

ULTRASTAR 36XP/18ZX/9LZX

Parallel SCSI Disk Drive Logical Interface Specification
Release 1.6

AS21-0224-16

Last changes: April 6, 1999

Notes:

IBM, Ultrastar 18ZX, 9LZX, 36XP, 18LP, No-ID and Predictive Failure Analysis are trademarks of International Business Machines Corporation in the United States and/or other Countries.

Preface

Product Features

"ANSI SCSI-3 Fast-40 Parallel Interface"	Refer to the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec.
Automatic Recommend Rewrite/Reallocate	Refer to "Automatic Rewrite/Reallocate-Recommend Rewrite/Reassign."
Automatic Rewrite/Reallocate	Refer to "Automatic Rewrite/Reallocate-Recommend Rewrite/Reassign."
Back to Back Writes	Refer to "Back-to-Back Writes."
Basic Assurance Tests	Refer to "Basic Assurance Tests (BATS)."
Command Queuing (both Tagged and Untagged)	Refer to "Command Queuing."
Command Reordering	Refer to "Command Queuing."
Down-loadable SCSI Firmware	Refer to "Write Buffer."
Full 32 byte SCSI Sense Data Format	Refer to Appendix A, "SCSI Sense Data Format."
LED support	Refer to "LED Pin."
Multiple Initiator Support	Refer to "Multiple Initiator Systems."
Multimode SE/LVD	Refer to "Synchronous Data Transfer Request"
"On the Fly" error correction capabilities	Refer to the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec for reliability features.
Optional ANSI SCSI commands supported	Refer to "SCSI Commands."
Performance counter information via Log Sense	Refer to "Log Sense."
Predictive Failure Analysis (TM) (PFA (TM))	Refer to "Predictive Failure Analysis."
Read Ahead support	Refer to "Read-Ahead."
Sector Slipping Defect Management	Refer to "Reassign Blocks."
Segmented Caching	Refer to "Segmented Caching."
Selectable reporting of Errors (QPE etc.)	Refer to "Page 0h - Vendor Unique Parameters."
Self-optimizing buffer ratios	Refer to "Page 2h - Disconnect/Reconnect Parameters."
Synchronous/wide data transfer support	Refer to "Supported Messages" and Page 0h - Vendor Unique Parameters."
Variable Block Lengths	Refer to the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec.
Write Caching (Volatile)	Refer to "Page 8h - Caching Parameters" regarding the WCE bit.

About This Manual

This manual is a description of the Small Computer System Interface (SCSI) function supported by the Drive.

The first three chapters cover SCSI commands, status, and messages.

Chapter 4 covers behavior under various operating conditions.

The Appendices cover sense data and error recovery.

For elements of SCSI not described in this document (that is, physical characteristics, bus phases), refer to the *Small Computer System Interface-2 (ANSI X3.131 : 1994)*.

For non-SCSI characteristics of this Drive such as physical dimensions, power requirements, and performance, refer to the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec.

Every attempt has been made to make this Drive compliant with the definition of the *Small Computer System Interface-2 (ANSI X3.131: 1994)* *Small Computer System Interface-3 Block Commands Revision 8b (ANSI NCITS.306-199x)*. *Small Computer System Interface-3 Architecture Model Revision 18 (ANSI x3.270-1996)*. *Small Computer System Interface-3 Primary Commands Revision 11a (ANSI x3.301-1997)*. *Small Computer System Interface-3 Interlocked Protocol Revision 10 (ANSI x3.292-1997)* standards.

SCSI-2 by design provides compatibility with SCSI devices that support bus parity and meet conformance level 2 of the American National Standard X3.131:1994. Accordingly, this Drive can coexist with earlier SCSI devices on the same bus. The Drive also operates well even when the Initiator does not support the SCSI-2 protocol extensions.

Note: This Document was printed from an on-line system and must only be used for reference purposes.

