense data is built by the Target provided a valid LUN address is known. Sense data is cleared by the Target upon receipt of any subsequent command to the LUN from the Initiator receiving the *Check Condition Status*.

## 4.1.2 Invalid LUN Processing

An invalid LUN may be identified by the Target after receiving the first Identify message or the CDB if no previous Identify message has been received. If an Identify message is received prior to the Command phase, then the LUN specified in the CDB is ignored. The Target accepts the first Identify message with an invalid LUN, without returning a Message Reject message. The Target also continues to the Command phase to receive the CDB from the Initiator.

The Target's response to an invalid LUN varies with the command, as follows:

<b>Inquiry</b>	Execute the command, return the Inquiry data that indicates unknown device type (byte 0 = 7Fh), and return <i>Good Status</i> . All other bytes are valid (see 1.3, “Inquiry” on page 28).
<b>Request Sense</b>	Execute the command, return the sense data with the Sense Key set to <i>Illegal Request</i> and the Additional Sense Code and Qualifier set to <i>Logical Unit Not Supported</i> , and return <i>Good Status</i> (see also 1.20, “Request Sense” on page 121).
<b>All Others</b>	Do not execute the command and return <i>Check Condition Status</i> . Since the Initiator is not addressing a valid LUN, no Sense data is built for the failed command.

In all cases, the Target's response to the command for an invalid LUN does not affect the current execution of a command on a valid LUN for this Initiator or any other Initiator.

An Identify message with a different LUN from the one in the first Identify message may be received before a Bus Free phase has occurred. In this case, the second LUN is considered to be invalid. This causes the Target to go to the Bus Free phase to signal a catastrophic error.



### 4.1.3 Incorrect Initiator Connection

It is an *Incorrect Initiator Connection* error if any of the following occurs:

- an Initiator attempts to establish an I\_T\_L nexus when an I/O process (either queued or active) with an I\_T\_L nexus already exists from a previous connection with the same Initiator

**Note:** It is not an Incorrect Initiator Connection to reconnect to an already established I\_T\_L nexus if it is in order to send an *Abort*, *Abort Tag*, *Continue I/O Process*, *Clear Queue*, or *Bus Device Reset* message during the same MESSAGE OUT phase as the *Identify* message.

- an Initiator attempts to establish an I\_T\_L\_Q nexus when an I\_T\_L nexus already exists from a previous connection with the same Initiator
- an Initiator attempts to establish an I\_T\_L nexus when an I\_T\_L\_Q nexus already exists from a previous connection with the same Initiator

**Note:** It is not an Incorrect Initiator Connection to send a Request Sense command without a Queue Tag message when sense is pending on the logical unit for the Initiator that issues the Request Sense command.

- an Initiator attempts to establish an I\_T\_L\_Q nexus when an I/O process (either queued or active) with the same I\_T\_L\_Q nexus already exists from a previous connection

**Note:** It is not an Incorrect Initiator Connection to reconnect to an already established I\_T\_L\_Q nexus if it is in order to send an *Abort*, *Abort Tag*, *Continue I/O Process*, *Clear Queue*, or *Bus Device Reset* message during the same MESSAGE OUT phase as the *Identify* message.

If any of the above errors occur, all queued I/O processes and active I/O processes associated with the issuing Initiator on the specified logical unit are terminated. The current I/O process is ended with a *Check Condition Status*, the sense key is set to ***Aborted Command*** and the additional sense code is set to ***Overlapped Commands Attempted***. Status is only returned for the current I/O process.

#### 4.1.4 Command Processing During Execution of Active I/O Process

When the Target is not executing any active I/O processes, a new I/O process is permitted to execute (unless execution is prevented by another internal Target condition listed in 4.1.1, “Priority of SCSI Status Byte Reporting” on page 179).

If an active I/O process does exist when the Target receives a new command, then the Target determines if:

- *Check Condition Status* with Sense Key = **Aborted Command** is returned for an **Overlapped Commands Attempted** error
- the command is permitted to execute
- the command is added to the command queue
- *Queue Full Status* is returned
- *Busy Status* is returned

If an active I/O process does exist when the Target receives a new command, then the Target determines how the new command should be handled based on the following rules:

- *Check Condition Status* is returned with Sense Key set to **Aborted Command** for an **Overlapped Commands Attempted** error if:
  - See 4.1.3, “Incorrect Initiator Connection” on page 181 for the conditions which cause this error.
- The command is permitted to execute if:
  - the command is an Inquiry or Request Sense command. (see 4.2, “Priority Commands” on page 203.)
- *Check Condition Status* is returned with Sense Key set to **Logical Unit Not Ready** if:
  - the start-up operation or format operation is an active process. See 4.1.6, “Command Processing During Start-up and Format Operations” on page 187 for the exact conditions which cause this response.  
**Note:** If a Unit Attention is pending when this condition exists, the Sense Key is set to **Unit Attention** rather than **Logical Unit Not Ready** since Unit Attention has a higher reporting priority. (See 4.1.1, “Priority of SCSI Status Byte Reporting” on page 179).
- The command is permitted to execute if:
  - the conditions to execute concurrently are met. (See 4.4, “Concurrent Command Processing” on page 207).
- The command is added to the command queue for an I\_T\_L nexus if:
  - no Queue Tag message was received during the connection which established the I/O process, and
  - disconnection is allowed for the current I/O process, and
  - there is no queued I/O process or active I/O process corresponding to the I\_T\_L nexus for the current I/O process, and
  - the command is not linked to a previous command.
- The command is added to the command queue for an I\_T\_L\_Q nexus if:

- a Queue Tag message was received during the connection which established the I/O process, and
  - Tagged Queuing is enabled (DQue = 0), and
  - an I/O process (either active or queued) exists at the Target for this Initiator, and
  - disconnection is allowed for the current I/O process, and
  - there is no queued I/O process or active I/O process corresponding to the I\_T\_L\_Q nexus for the current I/O process, and
  - the command is not linked to a previous command.
- *Queue Full Status* is returned if:
    - the command would otherwise be queued (according to the rules described above) but the command queue is full and all slots are utilized, or
    - the command would otherwise be queued (according to the rules described above) but all of the available command queue slots not reserved for use by another Initiator are utilized, or
    - Tagged Queuing is enabled (DQue = 0) and a Format Unit command was previously queued but has not yet begun execution, or
    - Tagged Queuing is enabled (DQue = 0) and a Start Unit command was previously queued but has not yet begun execution.
  - *Busy Status* is returned if:
    - Tagged Queuing is disabled (DQue = 1) and a Format Unit command was previously queued but has not yet begun execution, or
    - Tagged Queuing is disabled (DQue = 1) and a Start Unit command was previously queued but has not yet begun execution, or
    - the command would otherwise be queued (according to the rules described above) but disconnection is not allowed for the current I/O process.

If a command is queued, command execution may still be prevented at a later time when the command is dequeued to become an active I/O process. This occurs if command execution is prevented by another internal Target condition listed in 4.1.1, “Priority of SCSI Status Byte Reporting” on page 179 at the time the command is dequeued.

## 4.1.5 Unit Attention Condition

The Target generates a Unit Attention condition when one of the following occurs:

- Target has been reset

This includes Power-On reset, SCSI Bus reset, Bus Device Reset message, or Self Initiated reset. In all of these cases, a Unit Attention condition is generated for each Initiator. The Additional Sense Code and Additional Sense Code Qualifier reported is ***Power On, Reset, or Bus Device Reset occurred***.

- Mode Select command has been executed

In this case, a Unit Attention condition is generated for all Initiators except the one that issued the Mode Select command. The Additional Sense Code and Additional Sense Code Qualifier reported is ***Mode Parameters Changed***. The Unit Attention condition is generated if any of the current page parameters are set by the Mode Select command. The Target does not check to see that the old parameters are different from the new parameters. For example: If the Initiator issues a Mode Sense command with a page code to report the current values followed by a Mode Select command with the same parameter list, a Unit Attention condition is generated despite the fact that the current parameters were not changed from their previous value. However, if the Target detects an illegal parameter or error condition prior to modifying the current parameters, a Unit Attention condition is not generated since the parameters were not set. The Unit Attention condition is also not generated if the Mode Select command parameter list does not include any pages and only the header or header/block descriptor is present.

- Format Unit command has been executed

In this case, a Unit Attention condition is generated for all Initiators except the one that issued the Format Unit command. The Additional Sense Code and Additional Sense Code Qualifier reported is ***Not Ready To Ready Transition, (Medium may have changed)***. This indicates that the block descriptor parameters from the last Mode Select command have been used and are now considered current values.

- Write Buffer command to download microcode has been executed

In this case, a Unit Attention condition is generated for all Initiators except the one that issued the Write Buffer command. The Additional Sense Code and Additional Sense Code Qualifier reported is ***Microcode has been changed***.

- Commands Cleared by a Clear Queue Message

This Unit Attention condition is generated after an Initiator sends a Clear Queue Message. The Unit Attention condition is generated for all other Initiators with I/O processes that were either active or queued for the logical unit. The Additional Sense Code and Additional Sense Code Qualifier reported is ***Commands Cleared by Another Initiator***.

- Spindles Synchronized

This Unit Attention is generated when the Drive enters the spindle motor synchronization state. In this case, a Unit Attention condition is generated for all Initiators. The Additional Sense Code and Additional Sense Code Qualifier reported is ***Spindles Synchronized***. When this Unit Attention is generated, the Unit Attention for Spindles not Synchronized is cleared (they are mutually exclusive). See 4.5, “Motor Synchronization” on page 208 for more information on this Unit Attention.

- Spindles Not Synchronized

This Unit Attention is generated when the Drive could not enter into or falls out of the spindle motor synchronization state. If the Drive could not enter into the synchronization state, a Unit Attention condition is generated for all Initiators. If the Drive falls out of the synchronization state, a Unit Attention condition is generated for Initiators other than the first Initiator to detect the synchronization error. The Additional Sense Code and Additional Sense Code Qualifier reported is *Spindles Not Synchronized*. When this Unit Attention is generated, the Unit Attention for Spindles Synchronized is cleared (they are mutually exclusive). See 4.5, “Motor Synchronization” on page 208 for more information on this Unit Attention.

The Unit Attention condition persists for each Initiator until that Initiator clears the condition from the logical unit as described below. Several commands are handled as special cases during a Unit Attention condition. These cases are also discussed below.

If the Target receives a command from an Initiator before reporting a *Check Condition Status* for a pending Unit Attention condition for that Initiator, the Target's response varies with the command as follows:

<b>Inquiry</b>	Execute the command, return <i>Good Status</i> , and preserve the Unit Attention condition.
<b>Request Sense</b>	Execute the command, return any pending sense data, return <i>Good Status</i> , and preserve the Unit Attention condition. If there is not any pending sense data, the sense data associated with the highest priority Unit Attention condition is returned and the highest priority Unit Attention condition is cleared for this Initiator.
<b>All Others</b>	Do not execute the command, return a <i>Check Condition Status</i> , and clear the highest priority Unit Attention condition for this Initiator. If the following command from this Initiator is Request Sense, then the sense data associated with the highest priority Unit Attention condition is returned. Otherwise, it is lost.

The Target's response to the next command received from an Initiator after reporting a *Check Condition Status* for a pending Unit Attention condition for that Initiator, varies with the command as follows:

<b>Inquiry</b>	Execute the command, return <i>Good Status</i> , and preserve any lower priority Unit Attention condition(s) for that Initiator (sense data for the highest priority Unit Attention condition is lost).
<b>Request Sense</b>	Execute the command, return a Sense key of <i>Unit Attention</i> and Additional Sense Code and Qualifier that corresponds to the highest priority Unit Attention condition, return <i>Good Status</i> , and preserve any lower priority Unit Attention condition(s).
<b>All Others</b>	Execute the command unless prevented by another higher priority status (see 4.1.1, “Priority of SCSI Status Byte Reporting” on page 179 for more information) or another lower priority Unit Attention Condition, and return the

appropriate status for the command. (Sense data for the highest priority Unit Attention condition is lost).

More than one Unit Attention condition may be generated for an Initiator before that Initiator clears the Unit Attention condition. Each Unit Attention condition is reported one at a time until they all have been reported and cleared. The highest priority Unit Attention condition is reported first. The order in which the Target checks for each Unit Attention condition determines their priority (highest priority first) as follows:

1. Reset (Target has been reset)
2. Mode Select Parameters Changed (Mode Select command has been executed.)
3. Not Ready To Ready Transition (Medium May Have Changed) (Format Unit command has been executed.)
4. Microcode Has Been Changed (Write Buffer command to download microcode has been executed.)
5. Commands Cleared By Another Initiator (Commands Cleared by a Clear Queue Message)
6. Spindles Synchronized
7. Spindles Not Synchronized

## 4.1.6 Command Processing During Start-up and Format Operations

If the Target receives a command from an Initiator while the Target is executing a start-up or format operation, the Target's response varies with the command as follows:

<b>Inquiry</b>	Execute the command and return <i>Good Status</i> .
<b>Request Sense</b>	<p>Execute the command, return a Sense key of <i>Not Ready</i> and an Additional Sense code of <i>Logical Unit Not Ready</i> and return <i>Good Status</i>.</p> <p>The Additional Sense code Qualifier that is returned is dependent on type of I/O processes that are active:</p> <p>For the Start/Stop Unit command and the Auto-start operation, the qualifier returned is <i>logical unit is in the process of becoming ready</i>. For the Format Unit command, the qualifier returned is <i>Logical Unit Not Ready, Format in Progress</i>, and the Sense key specific bytes are set to return the progress indication.</p>
<b>Start/Stop Unit</b>	<p>If the start-up operation is an active process and a Start/Stop Unit command (either active or queued) does not exist at the Target from this Initiator, and disconnection is allowed for the current I/O process then:</p> <p style="padding-left: 40px;">The command is added to the command queue.</p> <p>Otherwise:</p> <p style="padding-left: 40px;">Do not execute the command and return <i>Check Condition Status</i>. The Sense data generated is described in Request Sense above.</p>
<b>All Others</b>	Do not execute the command and return <i>Check Condition Status</i> . The Sense data generated is described in Request Sense above.

## 4.1.7 Internal Error Condition

The Target generates an Internal Error condition for all Initiators when:

- an internally initiated operation completes with an unrecoverable error.

The following is a list of internally initiated error conditions:

- During the execution of the start-up sequence for Auto Start (to automatically start the spindle motor) after the SCSI bus has been enabled and prior to completion of the bring-up sequence.
  - Following a SCSI Hardware reset or a SCSI Bus Device Reset message if the reset was received during a start-up sequence with the Auto Start function enabled. The start-up sequence is executed if it has not been previously executed and completed.
- an unrecoverable error occurs during an internal Target idle time function

An Internal Error condition causes Sense data to be generated and saved for all Initiators. The Error Code field of the Sense data is set for a Current Error (70h) and the Sense Key is set to HARDWARE ERROR. Recovered errors are not reported. Any outstanding Deferred Error condition is cleared for all Initiators and the associated Sense data is lost. (The Internal Error condition and Deferred Error condition are mutually exclusive.)

The Internal Error condition persists for each Initiator until that Initiator clears the condition from the logical unit as described below. Several commands are handled as special cases during an Internal Error condition. These cases are also discussed.

If the Target receives a command from an Initiator while an Internal Error condition exists for that Initiator, the Target's response varies with the command as follows:

<b>Inquiry</b>	Execute the command, return <i>Good Status</i> , and do not clear the Internal Error condition.
<b>Request Sense</b>	Execute the command, return the sense data generated by the Internal Error condition, return <i>Good Status</i> , and clear the Internal Error condition for that Initiator.
<b>All Others</b>	Do not execute the command, return <i>Check Condition Status</i> , and clear the Internal Error condition. If the following command from this Initiator is Request Sense, then the sense data associated with the Internal Error condition is returned. Otherwise, it is lost.



## 4.1.8 Deferred Error Condition

The Target generates a Deferred Error condition for all Initiators when:

- an external initiated operation completes in error.

The following is a list of external conditions:

- During the execution of an immediate Start/Stop Unit command after *Good Status* has been returned and prior to completing the start unit operation.
- During the execution of an immediate Format Unit command after *Good Status* has been returned and prior to completing the format operation.
- In the case of write caching, during execution of a Write command after *Good Status* has been returned and prior to writing the data to the media.

A Deferred Error condition causes Sense data to be generated and saved for all Initiators. The Error Code field (byte 0) of the Sense data is set for a Deferred Error (71h) and the Sense Key is set to either RECOVERED ERROR, MEDIUM ERROR, or HARDWARE ERROR. Both recoverable and unrecoverable errors are reported. Any outstanding Internal Error condition is cleared for all Initiators and the associated Sense data is lost. (The Internal Error condition and Deferred Error condition are mutually exclusive.)

The Deferred Error condition persists for each Initiator until that Initiator clears the condition from the logical unit as described below. Several commands are handled as special cases during a Deferred Error condition. These cases are also discussed.

If the Target receives a command from an Initiator while Deferred Error condition exists for that Initiator, the Target's response varies with the command as follows:

<b>Inquiry</b>	Execute the command, return <i>Good Status</i> , and do not clear the Deferred Error condition.
<b>Request Sense</b>	Execute the command, return the Sense data generated by the Deferred Error condition, return <i>Good Status</i> , and clear the Deferred Error condition for that Initiator.
<b>All Others</b>	Do not execute the command, return <i>Check Condition Status</i> , and clear the Deferred Error condition. If the following command from this Initiator is Request Sense, then the Deferred Error Sense data is returned. Otherwise, it is lost.

Deferred Error conditions are not reported if the option pin is grounded (see 4.15, "Option Pin" on page 223).

## 4.1.9 Degraded Mode

There are certain errors or conditions that impair the Target's ability to function normally. Rather than fail hard, the Target is designed to be as responsive as possible. Also, in most cases, some action on the part of the Initiator may be used to restore normal operation. This mode of limited operation is called Degraded Mode.

The following sections address causes of Degraded Mode, characteristics and limitations the Target has in Degraded Mode, and actions that may be taken to restore normal operation.

### 4.1.9.1 Reasons for Entering Degraded Mode

The Target enters Degraded Mode for any of the reasons shown below. Not all reasons are error conditions (for example, the Target powers up in Degraded Mode and Degraded Mode does not clear until, among other requirements, the

spindle spins up).

- The spindle motor is not spinning or is not at its proper speed. (*Motor Stop Degraded*)

The motor may not be spinning for any of several reasons:

- The motor has not spun up yet, either because the point of Auto Start has not been reached (if enabled), or because a Start/Stop Unit command has not been issued to start the motor.
- The motor has been stopped by a Start/Stop Unit command.
- A spindle failure has caused the Target to stop the spindle.
- A Self Initiated Reset has occurred.

- The configuration records have not been read or are not readable. (*Configuration Sector Degraded*)

During the execution of the start-up sequence, the Target reads a portion of its Reserved Area called the configuration records. Errors prior to this read, or a failure of the read, result in entering Degraded Mode.

- RAM microcode is not loaded. (*Microcode Degraded*)

During the execution of the start-up sequence, the Target loads its microcode from the Reserved Area. The Target enters Degraded Mode if this load cannot be achieved. Degraded Mode is also entered after failures of a microcode download operation by way of the Write Buffer command.

- Failure of a Send Diagnostic self-test, a start-up sequence, abnormal termination of Track Squeeze Recovery in DRP, or other internal Target failures. (*Power on Self Test Degraded*)

There are several different reasons for entering this Degraded Mode. All of these reasons are related to error condition and failures within the Target. The reasons for entering this degraded mode include the following:

- The Target does extensive self testing as part of the start-up sequence or as a result of executing the Send Diagnostic command with the SlfTst option. A failure of either of these tests places the Target in this Degraded Mode.
- The Target periodically adjusts the track following position for each data head, to compensate for mechanical expansions and contractions due to temperature changes. If one of these adjustments fails, the Target enters this degraded mode to prevent writing data off-track.

- The Track Squeeze Recovery steps in DRP requires that sectors radially adjacent to the error sector be temporarily moved to the Reserved Area. If this DRP step is abnormally terminated this degraded mode is entered.
- A Format Unit or Reassign Blocks command, or an automatic reallocation failed or was abnormally terminated. (*Format Degraded, Reassign Degraded*)

Format Unit, Reassign Blocks, and automatic reallocation operations must complete successfully for the Drive to be useable. If one of these operations fail or is terminated by a reset, a Bus Device Reset or Abort message, a Self Initiated Reset, a loss of power, or certain SCSI errors, the Target enters Degraded Mode.

### 4.1.9.2 Limitations of the Drive While in Degraded Mode

Several commands are not executed by the Target while in Degraded Mode. If Degraded Mode does not allow the command to execute, the command is failed with a Check Condition status and the appropriate Sense Data is generated. The Sense Key is set to Not Ready. The Additional Sense Code and the Additional Sense Code Qualifier is dependent on the type of Degraded Mode as summarized below.

More than one Degraded Mode may exist at the same time. Only the highest priority Degraded Mode that prevents command execution is reported to the Initiator. The order in which the Target checks for each type of Degraded Mode determines their priority (highest priority first) as follows:

1. The spindle motor is not spinning or is not at its proper speed. (*Motor Stop Degraded*)

The Sense Key is *Not Ready*, with a Sense Code/Qualifier of *Logical Unit Not Ready initializing command required*.

2. The configuration records have not been read or are not readable. (*Configuration Sector Degraded*)

The Sense Key is *Not Ready*, with a Sense Code/Qualifier of *Logical Unit Failed Self-Configuration*.

3. RAM microcode is not loaded. (*Microcode Degraded*)

The Sense Key is *Not Ready*, with a Sense Code/Qualifier of *Logical Unit Failed Self-Configuration*.

4. Failure of a Send Diagnostic self-test, a start-up sequence, abnormal termination of Track Squeeze Recovery in DRP, or other internal Target failures. (*Power on Self Test Degraded*)

The Sense Key is *Not Ready*, with a Sense Code/Qualifier of *Diagnostic Failure*.

5. A Format Unit command failed or was abnormally terminated. (*Format Degraded*)

The Sense Key is *Not Ready*, with a Sense Code/Qualifier of *Medium Format Corrupted, Format Command Failed*.

6. Reassign Blocks command failed or was abnormally terminated. (*Reassign Degraded*)

The Sense Key is *Not Ready*, with a Sense Code/Qualifier of *Medium Format Corrupted, Reassign Failed*.

7. An Auto Reassign failed or was abnormally terminated. (*Reassign Degraded*)

The Sense Key is *Not Ready*, with a Sense Code/Qualifier of *Medium Format Corrupted, Reassign Failed*.

### 4.1.9.3 Summary of Target's response to each SCSI Command in Degraded Mode

A summary of the Target's response to each SCSI command that is received during Degraded Mode is shown in the following tables:

Inquiry, Read Buffer, Request Sense, Start/Stop Unit, Write Buffer (except download and save) commands		
Degraded Mode	Description	
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i></li> </ul>	Executes.	
	Success:	Reports Good status.
	Failure:	Generates Check Condition

Write Buffer (download & save) command		
Degraded Mode	Description	
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> </ul>	Does not execute. Generates Check Condition.	
<ul style="list-style-type: none"> <li>• <i>Microcode Degraded</i></li> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i></li> </ul>	Executes.	
	Success:	Reports Good status.
	Failure:	Generates Check Condition. Enters <i>Microcode Degraded</i> mode.

Reassign Blocks command	
Degraded Mode	Description
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> <li>• <i>Format Degraded</i></li> </ul>	Does not execute. Generates Check Condition.
<i>Reassign Degraded</i>	Executes.  <b>Note:</b> The LBA specified by the command must match the original LBA that was being re-assigned in order to resume the Reassign Blocks operation. Otherwise, Check Condition Status is returned and the command is not executed.
	Success: Reports Good status. Clears <i>Reassign Degraded</i> mode.
	Failure: Generates Check Condition. <i>Reassign Degraded</i> mode is not cleared.

Synchronize Cache, Write, Write Extended, Write Verify, Write Same, or Write Long command	
Degraded Mode	Description
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i></li> </ul>	Does not execute. Generates Check Condition.

<b>Test Unit Ready command</b>		
<b>Degraded Mode</b>	<b>Description</b>	
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> </ul>	Executes. Generates Check Condition.	
<ul style="list-style-type: none"> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i> (FRDD bit set to 0, see 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59.)</li> </ul>	Executes. Generates Check Condition.	
<ul style="list-style-type: none"> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i> (FRDD bit set to 1, see 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59.)</li> </ul>	Executes. Reports Good status or status other than Degraded Mode.	
	Success:	Reports Good status.
	Failure:	Generates Check Condition.

<b>Format Unit command</b>	
<b>Degraded Mode</b>	<b>Description</b>
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> </ul>	Does not execute. Generates Check Condition.
<i>Format Degraded</i>	Executes.
	Success: Reports Good status. Clears <i>Format Degraded</i> . Clears <i>Reassign Degraded</i> if it exists internally.
	Failure: Enters <i>Format Degraded</i> mode. Generates Check Condition.
<i>Reassign Degraded</i>	Executes.
	Success: Reports good status. Clears <i>Reassign Degraded</i> mode.
	Failure: Enters <i>Format Degraded</i> mode. Generates Check Condition.



<b>Send Diagnostics command</b>	
<b>Degraded Mode</b>	<b>Description</b>
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> </ul>	Does not execute. Generates Check Condition.
<i>Power on Self Test Degraded</i>	Executes.
	Success: Reports Good status. Clears <i>Power on Self Test Degraded</i> mode.
	Failure: Enters <i>Power on Self Test Degraded</i> mode. Generates Check Condition.
<i>Format Degraded</i>	Executes.
	Success: Reports Check Condition with <i>Format Degraded</i> mode.
	Failure: Enters <i>Power on Self Test Degraded</i> mode. Generates Check Condition.
<i>Reassign Degraded</i>	Executes.
	Success: Reports Good status. Clears <i>Reassign Degraded</i> mode.
	Failure: Generates Check Condition. Depending on the cause of the failure, <i>Power on Self Test Degraded</i> mode may be entered. <i>Reassign Degraded</i> mode is not cleared.

<b>Read, Read Extended, Read Capacity with PMI=1, Read Long, or Verify commands</b>	
<b>Degraded Mode</b>	<b>Description</b>
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> </ul>	Does not execute. Generates Check Condition.
<i>Power on Self Test Degraded</i>	Executes.
	Success: Reports Good status.
	Failure: Generates Check Condition.
<i>Format Degraded</i>	Does not execute. Generates Check Condition with <i>Format Degraded</i> .
<i>Reassign Degraded</i>	Executes.
	Success: Reports Good status.
	Failure: Generates Check Condition.

<b>Mode Select with Save Parameters bit = 1</b>	
<b>Degraded Mode</b>	<b>Description</b>
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> </ul>	Does not execute. Generates Check Condition.
<ul style="list-style-type: none"> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i></li> </ul>	Executes.
	Success: Reports Good status.
	Failure: Generates Check Condition.

<b>All Other commands</b>	
<b>Degraded Mode</b>	<b>Description</b>
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> </ul>	Does not execute. Generates Check Condition.
<ul style="list-style-type: none"> <li>• <i>Power on Self Test Degraded</i></li> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i></li> </ul>	Executes.
	Success: Reports Good status.
	Failure: Generates Check Condition.

Degraded Mode affects the operation of the automatic rewrite/reassignment function and various internal Target idle time functions as shown in the following tables:

<b>Automatic Rewrite/Reassignment function</b>	
<b>Degraded Mode</b>	<b>Description</b>
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i></li> </ul>	Does not execute.

<b>Update of Performance Counter Data in Reserved Area and Error Logging function</b>	
<b>Degraded Mode</b>	<b>Description</b>
<ul style="list-style-type: none"> <li>• <i>Motor Stop Degraded</i></li> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> </ul>	Does not execute.
<ul style="list-style-type: none"> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i></li> </ul>	Executes.

<b>Thermal Compensation and Disk Sweep function</b>	
<b>Degraded Mode</b>	<b>Description</b>
<i>Motor Stop Degraded</i>	Does not execute.
<ul style="list-style-type: none"> <li>• <i>Configuration Sector Degraded</i></li> <li>• <i>Microcode Degraded</i></li> <li>• <i>Power on Self Test Degraded</i></li> <li>• <i>Format Degraded</i></li> <li>• <i>Reassign Degraded</i></li> </ul>	Executes.

#### 4.1.9.4 Exiting Degraded Mode

There are one or more actions that cause the Target to attempt to exit each Degraded Mode. See Appendix C, “Recommended Initiator Error Recovery Procedures” on page 265 for a description of the recommended Initiator action for each Degraded Mode condition.

The following is a list of the actions and conditions that cause the Target to attempt to exit various Degraded Modes:

- The Target is reset by a Power-On reset or Self Initiated Reset, and the function to automatically start the spindle motor is enabled via the Auto Start jumper (see product Functional Specification).

When this occurs, the Target executes the start-up sequence. (see 4.18, “Bring-Up Sequence” on page 225). The Target attempts to clear the following Degraded Mode conditions:

- *Motor Stop Degraded*
- *Configuration Sector Degraded*
- *Microcode Degraded*
- *Power on Self Test Degraded*
- *Reassign Degraded*

- The Target is reset by a SCSI Bus reset or Bus Device Reset message, and the function to automatically start the spindle motor is enabled via the Auto Start jumper, and the Target has not previously completed the initial attempt to execute the start-up sequence.