- The Master document is the on-line version, ULTRADOC.LWP on M:\K2\DOCS\SCSISPEC\ on a LAN server in Rochester, Minnesota.
- To preserve the integrity of this document, please DO NOT remove any pages from this printed copy.
- The individual who receives a copy of this document is responsible for prompt disposal of the copy when it is superseded by a newer version of the document or by another document.
- If this document is received by other means than from the Development group, the individual who received this document should report to Development. Failure to do so prevents Development from informing the individual of future releases.

APPROVALS

This document is pending approval by:

Department 2D8 Manager

REVIEW PROCEDURES

The content of this specification is directly tied to the current microcode (firmware) release. Therefore, this document is reviewed when new levels of microcode are about to be released.

The next scheduled release will be when the Microcode changes the function of the drive.

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.

Revision History

<u>Version</u>	<u>Date</u>	<u>Comments</u>
1.0	05/22/98	Initial Version created from MATTERHORN, Version 6.0.
1.1	09/01/98	Release 1.1 Version of Ultrastar 18ZX/9LZX Logical Interface document.
1.2	10/06/98	Release 1.2 Version of Ultrastar 18ZX/9LZX/36XP/18LP Logical Interface document.
1.3	12/17/98	Release 1.3 Version of Ultrastar 18ZX/9LZX/36XP/18LP Logical Interface document.
1.4	02/08/99	Release 1.4 Version of Ultrastar 18ZX/9LZX/36XP/18LP Logical Interface document.
1.5	03/08/99	Release 1.5 Version of Ultrastar 18ZX/9LZX/36XP/18LP Logical Interface document.
1.6	04/05/99	Release 1.6 Version of Ultrastar 18ZX/9LZX/36XP/18LP Logical Interface document.

Contents

1.0 SCSI Commands	17
1.1.1 CDB Control Byte	18
1.1.2 Reserved Field Definitions	18
1.1.3 Reserved Value Definitions	18
1.1.4 Invalid Field Errors	18
1.2 Format Unit	19
1.3 Inquiry	25
1.3.1 Inquiry Data Format - EVPD = 0	27
1.3.2 Inquiry Data Format - EVPD = 1 - Page Code = 00	30
1.3.3 Inquiry Data Format - EVPD = 1 - Page Code = 01	31
1.3.4 Inquiry Data Format - EVPD = 1 - Page Code = 03	32
1.3.5 Inquiry Data Format - EVPD = 1 - Page Code = 80	33
1.3.6 Inquiry Data Format - EVPD = 1 - Page Code = 82	34
1.3.7 Inquiry Data Format - EVPD = 1 - Page Code = 83	35
1.4 Log Select	38
1.4.1 Log Select Page 01h	40
1.4.2 Log Select Page 02h	41
1.4.3 Log Select Page 03h	42
1.4.4 Log Select Page 05h	43
1.4.5 Log Select Page 06h	44
1.5 Log Sense	45
1.5.2 Log Sense Page 00h	47
1.5.3 Log Sense Page 01h	48
1.5.4 Log Sense Page 02h	49
1.5.5 Log Sense Page 03h	50
1.5.6 Log Sense Page 05h	51
1.5.7 Log Sense Page 06h	52
1.5.8 Log Sense Page 30h	53
1.5.9 Log Sense Page 32h	55
1.5.10 Log Sense Page 33h	57
1.5.11 Log Sense Page 35h	58
1.5.12 Log Sense Page 36h	59
1.6 Mode Select (6)	60
1.6.2 Page 0h - Vendor Unique Parameters	64
1.6.3 Page 1h - Error Recovery Parameters	71
1.6.4 Page 2h - Disconnect/Reconnect Parameters	77
1.6.5 Page 3h - Format Device Parameters	80
1.6.6 Page 4h - Rigid Disk Drive Geometry Parameters	82
1.6.7 Page 7h - Error Recovery Parameters for Verify	84
1.6.8 Page 8h - Caching Parameters	86
1.6.9 Page 09h - Peripheral Device Parameters	90
1.6.10 Page 0Ah - Control Mode Parameters	91
1.6.11 Page 0Ch - Notch Parameters	94
1.6.12 Page 1Ah - Power Conditions	96
1.6.13 Page 1Ch - Informational Exceptions Control	97
1.7 Mode Sense (6)	100