When this occurs, the Target executes the start-up sequence. The Target attempts to clear the following Degraded Mode conditions:

- *Motor Stop Degraded*
- *Configuration Sector Degraded*
- *Microcode Degraded*
- *Power on Self Test Degraded*
- *Reassign Degraded*

- A Start Unit command is executed and the Target has not previously completed the initial attempt to execute the start-up sequence.

When this occurs, the Target executes the start-up sequence. The Target attempts to clear the following Degraded Mode conditions:

- *Motor Stop Degraded*
- *Configuration Sector Degraded*
- *Microcode Degraded*
- *Power on Self Test Degraded*
- *Reassign Degraded*

- A Start Unit command is executed and the Target has previously completed the initial attempt to execute the start-up sequence.

When this occurs, the Target only starts the spindle motor and does not execute the complete start-up sequence. The Target attempts to clear the following Degraded Mode conditions:

- *Motor Stop Degraded*

- A Write Buffer command with the Download Microcode or Download Microcode and Save option is executed.

If the Write Buffer command completes successfully, the Target clears the following Degraded Mode conditions:

- ***Microcode Degraded***

- A Send Diagnostic command is executed.

If the Send Diagnostic command completes successfully, the Target clears the following Degraded Mode conditions:

- ***Power on Self Test Degraded***

- ***Reassign Degraded***

- A Format Unit command is executed.

If the Format Unit command completes successfully, the Target clears the following Degraded Mode conditions:

- ***Format Degraded***

- ***Reassign Degraded***

- A Reassign Blocks command is executed. The LBA specified by the Reassign Blocks command must be for the same LBA that was previously reassigned when the Reassign Degraded Mode was entered. Otherwise, the command is not executed and is ended with Check Condition Status.

If the Reassign Blocks command completes successfully, the Target clears the following Degraded Mode conditions:

- ***Reassign Degraded***

## 4.1.10 Command Processing While Reserved

A logical unit is reserved after successful execution of the Reserve command. Each time a Reserve command is executed successfully, the Target records the SCSI ID of the Initiator that made the reservation and the SCSI ID of the Initiator that is to receive the reservation. This information is needed to determine whether subsequent commands should be permitted or if the *Reservation Conflict Status* should be reported. The Initiator that made the reservation is the Initiator that issued the Reserve command. The Initiator to receive the reservation may be either the same or a different Initiator (third-party reservation).

**Note:** A third-party reservation (3rdPty bit equal to one) with the Third Party Device ID set equal to the SCSI ID of the Initiator that issued the Reserve command is not distinguishable from a Reserve command with the third party (3rdPty) bit equal to 0. In either case, the Target interprets both the SCSI ID of the Initiator that made the reservation and the Initiator to receive the reservation to be the SCSI ID of the Initiator that issued the Reserve command. In this case, a 3rdPty release is not required, but is allowed.

If the logical unit is reserved when a new command is received, the Target examines the command opcode and the SCSI ID of the issuing Initiator to determine whether a *Reservation Conflict Status* should be returned based on the following rules:

1. If the issuing Initiator is the one that made the reservation and also the one to receive the reservation then:
  - All commands are permitted.
2. If the issuing Initiator is neither the one that made the reservation nor the one to receive the reservation then:
  - A Request Sense or Inquiry command is permitted.
  - A Release command is permitted but is ignored.
  - Any other command results in a *Reservation Conflict Status*.
3. If the issuing Initiator is the one that made the reservation but is not the one to receive the reservation then:
  - An Inquiry, Request Sense, Reserve, or Release command is permitted.
  - Any other command results in a *Reservation Conflict Status*.
4. If the issuing Initiator is not the one that made the reservation but is the one to receive the reservation then:
  - A Reserve command results in a *Reservation Conflict Status*.
  - A Release command is permitted but is ignored.
  - Any other command is permitted.

If a *Reservation Conflict Status* is not reported and the command is permitted, then the Target checks the next highest priority internal condition to determine whether execution is allowed. See 4.1.1, “Priority of SCSI Status Byte Reporting” on page 179 for more information.

---

## 4.2 Priority Commands

Certain SCSI commands always execute without returning a *Busy Status*, *Reservation Conflict Status*, or *Queue Full Status* in response to the command. These commands are:

- Inquiry
- Request Sense

These commands do not disconnect from the SCSI bus prior to completion. They are executed prior to attempting to complete the execution of any other pending command that has disconnected from the SCSI bus. Therefore, a second priority command cannot be received during the execution of a priority command.

These commands are never queued whether or not the command is sent with a queue tag. However, the rules for an Incorrect Initiator Connection still apply to priority commands. (see 4.1.3, “Incorrect Initiator Connection” on page 181)

---

## 4.3 Command Queuing

The Target supports both Tagged and Untagged Queuing. An Initiator can use either Tagged or Untagged Queuing but cannot use both at the same time. Both tagged I/O processes and untagged I/O processes can coexist in the command queue at the same time, provided they are associated with different Initiators.

The command queue is allocated on a First-In First-Out (FIFO) basis. See 4.3.2, “Tagged Queuing” on page 205 for exceptions to the FIFO execution order.

Command Queuing is enabled and disabled using the DQue bit in Mode Select page 0Ah.

The following commands are never queued.

- Priority Commands. (i.e. Request Sense and Inquiry).
- Commands linked to previous commands. These are defined to be part of a single I/O process. (Linked commands are always executed immediately following the previous command from the same Initiator. No other Initiator's commands are allowed to be executed between two linked commands.)
- Commands for which disconnection is not allowed. (These may result in a *Busy Status*.)
- Commands in which a SCSI bus error occurred between selection and first disconnection following the receipt of the CDB.
- Commands for an invalid Logical Unit Number.
- Commands which cause an OVERLAPPED COMMANDS ATTEMPTED error. (see 4.1.3, “Incorrect Initiator Connection” on page 181)

In the above listed situations, the command queue is bypassed and the received command becomes an active I/O process immediately. If one of the above situations occurs when there are queued I/O processes, those queued I/O processes are preserved in the command queue in the order they were received.

At the completion of an active I/O process, the queued I/O process at the head of the queue is removed from the queue and becomes an active I/O process. (See 4.3.6, “Termination of I/O processes” on page 207 for the circumstances in which I/O processes can be removed from the queue without executing.)

### 4.3.1 Untagged Queuing

The Target supports queuing one I/O process from each SCSI Initiator. Untagged I/O processes are treated by the Target as though they were received with Simple Queue Tag messages for purposes of queuing.

**Note:** There is no guarantee that I/O processes are executed in the order they were received in a multiple initiator environment. Reserve and Release commands must be used to temporarily “lock out” other Initiators if any restrictions apply to the order of command processing in a multiple initiator environment. The Queue Algorithm Modifier parameter value of zero (restricted re-ordering) in Mode Select page 0Ah, only applies to commands from the same Initiator tagged with Simple Queue Tag message.



## 4.3.2 Tagged Queuing

The Target supports queuing multiple commands from one Initiator if Tagged Queuing is enable.

Initiators may add commands to the queue or delete commands from the queue. When a command is added to the queue, the Initiator may specify the command is to be execute next, in the order received, or allow the the Target to determine the order of execution. The target reordering of commands is controlled by the Mode Select Page 0Ah parameter QMOD. See 1.6.9, “Page 0Ah - Control Mode Parameters” on page 88 for more information.

Commands received with *Head of Queue* tags are executed Last-In First-Out (LIFO). (See 3.1.17, “Queue Tag Messages (20h, 21h, 22h)” on page 155 for more details concerning *Head of Queue* tags.)

When a command terminates with *Check Condition Status*, commands in the queue for that Initiator are not executed until a command is received by that Initiator. Execution of commands in the queue from other Initiators is not affected. See 4.3.5, “Effects of errors on Command Queuing” on page 206.

The Target can queue up to 32 I/O processes. The value of the Disable Queuing (DQue) bits determine how many commands an Initiator may queue.

When Tagged Queuing (DQue = 1) is disable, each Initiator may queued one command. An initiator with an active I/O process is not allowed to queue any commands.

When Tagged Queuing (DQue = 0) is enabled, the Target reserves 2 slots in the command queue for each Initiator the Target has received an I/O process from, once 2 slots become available for that Initiator. This prevents one Initiator from locking out other Initiators. Queue slot reservations are lost when the Target powers down.

The remaining queue slots are shared among all Initiators. The exact number of shared slots depend on the number of Initiators that have sent I/O processes to the Target. Any of the Initiators may have a queued I/O process in one of these shared slots. A single Initiator could use all the shared queue slots, if no other Initiator has more than 2 queued I/O process in the queue. The number of I/O processes a particular Initiator can queue varies between 2 and 32 I/O processes depending on the number of Initiators that have sent I/O process to the target.

## 4.3.3 Queue Full

When Tagged Queuing is enabled (DQue = 0), the Target responds with *Queue Full Status* when

- The command queue is full and all 32 slots are utilized.
- The command queue is not full but all of the available slots not reserved for use by another initiator are utilized.
- A Format Unit command has been queued but has not yet been executed.
- A Start Unit command has been queued but has not yet been executed.

### 4.3.4 Rules for Queuing commands

For a description of the rules for queuing commands, see 4.1.4, “Command Processing During Execution of Active I/O Process” on page 182.

### 4.3.5 Effects of errors on Command Queuing

When an I/O process ends with *Check Condition Status*, the Target builds sense for the Initiator associated with the I/O process which ended with *Check Condition Status*. When an Initiator has sense pending, all queued I/O processes associated with that Initiator that were in the command queue prior to the connection in which the *Check Condition Status* was sent, are held in the queue until the sense data is cleared. The SCSI operations of active I/O processes for that Initiator are suspended until the sense data is cleared. After the pending sense data is cleared, normal execution and dequeuing resumes. This handling of error conditions for Tagged Queuing, corresponds to  $QErr = 0$  in Mode Select page 0Ah.

**Note:** When the queue operation regarding queued I/O processes from other Initiators is not affected, I/O processes may be executed in a different order than they were received with respect to multiple Initiators. See 4.3.1, “Untagged Queuing” on page 204 for more information.

When the Target returns *Check Condition Status*:

- Execution of SCSI operations for active I/O processes of the affected Initiator are suspended until pending sense data is cleared.  
**Note:** the *affected Initiator* is the Initiator which received the *Check Condition Status*.
- Dequeuing is suspended for the affected Initiator until the pending sense data is cleared.
- Dequeuing of queued I/O processes from unaffected Initiators is not suspended, unless execution of SCSI operations for an active I/O process are suspended. In that case, dequeuing of queued I/O processes for all Initiators is suspended.
- Subsequent commands from the affected Initiator clear sense and restore normal queue operation for that Initiator.
  - A REQUEST SENSE command from the affected Initiator bypasses the queue as described in 4.2, “Priority Commands” on page 203, executes immediately, reports and clears the pending sense. A REQUEST SENSE command from the affected Initiator cannot cause an Incorrect Initiator Connection when sense is pending for that Initiator.
  - A command other than REQUEST SENSE from the affected Initiator will clear the pending sense and then be handled according to the protocol described in 4.1, “SCSI Protocol” on page 179.
- It is possible for queued I/O processes to not be executed in the order expected because some Initiators have dequeuing suspended while others do not.
- Relative execution order of all queued I/O processes for the affected Initiator is preserved.
- Relative execution order of all queued I/O processes for all unaffected Initiators is preserved.
- Enqueuing operation is not affected by pending sense.

### 4.3.6 Termination of I/O processes

I/O processes can be terminated in a number of ways:

- An I/O process ends normally either by being executed by the Target or by encountering an error.
- The current I/O process for a given Initiator is terminated by an *Abort Tag Message*.
- All I/O processes for a given Initiator on the specified logical unit are terminated by an *Abort Message*.
- All I/O processes for a given Initiator on the specified logical unit are terminated by an **Overlapped Commands Attempted** error. See 4.1.3, “Incorrect Initiator Connection” on page 181 for details of this error condition.
- All I/O processes from all Initiators on the specified logical unit are terminated by a *Clear Queue Message*
- All I/O processes from all Initiators are terminated by a *SCSI BUS RESET* or a *BUS DEVICE RESET MESSAGE*.

---

## 4.4 Concurrent Command Processing

Concurrent processing is when multiple I/O processes are active (not queued) on the same logical unit. The Target supports multiple active I/O processes executing on the same logical unit. The Initiator becomes aware an I/O process is active when the Data phase (Data In or Data Out) portion of that I/O process has started. The Initiator becomes aware multiple I/O processes are active when the Target has initiated the Data phase of another I/O process on the same logical unit as the first I/O process, before the first I/O process has completed the Status phase.

Two non-priority I/O processes are allowed to execute concurrently on the same logical unit. Priority commands are always allowed to execute concurrently regardless of the number of active I/O processes.

The following I/O processes are allowed to execute concurrently:

- Priority commands. (See 4.2, “Priority Commands” on page 203).
  - Inquiry
  - Request Sense

The following command is allowed to execute concurrently when the Concurrent Processing Enable (CPE) bit is 1. (See 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59)

- Back to Back write commands.

When an I/O process ends in *Check Condition Status*, other active I/O processes from the same initiator on the same logical unit will not reconnect and complete execution until the sense data has been cleared. (See 4.3.5, “Effects of errors on Command Queuing” on page 206.)

## 4.4.1 Back to Back Writes

Back to back writes allows multiple write commands which write sequential logical blocks, to be written to the media without losing a motor revolution between the ending of each write command and the beginning of the following write command.

The Target will write back to back under the following conditions:

- Two (or more) write commands (Operation Code 0Ah or Operation Code 2Ah) execute consecutively, without an intervening command.
- The write commands address consecutive logical block ranges. The logical block specified for the second write command is the next logical block following the last logical block to be written by the first write command.
- One logical block of data has been received in the buffer from the second write command in time to allow the media to be written before an additional revolution would be required.
- Concurrent Processing is enabled (CPE = 1) or Write Caching is enabled (WCE = 1, FUA = 0).

When the Target is required to write the data to the media before sending Status (writes not cached), the write commands execute concurrently. Following the completion of the Data Out phase and before the Status phase of the first write command, the Data Out phase of the second write command begins. When the first write command has completed writing the data to the media, and the second write command is disconnected, the first write command returns Status. The Target continues with writing data to the media for the second write command without losing a motor revolution.

When the Target is caching writes, *Good Status* is returned before writing all the data to the media. (See description of Write Cache Enable (WCE) bit in 1.6.8, “Page 8h - Read Caching Parameters” on page 85.) Execution of the second write command begins after the first write command returns status, but before the first write command has completed writing data to the media. When the Target completes writing data to the media for the first write command, data is written to the media for the second write command without losing a motor revolution.

Status for the first write command is always returned before Status is returned for the second write command. If the first write command returns *Check Condition Status* and the second write command is from the same initiator, execution of SCSI operations for the second write command are suspended. The second write command will resume execution when the pending sense data is cleared. Dequeuing and execution of queued commands for all initiators on the logical unit with the pending sense data are suspended until the pending sense data is cleared. (See 4.3.5, “Effects of errors on Command Queuing” on page 206.)

---

## 4.5 Motor Synchronization

Motor synchronization is controlled by the MODE SELECT and the START/STOP UNIT commands. See 1.6.6, “Page 4h - Rigid Disk Drive Geometry Parameters” on page 80 and 1.27, “Start/Stop Unit” on page 132 for complete descriptions of each command's respective role regarding Motor Synchronization.

The Target does not attempt to change the synchronization state until a motor start operation (start-up sequence) is initiated even if the RPL field in Page 4 of the

MODE SELECT command has been changed since the motor was started. The value of the RPL field in Page 4 of the MODE SELECT command *that is present when a motor start operation is initiated* is the value that determines motor synchronization state.

The actual state of synchronization can be altered by two different types of events -

1. Any event that attempts a start the spindle motor when the RPL field has a value other than 00b.

If an Initiator wishes to place the Drive into a synchronized mode other than the one that it is currently in, it must ensure that the current RPL field has the desired value and the proper signals are present on the Spindle Synchronization Cable (if required) before that attempt is made. For example, if an Initiator changes a Drive spindle synchronization mode from No Sync to Master Sync Control, it must ensure that another Drive is driving the Slave Sync line with a proper signal and that no other Drive is driving the Master Sync line before that attempt is made.

If changing from an unsynchronized state to a synchronized state, a Unit Attention condition is generated for all Initiators. The sense key is set to UNIT ATTENTION and the additional sense code/qualifier is set to SPINDLES SYNCHRONIZED.

If changing from a synchronized state to an unsynchronized state, a Unit Attention condition is also generated for all Initiators. The sense key is set to UNIT ATTENTION and the additional sense code/qualifier is set to SPINDLES NOT SYNCHRONIZED.

If the Target fails to achieve synchronization, a Unit Attention condition is generated for all Initiators. The sense key is set to UNIT ATTENTION and the additional sense code/qualifier is set to SPINDLES NOT SYNCHRONIZED.

See 4.1.5, “Unit Attention Condition” on page 184 for more information on Unit Attentions.

2. An error condition that causes the motor to fall out of synchronization subsequent to achieving synchronization is reported in two possible ways.
  - If the loss of synchronization is detected when an I/O process is not executing, the Target creates a unit attention condition for all initiators. The sense key is set to UNIT ATTENTION, the additional sense code/qualifier to SPINDLES NOT SYNCHRONIZED.
  - If detected by an I/O process, the first I/O process to detect the synchronization error, attempts to complete execution of the I/O process. The Target returns *Check Condition Status* for this I/O process. The sense key is set to RECOVERED ERROR if the Target is able to successfully complete the I/O process. The additional sense code/qualifier is set to SPINDLES NOT SYNCHRONIZED. The Target then generates a Unit Attention condition for all other Initiators. The sense key is set to UNIT ATTENTION and the additional sense code/qualifier is set to SPINDLES NOT SYNCHRONIZED. If in addition to the loss of synchronization, another error occurs, the sense data will be for the additional error and the Unit Attention will be set for all initiators.

The value of the RPL field Page 4 of the MODE SELECT command also affects the way the FORMAT UNIT command operates. The Target alters the method of track and cylinder skewing depending on whether the RPL field is set to synchronized or unsynchronized operation. When the motor synchronization mode is

changed a FORMAT UNIT command should be issued to change the track and cylinder skewing. The value of the RPL field in Page 4 of the MODE SELECT command *that is present when a FORMAT UNIT is initiated* is the value that determines the skewing method used.

**Note:** If the drive is operated in a synchronized mode but was formatted with the RPL field set to unsynchronized mode, *the Drive is effectively not synchronized* even though the spindles are synchronized.

In summary, changing the motor synchronization mode is done as follows:

- Issue a MODE SELECT command with the RPL field in Page 4 - Rigid Disk Drive Geometry Page set to the desired value.
- Reformat the Drive using the FORMAT UNIT command.
- Change the motor status by:
  - Issuing a START/STOP UNIT command with the Start bit = 1 (**note:** the status of the motor can be running or stopped) or
  - Cycling power with the Auto Start jumper configured to automatically start the spindle motor.

---

## 4.6 Error Logs

The target maintains error logs in RAM and in the reserved area for use in development and Automatic/Recommend Reassignment. When an error occurs, it is entered into the logs maintained in RAM and is analyzed for Automatic/Recommend Reassignment. The contents of these RAM logs are copied to the reserved area during idle time to prevent impacts to performance. If insufficient idle time is available to copy the log contents to the reserved area and the RAM logs are full, the copying will occur following the completion of the command on which the last error occurred. When the log contents are copied at the end of the command, the target will disconnect from the SCSI bus, if allowed, and then reconnect before ending the command.

---

## 4.7 Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign

The target supports Auto and Recommend Reallocate for READ, WRITE, WRITE and VERIFY, VERIFY, and WRITE SAME. VERIFY, and WRITE SAME. (See 1.6, “Mode Select (6)” on page 55 for a description of the bits used to control automatic reallocation.)

Automatic and Recommend Reallocate operate from within the read/write command. When an automatic reallocation occurs, the read or write command takes longer to complete. During this time, the target disconnects from the SCSI bus, if allowed, and reconnects before ending the command.

Following is a description of the target behavior for each setting of ARRE. ARRE settings affects all data errors (No Sector Found, Data Sync Byte Errors and Data ECC Errors.)

- ARRE = 1 - An error site determined to need rewriting or reassignment during a read is automatically rewritten or reallocated at the conclusion of the read and prior to sending ending status. If the site cannot be automatically rewritten or reallocated, then a recommendation for reassignment is given. The site will be automatically rewritten or reallocated only if it can be successfully read.
- ARRE = 0 - An error site determined to need rewriting or reassignment during a read is recommended for rewriting or reassignment at the conclusion of the read.

The setting of the ARRE bit is checked and the target will automatically rewrite/reallocate or recommend rewrite/reassign for the following commands.

- Read(6)
- Read(10)

The target will recommend rewrite/reallocate but will not auto rewrite/reallocate for the following commands.

- Verify
- Verify Portion of Write and Verify

For all other commands the ARRE setting is ignored and the target will not automatically rewrite/reallocate or recommend rewrite/reassign.

Following is a description of the target behavior for each setting of AWRE. AWRE settings affect only No Sector Found errors on writes.

- AWRE = 1 - An error site determined to need reassignment during a write is automatically reallocated at the conclusion of the write and prior to sending ending status. If the site cannot be automatically reallocated, then a recommendation for reassignment is given. The site is automatically reallocated only if the write recovery succeeded and the site can then be read.
- AWRE = 0 - An error site determined to need reassignment during a write will be recommended for reassignment at the conclusion of the write.

The setting of the AWRE bit is checked and the target will automatically reallocate or recommend reassign for the following commands.

- Write(6)
- Write(10)
- Write Same
- Write portion of Write and Verify

For all other commands the AWRE setting is ignored and the target will not automatically rewrite/reallocate or recommend rewrite/reassign.

Auto/Recommend Reallocate information is communicated via the sense data returned following a command during which a site was determined to need rewriting or reassignment. The LBA returned in the sense data is the LBA that was determined to need rewriting or reassignment. The Sense Key/Code/Qualifier combinations listed below is a list of all possible combinations that are valid for Auto Reallocate/Recommend Reassign.

It is important to note that Auto Reallocate/Recommend Reassign is affected by the PER, QPE, and RPFAE bits.

Table 111. Auto/Recommend Rewrite/Reallocate PER/QPE and RPFAE combinations.. Results of Various PER, QPE, RPFAE bit settings.			
PER	QPE	RPFAE	Result
0	x	0	Recovered Errors are not reported.
0	x	1	Only recovered errors with auto/recommend rewrite/reallocate sense data are reported.
1	0	x	All recovered errors are reported.
1	1	x	Recovered errors above the QPE threshold are reported and all recovered errors that have auto/recommend rewrite/reallocate sense data are reported.

**Note:** The x is a logical 'don't care'.

The sense data combinations associated with auto/recommend rewrite/reallocate are listed below.



Table 112. Auto/Recommend Rewrite/Reallocate Sense data combinations.. Sense data combinations for automatic or recommend rewrite or reallocate.			
Key	Code	Qual.	Description
1	0C	01	Recovered Write error - Auto Reallocated.
3	0C	02	Recovered Write error - auto reallocate failed.
1-3	0C	03	Write Error - Recommend Reassign.
3	11	04	Auto reallocate failed due to unrecovered read error.
3	11	0B	Unrecovered read error - Recommend Reassign.
1-3	14	05	Record Not Found - Recommend Reassign.
1	14	06	Record Not Found - data Auto-Reallocated.
1	16	01	Sync Byte Error - data Rewritten.
1	16	02	Sync Byte Error - recommend Rewrite.
1	16	03	Sync Byte Error - data auto-reallocated.
1-3	16	04	Sync Byte Error - Recommend Reassign.
1	17	06	Recovered data without ECC - Auto Reallocated.
1	17	07	Recovered data without ECC - Recommend Reassign.
1	17	08	Recovered data without ECC - Recommend Rewrite.
1	17	09	Recovered data without ECC - Data Rewritten.
1	18	02	Recovered data with ECC - Auto Reallocated.
1	18	05	Recovered data with ECC - Recommend Reassign
1	18	06	Recovered data with ECC - Recommend Rewrite.
1	18	07	Recovered data with ECC - Data Rewritten.

## 4.8 Predictive Failure Analysis

Predictive Failure Analysis (PFA) is an internal function of the Target. During Idle periods and as a part of Error Recovery, the Target performs a number of functions designed to predict drive failure conditions before they occur. The predictive actions performed by the Target are:

- Error Log Analysis
- Data channel parametric measurement and analysis
- Head flyheight analysis

When one of the predictive functions detects an anomaly and reporting of PFA is enabled by the appropriate Mode Select parameters, the Target will send Check Condition Status following a successful Read or Write command. The sense will indicate a Recovered Error and the sense Code/Qualifier will be ***Predictive Failure Analysis Threshold Reached on Recovered Error***.

The appropriate Mode Select parameters for enabling PFA reporting are shown in the RPF AE description in 1.6.2, “Page 0 - Vendor Unique Parameters” on page 59.

---

## 4.9 Segmented Caching

### 4.9.1 Overview

Segmented Caching divides the data buffer into several smaller buffer segments. Each buffer segment is used as a separate Read / Write / Read-Ahead / Pre-Fetch buffer.

At the start of a command, the target chooses which segment to use and where to start in the segment. The selection of the segment is based on whether or not the desired blocks are currently in the segment, the retention priority, and the relative age of the data now in the segment.

During the execution of a Read command, the target determines whether to transfer all the data directly from the cache segment, or to access the media to retrieve all or part of the requested data. If the media is accessed, the target also chooses how much Read-Ahead is to be done afterward.

### 4.9.2 Options Supported

- Log Sense Vendor Unique Page 35h: Cache Hit/Miss
  - Cumulative Cache Hits on Reads
  - Cumulative Cache Partial Hits on Reads
  - Cumulative Cache Misses on Reads
- Mode Select Page 8: Caching Parameters
  - RCD: Read Cache Disable
  - MF: Multiplication Factor
  - Demand Read Retention Priority
  - Write Retention Priority
  - Disable Pre-Fetch Transfer Length
  - Maximum Pre-Fetch
  - Maximum Pre-Fetch Ceiling
  - Number of Cache Segments
- Mode Select Page 0: Vendor Unique Parameters
  - ADC: Adaptive Caching
- Pre-Fetch Command
- Read-Ahead for Read commands
- Write Retention for Write and Write Verify commands
- DPO: Disable Page Out option on Read Commands
- FUA: Force Unit Access option on Read Commands
- DPO: Disable Page Out option on Write and Write Verify Commands

See 1.5.6, “Log Sense Page 35h” on page 54, 1.6.8, “Page 8h - Read Caching Parameters” on page 85, 1.10, “Read (10)” on page 103, and 1.32, “Write (10)” on page 139 for more details on the caching options.

### 4.9.3 Read-Ahead

The Read-Ahead function consists of reading data into the Target's data buffer that the Initiator has not yet requested. It is a form of anticipatory reading. This function is intended to improve performance for an Initiator that frequently accesses sequential data with successive Read commands.

The Read-Ahead function is controlled by the Mode Select Page 8 Caching parameters (See 1.6.8, “Page 8h - Read Caching Parameters” on page 85). When the

Read-Ahead function is enabled, a Read-Ahead is normally performed following the execution of every Read command. However, if all the data requested by the Read command already resides in a cache segment, then a Read-Ahead is not performed. For this case, the data is transferred directly from the cache segment, and the media is not accessed. If only a portion or none of the data requested by the Read command resides in a cache segment, then the media will be accessed and a Read-Ahead will be performed based on the Mode Select Page 8 Caching parameters and Page 0, ADC bit, as follows:

If the RCD (Read Cache Disable) bit from Mode Select Page 8 is on or if the number of requested blocks for the Read command is greater than the Disable Pre-Fetch Transfer Length from Mode Select Page 8, then a Read-Ahead operation is not performed. Otherwise the number of Read-Ahead blocks for the Read-Ahead operation is calculated based on the Mode Select Page 8 Caching parameters as follows:

If the MF (Multiplication Factor) bit in the Mode Select Page 8 is on, the Maximum Pre-Fetch parameter from Mode Select Page 8 is multiplied by the number of requested blocks from the Read command and used as the number of Read-Ahead blocks. If the MF bit is off, the Maximum Pre-Fetch parameter is used as the number of Read-Ahead blocks.

If ADC (Adaptive Caching) bit in Mode Select Page 0 is on, the number of Read-Ahead blocks maybe reduced if the drive is receiving non-sequential commands.

The number of Read-Ahead blocks is then reduced by any or all of the following factors:

- The number of Read-Ahead blocks is reduced so that it is not larger than the Maximum Pre-Fetch Ceiling parameter from Mode Select Page 8.
- The number of Read-Ahead blocks is reduced so that it is not larger than the size of the data buffer segment. The size of the data buffer segment is determined by the Number of Cache Segments parameter from Mode Select Page 8h.
- If the Demand Read Retention Priority from Mode Select Page 8 is set to Fh, the number of Read-Ahead blocks is reduced so that Read-Ahead data will not overwrite the Requested Data from the Read command. This occurs if the sum of the number of Read-Ahead blocks and Requested Data blocks is larger than the size of the data buffer segment.
- The number of Read-Ahead blocks is reduced so that the target will not read beyond the end of the media.

The Read-Ahead function works as follows:

- At the conclusion of a Read command, the Target continues to read data into its data buffer. The Target reads the data which immediately follows the data requested by the Initiator in the preceding Read command without missing a revolution of the disk. The Target may or may not overwrite the Read command data in the data buffer which has already been sent to the Initiator. This depends on the size of the number of Read-Ahead blocks for the Read-Ahead operation. The Read-Ahead operation continues until the required number of blocks have been read into the data buffer segment. The Target performs any necessary accesses (single cylinder seek and/or multiple head switches) in order to obtain more sequential data. As each block of data is stored in the data buffer during the read ahead operation, the Target maintains information

that identifies the LBAs in the data buffer and their location within the data buffer.

- If an error is encountered during the Read-Ahead operation:
  1. The Target terminates the Read-Ahead operation and does not attempt to recover the data.
  2. The error is not reported to the Initiator for the current Read command. (The error will be reported during the next command if the next command is a Read command that requests the block which encountered the error.)
  3. The blocks which were successfully read prior to the block in error are retained in the cache.
- If a command is received from an Initiator during a Read-Ahead operation, the Target's response varies with the command, as follows:
  - If the command is a priority command (see 4.2, “Priority Commands” on page 203), the Target:
    1. Continues the current Read-Ahead operation.
    2. Executes the priority command.
  - If the command is a Read command whose starting LBA is not in the data buffer segment for the current Read-Ahead operation and is not in any other data buffer segment, or the Force Unit Access (FUA) bit for the Read command is on, the Target:
    1. Terminates the current Read-Ahead operation.
    2. Retains the LBAs in the data buffer segment that had been cached during the preceding Read-Ahead operation.
    3. Selects a new data buffer segment. (See 4.9.4, “Segment Selection” on page 218 for more information.)
    4. Executes the Read command.
    5. Starts a new Read-Ahead operation at the conclusion of the Read command.
  - If the command is a Read command (with FUA off) whose starting LBA is not in the data buffer segment for the current Read-Ahead operation, but is in another data buffer segment, the Target:
    1. Terminates the current Read-Ahead operation.
    2. Retains the LBAs in the data buffer segment that had been cached during the preceding Read-Ahead operation.
    3. Selects the data buffer segment which contains the most data requested by the current Read command. (See 4.9.4, “Segment Selection” on page 218 for more information.)
    4. The target performs the following operations if the selected data buffer segment contains only a portion or none of the requested data:
      - a. Starts a new Read operation to read the required amount of data from the disk for the current Read command and the next Read-Ahead operation.
      - b. Executes the Read command by transferring the LBAs in the selected data buffer segment that had been cached during a previous Read, Read-Ahead, Pre-Fetch, Write, or Write Verify operation.
      - c. Continues with the new Read-Ahead operation at the conclusion of the Read command.

5. The target performs the following operations if the selected data buffer segment contains all of the requested data:
    - a. Transfers the requested data from the data buffer segment.
    - b. Does not access the media and does not start a Read-Ahead operation.
- The target then performs the following operations if the command is a Read command whose starting LBA is in the data buffer segment for the current Read-Ahead operation and FUA is off:
    1. Updates the current Read-Ahead operation to read the required amount of data from the disk for the current Read command and the next Read-Ahead operation.
    2. Executes the Read command by transferring the LBAs in the data buffer segment that had been cached during the preceding Read or Read-Ahead operation.
    3. Continues with the new Read-Ahead operation at the conclusion of the Read command.
  - If any other command is received, the Target:
    1. Terminates the current Read-Ahead operation.
    2. Retains the LBAs in the data buffer segment that had been cached during the preceding Read-Ahead operation.
    3. Executes the command.

**Note:** If the command is a media access command that requires use of the data buffer, a new data buffer segment is selected. (See 4.9.4, “Segment Selection” on page 218 for more information.)
  - If a command is received from an Initiator after a previous Read-Ahead operation has completed, the Target's response varies with the command as follows:
    - If the command is a priority command see also 4.2, “Priority Commands” on page 203 the Target:
      1. Preserves the LBAs in the data buffer that had been cached during the preceding Read-Ahead operation.
      2. Executes the priority command.
    - If the command is a Read command whose starting LBA is not in any data buffer segment, or the Force Unit Access (FUA) bit for the Read command is on, the Target:
      1. Retains the LBAs in the data buffer segment that had been cached during the preceding Read-Ahead operation.
      2. Selects a new data buffer segment. (See 4.9.4, “Segment Selection” on page 218 for more information.)
      3. Executes the Read command.
      4. Starts a new Read-Ahead operation at the conclusion of the Read command.
    - If the command is a Read command whose starting LBA is in the data buffer (any data buffer segment) and FUA is off, the Target:
      1. Retains the LBAs in the data buffer segment that had been cached during the preceding Read-Ahead operation.

2. Selects the data buffer segment which contains the most data requested by the current Read command. (See 4.9.4, “Segment Selection” on page 218 for more information.)
  3. The target performs the following operations if the selected data buffer segment contains only a portion or none of the requested data:
    - a. Starts a new read operation to read the required amount of data from the disk for the current Read command and the next Read-Ahead operation.
    - b. Executes the Read command by transferring the LBAs in the selected data buffer segment that had been cached during a previous Read, Read-Ahead, Pre-Fetch, Write, or Write Verify operation.
    - c. Continues with the new Read-Ahead operation at the conclusion of the Read command.
  4. The target performs the following operations if the selected data buffer segment contains all of the requested data:
    - a. Transfers the requested data from the data buffer segment.
    - b. Does not access the media and does not start a Read-Ahead operation.
- If any other command is received, the Target:
1. Retains the LBAs in the data buffer segment that had been cached during the preceding Read-Ahead operation.
  2. Executes the command.

**Note:** If the command is a media access command that requires use of the data buffer, a new data buffer segment is selected. (See 4.9.4, “Segment Selection” for more information.)

#### 4.9.4 Segment Selection

If the RCD (Read Cache Disable) bit in the Mode Select Page 8 is off, a data buffer segment must be selected before any operation that reads or writes the media can be started.

1. If the operation is one of the following:
  - Read Command with the FUA bit off
  - Pre-Fetch Command

The Cache is scanned to see if the start of the Requested Data is already in any of the segments. If the data is in more than one segment, and Read Ahead is enabled, the segment associated with the Read Ahead is selected. If Read Ahead is disabled the segment selected will be the segment with the most available data after the start of the Requested Data.

2. If the operation is one of the following:
  - Verify Command with the BytChk bit on
  - Write Command
  - Write and Verify Command
  - Write Long
  - Write Same

The Cache is scanned to see if the Requested Data is already in any of them. All segments that have data within the range of the Requested Data are emptied.

3. If the command is one of the following:
  - Mode Select Command with the Save Parameter bit on
  - Pre-Fetch Command and the Start of the Requested Data is not in the cache
  - Read Command and the Start of the Requested Data is not in the cache
  - Read Command with the FUA bit on
  - Read Long Command
  - Verify Command with the BytChk bit on
  - Write Command
  - Write and Verify Command
  - Write Long Command
  - Write Same Command

The following algorithm is used to select the segment to use:

- a. An Empty segment is chosen.
- b. If there are no Empty segments, the oldest Low Priority segment is chosen.
- c. If there are no Low Priority segments, the oldest Middle Priority segment is chosen.
- d. If there are no Middle Priority segments, the oldest High Priority segment is chosen.

Occasionally, the Drive requires a segment to perform some idle time function. When this happens, the preceding algorithm is used to select a segment.

## 4.9.5 Segment Prioritization

Initially, all data buffer segments are empty. When a segment is used, it is given a new priority level based on the type of command being executed and the Retention Priorities set in Mode Select Page 8 by the following algorithm:

1. If the command is one of the following, the segment is marked empty after the command completes:
  - Mode Select Command with the Save Parameter bit on
  - Read Long Command
  - Verify Command with the BytChk bit on
  - Write Long Command
  - Write Same Command
2. If the command is one of the following, all segments are marked empty:
  - Format Unit Command
  - Mode Select Command to Page 8 or Vendor Unique Page 0
  - Reassign Block Command
  - Send Diagnostic Command
  - Write Buffer Command
  - Read Defect Data (10) Command
  - Read Defect Data (12) Command
3. If the command is a Pre-Fetch, the segment is assigned Middle Priority.
4. If the command is a Write, or Write and Verify and:
  - a. The DPO bit is on, the segment is assigned Lowest Priority.
  - b. The DPO bit is off and the Write Retention Priority is:

- 0** The segment is assigned Middle Priority.
- 1** The segment is assigned Low Priority.
- Fh** The segment is assigned High Priority.

5. If the command is a Read and Read-Ahead will occur and:

- a. The DPO bit is on, the segment is assigned Middle Priority.
- b. The DPO bit is off and the Demand Read Retention Priority is:
  - 0** The segment is assigned Middle Priority.
  - 1** The segment is assigned Middle Priority.
  - Fh** The segment is assigned High Priority.

6. If the command is a Read and Read-Ahead will not occur and:

- a. The DPO bit is on, the segment is assigned Lowest Priority.
- b. The DPO bit is off and the Demand Read Retention Priority is:
  - 0** The segment is assigned Middle Priority.
  - 1** The segment is assigned Low Priority.
  - Fh** The segment is assigned High Priority.

Segments chosen for idle time functions are marked empty. All segments are marked empty if any of the following events occur:

- Automatic Block Reassign
- Bus Device Reset Message
- Power on Reset
- Self Initiated Reset
- SCSI Bus Reset



---

## 4.10 Reselection Timeout

A reselection timeout error occurs when the target attempts to reselect an initiator and the initiator does not respond within a Selection Timeout delay (250 mSec). If this occurs, the target releases the SCSI bus to the Bus Free phase (following the reselection timeout procedure) and then retries the reselection one more time. If a second reselection timeout error occurs, the target releases the SCSI bus to the Bus Free phase (following the reselection timeout procedure) and does not attempt to reselect the initiator again. This error condition causes the target to generate sense data with a Sense key of Aborted Command and an Additional Sense code of Select/Reselect failure (45h). If the second reselection is successful, command execution is resumed and no sense data is generated.

---

## 4.11 Single Initiator Selection

For single initiator systems, it is not an error to have only the target ID bit present during selection. Disconnection is not allowed for Single Initiator Selection with only one ID bit present during selection. The initiator must not send an Identify message with the disconnect permission bit (6) set.

---

## 4.12 Non-arbitrating Systems

The target cannot detect whether other SCSI devices on the SCSI bus use arbitration prior to selection. As a consequence, the target allows disconnect permission to be enabled by the Identify message independent of the initiators use of arbitration prior to selection. A non-arbitrating initiator must ensure that disconnect permission in the Identify message is disabled (bit 6=0) for proper operation.

---

## 4.13 Selection without ATN

If the target is selected without the ATN signal active, no Identify message is received from the initiator. In this case, the LUN is identified from the CDB and disconnect permission is disabled. The target does not perform any phase retries. The target still responds to a subsequent attention condition. However, the LUN is not considered to be known if a fatal error is detected during the Command phase. That is, a Command phase parity error or a fatal message error in response to attention condition during Command phase is handled as a Bus Free error with no sense data. The target also allows the use of linked commands if selected without ATN. The target does not initiate synchronous or wide data transfer negotiation if selected without ATN.

Phase retries and target initiated negotiations may be allowed if a subsequent Identify message is received.

---

## 4.14 Multiple Initiator Systems

This section describes how the target supports Multiple initiator systems.

### 4.14.1 Initiator Sense Data

A separate sense data area is reserved for each initiator. Each initiator's sense data is maintained independent of commands from other initiators. This allows a command from one initiator to complete with a Check Condition status and generated sense data, without being affected by a subsequent command from a different initiator. There is no requirement for the first initiator to send a Request Sense command to retrieve the sense data prior to target execution of a command from a different initiator.

### 4.14.2 Initiator Mode Select/Mode Sense Parameters

A single shared copy of the Mode Select/Mode Sense parameters is maintained by the target. This includes both the current and saved parameters. If a Mode Select command is executed, which updates the current parameters, a Unit Attention condition is generated for all initiators except the one that issued the Mode Select command. See 4.1.5, “Unit Attention Condition” on page 184 for more information.

### 4.14.3 Initiator Data Transfer Mode Parameters

A separate data transfer mode parameters area is reserved for each Initiator. Each initiator's data transfer mode (synchronous transfer period, REQ/ACK offset, and data transfer width) is maintained independent of Synchronous Data Transfer Request messages and Wide Data Transfer Request messages from other initiators. This allows multiple initiators to send commands to the target with different data transfer modes without the need to renegotiate the synchronous data transfer mode or the wide data transfer mode during each command.

---

## 4.15 Option Pin

The Drive provides an option pin which allows the integrator to select a Write Protect Mode of operation. The "Jumper Settings" section of the product Functional Specification shows the location of the option pin.

Grounding the option pin allows the Drive to control whether the Drive is write protected or not. With the option pin grounded, the WPEN (Write Protect Enable) controls whether the Drive is in a Write Protect Mode or not. Refer to 1.6.2, "Page 0 - Vendor Unique Parameters" on page 59 for more information regarding the WPEN bit and also for a description of what commands are prohibited in the Write Protect Mode.

**Note:** Grounding the Option Pin does not in itself put the Drive in a write protected mode. Grounding the Option Pin only enables the WPEN bit.

---

## 4.16 LED Pin

The Drive provides a LED pin which allows the integrator to attach an external LED to the drive, and use this pin output to control the LED. The MODE SELECT command allows a selection of various Drive 'conditions' to be reflected at the pin output. See 1.6.2, "Page 0 - Vendor Unique Parameters" on page 59 for information regarding the CMDAC and LED Mode fields. The "Jumper Settings" section of the *product Functional Specification* shows the location of the LED pin.

---

## 4.17 Reset

The Reset condition is used to clear all SCSI devices from the bus. This condition takes precedence over all other phases and conditions. After a reset condition is detected and the reset actions completed, the target returns to a 'SCSI bus enabled' state that allows the target to accept SCSI commands.

This device uses the Hard reset option as defined in the SCSI-2 standard.

### 4.17.1 Reset Sources

There are four sources of resets detected by the target:

Reset Name	Reset Source
Power-On reset	This is the signal generated by the hardware at initial power-on.
Self-Initiated reset	This is a software-generated reset that occurs when a catastrophic error is detected by the microcode (for example, microcode sanity error).
SCSI Bus reset	This is a reset generated when the SCSI bus control line RST goes active.
SCSI Bus Device Reset message	This is the reset generated by the SCSI Bus Device Reset Message (0Ch).

### 4.17.2 Reset Actions

The action taken by the Drive following a reset is dependent on the source of the reset.

#### 4.17.2.1 Power-On reset and Self-Initiated reset

These two reset conditions cause the following to be performed in the order shown.

1. A power-up sequence
2. A start-up sequence is necessary (see 4.18, "Bring-Up Sequence" on page 225) to put the Drive in a ready state

#### 4.17.2.2 SCSI Bus reset and SCSI Bus Device Reset message

These two reset conditions cause the following to be performed.

- If reset goes active while the power-up sequence is in progress, the power-up sequence is started over.
- If the Auto Start pin is grounded and a start-up sequence has not yet completed, a start-up sequence will be re-attempted from the beginning.  
**Note:** The power-up sequence, having already completed, is not rerun.
- If reset occurs while a physical sector is being written, the write operation is disabled after the current physical sector is written. Data is not lost as long as power stays valid until the physical sector being written is completed.

---

## 4.18 Bring-Up Sequence

Two sequences of events must occur after power is applied to the Drive (POR) before the Drive becomes ready. (See the command description of 1.29, “Test Unit Ready” on page 135 for the description of 'ready'.) A power-up sequence and a start-up sequence are required. The following is a chronological list of actions taken by the Target after POR in order to be ready.

The power-up sequence attempts to:

1. Perform BATS1 testing (see 4.19.1, “BATS1 Tests” on page 226)
2. Enable SCSI Bus

(The Target can only respond to SCSI commands after the power-up sequence is complete. )

**Note:** Refer to the *product Functional Specification* for information relating to execution time of the power-up sequence.

If the auto start motor is not enabled by the Auto Start pin being grounded, a Start/Stop Unit command is required to continue bring-up beyond this point. Else bring-up halts at this point until a Start/Stop Unit command is received.

The start-up sequence attempts to:

1. Start the spindle motor. Set ***Motor Stop Degraded*** if unable to spin up.
2. Read the Configuration Record from the Reserved Area (see 4.20, “Reserved Area” on page 227) Set ***Configuration Sector Degraded*** if the configuration record cannot be read.
3. Read the RAM Microcode from the Reserved Area (Microcode upload)
4. Read the remaining configuration information from the Reserved Area
5. Perform BATS2 testing (see 4.19.2, “BATS2 Tests” on page 226)
6. Set the spindle motor synchronization state (see 4.5, “Motor Synchronization” on page 208)
7. Check for an interrupted Format Unit operation.
8. Resume a checkpointed Reassign Blocks command or automatic reallocation (See 4.1.9.4, “Exiting Degraded Mode” on page 200)

---

## 4.19 Basic Assurance Tests (BATS)

Basic Assurance Tests (BATS) are a series of tests performed by the target to internally test the controller and Drive hardware. The BATS are broken into two functional groups, BATS1 and BATS2. The BATS1 tests are run during the power-up sequence. The BATS2 tests are run during the start-up sequence.

### 4.19.1 BATS1 Tests

BATS1 consists of the following tests:

- Microprocessor Functional Test
- Controller Hardware Test
- ROM Checksum Test
- Microprocessor RAM Test

### 4.19.2 BATS2 Tests

BATS2 consists of the following tests:

- Data Buffer Test
- Electronics/Disk Enclosure (DE) Compatibility Test
- Seek Test
- Head Offset Test
- Read Test (all heads)
- Write Test (all heads)
- Cyclic Redundancy Code (CRC) Test
- Error Correction Code (ECC) Test
- Longitudinal Redundancy Code (LRC) Test
- Predictive Failure Analysis (PFA) Test

---

## 4.20 Reserved Area

The target maintains a Reserved Area on the disk. This area is never used for customer data. An initiator has no direct access to this area.

The Reserved Area is used as follows:

- Power-on Self-Test

The Reserved Area is used during the self-test to verify that the Drive can read and write with each of its heads.

- Flags

A set of flags is used by the Drive to checkpoint critical operations. For example, the flags are used to determine whether the Drive lost power during a format or relocate physical sector operation. Operations such as these must be completed before normal processing can resume.

- Manufacturing SAT map (PList)

The map of disk defect sites created at manufacturing time, SAT (Surface Analysis Test), is contained in the Reserved Area. This map is used during format operations but is never altered.

- Grown defect map (GList)

The Reserved Area also contains a grown defect map. The GList map is empty at time of manufacture. Defect locations defined by the initiator with the Reassign Blocks, Format Unit command or Automatic Reallocation are maintained in the GList.

**Note:** The Format Unit command may purge, replace, preserve or add to the GList at the initiator's option.

- Error Logs

The Error Logs are used to maintain information for Predictive Failure Analysis functions such as Automatic/Recommend Block Rewrite/Reallocation.

- RAM Microcode

The Drive's RAM microcode load is contained in the Reserved Area.

- Configuration Information

- Configuration Record (Configuration data and Controller data)
- Inquiry (VPD) data
- Mode Select/Sense saved parameters

- Scratch areas for Reassignment

- Predictive Failure Test Areas

---

## 4.21 Microcode Unique Function

Functions listed below are available through unique loads of microcode. These functions cannot be added to the product through the use of the Write Buffer command. These functions are added in unique releases of ROS microcode.

Disable Target Initiated Synchronous Negotiation

Disable Power-on Reset Unit Attention

Auto Motor spin-up delay based on SCSI ID

Jumper to enable Write Protect Mode (inhibiting write operations)



## Appendix A. SCSI Sense Data Format

Following is the format of data returned by the Target in response to the Request Sense command.

Table 113. Format of Sense Data								
Byte	BIT							
	7	6	5	4	3	2	1	0
0	Valid	Error Code (70h or 71h)						
1	Reserved = 0							
2	RSVD = 0	ILI	RSVD = 0	Sense Key				
3	(MSB) Information (LSB)							
4								
5								
6								
7	Additional Sense Length = 18h							
8	(MSB) Command Specific Information (LSB)							
9								
10								
11								
12	Additional Sense Code							
13	Additional Sense Code Qualifier							
14	Field Replaceable Unit							
15	SKSV	Sense-Key Specific Bits						
16	(MSB) Sense-Key Specific Bytes (LSB)							
17								
18 - 19	Reserved = 0							
20 - 21	Unit Error Code							
22 - 23	Reserved = 0							
24 - 27	Physical Error Record (See Description)							
28 - 31	Reserved = 0							

## A.1.1 Sense Data Description

### Byte 0

#### Valid

The Valid Bit set to one indicates the Information field contains valid information. The Valid Bit, in conjunction with the ILI bit (byte 2) indicate what is in the Information field and the Physical Error Record field. The Valid Bit set to zero indicates the Information field does not contain valid information.

#### Error Code

Error Code set to 70h indicates an error for the current command. Error Code set to 71h indicates a deferred error. This indicates that the error is for a previous active LUN condition that returned a **Good** status. Such commands are associated with the use of the Immediate Bit. The Format Unit command is an example of a command that may cause deferred error sense data to be set.

### Byte 1

Not supported. Set to zero.

### Byte 2

#### ILI

Incorrect Length Indicator is valid for the Read Long and Write Long commands only. ILI set to one and Valid Bit set to one indicates that the requested logical block length did not match the logical block length of the data on the medium for a Read Long or Write Long command. The Information field contains residue information about the error ILI set to zero indicates there is no incorrect length condition.

#### Sense Key

SENSE KEY is the SCSI method of classifying sense information for operating system interpretation. Additional detail may be obtained from the additional sense bytes.

#### **0**     *No Sense*

There is no sense key information to be reported for the logical unit.

#### **1**     *Recovered Error*

The last command completed successfully with recovery action performed by the Target. More detailed information is available in the Additional Sense Code and Additional Sense Code Qualifier.

The contents of the Physical Error Record field (bytes 24 through 27) are valid for this sense key.

Exception: For errors associated with a Predictive Failure Analysis Threshold Reached, Additional Sense Code/Qualifier (5Dh/00h), the contents of the Physical Error Record field may be invalid.

#### **2**     *Not Ready*

The logical unit addressed cannot be addressed. More detailed information is available in the Additional Sense Code and Additional Sense Code Qualifier.

The contents of the Physical Error Record field (bytes 24 through 27) are zero for this sense key.

**3**      ***Medium Error***

The command terminated with a non-recoverable error condition caused by a flaw in the media or an error in the recorded data. More detailed information is contained in the Additional Sense Code and Additional Sense Code Qualifier. The contents of the Physical Error Record field (bytes 24 through 27) are valid for this sense key.

**4**      ***Hardware Error***

The Target detected a non-recoverable hardware error while performing a command or during a diagnostic test. More detailed information is contained in the Additional Sense Code and Additional Sense Code Qualifier. The contents of the Physical Error Record field (bytes 24 through 27) are valid for this sense key.

**5**      ***Illegal Request***

There was an illegal parameter in the Command Descriptor Block or additional parameter supplied as data. If the Target detects an invalid parameter in the CDB, then the command is terminated without altering the medium. If an invalid parameter is detected in parameters supplied as data, then the Target may already have altered the medium. The contents of the Physical Error Record field (bytes 24 through 27) are zero with this sense key.

**6**      ***Unit Attention***

This sense key is reported after an attention causing event.

**7**      ***Data Protect***

This sense key is reported to a Write type command when the target is in Write Protect Mode.

**8 - A**    Not Used

**B**      ***Aborted Command***

The Target aborted the command. The contents of the Physical Error Record field (bytes 24 through 27) are zero with this sense key.

**C - D**    Not Implemented

**E**      ***Miscompare***

The source data did not match the data read from the medium

**F**      Not Implemented

**Byte 3 - 6** Information

- ILI = 1 - Bytes 3 through 6 contain the difference (residue) of the requested length minus the actual length in bytes. Negative values are indicated by two's complement notation.
- ILI = 0 - Bytes 3 through 6 contain the unsigned Logical Block Address associated with the sense key. If the Valid Bit is on, the LBA reported will be within the LBA range of the command as defined in the Command Descriptor Block.

**Note:** An LBA other than the command LBA may be reported on the Reassign Blocks command.

**Byte 7** Additional Sense Length

Byte 7 contains the remaining number of bytes in the sense data. Sense data for this product is always 32 bytes.

**Byte 8 - 11** Command Specific Information

If a *Reassign Blocks* command fails, this field contains the first LBA from the defect descriptor block that was not reassigned.

If an *Auto-Reallocation* fails, this field contains the LBA that was not reassigned.

**Byte 12** Additional Sense Code

**Byte 13** Additional Sense Code Qualifier

**Note:** In Table 114, the Initiator recovery field is the recommended initiator recovery procedure for each Key/Code/Qualifier combination. The table has the Key/Code/Qualifier Combination with the SCSI description and Recommended Initiator Recovery Procedure, followed by a list of target error conditions which will return that combination.

**Note:** In Table 114, the Key, Code, and Qualifier fields are all hex values. (i.e. Sense Key E is Eh, Sense Code 0C is 0Ch, etc.)