1.7.1	Mode Sense Data Format (of Header & Block Descriptor)	102
1.7.2	Mode Sense Data Format (of All Pages)	104
1.8	Persistent Reserve In	112
1.9	Persistent Reserve Out	117
1.10	Pre-Fetch	120
1.11	Read (6)	122
1.12	Read (10)	123
1.13	Read Buffer	124
1.14	Read Capacity	126
1.15	Read Defect Data (10)	127
1.16	Read Defect Data (12)	130
1.17	Read Long	133
1.18	Reassign Blocks	134
1.19	Receive Diagnostic Results	136
1.19.1	Supported Diagnostic Pages - Receive Diagnostic	136
1.19.2	Receive Diagnostic Page 40h	136
1.19.3	Receive Diagnostic Page A0h	138
1.20	Release (6)	139
1.21	Release (10)	140
1.22	Report LUNs	141
1.23	Request Sense	143
1.24	Reserve (6)	144
1.25	Reserve (10)	145
1.26	Rezero Unit	147
1.27	Seek (6)	148
1.28	Seek (10)	148
1.29	Send Diagnostic	149
1.29.1	Send Diagnostic Page 00h	151
1.29.2	Send Diagnostic Page 40h	151
1.29.3	Send Diagnostic Page A0h	152
1.30	Start/Stop Unit	153
1.31	Synchronize Cache	154
1.32	Test Unit Ready	155
1.33	Verify	156
1.34	Write (6)	157
1.35	Write (10)	158
1.36	Write and Verify	159
1.37	Write Buffer	161
1.38	Write Long	164
1.39	Write Same	165
2.0	SCSI Status Byte	166
3.0	SCSI Message System	168
3.1	Supported Messages	168
3.1.1	Command Complete (00h)	169
3.1.2	Save Data Pointer (02h)	169
3.1.3	Restore Pointers (03h)	169
3.1.4	Disconnect (04h)	169
3.1.5	Initiator Detected Error (05h)	170
3.1.6	Abort (06h)	170

3.1.7	<i>Message Reject (07h)</i>	170
3.1.8	<i>No Operation (08h)</i>	171
3.1.10	<i>Linked Command Complete (0Ah)</i>	171
3.1.11	<i>Linked Command Complete with Flag (0Bh)</i>	171
3.1.12	<i>Bus Device Reset (0Ch)</i>	171
3.1.13	<i>Abort Tag (0Dh)</i>	171
3.1.14	<i>Clear Queue (0Eh)</i>	172
3.1.15	<i>Queue Tag Messages (20h, 21h, 22h)</i>	172
3.1.16	<i>Ignore Wide Residue (23h)</i>	173
3.1.17	<i>Identify (80h or C0h)</i>	173
3.1.18	<i>Extended Messages (01h,xxh)</i>	174
3.1.19	<i>Synchronous Data Transfer Request (01h,01h)</i>	174
3.1.20	<i>Wide Data Transfer Request (01h,03h)</i>	179
3.2	Supported Message Functions	182
3.3	Attention Condition	183
3.4	SCSI Bus Parity Error Detection	185
3.5	Message State/Action Table	186
3.6	SCSI Bus Related Error Handling Protocol	194
3.6.1	<i>Unexpected Bus Free Phase Error Condition</i>	194
3.6.2	<i>Message Out Phase Retry</i>	195
3.6.3	<i>Message In Phase Retry</i>	195
3.6.4	<i>Command Phase Retry</i>	196
3.6.5	<i>Status Phase Retry</i>	196
3.6.6	<i>Data Out Phase Retry</i>	197
3.6.7	<i>Data In Phase Retry</i>	197
3.6.8	<i>Identify Message In Phase Retry</i>	197
3.6.9	<i>Simple Queue Tag Message In Phase Retry</i>	197
4.0	Additional Information	199
4.1	SCSI Protocol	199
4.1.1	<i>Priority of SCSI Status Byte Reporting</i>	199
4.1.2	<i>Invalid LUN Processing</i>	199
4.1.3	<i>Incorrect Initiator Connection</i>	200
4.1.4	<i>Command Processing During Execution of Active I/O Process</i>	200
4.1.5	<i>Unit Attention Condition</i>	203
4.1.6	<i>Command Processing During Start-up and Format Operations</i>	205
4.1.7	<i>Internal Error Condition</i>	205
4.1.8	<i>Deferred Error Condition</i>	207
4.1.9	<i>Degraded Mode</i>	208
4.1.10	<i>Command Processing While Reserved</i>	217
4.2	Priority Commands	218
4.3	Command Queuing	219
4.3.1	<i>Untagged Queuing</i>	219
4.3.2	<i>Tagged Queuing</i>	219
4.3.3	<i>Queue Full</i>	220
4.3.4	<i>Rules for Queuing Commands</i>	220
4.3.5	<i>Effects of Errors on Command Queuing</i>	220
4.3.6	<i>Termination of I/O Processes</i>	221
4.4	Concurrent Command Processing	222
4.4.1	<i>Back-to-Back Writes</i>	222