Table 114 (Page 1 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
Sense Key = <i>No Sense</i>				
0	00	00	<i>No Additional Sense Information</i> No error.	NONE
Sense Key = <i>Recovered Error</i>				
1	01	00	<i>No Index/Sector Signal</i> Fake and Extra Index. Write with No Sector Pulses.	C.4.3, “Recovered drive error” on page 271

Table 114 (Page 2 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
1	02	00	<b>No Seek Complete</b> Servo error; Seek timeout. Servo error, Recalibrate timeout Three consecutive missing Servo IDs detected by Controller/Channel Hardware	C.4.3, “Recovered drive error” on page 271
1	03	00	<b>Peripheral Device Write Fault</b> Arm Electronics Not Ready. Arm Electronics error. Sector overrun error. Interface Processor write inhibit error. Microjog Write Inhibit IP Retract Error	C.4.3, “Recovered drive error” on page 271
1	09	00	<b>Track Following Error</b> Servo error; Loss of interrupts from the Controller/Channel Hardware Servo error; Settle timeout. Servo error; Coarse offtrack. Servo error; Three consecutive missing Servo IDs detected by Servo Processor	C.4.3, “Recovered drive error” on page 271
1	0C	01	<b>Write Error Recovered With Auto Reallocation</b> Recovered Write error, Auto Reallocated.	C.3.3, “Data error logging” on page 269
1	0C	03	<b>Write Error - Recommend Reassignment</b> Write Error, Recommend Reassignment	C.3.2, “Reassign a physical sector” on page 268
1	14	01	<b>Record Not Found</b> No sector found error (ID no sync. found).	C.3.3, “Data error logging” on page 269
1	14	05	<b>Record Not Found - Recommend Reassignment</b> No sector found error(ID no sync. found), Recommend Reassignment.	C.3.2, “Reassign a physical sector” on page 268
1	14	06	<b>Record Not Found - Data Auto-Reallocated</b> No sector found error(ID no sync. found), Data Auto Reallocated.	C.3.3, “Data error logging” on page 269

Table 114 (Page 3 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
1	15	00	<b>Random Positioning Error</b> Servo error; Unexpected Guardband detected. Servo error; Settle overshoot. Servo error; Maximum velocity exceeded. Servo error; Velocity too high at settle hand off.	C.4.3, “Recovered drive error” on page 271
1	15	02	<b>Positioning Error Detected by Read of Medium</b> Seek positioning error (ID miscompare).	C.4.3, “Recovered drive error” on page 271
1	16	00	<b>Data Synchronization Mark Error</b> No Data sync. found. Data Sync error detected while outside of the write band.	C.3.3, “Data error logging” on page 269
1	16	01	<b>Data Synchronization Mark Error - Data Rewritten</b> No Data sync. found, Data Rewritten	C.3.3, “Data error logging” on page 269
1	16	02	<b>Data Synchronization Mark Error - Recommend Rewrite</b> No Data sync. found, Recommend Rewrite	C.3.1, “Rewrite an LBA” on page 267
1	16	03	<b>Data Synchronization Mark Error - Data Auto-Reallocated</b> No Data sync. found, Data Auto Reallocated	C.3.3, “Data error logging” on page 269
1	16	04	<b>Data Synchronization Mark Error - Recommend Reassignment</b> No Data sync. found, Recommend Reassignment	C.3.2, “Reassign a physical sector” on page 268
1	17	01	<b>Recovered Data with Retries</b> ECC check corrected without using ECC correction. ECC Error Detected while outside of write band corrected without ECC.	C.3.3, “Data error logging” on page 269
1	17	02	<b>Recovered Data with Positive Head Offset</b> Data recovered using positive offsets.	C.3.3, “Data error logging” on page 269

Table 114 (Page 4 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
1	17	03	<b>Recovered Data with Negative Head Offset</b> Data recovered using negative offsets.	C.3.3, “Data error logging” on page 269
1	17	05	<b>Recovered Data using previous sector ID</b> Data recovered using No ID Recovery	C.3.2, “Reassign a physical sector” on page 268
1	17	06	<b>Recovered Data Without ECC - Data Auto-Reallocated</b> Recovered data without ECC, Auto Reallocated.	C.3.3, “Data error logging” on page 269
1	17	07	<b>Recovered Data Without ECC - Recommend Reassignment</b> Recovered data without ECC, Recommend Reassignment.	C.3.2, “Reassign a physical sector” on page 268
1	17	08	<b>Recovered Data Without ECC - Recommend Rewrite</b> Recovered data without ECC, Recommend Rewrite.	C.3.1, “Rewrite an LBA” on page 267
1	17	09	<b>Recovered Data Without ECC - Data Rewritten</b> Recovered data without ECC, Data Rewritten.	C.3.3, “Data error logging” on page 269
1	18	01	<b>Recovered Data with Error Correction and Retries Applied</b> Data correction applied to Drive data for a Data ECC check. ECC Error Detected while outside of write band corrected with ECC.	C.3.3, “Data error logging” on page 269
1	18	02	<b>Recovered Data - Data Auto-Reallocated</b> Recovered data with ECC, Auto Reallocated.	C.3.3, “Data error logging” on page 269
1	18	05	<b>Recovered Data - Recommend Reassignment</b> Recovered data with ECC, Recommend Reassignment	C.3.2, “Reassign a physical sector” on page 268

Table 114 (Page 5 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
1	18	06	<b>Recovered Data With ECC - Recommend Rewrite</b> Recovered data with ECC, Recommend Rewrite.	C.3.1, “Rewrite an LBA” on page 267
1	18	07	<b>Recovered Data With ECC - Data Rewritten</b> Recovered data with ECC, Data Rewritten.	C.3.3, “Data error logging” on page 269
1	1C	01	<b>Primary Defect List Not Found</b> Requested P List does match returned list format <b>(READ DEFECT DATA only)</b>	C.4.13, “Defect List Recovery” on page 278
1	1C	02	<b>Grown Defect List Not Found</b> Requested G List does match returned list format. <b>(READ DEFECT DATA only)</b>	C.4.13, “Defect List Recovery” on page 278
1	1F	00	<b>Partial Defect List Transferred</b> Defect list longer than 64k, 64k of data returned. <b>(READ DEFECT DATA only)</b>	C.4.13, “Defect List Recovery” on page 278



Table 114 (Page 6 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
1	44	00	<p><b>Internal Target Failure</b></p> <p>Servo Error; Invalid Servo Status Received by the Interface Processor</p> <p>Invalid SP Command Sequence.</p> <p>Illegal Head or Cylinder requested.</p> <p>A servo command is already active.</p> <p>Interface Processor detected Servo Sanity Error</p> <p>Controller/Channel Hardware detected Servo Sanity Error</p> <p>Servo error; Target Cylinder out of Range.</p> <p>Servo error; Command not accepted while in Retract.</p> <p>Servo error; Head number out of range.</p> <p>Servo error; Reference track is unreadable.</p> <p>Servo error; Invalid Command.</p> <p>Servo error, Offset out of range.</p> <p>Servo error, Loss of interrupts</p> <p>Servo error; Loss of interrupts</p> <p>Servo error, Loss of interrupts</p> <p>Servo error; Loss of interrupts</p> <p>Interrupt Occured with no interrupt bits set.</p> <p>Motor Speed Error.</p> <p>Channel module Register Write Error</p> <p>Temporary loss of Motor Synchronization.</p> <p>Channel Module Write Parity Error</p> <p>Channel Module Read Parity Error</p> <p>Tangential Correction(TC) RAM Parity Error</p> <p>Data Manager Write Parity Error</p>	C.4.3, "Recovered drive error" on page 271
1	5C	02	<p><b>Spindles Not Synchronized</b></p> <p>Motor Synchronization lost, motor speed maintained.</p>	C.4.10.7, "Spindles Not Synchronized" on page 276

Table 114 (Page 7 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
1	5D	00	<p><b><i>Predictive Failure Analysis Threshold Reached on Recovered Error</i></b></p> <p>Media Problem, Recommend Device Replacement</p> <p>Hardware Problem, Recommend Device Replacement</p> <p>Channel Noise Problem, Recommend Device Replacement</p> <p>Channel Assymetry Problem, Recommend Device Replacement</p> <p>Channel Precompensation Problem, Recommend Device Replacement</p> <p>Channel DC Offset Problem, Recommend Device Replacement</p> <p>Channel Timing Offset Problem, Recommend Device Replacement</p> <p>Fly Height Change Problem, Recommend Device Replacement</p> <p>Torque Amplification Problem, Recommend Device Replacement</p>	C.4.16, "Predictive failure analysis" on page 280
Sense Key = <i>Not Ready</i>				
2	04	00	<p><b><i>Logical Unit Not Ready Cause Not Reportable</i></b></p> <p>Motor is Stuck, Cannot be started.</p> <p>Motor timeout error.</p> <p>Motor Thermal Shutdown</p>	C.4.4, "Drive not ready" on page 271
2	04	01	<p><b><i>logical unit is in the process of becoming ready</i></b></p> <p>Unavailable while Start/Stop Unit Cmd active.</p> <p>Unavailable while Spinup active.</p>	C.4.1, "Drive busy" on page 270
2	04	02	<p><b><i>Logical Unit Not Ready, initializing command required</i></b></p> <p>Degraded Mode/Motor not running.</p>	C.4.6, "Degraded Mode" on page 272
2	04	04	<p><b><i>Logical Unit Not Ready, Format in Progress</i></b></p> <p>Unavailable while Format active.</p>	C.4.1, "Drive busy" on page 270
2	04	B0	<p>&amp;sk204b0.</p> <p>Self Init Reset, W/O Auto Motor Start</p> <p>IREC.2.04.B0</p>	
2	31	00	<p><b><i>Medium Format Corrupted Reassign Failed</i></b></p> <p>Degraded Mode/Reassign Block unsuccessful after pushdown started.</p>	C.4.6, "Degraded Mode" on page 272

Table 114 (Page 8 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
2	31	01	<b>Format Command Failed</b> Degraded Mode/Format unsuccessful.	C.4.6, “Degraded Mode” on page 272
2	40	80	<b>Diagnostic Failure</b> Degraded Mode/Bringup not successful.	C.4.6, “Degraded Mode” on page 272
2	40	85	<b>Diagnostic Failure</b> Degraded Mode/RAM Microcode not loaded, Download incomplete.	C.4.15, “Micro-code error” on page 279
2	4C	00	<b>Logical Unit Failed Self-Configuration</b> Degraded Mode/Configuration not loaded. Degraded Mode/RAM Microcode not loaded.	C.4.6, “Degraded Mode” on page 272
Sense Key = <i>Medium Error</i>				
3	0C	02	<b>Write Error - Auto-Reallocation Failed</b> Recovered Write error, Auto Reallocate failed.	C.3.4, “Reassign Blocks Recovery” on page 270
3	0C	03	<b>Write Error - Recommend Reassignment</b> Write Error, Recommend Reassignment	C.3.2, “Reassign a physical sector” on page 268
3	11	00	<b>Unrecovered Read Error</b> Data ECC Check Data ECC Check detected while outside of the write band.	C.3.3, “Data error logging” on page 269
3	11	04	<b>Unrecovered Read Error - Auto-Reallocation Failed</b> Recovered Read Error, Auto reallocate failed because of unreadable data.	C.3.4, “Reassign Blocks Recovery” on page 270
3	11	0B	<b>Unrecovered Read Error - Recommend Reassignment</b> Unrecovered read error, Recommend Reassignment	C.3.2, “Reassign a physical sector” on page 268
3	14	00	<b>Recorded Entity Not Found</b> Track characterization failure. Unable to determine sector LBA due to adjacent read ID failures, with one sector defective. Reassign(pushdown not started) or Log Sense.	C.3.4, “Reassign Blocks Recovery” on page 270

Table 114 (Page 9 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
3	14	01	<b>Record Not Found</b> No sector found error (ID no sync. found).	C.3.3, “Data error logging” on page 269
3	14	05	<b>Record Not Found - Recommend Reassignment</b> No sector found error(ID no sync. found), Recommend Reassignment.	C.3.2, “Reassign a physical sector” on page 268
3	16	00	<b>Data Synchronization Mark Error</b> No Data sync. found. Data Sync error detected while outside of the write band.	C.3.3, “Data error logging” on page 269
3	16	04	<b>Data Synchronization Mark Error - Recommend Reassignment</b> No Data sync. found, Recommend Reassignment	C.3.2, “Reassign a physical sector” on page 268
3	19	02	<b>Defect List Error in Primary List.</b> Error in Primary Defect list ( <b>READ DEFECT DATA only</b> )	C.4.7, “Reserved Area Hard Error” on page 273
3	19	03	<b>Defect List Error in Grown List.</b> Error in Grown Defect list ( <b>READ DEFECT DATA only</b> )	C.4.7, “Reserved Area Hard Error” on page 273
3	31	00	<b>Medium Format Corrupted Reassign Failed</b> Degraded Mode/Reassign Block unsuccessful after pushdown started. Unrecovered Read Error of Customer Data during Reassign after pushdown started.	C.3.4, “Reassign Blocks Recovery” on page 270
3	31	01	<b>Format Failed</b> Degraded Mode/Format unsuccessful.	C.4.6, “Degraded Mode” on page 272
3	44	00	<b>Internal Target Failure</b> Failure to successfully restore an LBA involved in Track Squeeze Recovery.	C.4.6, “Degraded Mode” on page 272
Sense Key = <b>Hardware Error</b>				

Table 114 (Page 10 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
4	01	00	<b>No Index/Sector Signal</b> No sector pulse found. Fake and Extra Index. Write with No Sector Pulses.	C.4.2, “Unrecovered drive error” on page 271
4	02	00	<b>No Seek Complete</b> Servo processor did not finish command in time. Servo error; Seek timeout. Servo error, Recalibrate timeout Three consecutive missing Servo IDs detected by Controller/Channel Hardware	C.4.2, “Unrecovered drive error” on page 271
4	03	00	<b>Peripheral Device Write Fault</b> Arm Electronics Not Ready. Arm Electronics error. Sector overrun error. Interface Processor write inhibit error. Microjog Write Inhibit IP Retract Error Write Gate not detected during Write	C.4.2, “Unrecovered drive error” on page 271
4	09	00	<b>Track Following Error</b> Servo error; Loss of interrupts from the Controller/Channel Hardware Servo error; Settle timeout. Servo error; Coarse offtrack. Servo error; Three consecutive missing Servo IDs detected by Servo Processor	C.4.2, “Unrecovered drive error” on page 271
4	11	00	<b>Unrecovered Read Error in Reserved Area</b> Data ECC Check (Reserved Area)	C.4.7, “Reserved Area Hard Error” on page 273
4	14	00	<b>Recorded Entity Not Found</b> No Sector Found caused by hardware fault or software.	C.4.2, “Unrecovered drive error” on page 271
4	14	01	<b>Record Not Found - Reserved Area</b> No sector found error (Reserved Area).	C.4.7, “Reserved Area Hard Error” on page 273

Table 114 (Page 11 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
4	15	00	<b>Random Positioning Error</b> Servo error; Unexpected Guardband detected. Servo error; Settle overshoot. Servo error; Maximum velocity exceeded. Servo error; Velocity too high at settle hand off.	C.4.2, “Unrecovered drive error” on page 271
4	15	02	<b>Positioning Error Detected by Read of Medium</b> Seek positioning error (ID miscompare).	C.4.2, “Unrecovered drive error” on page 271
4	16	00	<b>Data Synchronization Mark Error in Reserved Area</b> No Data sync. found.(Reserved Area)	C.4.7, “Reserved Area Hard Error” on page 273
4	19	02	<b>Defect List Error in Primary List.</b> Error in Primary Defect list.	C.4.7, “Reserved Area Hard Error” on page 273
4	19	03	<b>Defect List Error in Grown List.</b> Error in Grown Defect list (used by Format Unit and Reassign Block commands).	C.4.7, “Reserved Area Hard Error” on page 273
4	31	00	<b>Medium Format Corrupted Reassign Failed</b> Unrecovered Hardware or Reserved area Data error during reassign after pushdown started.	C.3.4, “Reassign Blocks Recovery” on page 270
4	32	00	<b>No Defect Spare Location Available</b> GLIST full. Cannot add more entries. Entire track of defective sectors. No spare sectors remaining.	C.4.5, “No defect spare” on page 271
4	32	01	<b>Defect list update failure</b> Defect list update failure.	C.4.7, “Reserved Area Hard Error” on page 273

Table 114 (Page 12 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
4	40	80	<p><b>Diagnostic Failure</b></p> <p>Microcode Check Sum error detected during ROS Test.</p> <p>Microcode Check Sum error detected during RAM Test.</p> <p>Servo Data not present in CSR.</p> <p>Reserved area sector valid check failed.</p> <p>Configuration Sector valid check failed.</p> <p>Configuration Sector uploaded but Check Sum error.</p> <p>Reserved area sector version check failed.</p>	C.4.6, “Degraded Mode” on page 272
4	40	85	<p><b>Diagnostic Failure</b></p> <p>Microcode Check Sum error detected during download of Microcode.</p> <p>Microcode Check Sum error detected during upload of Microcode.</p>	C.4.15, “Microcode error” on page 279
4	40	90	<p><b>Diagnostic Failure</b></p> <p>Servo Data Verify Failure.</p> <p>BATS#2 Error; Seek test failure.</p> <p>BATS#2 Error; Head Offset Test failure.</p> <p>BATS#2 Error. Palette RAM test failure.</p> <p>BATS#2 Error. Digital Filter RAM test failure.</p> <p>BATS#2 Error. Tangential Correction RAM test failure.</p>	C.4.6, “Degraded Mode” on page 272
4	40	A0	<p><b>Diagnostic Failure</b></p> <p>BATS#2 Error. Read write test failure.</p> <p>BATS#2 Error. ECC/CRC test failure.</p> <p>BATS#2 Error. LRC test failure.</p>	C.4.6, “Degraded Mode” on page 272
4	40	B0	<p><b>Diagnostic Failure</b></p> <p>Self Init Reset, W Auto Motor Start</p>	C.4.12, “Self Initiated Reset” on page 278
4	40	C0	<p><b>Diagnostic Failure</b></p> <p>Mismatch between the Servo Processor ROS and Interface Processor RAM.</p> <p>Mismatch between the Interface Processor RAM and DE.</p> <p>Mismatch between the Interface Processor ROS and RAM.</p> <p>Servo error; Mismatch between external Servo Processor RAM and internal Servo Processor ROS.</p>	C.4.11, “Components Mismatch” on page 276

Table 114 (Page 13 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
4	40	D0	<p><i>Diagnostic Failure</i></p> <p>Mismatch between the Servo Processor and the Reference Track Image.</p> <p>Mismatch between the Servo Processor ROS and DE.</p> <p>Mismatch between the Interface Processor ROS and the DE.</p>	<p>C.4.11, “Components Mismatch” on page 276</p>



Table 114 (Page 14 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
4	44	00	<p><b>Internal Target Failure</b></p> <p>Defect List Error prevented one or more defects from being used in a format Unit command or from being reported in a Read Defect Data command.</p> <p>Too few valid GEM measurements available to perform a GEM Predictive Failure Analysis.</p> <p>Mismatch between the Interface Processor ROS and Servo Processor ROS.</p> <p>Microcode error detected while trying to start SCSI data transfer or Missing Buffer Controller Chip interrupt detected.</p> <p>Buffer Controller Chip Channel A Error</p> <p>SCSI Controller Chip internal parity error.</p> <p>Attempt to write critical data without Thermal Updates allowed.</p> <p>SCSI Controller Chip detected an LRC error during read.</p> <p>Reassign could not find the target LBA.</p> <p>Servo Error; Invalid Servo Status Received by the Interface Processor</p> <p>Sanity Error during Read Capacity execution.</p> <p>Target unexpectedly went Bus Free. (<b>Bus Free</b>)</p> <p>SCSI interrupt invalid. (<b>Bus Free</b>)</p> <p>SP interrupt on but SP Status Valid bit is off.</p> <p>Format Track parameter error (number of sectors and number of ID's do not match).</p> <p>Invalid SP Command Sequence.</p> <p>Illegal Head or Cylinder requested.</p> <p>A servo command is already active.</p> <p>Interface Processor detected Servo Sanity Error</p> <p>Controller/Channel Hardware detected Servo Sanity Error</p> <p>Buffer too small to do a requested function.</p> <p>Servo error; Target Cylinder out of Range.</p> <p>Servo error; Command not accepted while in Retract.</p> <p>Servo error; Head number out of range.</p> <p>Servo error; Reference track is unreadable.</p> <p>Servo error; Invalid Command.</p> <p>Servo error, Offset out of range.</p>	C.4.2, "Unrecovered drive error" on page 271

Table 114 (Page 15 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
4	44	00	<p><b>Internal Target Failure</b></p> <p>Servo error, Loss of interrupts</p> <p>Servo error; Loss of interrupts</p> <p>Servo error, Loss of interrupts</p> <p>Servo error; Loss of interrupts</p> <p>SP lost.</p> <p>Interrupt Occured with no interrupt bits set.</p> <p>Motor Speed Error.</p> <p>Channel module Register Write Error</p> <p>Temporary loss of Motor Synchronization.</p> <p>Servo error; Multiple SID Offset Correction error during Tangential Correction.</p> <p>Buffer Controller Chip Sequencer Error</p> <p>Buffer Controller Chip Error</p> <p>Disk Manager Chip detected an LRC error during write.</p> <p>Channel Module Write Parity Error</p> <p>Channel Module Read Parity Error</p> <p>Tangential Correction(TC) RAM Parity Error</p> <p>Data Manager Write Parity Error</p> <p>Invalid UEC</p>	C.4.2, “Unrecovered drive error” on page 271
4	5C	02	<p><b>Spindles Not Synchronized</b></p> <p>Motor Synchronization lost, motor speed maintained.</p>	C.4.10.7, “Spindles Not Synchronized” on page 276
Sense Key = <i>Illegal Request</i>				
5	1A	00	<p><b>Parameter List Length Error</b></p> <p>Command parameter list length error.</p>	C.4.8, “Interface Protocol” on page 273
5	20	00	<p><b>Invalid Command Operation Code</b></p> <p>Invalid Op. code.</p>	C.4.8, “Interface Protocol” on page 273
5	21	00	<p><b>Logical Block Address out of Range</b></p> <p>Invalid LBA.</p>	C.4.8, “Interface Protocol” on page 273

Table 114 (Page 16 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
5	24	00	<b>Invalid Field in CDB</b> CDB Invalid. Data length error on Read Long or Write Long. Invalid Buffer ID in Write Buffer Command.	C.4.8, “Interface Protocol” on page 273
5	25	00	<b>Logical Unit Not Supported</b> Invalid LUN.	C.4.8, “Interface Protocol” on page 273
5	26	00	<b>Invalid Field in Parameter List</b> Command parameter data invalid. Interface Processor ROS and RAM mismatch during Write Buffer Command. Invalid field in Parameter Data, See Field Pointer Value. Invalid LBA in Reassign Command when Reassign degraded. Servo Processor ROS and Interface Processor RAM mismatch during Write Buffer Command. Interface Processor RAM and DE mismatch during Write Buffer Command.	C.4.8, “Interface Protocol” on page 273
5	3D	00	<b>Invalid Bits in Identify Message</b> Reserved bits in Identify message are non zero.	C.4.9, “Aborted Command” on page 274
<b>Sense Key = <i>Unit Attention</i></b>				
6	28	00	<b>Not Ready To Ready Transition, (Medium may have changed)</b> Unit Attention/Not Ready to Ready Transition(Format Completed)	C.4.10.1, “Not Ready to Ready Transition” on page 275
6	29	00	<b>Power On, Reset, or Bus Device Reset occurred</b> Unit Attention/POR. Unit Attention/Self Initiated Reset.	C.4.10.2, “Reset” on page 275
6	2A	01	<b>Mode Parameters Changed</b> Unit Attention/Mode Select Parameters have changed.	C.4.10.3, “Mode Parameters Changed” on page 275
6	2A	02	&sk62a02. Unit Attention/Log Parameters Changed. IREC.6.2A.02	

Table 114 (Page 17 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
6	2F	00	<i>Commands Cleared by Another Initiator</i> Unit Attention/Command cleared by another initiator.	C.4.10.5, “Commands Cleared by Another Initiator” on page 275
6	3F	01	<i>Microcode has been changed</i> Unit Attention/Write Buffer.	C.4.10.4, “Microcode has Changed” on page 275
6	5C	01	<i>Spindles Synchronized</i> Unit Attention/Spindles Synchronized	C.4.10.6, “Spindles Synchronized” on page 275
6	5C	02	<i>Spindles Not Synchronized</i> Unit Attention/Spindles not Synchronized.	C.4.10.7, “Spindles Not Synchronized” on page 276
Sense Key = <i>Data Protect</i>				
7	27	00	<i>Write Protected</i> Command not allowed while in Write Protect Mode.	1.6.2, “Page 0 - Vendor Unique Parameters” on page 59
Sense Key = <i>Aborted Command</i>				
B	1B	00	<i>Synchronous Data Transfer Error</i> Synchronous transfer error, Extra pulses on synchronous transfer.	C.4.9, “Aborted Command” on page 274
B	25	00	<i>Logical Unit Not Supported</i> Different LUN addressed (Identify message) from first selected. ( <b>Bus Free</b> )	C.4.9, “Aborted Command” on page 274

Table 114 (Page 18 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
B	43	00	<p><b>Message Error</b></p> <p>An unexpected CIOP message was received and disconnection was not allowed.</p> <p>A CIOP message was received on an initial connection. <b>(Bus Free)</b></p> <p>Innapropriate Message Reject message received. <b>(Bus Free)</b></p> <p>Attention dropped too late. <b>(Bus Free)</b></p> <p>Message parity error received when no message sent by Target. <b>(Bus Free)</b></p>	C.4.9, "Aborted Command" on page 274
B	44	00	<p><b>Internal Target Failure</b></p> <p>Command aborted due to Fatal Hardware error. Cannot resume the operation (Data transfer).</p>	C.4.9, "Aborted Command" on page 274
B	45	00	<p><b>Select or Reselect Failure</b></p> <p>Reselection timeout. <b>(Bus Free)</b></p>	C.4.9, "Aborted Command" on page 274
B	47	00	<p><b>SCSI Parity Error</b></p> <p>Unrecovered SCSI parity error detected by Target during a command or data phase.</p> <p>Unrecovered SCSI parity error detected by the Target during a MESSAGE OUT phase. <b>(Bus Free)</b></p> <p>Unrecovered SCSI parity error detected by the Initiator (Message Parity Error Message). <b>(Bus Free)</b></p>	C.4.9, "Aborted Command" on page 274
B	48	00	<p><b>Initiator Detected Error Message Received</b></p> <p>Initiator Detected Error for other than STATUS or linked COMMAND COMPLETE phase.</p> <p>Initiator Detected Error message for STATUS or Linked COMMAND COMPLETE phase. <b>(Bus Free)</b></p>	C.4.9, "Aborted Command" on page 274
B	49	00	<p><b>Invalid Message Error</b></p> <p>Invalid message or attention dropped before all bytes of an extended message are transferred. <b>(Bus Free)</b></p>	C.4.9, "Aborted Command" on page 274
B	4E	00	<p><b>Overlapped Commands Attempted</b></p> <p>Invalid Initiator Connection.</p>	C.4.9, "Aborted Command" on page 274
Sense Key = <i>Miscompare</i>				

Table 114 (Page 1 of 18). Valid Sense Key, Code, Qualifier Combinations used by the drive.				
Key	Code	Qual	Description	Initiator Recovery
E	1D	00	<i>Miscompare During Verify Operation</i> Miscompare during byte by byte verify.	C.4.14, “Miscompare recovery” on page 278

**Byte 14**

## Field Replaceable Unit Code

A Field Replaceable Unit Code of zero indicates that no specific mechanism or unit has been identified to have failed or that the data is not available.

A non-zero Field Replaceable Unit Code identifies the extended VPD page which describes the specific mechanism or unit that has failed. These extended VPD pages of the Inquiry command contain information about the field replaceable unit.

**Byte 15 - 17**

## Sense Key Specific

The meaning of the Sense-Key Specific Field, bytes 15-17, depends on which sense key is returned and whether the sense-key specific valid (SKSV) bit is one.

If the SKSV bit is zero, then sense-key specific field is unused and is zero.

If the sense key is **Recovered Error** or **Medium Error** or **Hardware Error**, and if the SKSV bit is one, then the Sense-Key Specific Field shall be defined as shown in Table 115 on page 250. These fields identify the actual number of retries used in attempting to recover from the error condition.

Table 115. Actual Retry Count Bytes								
Byte	BIT							
	7	6	5	4	3	2	1	0
15	SKSV = 1	RSVD = 0						
16 17	(MSB)	Actual Retry Count						(LSB)

The Actual Retry Count field returns the final step number of the DRP or ERP action that was used to attempt recovery from an error.

If the sense key is **Not Ready** and the SKSV bit is one, then the Sense-Key Specific Field shall be defined as shown in Table 116 on page 250. These fields define an indication of progress in completing the *Format Unit* command.

Table 116. Progress Indication Bytes								
Byte	BIT							
	7	6	5	4	3	2	1	0
15	SKSV = 1	RSVD = 0						
16 17	(MSB)	Progress Indication						(LSB)

The Progress Indication Field is a fraction complete indication in which the returned value is the numerator that has 65536 as its denominator.

Progress indication is only given for the *Format Unit* command with the Immed bit set to 1. Therefore, if the sense key is then the SKSV bit is only set to 1 if the additional sense code is Format In Progress.

If the sense key is *Illegal Request* and the SKSV bit is one, then the Sense-Key Specific Field shall be defined as shown in Table 117 on page 251. These fields designate the bytes and bits in error in the *Mode Select* command parameter bytes.

Table 117. Field Pointer Bytes								
Byte	BIT							
	7	6	5	4	3	2	1	0
15	SKSV = 1	C/D	RSVD = 0		BPV	Bit Pointer		
16 17	(MSB) Field Pointer (LSB)							

A Command Data (C/D) bit of one indicates that the illegal parameter is in the command descriptor block. A C/D bit of zero indicates that the illegal parameter is in the data parameter sent by the initiator during the DATA OUT phase.

A Bit Pointer Value (BPV) bit of zero indicates that the value in the bit pointer field is not valid. A BPV bit of one indicates that the bit pointer field specifies which bit of the byte designated by the Field Pointer Field is in error. When a multiple-bit field is in error, the bit pointer field shall point to the most significant (left most) bit of the field. When multiple fields in a byte are in error, the bit pointer field points to the most significant (left most) field in error.

The Field Pointer Field indicates which byte of the command descriptor block or the parameter data that was in error. Bytes are numbered starting from zero. When a multiple byte field is in error, the pointer shall point to the most significant (left most) byte of the field in error.

**Bytes 18 - 19**

Reserved

**Byte 20 - 21**

Unit Error Code

The UEC gives detailed information about the error. It contains a unique code which describes where the error was detected and which piece of hardware or microcode detected the error.

**Bytes 22 - 23**

Reserved



**Byte 24 - 27**

## Physical Error Record

- ILI = 1 - This field contains zeros.
- ILI = 0- These bytes contain the physical location of the error in cylinder, head and sector. Bytes 24 and 25 are Cylinder high and Cylinder low respectively. Byte 26 is the head number, and byte 27 is the sector. If the head and/or sector is undetermined, the value is set to FFh. If the Cylinder value is undetermined, both bytes 24 and 25 are set to FFh.

This field is valid with Sense Key 1, 3 and 4 only. If Cylinder, Head, and Sector have no relevance to the Error, Bytes 24, 25, 26, and 27 will all be set to FFh.

**Bytes 28 - 31**

## Reserved



---

## Appendix B. Recovery Procedures

The following sections describe the recovery procedures for each of the various types of errors for which recovery is attempted.

---

### B.1 Data Recovery Procedure for Data Field Errors

The following table lists the steps that have been defined for data recovery. These recovery actions are for Sync Byte, Data ECC and No Sector Found errors on read commands.

Definitions for the following tables.

- Reread - Read with no parameters altered.
- Rewrite - Write with no parameters altered.
- Read Bias - 0 = Normal Bias Current in Read Head, +1 = Elevated Bias Current in Read Head. -1 = Reduced Bias Current in Read Head,
- TO - Track Offset movement of the Read Head slightly off center. The numbers indicate relative degree of offset 1 = smallest 4 = largest.
- ECC Burst - Amount of ECC correction applied. Double Burst will correct a 41 bit error and under certain conditions can correct up to a 48 bit error. These are errors per physical sector.
- DA - Double burst ECC correction on adjacent error bytes.
- DR - Double burst ECC correction on adjacent or random error bytes.
- EQ - Equalizer Adjust, the Channel Hardware is set to a fixed mode with the equalizer adjusted based on the amount shown.
- TA Mode - Set the channel to thermal asperity recovery mode regardless of whether a thermal asperity was detected or not.
- NSF - No Sector Found error.
- NO ID Mode - Data recovery using tangentially adjacent sector IDs to locate the data in the NSF sector.
- HSC - Head State Change, A short write is done in the non-customer data area to change the residual head state characteristics.
- TU - A thermal update is done to correct for thermal movement in the radial and tangential directions since the data was written.
- TS Mode - Track Squeeze recovery mode in which data from radially adjacent sectors is temporarily moved to the Reserved Area to allow a cleaner read back signal on the error sector. The radially adjacent sectors are written with a DC pattern at the specified offset toward the error sector. Following recovery, the data is restored to the adjacent sectors.
- VCO Calibrate - Calibrates the Voltage Controlled Oscillator in the Channel Hardware.
- A/D Calibrate - Calibrates the Analog to Digital Convertor in the Channel Hardware.

## B.1.1 Read Commands

Table 118 (Page 3 of 7). Read Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
0	Initial Read	0		Count NSF.
1-3	Reread	0		Count NSF. HSC prior to step 3.
4	Reread	+1		Count NSF.HSC prior to step 4.
5	Reread	-1		Count NSF.HSC prior to step 5.
6	+1 TO	0		
7	-1 TO	0		
8	+2 TO	0		
9	-2 TO	0		
10	+3 TO	0		
11	-3 TO	0		
12	+1 EQ	0		
13	-1 EQ	0		
14	+2 EQ	0		
15	-2 EQ	0		
16	+4 TO	0		
17	-4 TO	0		
18-20	Reread	0		Perform Thermal Update, Recalculate cylinder and head and reseek prior to step 18 if NSF count is greater than 1. HSC prior to step 20. Count NSF.
21	Reread	+1		Count NSF.HSC prior to step 21.
22	Reread	-1		Count NSF.HSC prior to step 22.
23-25	Reread	0		Enable No ID Mode prior to step 23 if NSF count is greater than 1. HSC prior to step 25.
26	Reread	+1		HSC prior to step 26.
27	Reread	-1		HSC prior to step 27.
28	+1 TO	0		
29	-1 TO	0		
30	+1 TO	+1		
31	-1 TO	+1		
32	+1 TO	-1		
33	-1 TO	-1		
34	+2 TO	0		
35	-2 TO	0		
36	+2 TO	+1		
37	-2 TO	+1		
38	+2 TO	-1		
39	-2 TO	-1		

Table 118 (Page 4 of 7). Read Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
40	+ 3 TO	0		
41	-3 TO	0		
42	+ 3 TO	+1		
43	-3 TO	+1		
44	+ 3 TO	-1		
45	-3 TO	-1		
46	+ 1 EQ	0		
47	-1 EQ	0		
48	+ 1 EQ	+1		
49	-1 EQ	+1		
50	+ 1 EQ	-1		
51	-1 EQ	-1		
52	+ 2 EQ	0		Count TA
53	-2 EQ	0		Count TA
54	+ 2 EQ	+1		Count TA
55	-2 EQ	+1		Count TA
56	+ 2 EQ	-1		Count TA
57	-2 EQ	-1		Count TA
58	+ 4 TO	0		
59	-4 TO	0		
60	+ 4 TO	+1		
61	-4 TO	+1		
62	+ 4 TO	-1		
63	-4 TO	-1		
64-66	Reread	0		If the TA Count is greater than 2 then enable TA Mode prior to steps 64 through 66. HSC prior to step 66.
67	Reread	+1		If the TA Count is greater than 2 then enable TA Mode prior to step 67. HSC prior to step 67.
68	Reread	-1		If the TA Count is greater than 2 then enable TA Mode prior to step 68. HSC prior to step 68.
69-71	Reread	0	DA	If the TA Count is greater than 2 then enable TA Mode prior to steps 69 through 71. HSC prior to step 71.
72	Reread	+1	DA	If the TA Count is greater than 2 then enable TA Mode prior to step 72. HSC prior to step 72.
73	Reread	-1	DA	If the TA Count is greater than 2 then enable TA Mode prior to step 73. HSC prior to step 73.

Table 118 (Page 5 of 7). Read Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
74-76	Reread	0	DA	Perform Thermal Update, Recalculate cylinder and head and reseek prior to step 74. HSC prior to step 76.
77	Reread	+1	DA	HSC prior to step 77.
78	Reread	-1	DA	HSC prior to step 78.
79	+ 1 TO	0	DA	
80	-1 TO	0	DA	
81	+ 2 TO	0	DA	
82	-2 TO	0	DA	
83	+ 3 TO	0	DA	
84	-3 TO	0	DA	
85	+ 1 EQ	0	DA	
86	-1 EQ	0	DA	
87	+ 2 EQ	0	DA	
88	-2 EQ	0	DA	
89	+ 4 TO	0	DA	
90	-4 TO	0	DA	
91-93	Reread	0	DR	HSC prior to step 93.
94	Reread	+1	DR	HSC and VCO Calibrate prior to step 94.
95	Reread	-1	DR	HSC and A/D Calibrate prior to step 95.
96	+ 1 TO	0	DR	
97	-1 TO	0	DR	
98	+ 1 TO	+1	DR	
99	-1 TO	+1	DR	
100	+ 1 TO	-1	DR	
101	-1 TO	-1	DR	
102	+ 2 TO	0	DR	
103	-2 TO	0	DR	
104	+ 2 TO	+1	DR	
105	-2 TO	+1	DR	
106	+ 2 TO	-1	DR	
107	-2 TO	-1	DR	
108	+ 3 TO	0	DR	
109	-3 TO	0	DR	
110	+ 3 TO	+1	DR	
111	-3 TO	+1	DR	
112	+ 3 TO	-1	DR	
113	-3 TO	-1	DR	

Table 118 (Page 6 of 7). Read Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
114	+1 EQ	0	DR	
115	-1 EQ	0	DR	
116	+1 EQ	+1	DR	
117	-1 EQ	+1	DR	
118	+1 EQ	-1	DR	
119	-1 EQ	-1	DR	
120	+2 EQ	0	DR	
121	-2 EQ	0	DR	
122	+2 EQ	+1	DR	
123	-2 EQ	+1	DR	
124	+2 EQ	-1	DR	
125	-2 EQ	-1	DR	
126	+4 TO	0	DR	
127	-4 TO	0	DR	
128	+4 TO	+1	DR	
129	-4 TO	+1	DR	
130	+4 TO	-1	DR	
131	-4 TO	-1	DR	
132-134	Reread	0	DR	Enable TS Mode with offset 0 prior to step 132. HSC prior to step 134.
135	Reread	+1	DR	HSC prior to step 135.
136	Reread	-1	DR	HSC prior to step 136.
137	+1 TO	0	DR	
138	-1 TO	0	DR	
139	+1 TO	+1	DR	
140	-1 TO	+1	DR	
141	+1 TO	-1	DR	
142	-1 TO	-1	DR	
143	+2 TO	0	DR	
144	-2 TO	0	DR	
145	+2 TO	+1	DR	
146	-2 TO	+1	DR	
147	+2 TO	-1	DR	
148	-2 TO	-1	DR	
149	+3 TO	0	DR	
150	-3 TO	0	DR	
151	+3 TO	+1	DR	
152	-3 TO	+1	DR	
153	+3 TO	-1	DR	
154	-3 TO	-1	DR	

Table 118 (Page 7 of 7). Read Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
155	+ 1 EQ	0	DR	
156	-1 EQ	0	DR	
157	+ 1 EQ	+1	DR	
158	-1 EQ	+1	DR	
159	+ 1 EQ	-1	DR	
160	-1 EQ	-1	DR	
161	+ 2 EQ	0	DR	
162	-2 EQ	0	DR	
163	+ 2 EQ	+1	DR	
164	-2 EQ	+1	DR	
165	+ 2 EQ	-1	DR	
166	-2 EQ	-1	DR	
167	+ 4 TO	0	DR	
168	-4 TO	0	DR	
169	+ 4 TO	+1	DR	
170	-4 TO	+1	DR	
171	+ 4 TO	-1	DR	
172	-4 TO	-1	DR	
173-175	Reread	0	DR	Enable TS Mode with offset 1 prior to step 173. HSC prior to step 175.
176	Reread	+1	DR	HSC prior to step 176.
177	Reread	-1	DR	HSC prior to step 177.
178	+ 1 TO	0	DR	
179	-1 TO	0	DR	
180	+ 1 TO	+1	DR	
181	-1 TO	+1	DR	
182	+ 1 TO	-1	DR	
183	-1 TO	-1	DR	
184	+ 2 TO	0	DR	
185	-2 TO	0	DR	
186	+ 1 TO	+1	DR	
187	-2 TO	+1	DR	
188	+ 2 TO	-1	DR	
189	-2 TO	-1	DR	
190	+ 3 TO	0	DR	
191	-3 TO	0	DR	
192	+ 3 TO	+1	DR	
193	-3 TO	+1	DR	
194	+ 3 TO	-1	DR	
195	-3 TO	-1	DR	



Table 118 (Page 1 of 7). Read Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
196	+1 EQ	0	DR	
197	-1 EQ	0	DR	
198	+1 EQ	+1	DR	
199	-1 EQ	+1	DR	
200	+1 EQ	-1	DR	
201	-1 EQ	-1	DR	
202	+2 EQ	0	DR	
203	-2 EQ	0	DR	
204	+2 EQ	+1	DR	
205	-2 EQ	+1	DR	
206	+2 EQ	-1	DR	
207	-2 EQ	-1	DR	
208	+4 TO	0	DR	
209	-4 TO	0	DR	
210	+4 TO	+1	DR	
211	-4 TO	+1	DR	
212	+4 TO	-1	DR	
213	-4 TO	-1	DR	
214-216	Reread	0	DR	Enable TS mode with offset 2 prior to step 214. HSC prior to step 216. Enable TA Mode prior to step 214 through 216.
217	Reread	+1	DR	HSC prior to step 217. Enable TA Mode prior to step 217.
218	Reread	-1	DR	HSC prior to step 218. Enable TA Mode prior to step 218.
219	+1 TO	0	DR	Enable TA Mode prior to step 219.
220	-1 TO	0	DR	Enable TA Mode prior to step 220.
221	+1 TO	+1	DR	Enable TA Mode prior to step 221.
222	-1 TO	+1	DR	Enable TA Mode prior to step 222.
223	+1 TO	-1	DR	Enable TA Mode prior to step 223.
224	-1 TO	-1	DR	Enable TA Mode prior to step 224.
225	+2 TO	0	DR	Enable TA Mode prior to step 225.
226	-2 TO	0	DR	Enable TA Mode prior to step 226.
227	+2 TO	+1	DR	Enable TA Mode prior to step 227.
228	-2 TO	+1	DR	Enable TA Mode prior to step 228.
229	+2 TO	-1	DR	Enable TA Mode prior to step 229.
230	-2 TO	-1	DR	Enable TA Mode prior to step 230.
231	+3 TO	0	DR	Enable TA Mode prior to step 231.
232	-3 TO	0	DR	Enable TA Mode prior to step 232.
233	+3 TO	+1	DR	Enable TA Mode prior to step 233.

Table 118 (Page 1 of 7). Read Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
234	-3 TO	+1	DR	Enable TA Mode prior to step 234.
235	+3 TO	-1	DR	Enable TA Mode prior to step 235.
236	-3 TO	-1	DR	Enable TA Mode prior to step 236.
237	+1 EQ	0	DR	Enable TA Mode prior to step 237.
238	-1 EQ	0	DR	Enable TA Mode prior to step 238.
239	+1 EQ	+1	DR	Enable TA Mode prior to step 239.
240	-1 EQ	+1	DR	Enable TA Mode prior to step 240.
241	+1 EQ	-1	DR	Enable TA Mode prior to step 241.
242	-2 EQ	-1	DR	Enable TA Mode prior to step 242.
243	+2 EQ	0	DR	Enable TA Mode prior to step 243.
244	-2 EQ	0	DR	Enable TA Mode prior to step 244.
245	+2 EQ	+1	DR	Enable TA Mode prior to step 245.
246	-2 EQ	+1	DR	Enable TA Mode prior to step 246.
247	+2 EQ	-1	DR	Enable TA Mode prior to step 247.
248	-2 EQ	-1	DR	Enable TA Mode prior to step 248.
249	+4 TO	0	DR	Enable TA Mode prior to step 249.
250	-4 TO	0	DR	Enable TA Mode prior to step 250.
251	+4 TO	+1	DR	Enable TA Mode prior to step 251.
252	-4 TO	+1	DR	Enable TA Mode prior to step 252.
253	+4 TO	-1	DR	Enable TA Mode prior to step 253.
254	-4 TO	-1	DR	Enable TA Mode prior to step 254.
255	Reread	0	DR	Enable TA Mode prior to step 255.

## B.1.2 Verify Commands

The recovery procedures for *Write and Verify (2Eh)* and *Verify (2Fh)* commands are as follows. It is important to understand two items relative to verify.

1. A failed command that does not verify is considered a hard error if the byte-by-byte option is in effect and a miscompare, error confirmed to be non-ECC detected, occurs. The sense key reported for this error is Eh (Miscompare Error during Verify byte-by-byte Operation).
2. In all other cases, a failed command that does not verify is not considered a hard error. Although the Drive will return a sense key of 3h (Medium Error) the error is not unrecoverable since Verify recovery is not nearly as extensive as normal read recovery. A subsequent read with full recovery should successfully read the data.

**Note:** "On the Fly ECC" correction is not done when the byte-by-byte option is in effect.

### B.1.2.1 Write and Verify

Table 119. Write and Verify Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
0	Initial Read	0		Count NSF.
1-3	Reread	0		Count NSF.HSC prior to step 3.
4	Reread	+1		Count NSF.HSC prior to step 4.
5	Reread	-1		Count NSF.HSC prior to step 5.
6-8	Reread	0		Perform Thermal Update, Recalculate cylinder and head and reseek if NSF count is greater than 1 prior to step 6. HSC prior to step 8.
9	Reread	+1		HSC prior to step 9.
10	Reread	-1		HSC prior to step 10.

### B.1.2.2 Verify

Table 120 (Page 1 of 2). Verify Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
0	Initial Read	0		Count NSF.
1-3	Reread	0		Count NSF. HSC prior to step 3.
4	Reread	+1		Count NSF. HSC prior to step 4.
5	Reread	-1		Count NSF. HSC prior to step 5.
6-8	Reread	0		Perform Thermal Update, Recalculate cylinder and head and reseek if NSF count is greater than 1 prior to step 6. HSC prior to step 8.
9	Reread	+1		HSC prior to step 9.
10	Reread	-1		HSC prior to step 10.
11	+1 TO	0		
12	-1 TO	0		
13	+1 TO	+1		
14	-1 TO	+1		
15	+1 TO	-1		
16	-1 TO	-1		
17	+2 TO	0		
18	-2 TO	0		
19	+2 TO	+1		
20	-2 TO	+1		
21	+2 TO	-1		
22	-2 TO	-1		
23	+3 TO	0		
24	-3 TO	0		
25	+3 TO	+1		
26	-3 TO	+1		

Table 120 (Page 1 of 2). Verify Recovery Steps				
DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
27	+ 3 TO	-1		
28	-3 TO	-1		
29	+ 1 EQ	0		
30	-1 EQ	0		
31	+ 1 EQ	+ 1		
32	-1 EQ	+ 1		
33	+ 1 EQ	-1		
34	-1 EQ	-1		
35	+ 2 EQ	0		VCO Calibrate prior to step 35.
36	-2 EQ	0		A/D Calibrate prior to step 36.
37	+ 2 EQ	+ 1		
38	-2 EQ	+ 1		
39	+ 2 EQ	-1		
40	-2 EQ	-1		
41	+ 4 TO	0		
42	-4 TO	0		
43	+ 4 TO	+ 1		
44	-4 TO	+ 1		
45	+ 4 TO	-1		
46	-4 TO	-1		
47	Reread	0		

### B.1.3 Write Commands (No Sector Found)

Table 121. Write Recovery Steps			
Step	Action	Read Bias	Miscellaneous
0	Initial Write	0	
1-3	Rewrite	0	HSC prior to step 3.
4	Rewrite	+ 1	HSC prior to step 4.
5	Rewrite	-1	HSC prior to step 5.
6-8	Rewrite	0	Perform Thermal Update prior to step 6. Recalculate cylinder and head and reseek prior to step 6. HSC prior to step 8.
9	Rewrite	+ 1	HSC prior to step 9.
10	Rewrite	-1	HSC prior to step 10.

---

## B.2 Error Recovery for Non-Data Errors

Errors other than data errors as described above invoke Error Recovery Procedures called ERP. ERP errors fall into 4 classes:

- Drive Fault
- Position (Servo)
- Spindle Motor
- Spindle Synchronization Errors.

### B.2.1 Position Error (Servo Error Recovery)

Step	Action	Read Bias	Miscellaneous
0	Initial Servo Operation	0	
1-3	Retry Servo Operation	0	Recalibrate, TU, Seek to last good location, prior to step 1-3.

### B.2.2 Drive Fault

Step	Action	Read Bias	Miscellaneous
0	Initial Read/Write	0	
1-9	Reread/Rewrite	0	Recalibrate, TU, Seek to last good location prior to step 9.

### B.2.3 Spindle Motor Error

A spindle motor error is generated when motor speed can not be maintained. The motor will be stopped and a Not Ready sense key is returned. There is no self initiated recovery taken by the Drive for this error.

### B.2.4 Spindle Synchronization Error

A spindle synchronization error occurs when the target spindle loses synchronization with the master. For information on how this condition is reported, see 4.5, “Motor Synchronization” on page 208. When this condition occurs, the target reverts to unsynchronized operation. Read and write operations are not affected by the synchronization loss.

---

## B.3 Priority of Error Reporting

Multiple errors can occur during the execution of a command. Since only one error is reported in the sense data, a priority scheme for determining which error to report is used. Following are the rules for reporting sense data:

- Hard Errors
  - An unrecovered (hard) error will always be reported in place of any soft error that may have occurred during the same operation.

- If multiple hard error conditions exist, the first hard error detected will be reported.
- Soft Errors
  - When PER=0 (*Mode Select* data format Page 1), recovered data and non-data errors are not reported.
  - When PER=1 (*Mode Select* data format Page 1), the last error detected will be reported.

If an unrecoverable error occurs during idle time, it is reported by the next command processed even though the error may seem inappropriate for that command.

---

## Appendix C. Recommended Initiator Error Recovery Procedures

The Drive's design points for error reporting to the system assumes certain system action for the error return codes. These assumptions are:

1. SCSI-2 protocol will be the first priority in reporting errors.
2. The system will maintain a log of all reported errors.
3. System architecture should include all error handling recommendations made in this appendix. Deviations should have mutual agreement between Drive development and system integration.

This section of the appendix is directed toward documenting the assumptions made by the Drive that the system is expected to implement. The two error classes that the system should be concerned with are DATA and NON-DATA errors.

Data errors are those errors that deal with the handling of data to and from the MEDIA and are identified by the additional sense code contained in the sense data. The additional sense codes for data errors are

- 0C - Write error
- 11 - Unrecovered read error
- 14 - No record found
- 16 - Data Synchronization mark error
- 17 - Recovered read error without ECC correction
- 18 - Recovered read error with ECC correction

Typically, data errors do not include positioning of the heads or the data path through the electronics.

Non-data errors are those errors that do not have a direct relationship with transferring data to and from the media. Non-data errors can include data handling if the media is not associated with the error (i.e. interface errors).

The system action assumed for each class of error is outlined here. System integrators should be aware that deviating from these recommendations can affect drive performance and the system service strategy.

---

### C.1 Drive service strategy

The Drive service strategy is defined so the customer will be able to use the system as soon after a failure is detected as possible. The first priority is to replace the entire drive to make the system operational with minimal service time. The service representative should:

1. Back up all the customer data on this drive if possible
2. Replace the complete drive
3. Restore the customer data
4. Return the drive to customer service

Drive development recognizes the need to preserve customer data. Therefore, there is a procedure that will allow the service representative to replace the electronics portion of the drive at the customer location. This procedure should only be attempted when customer data is critical. If customer data is not critical, then complete drive replacement is recommended. To replace the electronics assembly, the service representative should:

1. Since data cannot be backed up, replace the electronics assembly per the instructions
2. Restore the drive to service
3. If drive performance is questionable after replacing the electronics assembly, back up customer data if possible, then replace the complete drive. Questionable performance can be a high number of recovered errors or reduced data throughput.

The service representative should **NEVER** attempt to replace only the head/disk assembly (HDA).

---

## C.2 Recommendations for System Error Log

The system error log should contain information about the Drive error that will allow recovery actions. The system error logs should contain all the error information returned in the sense data. At a minimum the following information about each error occurrence should be logged.

- Valid bit and error code (Sense byte 0)
- Sense Key (Sense byte 2)
- Information bytes (Sense bytes 3 thru 6)
- Command specific information (Sense bytes 8 thru 11)
- Additional Sense code (sense byte 12)
- Additional Sense code qualifier (sense byte 13)
- Field Replaceable Unit (Sense byte 14)
- Sense Key Specific (sense bytes 15, 16, and 17)
- UEC (Sense bytes 20 & 21)
- Error Record - Physical Block Address (Sense bytes 24 thru 27)

---

## C.3 Data Recovery Procedure

Statistically, most data error activity is noise related and has nothing to do with defects in the media. It is wrong for the system to assume that every data error reported occurred because of a defect in the media. It is also wrong for the system to assume that every data error that occurred because of a media defect rendered the Drive unusable.

Recurring data error activity at the same physical location is an indication of a problem. The problem can be due to a media defect or magnetic damage. A media defect is physical damage to the recording capability of the media while magnetic damage is a defect in the bit pattern written to the media.



The system recovery action to correct these types of defects differs. In both cases, the error can be corrected without replacing the unit. For media defects, the physical sector may require relocation. For magnetic defects, a rewrite of the failing LBA may be all that is required. The Drive determines the need to either rewrite or reassign a sector. The Mode Select page 1 option bits, 1.6, “Mode Select (6)” on page 55, AWRE/ARRE active allows the Drive to rewrite or relocate recovered data errors. Non-recovered data errors and AWRE/ARRE inactive will have additional sense codes returned to recommend rewrite or reassignment of sectors.

**For the cases where a rewrite is recommended, the initiator should write the data using the *Write and Verify (2E)* command.** The verify portion of the *Write and Verify (2E)* command uses limited error recovery, which is desirable in this case. The initiator may elect to verify the write with byte check (*Write and Verify (2E)* command CDB byte 1, bit 1) enabled or disabled.

The need to reassign a sector should be infrequent. Sites not meeting error rate criteria are removed from use during SAT (Surface Analysis Test) in Drive manufacturing. With the exception of some early life SAT escapes (sites that were marginally missed during SAT), reassigning defective sectors should be rare.

Frequent sector reassignment may be an (early) indication of another type of failure. Sector reassignments are monitored as part of the predictive failure analysis. When a threshold is exceeded, the Drive will notify the initiator that a scheduled service action is required.

Drive soft error rates are based on extraneous random faults that are not predictable. Media defects discovered after the Drive completes manufacturing final test need to be relocated so that soft error rates are not influenced by predictable known error sites. Failure of the system to properly relocate defective media sites can have a direct influence on system throughput and drive error rates.

### C.3.1 Rewrite an LBA

The Drive determines the need to rewrite a logical block address (LBA) based on error activity. Once a LBA requires rewriting, the Drive will either rewrite the LBA, or recommend to the initiator that the LBA be rewritten.

When the following sense key, additional sense code, and additional sense code qualifier combinations are returned, the initiator should rewrite the LBA reported in the sense at the next opportunity.

**NOTE:** In Table 124 on page 267 the Key, Code, and Qualifier fields are all hex values. (i.e. Sense key 1 is 1h, sense code 17 is 17h, etc)

Table 124. Recommend Rewrite errors. Sense data combinations for recommend rewrite of an LBA.			
Key	Code	Qual.	Description
1	16	02	Data Synchronization Byte Error - recommend Rewrite.
1	17	08	Recovered data without ECC - Recommend Rewrite.
1	18	06	Recovered data with ECC - Recommend Rewrite.

To rewrite a LBA that has sense data recommending a rewrite, the initiator should rewrite the data using the *WRITE and VERIFY (2E)* command. If the *WRITE and VERIFY (2E)* command completes:

- Successfully (GOOD status) or check condition status for recovered data error, log the error in the system error log.
- Unsuccessfully (Check Condition status) for a medium error, reassign the LBA using the *Reassign Blocks (07)* command.
- Unsuccessfully (Check Condition status) for any error other than recovered data error or medium error, follow the recommendations for the error codes reported.

### C.3.2 Reassign a physical sector

The Drive determines the need to reassign physical sectors based on error activity. Once a physical sector requires reassignment, the Drive will either reassign the physical sector, or recommend to the initiator that the LBA associated with the physical sector be reassigned.

When the following sense key, additional sense code, and additional sense code qualifier combinations are returned, the initiator should reassign the LBA reported at the next opportunity.

**NOTE:** In Table 125 on page 268 the Key, Code, and Qualifier fields are all hex values. (i.e. Sense key 1 is 1h, sense code 17 is 17h, etc)

Key	Code	Qual.	Description
1	0C	03	Write Error - Recommend Reassignment.
1	14	05	Record Not Found - Recommend Reassignment.
1	16	04	Sync Byte Error - Recommend Reassignment.
1	17	05	Recovered data using previous ID
1	17	07	Recovered data without ECC - Recommend Reassignment.
1	18	05	Recovered data with ECC - Recommend Reassignment.
3	0C	03	Write Error - Recommend Reassignment.
3	11	0B	Unrecovered read error - Recommend Reassignment.
3	14	05	Record Not Found - Recommend Reassignment.
3	16	04	Sync Byte Error - Recommend Reassignment.

To reassign an LBA that has sense data recommending a reassignment, the initiator should:

1. Attempt to recover the data from the sector being reassigned with a *Read (08) or Read (28)* command.
2. Reassign the LBA using the *Reassign Blocks (07)* command. If the reassignment completes:
  - Successfully (GOOD status), log the error in the system error log.

- Unsuccessfully (Check Condition status), follow the C.3.4, “Reassign Blocks Recovery” procedure.

3. Write the LBA that was reassigned.

When a LBA is reassigned, the physical location of all LBAs from the LBA being reassigned to the location of the next available spare changes. Initiators wanting to reassign more than one LBA should reassign the largest LBA first and the following LBAs in descending order. All candidates for reassignment are listed in page 32 of log sense data. Refer to 1.5, “Log Sense” on page 44.

### C.3.3 Data error logging

The Drive will report data errors to the initiator that do not require immediate action (successful auto reallocation, successful auto rewrite, or no action needed on this occurrence). The initiator should log these errors in the system error log. No other action is required.

**NOTE:** In Table 126 on page 269 the Key, Code, and Qualifier fields are all hex values. (i.e. Sense key 1 is 1h, sense code 17 is 17h, etc)

Table 126. Log Only errors. Sense data combinations for automatic rewrite or reallocate or that requires no system action except log.			
Key	Code	Qual.	Description
1	0C	01	Recovered Write error - Auto Reallocated.
1	14	01	Record not found
1	14	06	Record Not Found - data Auto-Reallocated.
1	16	00	Data synchronization mark error
1	16	01	Sync Byte Error - data Rewritten.
1	16	03	Sync Byte Error - data auto-reallocated.
1	17	01	Recovered data with retries
1	17	02	Recovered data with positive offset
1	17	03	Recovered data with negative offset
1	17	06	Recovered data without ECC - Auto Reallocated.
1	17	09	Recovered data without ECC - Data Rewritten.
1	18	01	Recovered data with error correction and retries applied
1	18	02	Recovered data with ECC - Auto Reallocated.
1	18	07	Recovered data with ECC - Data Rewritten.
3	11	00	Unrecovered read error - Limited DRP set by initiator
3	14	01	Record not found
3	16	00	Data synchronization mark error

### C.3.4 Reassign Blocks Recovery

The Drive provides the capability to remove media defects without reducing capacity. If the mode parameter bits ARRE/AWRE are active, the Drive will automatically reallocate LBAs determined to be defective. For those LBAs where the error is unrecoverable or the initiator elects to not have the Drive automatically reallocate LBAs, the Drive will recommend reassignment of the LBA.

Recovery from a failed reassignment is to:

- Retry the *Reassign Blocks (07)* command to the same LBA as the failed *Reassign Blocks (07)* command. Note: the LBA of the failed *Reassign Blocks (07)* command is in the command specific bytes of the sense data.
- If the retried *Reassign Blocks (07)* command completes successfully, return to normal processing.
- If the retried *Reassign Blocks (07)* command fails, service the drive using the service guidelines recommended in the C.1, “Drive service strategy” on page 265.

---

## C.4 Non-Data Error Recovery Procedure

The Drive will follow a logical recovery procedure for non-data errors. The initiator options for non-data errors are limited to logging the error, retrying the failing command, or replacing the drive.

These recovery procedures assume the initiator practices data back-up and logs errors at the system level for interrogation by service personnel.

### C.4.1 Drive busy

The Drive is busy performing an operation. Refer to the *product Functional Specification* for time-out limits. **This is not an error condition.** The initiator can test for completion of the operation by issuing a *Test Unit Ready (00)* (media access) command.

**Note** - The **Test Unit Ready (00)** command may return **Good** status when error conditions occur and the option pin is low (-- Heading 'OPTLOW' unknown --). A media access command may be required to test for **Busy** with the option pin low.