4.5 Error Logs	224
4.6 Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign (ARRR)	224
4.7 Predictive Failure Analysis (PFA)	226
4.8 Segmented Caching	227
4.8.1 Overview	227
4.8.2 Options Supported	227
4.8.3 Read-Ahead	228
4.8.4 Write Cache	231
4.8.5 Synchronize Cache on SCSI Reset	232
4.8.6 Segment Selection	232
4.8.7 Data Discard	233
4.8.8 Segment Prioritization	233
4.9 Reselection Timeout	235
4.10 Single Initiator Selection	235
4.11 Nonarbitrating Systems	235
4.12 Selection Without ATN	235
4.13 Multiple Initiator Systems	236
4.13.1 Initiator Sense Data	236
4.13.2 Initiator Mode Select/Mode Sense Parameters	236
4.13.3 Initiator Data Transfer Mode Parameters	236
4.14 Options Jumper Block	237
4.15 LED Pin	237
4.16 Reset	237
4.17 Bring-Up Sequence	238
4.18 Basic Assurance Tests (BATS)	239
4.18.1 BATS1 Tests	240
4.18.2 BATS2 Tests	240
4.19 Reserved Area	240
Appendix A. SCSI Sense Data Format	242
A.1.1 Sense Data Description	243
Appendix B. Recovery Procedures	259
B.1 Data Recovery Procedure for Data Field Errors	259
B.1.1 Read Command Data Recovery Procedure Steps	260
B.1.2 Verify Commands	264
B.2 Error Recovery for Non-Data Errors	269
B.2.1 Position Error (Servo Error Recovery)	269
B.2.2 Drive Fault	269
B.2.3 Spindle Motor Error	269
B.3 Priority of Error Reporting	269
Appendix C. Recommended Initiator ERP	270
C.1 Drive Service Strategy	270
C.2 Recommendations for System Error Log	271
C.3 Data Recovery Procedure	271
C.3.1 Rewrite an LBA	272
C.3.2 Reassign a Physical Sector	272
C.3.3 Data Error Logging	274
C.3.4 Reassign Blocks Recovery	274
C.4 Nondata Error Recovery Procedure	275
C.4.1 Drive Busy	275