- If the *Test Unit Ready (00)* (or media access) command completes with **Check Condition** status then issue a *Request Sense (03)* command.
  - If the specified recovery procedure for the sense data is for a condition other than drive busy, follow the recovery procedure for the condition reported.
  - If the specified recovery procedure for the sense data is for a drive busy condition, and
    - If the drive is not busy for longer than the time specified in the *product Functional Specification*, test for busy again. This is not an error condition.
    - If the drive has been busy for longer than the limit specified in the *product Functional Specification*, then service the drive using the service guidelines recommended in C.1, “Drive service strategy” on page 265.
- If the *Test Unit Ready (00)* (or media access) command completes with **Good** status then return to normal processing.

## C.4.2 Unrecovered drive error

The initiator should:

1. Retry the failing command.
2. If the retry of the failing command completes with
  - a. Good status or recovered sense key, follow the recovery procedure in C.4.3, “Recovered drive error” on page 271.
  - b. Hardware error sense, verify there is no outside cause (e.g. power supply) for the failure, then
    - 1) Retry the failing command.
    - 2) If the retry of the failing command completes with
      - a) Good status, follow the recovery procedure in C.4.3, “Recovered drive error” on page 271.
      - b) Recovered sense or Hardware error sense, then service the drive using the service guideline recommended in C.1, “Drive service strategy” on page 265.

## C.4.3 Recovered drive error

The initiator should log the error as soft with the recovery level.

## C.4.4 Drive not ready

The initiator should

1. Issue a *Start/Stop Unit (1B)* command.
2. Verify the drive comes ready within the time specified in the *product Functional Specification*.

If the drive fails to come ready within the specified time, service the drive using the service guidelines specified in C.1, “Drive service strategy” on page 265.
3. Retry the failing command.
4. If the failing command completes with
  - a. Good status, log the error as recovered.
  - b. Not Ready sense, verify there is no outside cause (e.g. power supply) then service the drive using the service guidelines specified in C.1, “Drive service strategy” on page 265.

## C.4.5 No defect spare

Three conditions can exist that will cause this error. The three conditions are:

1. When the *Reassign Blocks (07)* command is issued and all the available spares on the cylinder with the sector in error and all the spares on the two subsequent cylinders are used. In this case, there are no spares available for the Drive to use for the relocation requested.
2. When last usable physical sector on a track is the sector being reassigned. This reassignment would make all the sectors on a track reassigned. The Drive requires at least one (1) usable sector on each track.
3. When the Glist is full and the sector to be reassigned cannot be added.

Service the Drive following the C.1, “Drive service strategy” on page 265.

## C.4.6 Degraded Mode

Refer to 4.1.9, “Degraded Mode” on page 190 for the definition of this state. There are five (5) causes for entering degraded mode. In all cases the Sense Key is *Not Ready*. They are:

1. Sense Code/Qualifier of **Logical Unit Not Ready, initializing command required**. The spindle motor not spinning or not at the proper speed.

This may not be an error condition. The initiator should issue a *Start Unit (1B)* command to start the spindle motor.

If the Drive fails to come ready in the specified time (reference the *product Functional Specification*), service the drive using the service guideline recommended in C.1, “Drive service strategy” on page 265.

2. Sense Code/Qualifier of **Logical Unit Failed Self-Configuration**. File configuration record has not been read or is not readable This condition is the result of the configuration data located in the reserved area not being read because of an error in the configuration data records or an error prior to the process step that reads the configuration data. In either case, recovery requires the Drive to successfully read the configuration data. This can be done one of two (2) ways.

First, power may be cycled. This will initiate the power up process that includes reading in the configuration data. If the error that caused the configuration data to not be read occurs again, then the configuration data will not be loaded.

The second method to load configuration data is to execute a *Start Unit (1B)* command.

3. Sense Code/Qualifier of **Logical Unit Failed Self-Configuration**. RAM microcode not loaded. The RAM microcode is loaded as part of the spindle start up. If an error occurs during the spin up or as part of the test process after the motor has reached speed, then the RAM microcode may not be loaded.

Recovery for this step is the same as for the configuration data not being loaded. Either a power cycle or the successful completion of a *Start Unit (1B)* command will cause the Drive to attempt to load the RAM microcode.

In addition to the power cycle or *Start Unit (1B)* command, the initiator may attempt to load microcode (reference 1.34, “Write Buffer” on page 142).

4. Sense Code/Qualifier of **Diagnostic Failure**. Failure of a Send Diagnostic self test, a start up sequence, or other internal target failures
  - Failure of a send diagnostic self test or a start up sequence. This failure is the result of the diagnostics that are executed during power on or when the *Send Diagnostic (1D)* command is executed detecting a failure. As with the RAM code not loaded and the configuration data not loaded, the recovery is either a power cycle or issuing the *Send Diagnostic (1d)* command with the self test bit set active.
    - Recovery for a failed *Send Diagnostic (1D)* command is either:
      - a. Execute the *Send Diagnostic (1D)* command, or
      - b. Power cycle the drive.
    - If the failure repeats, service the drive using the service guideline recommended in C.1, “Drive service strategy” on page 265.

- Recovery for a failed power up sequence is either:
  - a. Issue a *Start Unit (1B)* command, or
  - b. Power cycle the drive.

If the failure repeats, service the drive using the service guideline recommended in C.1, “Drive service strategy” on page 265.

- Internal target failures

The file periodically adjusts the track following for each head to compensate for expansion and contraction of the disks due to temperature changes. If one of these adjustments fails, the file will enter a degraded mode to prevent writing data off track.

Recovery of this condition is either a power cycle or successful completion of the *Send Diagnostic (1D)* command. Service the drive using the recommended service guidelines specified in C.1, “Drive service strategy” on page 265 if the power cycle or the *Send Diagnostic (1D)* command fail to complete successfully.

5. Sense Code/Qualifier of ***Format Command Failed Format Unit (04)***, Sense Code/Qualifier of ***Medium Format Corrupted Reassign Failed Reassign Blocks (07)*** command, or an automatic reallocation failed or was abnormally terminated.

Recovery from a reassign degraded condition is to retry the failed *Reassign Blocks (07)* command (see C.3.4, “Reassign Blocks Recovery” on page 270). The sense data returned for a reassign degraded condition contains the LBA that was being reassigned in the command specific bytes.

Recovery from a failed *Format Unit (04)* command is to retry the command. If the command fails a second time, service the drive following the procedure defined in C.1, “Drive service strategy” on page 265.

If the above defined recovery procedures fail to clear the degraded mode condition, the the Drive should be replaced. Follow the procedure C.1, “Drive service strategy” on page 265 when replacing the drive.

### C.4.7 Reserved Area Hard Error

Sectors found defective in the reserved area of the disk cannot be reassigned after the Drive leaves the factory. The data in the reserved area is not directly accessible by the initiator. For this reason, the reserved area has all data except the *Reassign Blocks (07)* command work area duplicated. A data error must occur in both copies of the data record before the Drive considers a reserved area read error. When this happens, the integrity of the drive is questionable.

Service the Drive using the C.1, “Drive service strategy” on page 265 procedure.

### C.4.8 Interface Protocol

For all interface protocol errors, the initiator should:

1. Correct the parameter that caused the illegal request
2. Retry the failing command.
3. If the first retry of the failing command completes with
  - a. Good status, log the error as recovered

- b. Check condition status with sense data for an illegal request, verify there is no outside cause (e.g. power supply) for the failure
- c. Other, follow the recommendations for the error condition reported.
  - 1) Retry the failing command.
  - 2) If this retry of the failing command completes with
    - a) Good status, log the error as recovered
    - b) Check condition status with sense data for an illegal request, service the drive using the service guideline recommended in C.1, “Drive service strategy” on page 265.
    - c) Other, follow the recommendations for the error condition reported.

**Note:** - During a microcode download, the Drive verifies the level of microcode being loaded with the level resident in ROS. If the two microcode levels are incompatible, the command terminates with check condition status and sense for illegal request and the additional sense code for invalid field in parameter list.

If all the command parameters are correct, the command is a *Write Buffer (3B)* command with a download or download and save microcode mode selected, and the additional sense code is 26, the cause of the illegal request may be an incompatible LID. To verify the LID is correct, the initiator should:

1. Issue an *Inquiry (12)* command with the EVPD option set to 1 and the identifier set to 03.
2. Determine the correct level of microcode from the inquiry data. Refer to 1.3, “Inquiry” on page 28 for the location of the LID.
3. Load the correct level of microcode to the drive using the *Write Buffer (3B)* command with the correct mode.
4. If the error repeats or the *Write Buffer (3B)* command fails to complete successfully, service the drive using the service guideline recommended in C.1, “Drive service strategy” on page 265.

## C.4.9 Aborted Command

The initiator should determine the cause from the additional sense code (byte 12)

- Sense key = B (Aborted Command) with additional sense codes of 1B, 25, 43, 45, 49, and 4E and sense key = 5 (Illegal Request) with additional additional sense code of 3D are initiator caused abort conditions. The initiator should correct the condition that caused the abort and retry the failing command.
- Sense key = B (Aborted Command) with additional sense code of 44 or 48 are Drive caused abort conditions The initiator should:
  1. Retry the failing command.
  2. If the retry of the failing command completes with
    - a. Good status, log the error as recovered
    - b. Abort command sense, verify there is no outside cause (e.g. power supply) for the failure.
  3. Retry the failing command.



4. If the retry of the failing command completes with
  - a. Good status, log the error as recovered
  - b. Abort command sense, then service the drive using the service guideline recommended in C.1, “Drive service strategy” on page 265.
- Sense key = B (Aborted Command) and an additional sense code of 47 can be an initiator or Drive caused abort condition. The initiator should follow the above procedure for initiator caused abort conditions if the Drive detected the SCSI bus parity error. The initiator should follow the above procedure for Drive caused abort conditions if the initiator detected the SCSI bus parity error.

## **C.4.10 Unit Attention**

Unit Attention conditions are not errors. They alert the initiator that the Drive had an action that may have changed an initiator controlled state in the drive. These conditions are:

### **C.4.10.1 Not Ready to Ready Transition**

Not ready to ready transition, unit formatted. This unit attention condition will not be reported to the initiator that issued the *Format Unit (04)* command.

### **C.4.10.2 Reset**

Reset - This means the drive was reset by either a power-on reset, Bus reset, a Bus Device Reset message, or an internal reset.

### **C.4.10.3 Mode Parameters Changed**

A *Mode Select (15)* command successfully completed - This means that the mode parameters that are the current value may have changed. The parameters may or may not have changed but the command to change the parameters successfully completed. The Drive does not actually compare the old current and the new current parameters to determine if the parameters changed. This unit attention condition will not be reported to the initiator that issued the *Mode Select (15)* command.

### **C.4.10.4 Microcode has Changed**

*Write Buffer (3B)* command to download microcode has successfully completed. This means that the microcode that controls the Drive has been changed. The code may or may not be the same as the code resident on the media. The Drive does not compare old level code with new code.

### **C.4.10.5 Commands Cleared by Another Initiator**

Tagged commands cleared by a clear queue message. This means that the command queue has been cleared. The unit attention condition is not reported to the initiator that issued the clear queue message. Unit attention condition is reported to all initiators that had commands active or queued.

Reissue any outstanding command.

### **C.4.10.6 Spindles Synchronized**

Spindles synchronized - This means that the spindle of this Drive achieved synchronization with the spindle of another device. This unit attention condition is issued to all initiators including the initiator requesting spindle synchronization.

NOTE: If the saved mode parameters are for a synchronized mode, unit attention condition will not be reported when the spindle comes into synchronization from a start up sequence.

#### C.4.10.7 Spindles Not Synchronized

Spindles not synchronized - This means that the Drive's spindle was synchronized with another device and that synchronization is lost. The two conditions that will cause loss of spindle synchronization are a hardware fault or a *Start/Stop Unit (1B)* command with the mode parameter RPL bits changed to the no synchronization state (00). If the initiator requires spindle synchronization, recovery from loss of synchronization is the same for either cause.

The initiator should:

1. Issue a *Mode Select (15)* command, page 4, with the desired RPL mode bits and rotational offset selected.
2. Issue a *Start/Stop Unit (1B)* command with the start bit active.
3. Issue a *Test Unit Ready (00)* command followed by a *Request Sense (03)* command. If the *Test Unit Ready (00)* command or the *Request Sense (03)* command complete with sense for
  - Not ready, logical unit in process of becoming ready, verify that the attempt to achieve synchronization is still within the specified time limit (4.5, “Motor Synchronization” on page 208) then return to the beginning of this step.
  - Unit attention condition, spindles not synchronized, the loss of synchronization is probably caused by a *Start/Stop Unit (1B)* command with the RPL mode bits selected for no synchronization (00). Return to normal processing.
  - Unit attention condition, spindles not synchronized, a hardware fault is probably the cause of failure. Locate and correct the cause of loss of synchronization.
  - Good status with no sense data, this is an error condition. The Drive will report check condition status with sense data for a unit attention condition before good status is reported. If good status is reported before the unit attention condition, service the drive following the service guideline recommendations in C.1, “Drive service strategy” on page 265
  - Any other status and sense combination, follow the recommendations for the returned sense.

#### C.4.10.8 Log Select Parameters Changed

A Log Select (4C) command successfully completed - This means that the the Log Select command cleared statistical information successfully (See Log Select command). This unit attention condition is reported to all initiators excluding the initiator that issued the Log Select command.

#### C.4.11 Components Mismatch

A compatibility test is performed during BATS-2. The compatibility test verifies that the electronics assembly is compatible with the HDA. When the Drive detects a mismatch in the components, the most likely cause is the result of incorrect parts used during a service action.

If the error reported is

**Key/code/qualifier 4/40/C0** - Diagnostic failure, Load IDs do not match on bring-up

The initiator should determine

1. The proper level of microcode from page 3 of the EVPD inquiry data. Refer to 1.3, "Inquiry" on page 28.
2. Load the proper level of microcode to the media using the *Write Buffer (3B)* command with the download and save option.

To load new microcode, the initiator should:

- a. Issue an *Inquiry (12)* command for EVPD page 3. EVPD page 3 of the inquiry data contains the valid LID.
  - b. Issue a *Write Buffer (3B)* command with the download and save option. The microcode used during the data out phase should have the same LID as the LID reported in page 3 of the inquiry data.
  - c. If the *Write Buffer (3B)* command completes with:
    - GOOD status, continue to next step.
    - Check Condition status, service the drive using the service guidelines recommended in C.1, "Drive service strategy" on page 265.
3. Execute the *Send Diagnostic (1D)* command with the self test bit active.
  4. Check the send diagnostic end status. If the status is
    - GOOD, Return to normal processing.
    - CHECK CONDITION, issue a *Request Sense (03)* command and follow the recommendations for the sense data returned unless the sense data is for a component mismatch. If the sense data is for component mismatch, service the drive using the service guideline recommended in C.1, "Drive service strategy" on page 265.

**Key/code/qualifier 4/40/D0** - Diagnostic failure, Electronics mismatch on bring-up.

The initiator should determine

1. The proper level of electronics from page 2 of the EVPD inquiry data. Refer to 1.3, "Inquiry" on page 28.
2. Replace the electronics with the correct part number electronics.
3. If bring-up complete status is
  - GOOD, Return to normal processing.
  - CHECK CONDITION, issue a *Request Sense (03)* command and follow the recommendations for the sense data returned unless the sense data is for a component mismatch. If the sense data is for component mismatch, service the drive using the service guideline recommended in C.1, "Drive service strategy" on page 265.

## C.4.12 Self Initiated Reset

The Drive will initiate a self reset when the condition of the Drive cannot be determined. The internal reset will terminate any outstanding commands, release any reserved initiators, execute BATS, and stop the spindle motor. The initiator can recover by:

- Sense key = 2,
  1. Log the error
  2. Issue a *Start/Stop Unit (1B)* command
  3. Retry the failing command. If the failing command completes with
    - Good status, return to normal processing.
    - Self initiated reset sense, service the drive following the guidelines recommended in C.1, “Drive service strategy” on page 265.
    - Other, follow the recommendations for the error reported.
- Sense key = 4,
  1. Log the error
  2. Retry the failing command. If the failing command completes with
    - Good status, return to normal processing.
    - Self initiated reset sense, service the drive following the guidelines recommended in C.1, “Drive service strategy” on page 265.
    - Other, follow the recommendations for the error reported.

## C.4.13 Defect List Recovery

**This is not an error condition.**

The initiator either requested a defect list in a format (block or vendor specific) that the Drive does not support or the requested defect list(s) exceed the maximum list length that can be returned. If the sense key/code/qualifier are:

1/1F/00, the requested list(s) exceed the maximum length that can be supported. The initiator should request one list at a time. If a single list exceeds the maximum returnable length, this may be an indication of a marginally operational drive. Service the drive using following the service guidelines in C.1, “Drive service strategy” on page 265.

1/1C/01 or 1/1C/02, the requested defect list is not in the format that the Drive supports. The requested defect list is returned in the physical (cylinder, sector, head) format. This is the default format. There is no initiator action required for this condition.

## C.4.14 Miscompare recovery

A miscompare can occur on a *Verify (2F)* command or a *Write and Verify (2E)* command with the byte check (BytChk) bit active. Recovery for a miscompare error is different for the two commands.

### **Verify command**

The initiator should:

1. Verify that the data sent to the drive is the correct data for the byte-by-byte compare.
2. Read the data from the media with a *Read (08)* or *Read (28)* command and verify that the data from the media is the expected data for the byte-by-byte compare.
  - If all data are correct, this is an indication that the data may have been read from the media incorrectly without an error detected. Service the drive using the procedure specified in C.1, “Drive service strategy” on page 265.
  - If all data are not correct, this is an indication that the data on the media is not the data the initiator expected. Rewrite the correct data to the media.

#### **Write and Verify command**

The drive uses the same data in the data buffer to write then read and compare. A miscompare error on the *Write and Verify (2E)* command is an indication that the drive cannot reliability write or read the media. Service the drive using the procedures specified in C.1, “Drive service strategy” on page 265.

### **C.4.15 Microcode error**

The microcode loaded from the media or from the interface is validated before the device operates using that microcode. When the validation detects incorrect or incomplete data, the Drive enters degraded mode.

If the initiator attempted to load microcode using the *Write Buffer (3B)* command, retry the *Write Buffer (3B)* command. If the command completes with

- Good status - return to normal processing
- Check condition status - service the drive using the service guidelines recommended in C.1, “Drive service strategy” on page 265.

If the check sum error occurred during normal processing, the initiator may attempt to load microcode to the media before deciding to service the drive using the service guidelines recommended in C.1, “Drive service strategy” on page 265.

To load new microcode, the initiator should:

1. Issue an *Inquiry (12)* command for EVPD page 3. EVPD page 3 of the inquiry data contains the valid LID.
2. Issue a *Write Buffer (3B)* command with the download and save option. The microcode used during the data out phase should have the same LID as the LID reported in page 3 of the inquiry data.
3. If the *Write Buffer (3B)* command completes with:
  - GOOD status - return to normal processing

Retry the failing command. If the command complete with

- Good status - Continue normal processing.
- Check condition status for check sum error - Service the drive using the service guidelines recommended in C.1, “Drive service strategy” on page 265.

- Check condition for any other error - follow the recommended recovery procedure for the error reported.
- Check Condition status for Check sum error, service the drive using the service guidelines recommended in C.1, “Drive service strategy” on page 265.
- Check condition status for any other error, follow the recommendations for the returned sense data.

#### **C.4.16 Predictive failure analysis**

The Drive performs error log analysis and will alert the initiator of a potential failure. The initiator should determine if this device is the only device with error activity.

If this drive is the only drive attached to the initiator with error activity, service the drive using the procedures specified in C.1, “Drive service strategy” on page 265. Note, service for this drive can be deferred. The longer service is deferred, the more probable a failure can occur that will require immediate service.

If more than this drive is experiencing error activity, the drive is probably not at fault. Locate and service the outside source causing error activity on this drive.

## Appendix D. UEC List

Following is the list of Unit Error Codes and associated descriptions. The Unit Error Codes are returned by the target in sense data bytes 20-21 in response to the Request Sense command.

**Note:** The list of Unit Error Codes and descriptions does not have a direct correlation to the error descriptions and Sense Key/Code/Qualifier descriptions in Appendix A, “SCSI Sense Data Format” on page 229.

Table 127 (Page 3 of 8). Unit Error Codes with Descriptions.	
Unit Error Code	Description
00 00	No error.
01 01	Degraded Mode/Motor not running.
01 02	Unavailable while Start/Stop Unit Cmd active.
01 03	Unavailable while Spinup active.
01 04	Unavailable while Format active.
01 05	Synchronous transfer error, Extra pulses on synchronous transfer.
01 06	Requested P List does match returned list format ( <b>READ DEFECT DATA only</b> )
01 07	Requested G List does match returned list format. ( <b>READ DEFECT DATA only</b> )
01 08	Defect List Error prevented one or more defects from being used in a ormat Unit command or from being reported in a Read Defect Data command.
01 0A	Defect list longer than 64k, 64k of data returned. ( <b>READ DEFECT DATA only</b> )
01 10	Too few valid GEM measurements available to perform a GEM Predictive Failure Analysis.
01 11	Degraded Mode/Reassign Block unsuccessful after pushdown started.
01 12	Degraded Mode/Format unsuccessful.
01 13	Degraded Mode/Configuration not loaded.
01 14	Degraded Mode/RAM Microcode not loaded.
01 15	Degraded Mode/RAM Microcode not loaded, Download incomplete.
01 1C	Command not allowed while in Write Protect Mode.
01 1D	An unexpected CIOP message was received and disconnection was not allowed.
01 1E	A CIOP message was received on an initial connection. ( <b>Bus Free</b> )
01 1F	Mismatch between the Servo Processor and the Reference Track Image.
01 20	Microcode Check Sum error detected during download of Microcode.
01 21	Mismatch between the Interface Processor ROS and Servo Processor ROS.

Table 127 (Page 4 of 8). Unit Error Codes with Descriptions.	
Unit Error Code	Description
01 22	Degraded Mode/Bringup not successful.
01 23	Microcode error detected while trying to start SCSI data transfer or Missing Buffer Controller Chip interrupt detected.
01 24	Mismatch between the Servo Processor ROS and DE.
01 25	Mismatch between the Servo Processor ROS and Interface Processor RAM.
01 26	Mismatch between the Interface Processor RAM and DE.
01 27	Buffer Controller Chip Channel A Error - Parity error during transfer in.
01 28	Buffer Controller Chip Channel A Error - Parity error during transfer out.
01 29	Buffer Controller Chip Channel A Error - Programmed IO Parity error.
01 2A	Buffer Controller Chip Channel A Error - Unexpected error.
01 2B	Command aborted due to Fatal Hardware error.
01 2C	SCSI Controller Chip internal parity error.
01 2D	Cannot resume the operation (Data transfer).
01 2E	Mismatch between the Interface Processor ROS and RAM.
01 2F	Mismatch between the Interface Processor ROS and the DE.
01 30	Invalid Op. code.
01 31	Invalid LBA.
01 32	CDB Invalid.
01 33	Invalid LUN.
01 34	Command parameter data invalid.
01 35	Command parameter list length error.
01 36	Interface Processor ROS and RAM mismatch during Write Buffer Command.
01 37	Data length error on Read Long or Write Long.
01 38	Invalid field in Parameter Data, See Field Pointer Value.
01 39	Invalid LBA in Reassign Command when Reassign degraded.
01 3A	Invalid Buffer ID in Write Buffer Command.
01 3B	Servo Processor ROS and Interface Processor RAM mismatch during Write Buffer Command.
01 3C	Interface Processor RAM and DE mismatch during Write Buffer Command.
01 3E	Attempt to write critical data without Thermal Updates allowed.
01 3F	SCSI Controller Chip detected an LRC error during read.
01 40	Unit Attention/Not Ready to Ready Transition(Format Completed)



Table 127 (Page 5 of 8). Unit Error Codes with Descriptions.	
Unit Error Code	Description
01 41	Unit Attention/POR.
01 42	Unit Attention/Mode Select Parameters have changed.
01 43	Unit Attention/Write Buffer.
01 44	Unit Attention/Command cleared by another initiator.
01 45	Unit Attention/Self Initiated Reset.
01 47	Unit Attention/Spindles not Synchronized.
01 48	Unit Attention/Spindles Synchronized
01 49	Unit Attention/Log Parameters Changed.
01 50	Microcode Check Sum error detected during ROS Test.
01 51	Microcode Check Sum error detected during RAM Test.
01 52	Microcode Check Sum error detected during upload of Microcode.
01 53	Motor Synchronization lost, motor speed maintained.
01 56	GLIST full. Cannot add more entries.
01 57	Entire track of defective sectors.
01 58	No sector pulse found.
01 59	Defect list update failure.
01 5A	Motor is Stuck, Cannot be started.
01 5C	Reassign could not find the target LBA.
01 5D	No Sector Found caused by hardware fault or software.
01 5E	No spare sectors remaining.
01 5F	Error in Primary Defect list.
01 60	Initiator Detected Error for other than STATUS or linked COMMAND COMPLETE phase.
01 61	Unrecovered SCSI parity error detected by Target during a command or data phase.
01 62	Invalid Initiator Connection.
01 63	Media Problem, Recommend Device Replacement
01 64	Hardware Problem, Recommend Device Replacement
01 65	Error in Primary Defect list ( <b>READ DEFECT DATA only</b> )
01 66	Error in Grown Defect list ( <b>READ DEFECT DATA only</b> )
01 68	Failure to successfully restore an LBA involved in Track Squeeze Recovery.
01 6A	Servo Error; Invalid Servo Status Received by the Interface Processor
01 6B	Arm Electronics Not Ready.
01 6C	Sanity Error during Read Capacity execution.
01 6D	Target unexpectedly went Bus Free. ( <b>Bus Free</b> )
01 6E	Servo Data not present in CSR.
01 6F	Servo Data Verify Failure.
01 71	Different LUN addressed (Identify message) from first selected. ( <b>Bus Free</b> )

Table 127 (Page 6 of 8). Unit Error Codes with Descriptions.	
Unit Error Code	Description
01 72	Innapropriate Message Reject message received. <b>(Bus Free)</b>
01 73	Reselection timeout. <b>(Bus Free)</b>
01 74	Unrecovered SCSI parity error detected by the Target during a MESSAGE OUT phase. <b>(Bus Free)</b>
01 75	Initiator Detected Error message for STATUS or Linked COMMAND COMPLETE phase. <b>(Bus Free)</b>
01 76	Invalid message or attention dropped before all bytes of an extended message are transferred. <b>(Bus Free)</b>
01 77	Attention dropped too late. <b>(Bus Free)</b>
01 78	Message parity error received when no message sent by Target. <b>(Bus Free)</b>
01 79	Reserved bits in Identify message are non zero.
01 7A	Unrecovered SCSI parity error detected by the Initiator (Message Parity Error Message). <b>(Bus Free)</b>
01 7B	SCSI interrupt invalid. <b>(Bus Free)</b>
01 80	SP interrupt on but SP Status Valid bit is off.
01 81	Error in Grown Defect list (used by Format Unit and Reassign Block commands).
01 82	Format Track parameter error (number of sectors and number of ID's do not match).
01 83	Seek positioning error (ID miscompare).
01 84	Invalid SP Command Sequence.
01 85	Illegal Head or Cylinder requested.
01 86	A servo command is already active.
01 87	Interface Processor detected Servo Sanity Error
01 88	Controller/Channel Hardware detected Servo Sanity Error
01 89	Reserved area sector valid check failed.
01 8A	Servo processor did not finish command in time.
01 8B	Motor timeout error.
01 8C	Configuration Sector valid check failed.
01 8D	Configuration Sector uploaded but Check Sum error.
01 8E	Reserved area sector version check failed.
01 8F	Buffer too small to do a requested function.
01 90	Self Init Reset, W/O Auto Motor Start - Invalid Input
01 90	Self Init Reset, W Auto Motor Start - Invalid Input
01 91	Track characterization failure. Unable to determine sector LBA due to adjacent read ID failures, with one sector defective. Reassign(pushdown not started) or Log Sense.
01 92	Miscompare during byte by byte verify.
01 93	BATS#2 Error. Read write test failure.
01 94	BATS#2 Error. ECC/CRC test failure.
01 95	BATS#2 Error; Seek test failure.
01 96	BATS#2 Error; Head Offset Test failure.

Table 127 (Page 7 of 8). Unit Error Codes with Descriptions.	
Unit Error Code	Description
01 97	Self Init Reset, W/O Auto Motor Start - No task available
01 97	Self Init Reset, W Auto Motor Start - No task available.
01 98	Self Init Reset, W/O Auto Motor Start - Cause Unknown
01 98	Self Init Reset, W Auto Motor Start - Cause Unknown
01 99	Self Init Reset, W/O Auto Motor Start - SCSI Controller Chip Reset unsuccessful
01 99	Self Init Reset, W Auto Motor Start - SCSI Controller Chip Reset unsuccessful
01 9A	Self Init Reset, W/O Auto Motor Start - Buffer Controller Chip Reset unsuccessful
01 9A	Self Init Reset, W Auto Motor Start - Buffer Controller Chip Reset unsuccessful
01 9B	Self Init Reset, W/O Auto Motor Start - Zero Divide Error
01 9B	Self Init Reset, W Auto Motor Start - Zero Divide Error
01 9C	Self Init Reset, W/O Auto Motor Start - Control Store Address Fault
01 9C	Self Init Reset, W Auto Motor Start - Control Store Address Fault.
01 9D	Self Init Reset, W/O Auto Motor Start - Unused Op Code
01 9D	Self Init Reset, W Auto Motor Start - Unused OP Code
01 9E	Motor Thermal Shutdown
01 9F	Self Init Reset, W/O Auto Motor Start - Invalid Queue Operation
01 9F	Self Init Reset, W Auto Motor Start - Invalid Queue Operation
01 A1	Servo error; Loss of interrupts from the Controller/Channel Hardware
01 A2	Servo error; Settle timeout.
01 A3	Servo error; Coarse offtrack.
01 A4	Servo error; Three consecutive missing Servo IDs detected by Servo Processor
01 A5	Servo error; Unexpected Guardband detected.
01 A6	Servo error; Settle overshoot.
01 A7	Servo error; Seek timeout.
01 A8	Servo error; Target Cylinder out of Range.
01 A9	Servo error; Command not accepted while in Retract.
01 AB	Servo error; Maximum velocity exceeded.
01 AC	Servo error; Head number out of range.
01 AD	Servo error; Reference track is unreadable.
01 AE	Servo error; Invalid Command.
01 AF	Servo error; Velocity too high at settle hand off.
01 B0	Servo error, Offset out of range.

Table 127 (Page 8 of 8). Unit Error Codes with Descriptions.	
Unit Error Code	Description
01 B1	Servo error, Recalibrate timeout - State 0, No OD guardband. ID guardband seen.
01 B2	Servo error, Recalibrate timeout - State 0, No OD guardband. Data band seen.
01 B3	Servo error, Recalibrate timeout - State 0, No OD guardband. ID and Data band seen.
01 B4	Servo error, Recalibrate timeout - State 0, No OD guardband. ID and Data band not seen.
01 B5	Servo error, Recalibrate timeout - while looking for 6 consecutive outside diameter guardband sectors.
01 B6	Servo error, Recalibrate timeout - while looking for 6 consecutive data bands.
01 B7	Servo error, Recalibrate timeout - while looking for the first data band track type zero.
01 B8	Servo error, Loss of interrupts - motor start bit not active.
01 B9	Servo error, Loss of interrupts - Servo Processor Lost
01 BA	Servo error, Loss of interrupts - Interface Processor initiated
01 BB	Servo error, Loss of interrupts - Servo Processor initiated retract.
01 BC	Servo error; Loss of interrupts - Controller/Channel Hardware reset.
01 BD	Servo error, Loss of interrupts - Bad Coherence.
01 BE	Servo error, Loss of interrupts - Conversion Too Long
01 BF	Servo error; Loss of interrupts - Controller/Channel Hardware Demod Conversion in Progress
01 C0	Three consecutive missing Servo IDs detected by Controller/Channel Hardware
01 C1	Arm Electronics error.
01 C2	Fake and Extra Index.
01 C3	SP lost.
01 C4	Sector overrun error.
01 C5	Interface Processor write inhibit error.
01 C7	Microjog Write Inhibit
01 C8	Interrupt Occured with no interrupt bits set.
01 C9	Write with No Sector Pulses.
01 CB	Motor Speed Error.
01 CC	Channel module Register Write Error
01 CD	IP Retract Error
01 CE	Temporary loss of Motor Synchronization.
01 D0	No sector found error (ID no sync. found).
01 D1	No Data sync. found.
01 D2	Data ECC Check
01 D3	Data correction applied to Drive data for a Data ECC check.

Table 127 (Page 1 of 8). Unit Error Codes with Descriptions.	
Unit Error Code	Description
01 D4	ECC check corrected without using ECC correction.
01 D5	Data Sync error detected while outside of the write band.
01 D6	Data ECC Check detected while outside of the write band.
01 D7	ECC Error Detected while outside of write band corrected with ECC.
01 D8	ECC Error Detected while outside of write band corrected without ECC.
01 D9	Data recovered using positive offsets.
01 DA	Data recovered using negative offsets.
01 DB	Data recovered using No ID Recovery
01 E3	Servo error; Multiple SID Offset Correction error during Tangential Correction.
01 E5	Servo error; Mismatch between external Servo Processor RAM and internal Servo Processor ROS.
01 F0	Buffer Controller Chip Sequencer Error - Check Sum error when loading
01 F1	Buffer Controller Chip Sequencer Error - Not stopped when loading.
01 F2	Buffer Controller Chip Error - Invalid interrupt error.
01 F3	Buffer Controller Chip Error - Invalid read SEQSTOP.
01 F6	Write Gate not detected during Write
01 F8	Buffer Controller Chip Error - Channel parity error on read.
01 F9	Buffer Controller Chip Error - Channel parity error on write.
01 FB	Buffer Controller Chip Error - Channel B was busy before the start of a data transfer.
01 FC	Buffer Controller Chip Error - Channel error during a transfer from the Data buffer to the Control Store RAM (CSR)
01 FD	Buffer Controller Chip Error - Channel error during a transfer from the Control Store RAM to the Data Buffer
02 00	Buffer Controller Chip Error - ECC On The Fly timeout
02 01	Buffer Controller Chip Error - Pipeline already full
02 02	Buffer Controller Chip Error - FIFO overrun/underun
02 03	Disk Manager Chip detected an LRC error during write.
02 10	Channel Module Write Parity Error
02 11	Channel Module Read Parity Error
02 12	Tangential Correction(TC) RAM Parity Error
02 13	Data Manager Write Parity Error
02 20	Channel Noise Problem, Recommend Device Replacement
02 21	Channel Assymetry Problem, Recommend Device Replacement

Table 127 (Page 1 of 8). Unit Error Codes with Descriptions.	
Unit Error Code	Description
02 22	Channel Precompensation Problem, Recommend Device Replacement
02 23	Channel DC Offset Problem, Recommend Device Replacement
02 24	Channel Timing Offset Problem, Recommend Device Replacement
02 25	Fly Height Change Problem, Recommend Device Replacement
02 26	Torque Amplification Problem, Recommend Device Replacement
02 30	BATS#2 Error. LRC test failure.
02 31	BATS#2 Error. Palette RAM test failure.
02 32	BATS#2 Error. Digital Filter RAM test failure.
02 33	BATS#2 Error. Tangential Correction RAM test failure.
4x xx	Thermal Asperity Detected during error.
8x xx	Invalid UEC - x xx is Invalid UEC.

---

# Glossary

**Active I/O process.** An I/O process which is executing (not queued).

**Active LUN condition.** Under certain Target conditions a LUN is active to all Initiators without a command being in progress (that is, without status being outstanding for a command). The Active LUN condition is initiated by either an internal Target action (internally initiated) or a command that was received from an Initiator (externally initiated).

**Cache.** A storage area where data can be retrieved from faster than from the main storage area. The cache contains copies of parts of the main storage.

**Command queue.** The queue in which processes are queued for execution in the target. The **head of the queue** refers to the next queued I/O process to be executed. The **tail of the queue** refers to the process that will be the last process to be executed.

**Current I/O process.** The I/O process which is currently connected to SCSI Bus.

**Dequeuing.** The removal of a queued I/O process from the command queue. The queued I/O process thus removed becomes the active I/O process.

**DRP.** Data Recovery Procedure for:

- No Data Sync Byte
- Data ECC Check
- No Sector Found errors on read commands.

**ERP.** Error Recovery Procedures. ERP errors fall into 4 classes:

- Drive Fault
- Position (Servo)
- Spindle Motor
- Spindle Synchronization Errors.

**Enqueuing.** The placing, by the target, of the current I/O process into the command queue.

**FRU.** Field Replaceable Unit. Refers to any part of the file that can be separately replaced in the field.

**I/O process.** A target process started by an initiator selecting the target. A CDB may or may not have been transferred. If a CDB is transferred, the I/O process persists until the CDB execution is complete or the I/O process has been abnormally terminated. Multiple linked commands are considered to be a single I/O process. If a CDB is not transferred, the I/O process is considered complete at the next BUS FREE phase.

**I\_T\_L nexus.** The combination of *Initiator*, *Target*, and *LUN* that uniquely identifies an *untagged I/O*

*process*. The *I\_T\_L nexus* is established by a connection or reconnection that has an *Identify* message associated with it but no *Queue Tag* message.

**Note:** The term *I\_T\_L nexus* is also used to describe the entire class of *I\_T\_L\_Q nexus* I/O processes which have the same I, T, and L values.

**I\_T\_L\_Q nexus.** The combination of *Initiator*, *Target*, *LUN*, and *Queue Tag* that uniquely identifies a tagged I/O process. The *I\_T\_L\_Q nexus* is established by a connection or reconnection that has both an *Identify* and a *Queue Tag* message associated with it.

**LRC.** Longitudinal Redundancy Code. An error detecting code that detects data buffer address faults and ensures large block coherency.

**PFA.** Predictive Failure Analysis. The process by which a condition is detected that could possibly cause a failure in the future, thereby allowing the replacement of the FRU prior to failure.

**POR.** Power On Reset

**PRDF.** Partial Response Digital Filter detection scheme. This hardware is used to detect the data coming off a magnetic disk.

**Pre-Fetch.**

1. A command where the Initiator informs the Target that it should read data into its cache storage. The Initiator may issue such a command if it knows it will request this data in the near future, but it is not currently ready for it.
2. An operation where the Target automatically reads data into its cache in anticipation of a future request by an Initiator.

**Queue tag.** The value associated with an I/O process that uniquely identifies it from other tagged I/O processes on the same logical unit for the same initiator.

**Queue tag message.** A Simple Queue Tag, Ordered Queue Tag, or Head of Queue Tag message.

**Queued I/O process.** An I/O process that is in the command queue but has not yet begun execution. A queued I/O process can't be the *active* I/O process.

**Read Ahead.** An operation where the Target continues reading beyond what was requested by the Initiator in anticipation that the Initiator will request the next sequential data soon.

**Read Retention.** A process where data just read is saved in a cache storage in anticipation that an Initiator will request that the data be read again soon.

**Tagged I/O process.** An I/O process which has a *Queue Tag* associated with it.

**Thermal Asperity Event.** A media error associated with the head/disk interface used in the target. The target

contains special circuitry to compensate and recover from Thermal Asperity Events.

**Untagged I/O process.** An I/O process which *does not* have a *Queue Tag* associated with it.



# Index

## A

- Abort message 151
- Abort Tag message 153
- Aborted Command sense key 231
- active LUN condition 291
- Actual Retry Count bytes 251
- additional length 33, 34, 36, 40
- additional sense code 231
- additional sense code qualifier 231
- additional sense code qualifier byte 12 and 13 231
- additional sense length 231
- allocation length 33, 34, 36, 40
- ARRE
  - See automatic read reallocation enabled
- ASCII (American Standard Code for Information Interchange) 40, 41
- ASCII data 5
- attention condition 164
- automatic block reassignment 211
- automatic read reallocation enabled (ARRE) 66
- automatic write reallocation enabled (AWRE) 66
- AWRE
  - See automatic write reallocation enabled

## B

- Basic Assurance Tests (BATS)
  - BATS1 tests 226
  - BATS2 tests 226
  - description 226
- BATS
  - See Basic Assurance Tests (BATS)
- block reassignment, automatic 211
- bring-up sequence 225
- buffer ratios, choosing 75
- bus
  - free phase error condition, unexpected 174
  - parity error detection 167
  - related error handling protocol 174
- Bus Device Reset message 153
- busy status 147

## C

- Caching
  - Controls
    - Demand Read Retention Priority 214, 215, 220
    - Disable Pre-Fetch Transfer Length 214, 215
    - DPO: Disable Page Out 214, 219, 220
    - FUA: Force Unit Access 214
    - Maximum Pre-Fetch 214, 215
    - Maximum Pre-Fetch Ceiling 214, 215
    - MF: Multiplication Factor 214, 215
    - Number of Cache Segments 214
    - RCD: Read Cache Disable 214, 215, 218

- Caching (*continued*)
  - Controls (*continued*)
    - Write Retention Priority 214, 219
- check condition status 147
- Clear Queue message 153
- command
  - complete 150
  - descriptor block 20
  - phase retry message 176
  - processing during execution of active I/O
    - process 182
    - processing while busy 182
    - processing while reserved 202
- command aging enable 64
- command queuing 204
- commands
  - Format Unit 21
  - Inquiry 28
  - Mode Select 55
  - Mode Sense 90
  - Pre-Fetch 100
  - priority 203
  - Read 74, 102
  - Read Buffer 104
  - Read Capacity 106
  - Read Defect Data 107, 110
  - Read Extended 103
  - Read Long 113
  - Reassign Blocks 114
  - Receive Diagnostic 116
  - redundant, to an active logical unit 181
  - Release (10) 120
  - Release (6) 119
  - Request Sense 121
  - Reserve (10) 124
  - Reserve (6) 123
  - Rezero Unit 126
  - Seek 127
  - Seek Extended 128
  - Send Diagnostic 129
  - Start/Stop Unit 132
  - summary 19
  - Synchronize Cache 134
  - Test Unit Ready 135
  - Verify 136
  - Write 75, 138
  - Write and Verify 140
  - Write Buffer 142
  - Write Extended 139
  - Write Long 145
  - Write Same 146
- condition
  - attention 164
  - deferred error 189

**condition** (*continued*)  
externally initiated active LUN 189  
internal error 188  
reset 224  
Unit Attention 184  
**condition met status** 147  
**condition, deferred error** 189  
**condition, Internal Error** 188  
**configuration information** 227  
**Continue I/O Process message** 154  
**control** 88

## D

**DASD (Direct Access Storage Device)** 33, 34, 36, 40  
**data**  
ASCII 5  
**data in phase retry** 177  
**data out phase retry** 177  
**data phase save data pointer** 63  
**DCR**  
*See* disable correction  
**defect map, grown** 227  
**deferred error condition** 189  
**Definition of Terms** 291  
**degraded mode.** 190  
**degraded mode**  
description 190  
exiting 200  
limitations of the Drive 192  
reasons for entering 190  
**disable correction (DCR)** 68  
**disable page out (DPO)** 103  
**disable synchronous negotiations** 63  
**disable transfer on error (DTE)** 68  
**Disconnect message** 151  
**disconnect permission** 221  
**disconnect/reconnect parameters** 74  
download and save 40  
download microcode 142  
**DPO**  
*See* disable page out  
**Drive Microcode** 227  
**DTE**  
*See* disable transfer on error

## E

**EBCDIC (Extended Binary-Coded Decimal Interchange Code)** 34, 41  
**ECC**  
*See* error correction code  
**ECMA**  
*See* European Computer Manufactures Association  
**enable early recovery (ERR)** 68  
**enable vital product data (EVPD)** 28, 33, 34, 36, 38, 39, 40  
**end inquiry command** 40

**end inquiry command EVPD = 1** 40  
**end SCSI sense data** 253  
**end SCSI sense data description** 253  
**ERP**

*See* error recovery procedures

## ERR

*See* enable early recovery

## error

condition, deferred 189  
condition, internal 188  
recovery parameters 66  
recovery parameters for verify 83

**error correction code (ECC)** 136

**error logs** 211, 227

**error recovery procedures (ERP)** 102

**error reporting, priority of** 265

**European Computer Manufacturers Association (ECMA)** 31

## EVPD

*See* enable vital product data

**exiting degraded mode** 200

**extended message table** 158, 161

**Extended messages** 157

## F

**flag and link bits** 20

**flags** 227

**force unit access (FUA)** 103

**format and reassign degraded disable** 63

**format in progress** 252

**Format Unit command** 21

## FUA

*See* force unit access

## G

**GList** 21, 227

**good status** 147

**grown defect map** 227

## H

**Hardware Error sense key** 230

**Head of Queue Tag message** 155

**hex values** 5

## I

**Identify message** 156

**identify message in phase retry** 178

**Illegal Request sense key** 231

**Immed bit** 24

**immed Not Ready,** 252

**incorrect initiator connection** 181

**Initiator Data Transfer Mode parameters** 222

**Initiator Detected Error message** 151

**Initiator Mode Select/Mode Sense parameters** 222

- Initiator selection, single 221
- Initiator sense data 222
- Initiator systems, multiple 222
- Inquiry command 28
  - inquiry command EVPD = 1 page code = 01 34
  - inquiry command EVPD = 1 page code = 02 36
  - inquiry command EVPD = 1 page code = 03 38
  - inquiry command EVPD = 1 page code = 80 39
  - inquiry command EVPD = 1 page code = 82 40
- inquiry data format 30
  - inquiry data format - EVPD = 1 page code = 01 34
  - inquiry data format download and save 40
  - inquiry data format EVPD = 1 page code = 82 40
  - inquiry data format EVPD = 1 34, 36, 40, 41
  - inquiry data format EVPD = 1 page code = 02 36
  - inquiry data format EVPD = 1 Page Code = 03 38
  - inquiry data format EVPD = 1 Page Code = 80 39
- Interleave factor 21
- intermediate/condition met 148
- intermediate/good status 148
- internal error condition 188
- invalid LUN processing 180

## L

- LBA**
  - See* logical block address
- LED pin 60, 223
- Limit Idle Time Function 64
- link and flag bits 20
- Linked Command Complete message 152
- Linked Command Complete with Flag message 153
- logical block address (LBA) 106, 231
  - logical unit number (LUN) 34
- LUN
  - See* logical unit number
- LUN (logical unit number) 28, 121, 180
- LUN processing, invalid 180

## M

- manufacturing SAT map (PList) 227
- Medium Error sense key 230
- message
  - action legend 170
  - functions, supported 163
  - in phase retry 176
  - out phase retry 175
  - phase retry 176
  - state/action table 168
  - system 149
- Message Parity Error message 152
- Message Reject during Target disconnection 164
- Message Reject message 152
- messages
  - Abort 151
  - Abort Tag 153
  - Bus Device Reset 153
  - Clear Queue 153

- messages (*continued*)
  - Command Complete 150
  - Continue I/O Process 154
  - Disconnect 151
  - Extended messages 157
  - Head of Queue Tag 155
  - Identify 156
  - Initiator Detected Error 151
  - Linked Command Complete 152
  - Linked Command Complete with Flag 153
  - Message Parity Error 152
  - Message Reject 152
    - multiple-byte extended 150
  - No Operation 152
  - Ordered Queue Tag 156
  - Queue Tag 155
    - Restore Pointers 150
    - Save Data Pointer 150
    - Simple Queue Tag 155
    - supported 163
    - supported, single byte 149
    - Synchronous Data Transfer Request 157
    - table of 149
    - Target Transfer Disable 154
    - two-byte extended 149
    - Wide Data Transfer Request 161

- microcode unique function 227

- microcode, download 142

- Mode Select (6) 55

- Mode Select command 55

- Mode Sense (6) 90

- Mode Sense command 90

- Mode Sense data format 92

- Motor Synchronization

- achieving synchronization 208

- Format considerations 209

- overview 208

- synchronization parameters 80

- multiple Initiator systems 222

## N

- No Operation message 152

- No Sense sense key 230

- non-arbitrating systems 221

- Not Ready sense key 230

## O

- option pin 189, 223

- Ordered Queue Tag message 156

## P

- parameters

- control 88

- disconnect/reconnect 74

- error recovery 66

- error recovery, for verify 83

**parameters** (*continued*)

- Initiator data transfer mode 222
- Initiator mode select/mode sense 222
- medium format 77
- read caching 85
- spindle synchronization 80

**partial medium indicator (PMI)** 106

**PC meaning** 90

**PE**

- See* post error

**peripheral device type** 40

**phase retry**

- command 176
- data in 177
- data out 177
- identify message in 178
- message in 176
- message out 175
- simple queue tag message in 178
- status 177

**physical block address** 252

**PList** 21, 227

**PMI**

- See* partial medium indicator

**post error (PER)** 68

**power-on reset** 224

**power-on self test** 227

**power-up sequence** 225

**Pre-Fetch command** 100

**predictive failure analysis errors, reporting of (RPFAE)** 61

**priority commands** 203

**priority of error reporting** 265

**progress indication bytes** 252

**protocol, SCSI** 179

**Q**

- qualify post error 59
- queue full status 148
- queue tag message table 155
- Queue Tag messages 155
- queuing, command 204

**R**

- RAM code revision level 32
- ratios, buffer 75

**RC**

- See* read continuous

**RCD**

- See* read cache disable

**read (10)** 103

**read (6)** 102

**Read Buffer command** 104

**read cache disable (RCD)** 85

**Read Caching parameters** 85

**Read Capacity command** 106

**Read command** 74, 102

**read continuous (RC)** 67

**Read Defect Data command** 107, 110

**Read Extended command** 103

**Read Long command** 113

**Read-Ahead Function** 214

**Reassign Blocks command** 114

**reassignment, automatic block** 211

**Receive Diagnostic command** 116

**reconnecting to Read command** 74

**reconnecting to Write command** 75

**Recovered Error sense key** 230

**Release (6) command** 119, 120

**removable media bit (RMB)** 31

**Request Sense command** 121

**reselection timeout** 221

**reservation conflict status** 148

**Reserve (10) command** 124

**Reserve (6) command** 123

**reserved area** 227

**reset actions** 224

**reset condition** 224

**reset sources** 224

**Restore Pointers message** 150

**retry** 251

**Rezero Unit command** 126

**RMB**

- See* removal media bit

**ROM code revision level** 32

**S**

- SAT map, manufacturing 227
- Save Data Pointer message 60, 150
- scratch areas for reassignment 227

**SCSI bus**

- parity error detection 167
- related error handling protocol 174

**SCSI bus device reset message** 224

**SCSI bus reset** 224

**SCSI commands** 19

**SCSI protocol** 179

**SCSI sense data description** 230

**SCSI sense data description Actual Retry Count bytes** 251

**SCSI sense data description byte 1** 230

**SCSI sense data description byte 12** 232

**SCSI sense data description byte 14** 251

**SCSI sense data description byte 15 - 16** 251

**SCSI sense data description byte 18 - 19** 252

**SCSI sense data description byte 2** 230

**SCSI sense data description byte 24 - 27** 253

**SCSI sense data description byte 28 - 31** 253

**SCSI sense data description byte 3 - 6** 231

**SCSI sense data description byte 7** 232

**SCSI sense data description byte 8 - 11** 232

**SCSI sense data description progress indication bytes** 252

- SCSI sense data format 229
- SCSI status byte reporting, priority of 179
- seek (10) 128
- Seek (6) 127
- Seek command 127
- Seek Extended command 128
- selection without ATN 221
- self-initiated reset 224
- Send Diagnostic command 129
- sense
  - sense code
    - See* additional sense code
  - sense code qualifier
    - See* additional sense code qualifier
  - sense data description 231
  - sense data format 229
  - sense key specific bytes 251
  - sense key specific bytes (SKSV) 251
  - sense keys 230, 231
  - sense keys *Aborted Command* 231
  - sense keys additional sense code 232
  - sense keys additional sense length 232
  - sense keys *Hardware Error* 230, 231
  - sense keys *Illegal Request* 231
  - sense keys *Medium Error* 230, 231
  - sense keys *Miscompare* 231
  - sense keys *No Sense* 230
  - sense keys not implemented 231
  - sense keys *Not Ready* 230
  - sense keys *Recovered Error* 230
  - sense keys *Unit Attention* 231
- Simple Queue Tag message 155
- simple queue tag message in phase retry 178
- single Initiator selection 221
- spin-up 225
  - start commands log sense 42, 44
  - start log sense 42, 44
- start SCSI sense data 229
- start SCSI sense data description 230
- start-up sequence 225
- Start/Stop Unit command 132
- Status Byte 147
- status byte reporting, priority of SCSI 179
- status phase retry 177
- statuses
  - busy 147
  - check condition 147
  - condition met 147
  - good 147
  - intermediate/condition met 148
  - intermediate/good 148
  - queue full 148
  - reservation conflict 148
- summary of SCSI commands 19
- supported message functions 163
- supported messages, single byte 149
- Synchronize Cache command 134
- synchronous
  - data transfer request message definition 158

- synchronous (*continued*)
  - negotiation started by the Initiator 158
  - negotiation started by the target 159
- Synchronous Data Transfer Request message 157
- systems, non-arbitrating 221

## T

- Target Transfer Disable message 154
- TB
  - See* transfer block
- Test Unit Ready command 135
- thermal compensation control 62
- thermal compensation head control 62
- timeout, reselection 221
- transfer block (TB) 67

## U

- Unit Attention condition 184
- Unit Attention* sense key 231
- Unit Error Code (UEC) 252

## V

- vendor unique 40
- Verify command 136
- vital product data (VPD) 33, 34, 36, 40

## W

- Wide Data Transfer Request message 161
- write (10) 139
- write (6) 138
- Write and Verify command 140
- Write Buffer command 142
- Write command 75, 138
- Write Extended command 139
- Write Long command 145
- write protect enable 63
- Write Same command 146



<b>Table Definitions</b>
--------------------------

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
THEAD	SPECTBLS	1	
THEAD2	SPECTBLS	1	
THEAD3	SPECTBLS	1	
THEADX	SPECTBLS	1	21, 22, 23, 25, 26, 28, 30, 33, 34, 36, 38, 39, 40, 42, 44, 46, 47, 50, 51, 52, 52, 54, 55, 57, 58, 58, 59, 66, 74, 77, 80, 83, 85, 88, 90, 92, 92, 94, 94, 95, 96, 97, 97, 98, 98, 100, 102, 103, 104, 106, 106, 107, 108, 108, 109, 109, 110, 111, 111, 112, 112, 113, 114, 115, 115, 116, 117, 119, 120, 121, 123, 124, 126, 127, 128, 129, 131, 132, 134, 135, 136, 138, 139, 140, 142, 145, 146, 147, 156, 229, 251, 251, 252
TBLX8	SPECTBLS	1	1, 1, 1, 1
TBLX8X	SPECTBLS	1	23, 30
TBLNO	SPECTBLS	1	1
TBLR2	SPECTBLS	1	
TBLR2R	SPECTBLS	1	
TBLR2RR	SPECTBLS	1	
PADFOOT	SPECTBLS	1	30
TBLHD	SPECTBLS	1	
TBLXX	SPECTBLS	1	21, 21, 21, 23, 23, 23, 23, 26, 26, 26, 26, 26, 26, 28, 28, 30, 30, 33, 34, 36, 38, 39, 40, 42, 42, 44, 44, 44, 46, 47, 47, 47, 50, 50, 50, 51, 51, 52, 52, 52, 52, 52, 54, 54, 54, 55, 55, 57, 57, 58, 58, 59, 59, 59, 59, 59, 59, 66, 66, 74, 74, 74, 77, 77, 80, 80, 80, 83, 83, 85, 85, 85, 88, 88, 90, 90, 92, 92, 92, 94, 94, 94, 94, 94, 94, 94, 94, 94, 95, 95, 95, 96, 96, 97, 97, 97, 97, 97, 98, 98, 98, 98, 98, 100, 100, 102, 102, 103, 103, 104, 104, 106, 106, 106, 107, 107, 108, 108, 108, 108, 109, 109, 109, 109, 109, 109, 110, 110, 110, 110, 110, 111, 111, 111, 111, 111, 111, 112, 112, 112, 112, 112, 112, 113, 113, 114, 114, 115, 115, 116, 116, 117, 117, 117, 117, 117, 117, 119, 119, 120, 120, 121, 121, 123, 123, 124, 124, 126, 126, 127, 127, 128, 128, 129, 129, 131, 131, 131, 131, 132, 132, 134, 134, 135, 135, 136, 136, 138, 138, 139, 139, 140, 140, 142, 142, 145, 145, 146, 146, 156, 156, 229, 229, 229, 251, 251, 252
TBL1CH	SPECTBLS	1	117
TBL1DH	SPECTBLS	1	117, 131, 131
TBLXV	SPECTBLS	1	
TBLXXH	SPECTBLS	1	
TBLYY	SPECTBLS	1	
TBL1	SPECTBLS	1	
TBL2	SPECTBLS	1	30, 33, 34, 36, 38, 39, 40, 106, 107, 114, 116, 121, 126, 128, 135
TBL2X	SPECTBLS	1	102, 127, 138
TBL3	SPECTBLS	1	21, 28, 42, 44, 55, 90, 100, 102, 103, 104, 106, 107, 110, 113, 114, 116, 119, 120, 121, 123, 124, 126, 127, 128, 129, 132, 134, 135, 136, 138, 139, 140, 142, 145, 146
TBL4	SPECTBLS	1	21, 107, 108, 110, 111
TBL5	SPECTBLS	1	88, 98, 106
TBL6	SPECTBLS	1	
TBL7	SPECTBLS	1	28, 132, 145
TBL7X	SPECTBLS	1	147
TBL8	SPECTBLS	1	