<i>C.4.2 Unrecovered Drive Error</i>	275
<i>C.4.3 Recovered Drive Error</i>	276
<i>C.4.4 Drive Not Ready</i>	276
<i>C.4.5 No Defect Spare</i>	276
<i>C.4.6 Degraded Mode</i>	276
<i>C.4.7 Reserved Area Hard Error</i>	277
<i>C.4.8 Interface Protocol</i>	278
<i>C.4.9 Aborted Command</i>	278
<i>C.4.10 Unit Attention</i>	280
<i>C.4.11 Components Mismatch</i>	280
<i>C.4.12 Self Initiated Reset</i>	281
<i>C.4.13 Defect List Recovery</i>	282
<i>C.4.14 Miscompare Recovery</i>	282
<i>C.4.15 Microcode Error</i>	283
<i>C.4.16 Predictive Failure Analysis</i>	283
Appendix D. UEC List	284

Figures

Figure 1. Attention Condition during Asynchronous Information Transfer phase	183
Figure 2. Attention Condition during Synchronous Data Transfer phase	184

Tables

Table 1. SCSI Commands Supported	17
Table 2. Format Unit Command Descriptor Block	19
Table 3. Format of Defect List - Header	20
Table 4. Format of Defect List - Descriptor(s)	20
Table 5. Effect of Options on Format Operations	22
Table 6. Defect Descriptor - Bytes From Index Format	23
Table 7. Defect Descriptor - Physical Sector Format	23
Table 8. Inquiry Command Descriptor Block	25
Table 9. Standard Inquiry Data - EVPD = 0	27
Table 10. Inquiry Data - EVPD = 1 - Page Code = 00	30
Table 11. Inquiry Data - EVPD = 1 - Page Code = 01	31
Table 12. Inquiry Data - EVPD = 1 - Page Code = 03	32
Table 13. Inquiry Data - EVPD = 1 - Page Code = 80	33
Table 14. Inquiry Data - EVPD = 1 - Page Code = 82	34
Table 15. Inquiry Data - EVPD = 1 - Page Code = 83	35
Table 16. Code Set Values	36
Table 17. Identifier Type Values	36
Table 18. Inquiry Data - EVPD = 1 - Page Code = D1	37
Table 19. Inquiry Data - EPVD = 1 - Page Code = D2	37
Table 20. Log Select Command Descriptor Block	38
Table 21. Log Select Page 01h	40
Table 22. Log Select Page 01h - Parameter Code 0000h	40
Table 23. Log Select Page 01h - Parameter Code 0001h	40
Table 24. Log Select Page 02h	41
Table 25. Log Select Page 02h - Parameter Code 0002h	41
Table 26. Log Select Page 02h - Parameter Code 0003h	41
Table 27. Log Select Page 02h - Parameter Code 0006h	41
Table 28. Log Select Page 03h	42
Table 29. Log Select Page 03h - Parameter Code 0000h	42
Table 30. Log Select Page 03h - Parameter Code 0002h	42
Table 31. Log Select Page 03h - Parameter Code 0003h	42
Table 32. Log Select Page 03h - Parameter Code 0006h	43
Table 33. Log Select Page 05h	43
Table 34. Log Select Page 05h - Parameter Code 0000h	43
Table 35. Log Select Page 05h - Parameter Code 0002h	43
Table 36. Log Select Page 05h - Parameter Code 0003h	44
Table 37. Log Select Page 05h - Parameter Code 0006h	44
Table 38. Log Select Page 06h	44
Table 39. Log Sense Command Descriptor Block	45
Table 40. Log Sense Page 00h	47
Table 41. Log Sense Page 01h	48
Table 42. Log Sense Page 01h - Parameter Code 0000h	48
Table 43. Log Sense Page 01h - Parameter Code 0001h	48
Table 44. Log Sense Page 02h	49
Table 45. Log Sense Page 02h - Parameter Code 0002h	49
Table 46. Log Sense Page 02h - Parameter Code 0003h	49