TBL9	SPECTBLS	1	55, 119, 120, 123, 124
TBL9X	SPECTBLS	1	42, 44, 46, 47, 50, 52, 54, 90
TBL10	SPECTBLS	1	30
TBL11	SPECTBLS	1	30, 229, 229, 251, 251
TBL12	SPECTBLS	1	59, 66, 94, 94
TBL13	SPECTBLS	1	83, 97
TBL14	SPECTBLS	1	58, 59, 66, 74, 77, 80, 83, 85, 88, 94, 94, 95, 96, 97, 97, 98, 98
TBL15	SPECTBLS	1	85, 98
TBL16	SPECTBLS	1	
TBL17	SPECTBLS	1	51, 52, 132
TBL18	SPECTBLS	1	30
T1185	SPECTBLS	1	
TBL19	SPECTBLS	1	90
TBL20	SPECTBLS	1	57, 92
TBL21	SPECTBLS	1	129
TBL22	SPECTBLS	1	103, 139
TBL23	SPECTBLS	1	136, 140
TBL24	SPECTBLS	1	104, 142
TBL25	SPECTBLS	1	146
TBL26	SPECTBLS	1	
TBL27	SPECTBLS	1	85, 98
TBL28	SPECTBLS	1	
PADHD	SPECTBLS	1	
TBLF0H	SPECTBLS	1	59
TBLF0A	SPECTBLS	1	59, 77, 94, 96
TBLF0AH	SPECTBLS	1	
TBL03H	SPECTBLS	1	
TBL0FH	SPECTBLS	1	
TBL3FH	SPECTBLS	1	
TBL93H	SPECTBLS	1	42, 44, 100, 113, 134
TBLFBH	SPECTBLS	1	
TBLFFH	SPECTBLS	1	
TBLB8H	SPECTBLS	1	94, 229
TBLECH	SPECTBLS	1	252
TBL03	SPECTBLS	1	74, 80, 95, 97
TBL29	SPECTBLS	1	88, 98
TBLE0H	SPECTBLS	1	85, 94, 98
TBLF1H	SPECTBLS	1	59, 94
TBLF8H	SPECTBLS	1	
TBLFCH	SPECTBLS	1	
TBL87H	SPECTBLS	1	88, 98
TBLC3H	SPECTBLS	1	







MSGBR40	CHAPTER3	172	173
MSGBR1E	CHAPTER3	172	173
MSGBR5F	CHAPTER3	172	173
TBLSHD	APPENDX3	269	269, 270, 271
TBLSBD	APPENDX3	269	269, 270, 270, 270, 270, 270, 270, 270, 270, 270, 271

Figures
---------

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
ATTN1	CHAPTER3	165	1 164
ATTN2	CHAPTER3	166	2 164

Headings
----------

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
WARN	SCSIATM2	5	Data Representation
SCSICDS	CHAPTER1	19	1.0, SCSI Commands 2
WARN1	CHAPTER1	20	1.1.1, CDB Control Byte
RSVDEF	CHAPTER1	20	1.1.2, Reserved field definitions
RSVDVAL	CHAPTER1	20	1.1.3, Reserved value definitions
INVFLD	CHAPTER1	20	1.1.4, Invalid field Errors
FMTUNT	FORMAT	21	1.2, Format Unit 77
INQUIRY	INQRYCMD	28	1.3, Inquiry 180, 276, 279, 279
INQSTAN	INQRYCMD	30	1.3.1, Inquiry Data Format - EVPD = 0 33, 34, 36, 38, 39, 40, 40, 41
CMD4C	LOGSECMD	42	1.4, Log Select
CMD4D	LOGSNCMD	44	1.5, Log Sense 3, 271
LSP35H	LOGSNCMD	54	1.5.6, Log Sense Page 35h 214
MSELECT	MDSELCMD	55	1.6, Mode Select (6) 91, 99, 211, 269
MODHDR	MDSELCMD	57	1.6.1.1, Header 92
MODBDR	MDSELCMD	58	1.6.1.2, Block Descriptor 22
MODPDR	MDSELCMD	58	1.6.1.3, Page Descriptor
PAGE0H	MDSELCMD	59	1.6.2, Page 0 - Vendor Unique Parameters 2, 3, 3, 68, 89, 126, 150, 161, 163, 195, 195, 207, 213, 223, 223, 248
PAGE1H	MDSELCMD	66	1.6.3, Page 1 - Error Recovery Parameters 61, 102, 138
PAGE2H	MDSELCMD	74	1.6.4, Page 2 - Disconnect/Reconnect Parameters 2
PAGE3H	MDSELCMD	77	1.6.5, Page 3 - Format Device Parameters 22, 58

PAGE4H	MDSELCMD	80	1.6.6, Page 4h - Rigid Disk Drive Geometry Parameters 22, 208
PAGE7H	MDSELCMD	83	1.6.7, Page 7h - Error Recovery Parameters for Verify 59, 136, 140
PAGE8H	MDSELCMD	85	1.6.8, Page 8h - Read Caching Parameters 2, 103, 139, 141, 208, 214, 214
PAGEAH	MDSELCMD	88	1.6.9, Page 0Ah - Control Mode Parameters 64, 64, 205
MSENSE	MDSNSCMD	90	1.7, Mode Sense (6) 55, 55, 56, 57
PREFET	PRFCHCMD	100	1.8, Pre-Fetch 147
READ10	READCMDS	103	1.10, Read (10) 214
READDEF	RDDEFCMD	107	1.13, Read Defect Data (10)
REDDEF2	RDDEFCMD	110	1.14, Read Defect Data (12)
REASGN	REASSCMD	114	1.16, Reassign Blocks 2
RELEASE	RELCMDS	119	1.18, Release (6) 148
REL10	RELCMDS	120	1.19, Release (10)
REQSNS	REQSNCMD	121	1.20, Request Sense 180
RESRV	RESCMDS	123	1.21, Reserve (6) 119, 120, 148
RESRV10	RESCMDS	124	1.22, Reserve (10)
RZERO	RZEROCMD	126	1.23, Rezero Unit
SEEK10	SEEKCMDS	128	1.25, Seek (10)
STRSTP	SUNITCMD	132	1.27, Start/Stop Unit 81, 208
TURDY	TURDYCMD	135	1.29, Test Unit Ready 225
VERIFY	VERIFCMD	136	1.30, Verify 84
WRITEX	WRITCMDS	139	1.32, Write (10) 214
WRITEV	WRTVYCMD	140	1.33, Write and Verify 84
WRTBUFR	WRBUFCMD	142	1.34, Write Buffer 2, 38, 274
STATBYT	CHAPTER2	147	2.0, SCSI Status Byte
SPTMSG	CHAPTER3	149	3.1, Supported Messages 3, 174
DISCMSG	CHAPTER3	151	3.1.4, Disconnect (04h) 150
CIOPMSG	CHAPTER3	154	3.1.15, Continue I/O Process (12h) 154
TTDMSG	CHAPTER3	154	3.1.16, Target Transfer Disable (13h) 60, 63, 154
QTAGMSG	CHAPTER3	155	3.1.17, Queue Tag Messages (20h, 21h, 22h) 205
C3IDMSG	CHAPTER3	156	3.1.19, Identify (80h or C0h) 151
SYNCI	CHAPTER3	158	3.1.21.2, Synchronous Negotiation Started by the Initiator

			161	
SYNCT	CHAPTER3	159	3.1.21.3, Synchronous Negotiation Started by the Target	161
WIDEI	CHAPTER3	162	3.1.22.2, Transfer Width Negotiation Started by the Initiator	163
WIDET	CHAPTER3	162	3.1.22.3, Transfer Width Negotiation Started by the Target	163
MSGFUNC	CHAPTER3	163	3.2, Supported Message Functions	151, 152, 152, 157
ATNCOND	CHAPTER3	164	3.3, Attention Condition	151, 152, 152, 159, 162, 167, 175
MSGTBL	CHAPTER3	168	3.5, Message State/Action Table	152, 152, 156, 161, 163
BUSERRS	CHAPTER3	174	3.6, SCSI Bus Related Error Handling Protocol	150, 151, 152, 152
BUSFREE	CHAPTER3	174	3.6.1, Unexpected Bus Free Phase Error Condition	154
MORTY	CHAPTER3	175	3.6.2, Message Out Phase Retry	161, 163, 167, 168
MIRTY	CHAPTER3	176	3.6.3, Message In Phase Retry	159, 160, 162, 163, 167
CORTY	CHAPTER3	176	3.6.4, Command Phase Retry	167
SIRTY	CHAPTER3	177	3.6.5, Status Phase Retry	167
DORTY	CHAPTER3	177	3.6.6, Data Out Phase Retry	
DIRTY	CHAPTER3	177	3.6.7, Data In Phase Retry	167
PROTOCO	SCSIPROT	179	4.1, SCSI Protocol	206
STPRIOR	SCSIPROT	179	4.1.1, Priority of SCSI Status Byte Reporting	132, 182, 182, 183, 185, 202
BADLUN	SCSIPROT	180	4.1.2, Invalid LUN Processing	156, 179
TWOCMDS	SCSIPROT	181	4.1.3, Incorrect Initiator Connection	155, 174, 179, 182, 203, 204, 207
EXBUSY	SCSIPROT	182	4.1.4, Command Processing During Execution of Active I/O Process	148, 179, 206
UNITATN	SCSIPROT	184	4.1.5, Unit Attention Condition	22, 56, 56, 121, 144, 153, 179, 209, 222
CPDSAF	SCSIPROT	187	4.1.6, Command Processing During Start-up and Format Operations	22, 132, 179, 182
INTERR	SCSIPROT	188	4.1.7, Internal Error Condition	179
BRUPERR	SCSIPROT	189	4.1.8, Deferred Error Condition	24, 132, 179
DEGRAD	DEGRADE4	190	4.1.9, Degraded Mode	22, 114, 129, 179, 274
DEGRENT	DEGRADE4	190	4.1.9.1, Reasons for Entering Degraded Mode	
DEGLRIM	DEGRADE4	192	4.1.9.2, Limitations of the Drive While in Degraded Mode	
RESTOR	DEGRADE4	200	4.1.9.4, Exiting Degraded Mode	225
EXRSVD	SCSIPROT	202	4.1.10, Command Processing While Reserved	119, 120, 123, 124, 148, 179
PRICMDS	SCSIPROT			

		203	4.2, Priority Commands 62, 182, 206, 207, 216, 217
QSTUFF	COMQUEUE		
		204	4.3, Command Queuing 2, 2, 22, 28, 122
UNTAGQ	COMQUEUE		
		204	4.3.1, Untagged Queuing 60, 206
TAGQ	COMQUEUE		
		205	4.3.2, Tagged Queuing 88, 148, 204
QRULES	COMQUEUE		
ERRONQ	COMQUEUE		
		206	4.3.4, Rules for Queuing commands
		206	4.3.5, Effects of errors on Command Queuing 205, 207, 208
QTERM	COMQUEUE		
		207	4.3.6, Termination of I/O processes 204
SAMETIM	BAK2BAK		
		207	4.4, Concurrent Command Processing 182
BAK2BAK	BAK2BAK		
		208	4.4.1, Back to Back Writes 2, 62
SYNCDES	MTRSYNC		
		208	4.5, Motor Synchronization 2, 22, 78, 81, 133, 184, 185, 225, 265, 278
ERRLOGS	ERRLOGS		
		211	4.6, Error Logs 88
ATRCRES	ERRLOGS		
		211	4.7, Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign 2, 2, 21, 51, 52, 66, 67
PREFA	PFA		
		213	4.8, Predictive Failure Analysis 2, 62
SEGCAC	SEGCACH		
		214	4.9, Segmented Caching 2, 54, 85, 86, 87, 93, 103, 139, 141, 144
RLBA	SEGCACH		
		214	4.9.3, Read-Ahead 2, 100, 103
SEGSEL	SEGCACH		
		218	4.9.4, Segment Selection 216, 216, 217, 217, 218, 218
REPRI	SEGCACH		
SINIT	MISCHAP4		
		219	4.9.5, Segment Prioritization
MULTINT	MISCHAP4		
		221	4.11, Single Initiator Selection
		222	4.14, Multiple Initiator Systems 3
OPTPIN	MISCHAP4		
		223	4.15, Option Pin 63, 189
LEDPIN	MISCHAP4		
		223	4.16, LED Pin 3
RESET	MISCHAP4		
		224	4.17, Reset
RSTSRC	MISCHAP4		
		224	4.17.1, Reset Sources
RSTACTN	MISCHAP4		
		224	4.17.2, Reset Actions
BRINGUP	MISCHAP4		
		225	4.18, Bring-Up Sequence 114, 132, 200, 224
BATS	MISCHAP4		
		226	4.19, Basic Assurance Tests (BATS) 3
BTS1	MISCHAP4		
		226	4.19.1, BATS1 Tests 225
BTS2	MISCHAP4		
		226	4.19.2, BATS2 Tests 225
RSRVD	MISCHAP4		
		227	4.20, Reserved Area 225
SENSDAT	APPENDX1		
		229	Appendix A, SCSI Sense Data Format 3, 56, 121, 132, 283
APNDX2	APPENDX2		

DRPDESC	APPENDX2	253	Appendix B, Recovery Procedures
		253	B.1, Data Recovery Procedure for Data Field Errors 59, 73, 102, 138
RRECDES	APPENDX2	254	B.1.1, Read Commands
VRPDESC	APPENDX2	260	B.1.2, Verify Commands 84, 84, 136, 140
WRPDESC	APPENDX2	262	B.1.3, Write Commands (No Sector Found)
SRPDESC	APPENDX2	263	B.2.1, Position Error (Servo Error Recovery)
PRPDESC	APPENDX2	263	B.2.2, Drive Fault
ERRPTY	APPENDX2	263	B.3, Priority of Error Reporting 70, 70, 71, 71, 71, 72, 72, 72, 135
SCTRREL	APPENDX3	265	Appendix C, Recommended Initiator Error Recovery Procedures 200
SERVICE	APPENDX3	265	C.1, Drive service strategy 272, 272, 273, 273, 273, 274, 274, 274, 275, 275, 275, 275, 275, 276, 276, 277, 278, 279, 279, 279, 280, 280, 280, 281, 281, 281, 281, 281, 282, 282
ERRLOG	APPENDX3	266	C.2, Recommendations for System Error Log
DRP11	APPENDX3	266	C.3, Data Recovery Procedure
DRP1	APPENDX3	267	C.3.1, Rewrite an LBA 234, 235, 236
DRP3	APPENDX3	268	C.3.2, Reassign a physical sector 233, 233, 234, 235, 235, 235, 239, 239, 240, 240
DRP2	APPENDX3	269	C.3.3, Data error logging 233, 233, 233, 234, 234, 234, 234, 234, 235, 235, 235, 235, 235, 236, 239, 240, 240
ASSIGN	APPENDX3	270	C.3.4, Reassign Blocks Recovery 239, 239, 239, 240, 242, 271, 275
NONDATA	APPENDX3	270	C.4, Non-Data Error Recovery Procedure
NOERR	APPENDX3	270	C.4.1, Drive busy 238, 238
OPTLOW	?	?	?
ERP1	APPENDX3	271	270
ERP2	APPENDX3	271	C.4.2, Unrecovered drive error 241, 241, 241, 241, 241, 242, 242, 245, 246
ERP3	APPENDX3	271	C.4.3, Recovered drive error 232, 233, 233, 233, 234, 234, 237, 273, 273
ERP7	APPENDX3	271	C.4.4, Drive not ready 238
CRPTFMT	APPENDX3	272	C.4.5, No defect spare 242
DRP	APPENDX3	273	C.4.6, Degraded Mode 238, 238, 239, 239, 239, 240, 240, 243, 243, 243
ERP9	APPENDX3	273	C.4.7, Reserved Area Hard Error 240, 240, 241, 241, 242, 242, 242, 242
ERP10	APPENDX3	274	C.4.8, Interface Protocol 246, 246, 246, 247, 247, 247
ERP5	APPENDX3	275	C.4.9, Aborted Command 247, 248, 248, 249, 249, 249, 249, 249, 249
UANRTR	APPENDX3	275	C.4.10, Unit Attention
UAPOR	APPENDX3	275	C.4.10.1, Not Ready to Ready Transition 247
UAMPC	APPENDX3	275	C.4.10.2, Reset 247

		275	C.4.10.3, Mode Parameters Changed 247
UAMCCH	APPENDX3	275	C.4.10.4, Microcode has Changed 248
UACCAI	APPENDX3	275	C.4.10.5, Commands Cleared by Another Initiator 248
UASS	APPENDX3	275	C.4.10.6, Spindles Synchronized 248
UASNS	APPENDX3	276	C.4.10.7, Spindles Not Synchronized 237, 246, 248
UALPC	APPENDX3	276	C.4.10.8, Log Select Parameters Changed
ERP13	APPENDX3	276	C.4.11, Components Mismatch 243, 244
ERP14	APPENDX3	278	C.4.12, Self Initiated Reset 243
DEFLIST	APPENDX3	278	C.4.13, Defect List Recovery 236, 236, 236
MISCOMP	APPENDX3	278	C.4.14, Miscompare recovery 250
CHKSUM	APPENDX3	279	C.4.15, Microcode error 239, 243
PREDICT	APPENDX3	280	C.4.16, Predictive failure analysis 238
UECLIST	APPENDX4	281	Appendix D, UEC List

Index Entries
---------------

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
CMDS	SCSIINX2	1	(1) commands 19, 21, 28, 55, 74, 75, 90, 100, 102, 103, 104, 106, 107, 110, 113, 114, 116, 119, 120, 121, 123, 124, 126, 127, 128, 129, 132, 134, 135, 136, 138, 139, 140, 142, 145, 146, 181, 203
PARM	SCSIINX2	1	(1) parameters 66, 74, 77, 80, 83, 85, 88, 222, 222
MSGS	SCSIINX2	1	(1) messages 149, 149, 149, 150, 150, 150, 150, 151, 151, 151, 152, 152, 152, 152, 153, 153, 153, 153, 154, 154, 155, 155, 155, 156, 156, 157, 157, 161, 163
MSG	SCSIINX2	1	(1) message 149, 163, 168, 170, 175, 176, 176
CMD	SCSIINX2	1	(1) command 20, 150, 176, 182, 182, 202
SYNC	SCSIINX2	1	(1) synchronous 158, 158, 159
SCSIBUS	SCSIINX2	1	(1) SCSI bus 167, 174
BUS	SCSIINX2	1	(1) bus 167, 174, 174
PHASE	SCSIINX2	1	(1) phase retry 175, 176, 176, 177, 177, 177, 178, 178
COND	SCSIINX2	1	(1) condition 164, 184, 188, 189, 189, 224
DEGR	SCSIINX2	1	(1) degraded mode 190, 190, 192, 200
BATS	SCSIINX2	1	(1) Basic Assurance Tests (BATS) 1, 226, 226, 226
SENSE	SCSIINX2		







R10S1	SETUP	1		
R11S1	SETUP	1		92, 92, 247, 247, 278, 278, 285, 285
R12S1	SETUP	1		2, 2, 19, 19, 23, 23, 59, 59, 60, 60, 60, 61, 64, 64, 64, 65, 67, 67, 86, 86, 86, 87, 94, 94, 116, 118, 129, 129, 129, 129, 129, 131, 156, 156, 171, 171, 171, 171, 171, 171, 172, 172, 173, 173, 182, 182, 183, 183, 183, 183, 187, 187, 190, 190, 204, 204, 204, 204, 204, 204, 204, 204, 205, 205, 214, 214, 215, 215, 215, 215, 247, 247, 286, 286
R13S1	SETUP	1		4, 4, 30, 30, 31, 31, 41, 41, 59, 59, 116, 116, 116, 116, 116, 116, 119, 119, 120, 120, 123, 123, 124, 124, 129, 129, 129, 129, 129, 130, 133, 133, 134, 134, 145, 145, 145, 145, 146, 146
R3S2	SETUP	1		
I3	?	?	?	59, 59, 170, 170
I1	?	?	?	39
R2	?	?	?	61
I6	?	?	?	85, 85
R1S11	?	?	?	172, 172, 172, 172, 172, 172
R1S2	?	?	?	205, 205
				207, 207

Tables			
--------	--	--	--

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
DLHD	FORMAT	23	3 21, 23
DLDSC	FORMAT	23	4 21
DFDSTBL	FORMAT	25	5 21, 25
PHYFRMT	FORMAT	26	7 118, 131
MSEL	MDSELCMD	57	26 55
PAGE0	MDSELCMD	59	29 59
PAGE1	MDSELCMD	66	31 66, 66
PAGE2	MDSELCMD	74	32 74, 74
PAGE3	MDSELCMD	77	33
PAGE4	MDSELCMD	80	34
SPIN	MDSELCMD	81	35 81
PAGE7	MDSELCMD	83	36 83
PAGE8	MDSELCMD	85	37 85
PAGEA	MDSELCMD		

		88	38	88, 88
MDSNP0	MDSNSCMD			
MDSNP1	MDSNSCMD	94	42	
MDSNP2	MDSNSCMD	94	43	
MDSNP3	MDSNSCMD	95	44	
MDSNP4	MDSNSCMD	96	45	
MDSNP7	MDSNSCMD	97	46	
MDSNP8	MDSNSCMD	97	47	
MDSNPA	MDSNSCMD	98	48	
STATUS	CHAPTER2	98	49	
		147	92	
				147
MSGLGNT	CHAPTER3			
MSGTBL1	CHAPTER3	169	106	
MSGTBL2	CHAPTER3	171	108	
MSGTBL3	CHAPTER3	172	109	
SNSCOMB	APPENDX1	173	110	
		232	114	
LRETRYC	APPENDX1			232, 232
		250	115	
LPROGIN	APPENDX1			251
		250	116	
				251
LFLDPTR	APPENDX1			251
		251	117	
				252
WRTCODE	APPENDX3			252
		267	124	
				269
RSGCODE	APPENDX3			269
		268	125	
				270
LOGCODE	APPENDX3			270
		269	126	
				271

<b>Processing Options</b>
---------------------------

## Runtime values:

Document fileid .....	DOCUMENT S10SPEC
Document type .....	USERDOC
Document style .....	ISO9000
Profile .....	EDFPRF40
Service Level .....	0024
SCRIPT/VS Release .....	4.0.0
Date .....	94.07.12
Time .....	09:52:17
Device .....	3820A
Number of Passes .....	2
Index .....	YES
SYSVAR G .....	INLINE
SYSVAR H .....	Y
SYSVAR R .....	ALLICAT
SYSVAR W .....	ALLICAT
SYSVAR X .....	XONLY
SYSVAR 1 .....	OEM

## Formatting values used:

Annotation .....	NO
Cross reference listing .....	YES
Cross reference head prefix only .....	NO
Dialog .....	LABEL
Duplex .....	YES
DVCF conditions file .....	(none)
DVCF value 1 .....	OEM
DVCF value 2 .....	(none)

DVCF value 3 .....	(none)
DVCF value 4 .....	(none)
DVCF value 5 .....	(none)
DVCF value 6 .....	(none)
DVCF value 7 .....	(none)
DVCF value 8 .....	(none)
DVCF value 9 .....	(none)
Explode .....	NO
Figure list on new page .....	YES
Figure/table number separation .....	YES
Folio-by-chapter .....	NO
Head 0 body text .....	(none)
Head 1 body text .....	(none)
Head 1 appendix text .....	Appendix
Hyphenation .....	YES
Justification .....	NO
Language .....	ENGL
Keyboard .....	395
Layout .....	OFF
Leader dots .....	YES
Master index .....	(none)
Partial TOC (maximum level) .....	4
Partial TOC (new page after) .....	RHPAGE
Print example id's .....	NO
Print cross reference page numbers .....	YES
Process value .....	(none)
Punctuation move characters .....	..
Read cross-reference file .....	ALLICAT
Running heading/footing rule .....	NONE
Show index entries .....	NO
Table of Contents (maximum level) .....	3
Table list on new page .....	YES
Title page (draft) alignment .....	RIGHT
Write cross-reference file .....	ALLICAT

<b>Named Files</b>
--------------------

<u>Fileid</u>	<u>Page</u>	<u>Data Set Name</u>
CORNERS	1	CORNERS SCRIPT
DEC2HEX	1	DEC2HEX SCRIPT
COMMAS	1	COMMAS SCRIPT
SYMBOLS	1	SYMBOLS ALLICAT
DOCLIST	1	DOCLIST ALLICAT
SPECTBLS	1	SPECTBLS S10SPEC
CHAPTER1	19	CHAPTER1 S10SPEC
FORMAT	21	FORMAT S10SPEC
INQRYCMD	27	INQRYCMD S10SPEC
LOGSECMD	41	LOGSECMD S10SPEC
LOGSNCMD	44	LOGSNCMD S10SPEC
MSELCMD	55	MSELCMD S10SPEC
MDSNSCMD	90	MDSNSCMD S10SPEC
PRFCHCMD	99	PRFCHCMD S10SPEC
READCMDS	101	READCMDS S10SPEC
RDBUFCMD	103	RDBUFCMD S10SPEC
RDCAPCMD	105	RDCAPCMD S10SPEC
RDDEFCMD	106	RDDEFCMD S10SPEC
RDLNGCMD	112	RDLNGCMD S10SPEC
REASSCMD	113	REASSCMD S10SPEC
RDIAGCMD	115	RDIAGCMD S10SPEC
RELCMDS	118	RELCMDS S10SPEC
REQSNCMD	120	REQSNCMD S10SPEC
RESCMDS	122	RESCMDS S10SPEC
RZEROCMD	125	RZEROCMD S10SPEC
SEEKCMDS	126	SEEKCMDS S10SPEC
SDIAGCMD	128	SDIAGCMD S10SPEC
SUNITCMD	131	SUNITCMD S10SPEC
SYNCHCMD	133	SYNCHCMD S10SPEC
TURDYCMD	134	TURDYCMD S10SPEC
VERIFCMD	135	VERIFCMD S10SPEC
WRITCMDS	137	WRITCMDS S10SPEC
WRTVYCMD	139	WRTVYCMD S10SPEC
WRBUFCMD	141	WRBUFCMD S10SPEC
WRLNGCMD	144	WRLNGCMD S10SPEC
WRSMECMD	145	WRSMECMD S10SPEC
CHAPTER2	146	CHAPTER2 S10SPEC
CHAPTER3	148	CHAPTER3 S10SPEC
CHAPTER4	178	CHAPTER4 S10SPEC
SCSIPROT	178	SCSIPROT S10SPEC
COMQUEUE	178	COMQUEUE S10SPEC
BAK2BAK	178	BAK2BAK S10SPEC
MTRSYNC	178	MTRSYNC S10SPEC
ERRLOGS	178	ERRLOGS S10SPEC

PFA	178	PFA S10SPEC
SEGCACH	178	SEGCACH S10SPEC
LGBLKSZE	178	LGBLKSZE S10SPEC
LRC	178	LRC S10SPEC
MISCHAP4	178	MISCHAP4 S10SPEC
STRIPED	178	OPTSTR S2SPEC
DEGRADE4	178	DEGRADE4 S10SPEC
PRIORITY	178	PRIORITY S10SPEC
ARRECREL	211	ARRECREL S10SPEC
APPENDX1	229	APPENDX1 S10SPEC
SNSCOMBO	232	SNSCOMBO S10SPEC
APPENDX2	253	APPENDX2 S10SPEC
DRPTABLE	256	DRPTABLE S10SPEC
VWPTABLE	263	VWPTABLE S10SPEC
VRPTABLE	263	VRPTABLE S10SPEC
WRPTABLE	264	WRPTABLE S10SPEC
SRPTABLE	265	SRPTABLE S10SPEC
PRPTABLE	265	PRPTABLE S10SPEC
APPENDX3	266	APPENDX3 S10SPEC
APPENDX4	282	APPENDX4 S10SPEC
UECLIST	283	UECLIST S10SPEC
GLOSSARY	291	GLOSSARY S10SPEC
SYMBOLS	291	SYMBOLS ALLICAT
BIBLIOG	292	BIBLIOG ALLICAT

Imbed Trace
-------------

Page 0	CORNERS
Page 0	DEC2HEX
Page 0	COMMAS
Page 0	SETUP
Page 0	SYMBOLS
Page 0	DOCLIST
Page 0	SPECTBLS
Page 0	SCSIINX2
Page 2	FEATURES
Page 3	SCSIATM2
Page 18	CHAPTER1
Page 21	FORMAT
Page 27	INQRYCMD
Page 41	LOGSECMD
Page 44	LOGSNCMD
Page 55	MDELCMD
Page 90	MDSNSCMD
Page 99	PRFCHCMD
Page 101	READCMDS
Page 103	RDBUFCMD
Page 105	RDCAPCMD
Page 106	RDDEF CMD
Page 112	RDLNGCMD
Page 113	REASSCMD
Page 115	RDIAGCMD
Page 118	RELCMDS
Page 120	REQSNCMD
Page 121	RESCMDS
Page 124	RZEROCMD
Page 126	SEEKCMDS
Page 128	SDIAGCMD
Page 131	SUNITCMD
Page 133	SYNCHCMD
Page 134	TURDYCMD
Page 135	VERIFCMD
Page 136	WRITCMDS
Page 139	WRTVYCMD
Page 141	WRBUFCMD
Page 144	WRLNGCMD
Page 145	WRSMECMD
Page 146	CHAPTER2
Page 148	CHAPTER3
Page 178	CHAPTER4
Page 178	SCSIPROT
Page 190	DEGRADE4
Page 203	COMQUEUE
Page 207	BAK2BAK
Page 208	MTRSYNC
Page 210	ERRLOGS
Page 211	ARRECREL
Page 213	PFA
Page 213	SEGCACH
Page 220	MISCHAP4
Page 227	MUNIQUE S10SPEC

Page 228	APPENDX1
Page 232	SNSCOMBO
Page 253	APPENDX2
Page 256	DRPTABLE
Page 263	VWPTABLE
Page 263	VRPTABLE
Page 264	WRPTABLE
Page 265	SRPTABLE
Page 265	PRPTABLE
Page 266	APPENDX3
Page 282	APPENDX4
Page 283	UECLIST
Page 290	GLOSSARY
Page 290	SYMBOLS