Table 47. Log Sense Page 02h - Parameter Code 0006h	49
Table 48. Log Sense Page 03h	50
Table 49. Log Sense Page 03h - Parameter Code 0000h	50
Table 50. Log Sense Page 03h - Parameter Code 0002h	50
Table 51. Log Sense Page 03h - Parameter Code 0003h	50
Table 52. Log Sense Page 03h - Parameter Code 0006h	51
Table 53. Log Sense Page 05h	51
Table 54. Log Sense Page 05h - Parameter Code 0000h	51
Table 55. Log Sense Page 05h - Parameter Code 0002h	51
Table 56. Log Sense Page 05h - Parameter Code 0003h	51
Table 57. Log Sense Page 05h - Parameter Code 0006h	52
Table 58. Log Sense Page 06h	52
Table 59. Log Sense Page 30h	53
Table 60. Log Sense Page 32h	55
Table 61. Log Parameter Format - Page 32h	56
Table 62. Log Sense Page 33h	57
Table 63. Log Parameter Format - Page 33h	57
Table 64. Log Sense Page 35h	58
Table 65. Log Sense Page 36h	59
Table 66. Mode Select Command Descriptor Block	60
Table 67. Mode Select Parameter List - Header	61
Table 68. Mode Select Parameter List - Block Descriptor	62
Table 69. Mode Select Parameter List - Page Descriptor(s)	63
Table 70. Mode Select Data Format Page 00h	64
Table 71. Error Reporting Modes	66
Table 72. Mode Select Data Format Page 01h	71
Table 73. Mode Select Data Format Page 02h	77
Table 74. Mode Select Data Format Page 03h	80
Table 75. Mode Select Data Format Page 04h	82
Table 76. Mode Select Data Format Page 07h	84
Table 77. Mode Select Data Format Page 08h	86
Table 78. Mode Select Data Format Page 09h	90
Table 79. Mode Select Data Format Page 0Ah	91
Table 80. Mode Select Data Format Page 0Ch	94
Table 81. Mode Select Data Format Page 1Ah	96
Table 82. Mode Select Data Format Page 1Ch	97
Table 83. Method of Reporting Informational Exception Fields	99
Table 84. Mode Sense Command Descriptor Block	100
Table 85. Mode Sense Data Format - Header	102
Table 86. Mode Sense Data Format - Block Descriptor	102
Table 87. Mode Sense Data Format Page 00h	104
Table 88. Mode Sense Data Format Page 01h	104
Table 89. Mode Sense Data Format Page 02h	105
Table 90. Mode Sense Data Format Page 03h	106
Table 91. Mode Sense Data Format Page 04h	107
Table 92. Mode Sense Data Format Page 07h	108
Table 93. Mode Sense Data Format Page 08h	108
Table 94. Mode Sense Data Format Page 09h	109
Table 95. Mode Sense Data Format Page 0Ah	109

Table 96. Mode Sense Data Format Page 0Ch	110
Table 97. Mode Sense Data Format Page 1Ah	110
Table 98. Mode Sense Data Format Page 1Ch	111
Table 99. Persistent Reserve In command	112
Table 100. Persistent Reserve In Service Action codes	112
Table 101. Persistent Reserve In parameter data for Read Keys	113
Table 102. Persistent Reserve In parameter data for Read Reservation	113
Table 103. Persistent Reserve In Reservation Descriptor	114
Table 104. Persistent Reservation SCOPE Codes	115
Table 105. Persistent Reservation Type Codes	115
Table 106. Persistent Reserve Out command	117
Table 107. Persistent Reserve Out service action codes	118
Table 108. Persistent Reserve Out parameter list	118
Table 109. Pre-Fetch Command Descriptor Block	120
Table 110. Read (6) Command Descriptor Block	122
Table 111. Read (10) Command Descriptor Block	123
Table 112. Read Buffer Command Descriptor Block	124
Table 113. Read Capacity Command Descriptor Block	126
Table 114. Format of Read Capacity Data Returned	126
Table 115. Read Defect Data Command Descriptor Block	127
Table 116. Read Defect Data - Header	128
Table 117. Read Defect Data - Defect Descriptor(s)	128
Table 118. Defect Descriptor - Bytes From Index Format	129
Table 119. Defect Descriptor - Physical Sector Format	129
Table 120. Read Defect Data Command Descriptor Block	130
Table 121. Read Defect Data - Header	131
Table 122. Read Defect Data - Defect Descriptor	131
Table 123. Defect Descriptor - Bytes From Index Format	132
Table 124. Defect Descriptor - Physical Sector Format	132
Table 125. Read Long Command Descriptor Block	133
Table 126. Reassign Blocks Command Descriptor Block	134
Table 127. Format of Reassign Blocks Data - Header	135
Table 128. Format of Reassign Blocks Data - Descriptor(s)	135
Table 129. Receive Diagnostic Results Command Descriptor Block	136
Table 130. Supported Diagnostic Pages - Receive Diagnostic	136
Table 131. Translate Address Page - Receive Diagnostic	137
Table 132. Device LED Control Page - Receive Diagnostic	138
Table 133. Release (6) Command Descriptor Block	139
Table 134. Release (10) Command Descriptor Block	140
Table 135. Report LUNs Command Descriptor Block	141
Table 136. LUN Reporting Parameter List Format	141
Table 137. Request Sense Command Descriptor Block	143
Table 138. Reserve (6) Command Descriptor Block	144
Table 139. Reserve (10) Command Descriptor Block	145
Table 140. Rezero Unit Command Descriptor Block	147
Table 141. Seek (6) Command Descriptor Block	148
Table 142. Seek (10) Command Descriptor Block	148
Table 143. Send Diagnostic Command Descriptor Block	149
Table 144. Supported Diagnostics Pages - Send Diagnostic	151

Table 145. Translate Address Page - Send Diagnostic	151
Table 146. Device LED Control Page - Send Diagnostic	152
Table 147. Start/Stop Unit Command Descriptor Block	153
Table 148. Synchronize Cache Command Descriptor Block	154
Table 149. Test Unit Ready Command Descriptor Block	155
Table 150. Verify Command Descriptor Block	156
Table 151. Write (6) Command Descriptor Block	157
Table 152. Write (10) Command Descriptor Block	158
Table 153. Write and Verify Command Descriptor Block	159
Table 154. Write Buffer Command Descriptor Block	161
Table 155. Write Long Descriptor Block	164
Table 156. Write Same Command Descriptor Block	165
Table 157. SCSI Status Byte	166
Table 158. Status Code Bit Definitions	166
Table 159. Supported One-Byte Messages	168
Table 160. Supported Two-Byte Messages	168
Table 161. Supported Multiple-Byte Messages	169
Table 162. Queue Tag Messages	172
Table 163. Ignore Wide Residue Message Format	173
Table 164. Extended Message Format	174
Table 165. Synchronous Data Transfer Request Message Definition	175
Table 166. Synchronous Data Transfer Period Factor	175
Table 167. Initiator Request/Target Response	176
Table 168. Target Response to Initiator's Transfer Period	177
Table 169. Wide Data Transfer Request Message Definition	179
Table 170. Initiator Request/Target Response	179
Table 171. Target Response to Initiator's Transfer Period	180
Table 172. Action Table Example	186
Table 173. Message Action Legend	187
Table 174. Message State/Action Table (Part 1)	188
Table 175. Message State/Action Table (Part 2)	190
Table 175. Message State/Action Table (Part 2, cont.)	192
Table 176. Message State/Action Table (Part 3)	193
Table 177. Auto/Recommend Rewrite/Reallocate Sense Data Combinations.	226
Table 178. Format of Sense Data	242
Table 179. Valid Sense Key, Code, Qualifier Combinations Used by the Drive.	245
Table 180. Actual Retry Count Bytes	256
Table 181. Progress Indication Bytes	256
Table 182. Field Pointer Bytes	257
Table 183. Read Recovery Steps	260
Table 184. TA Branch of Read Recovery	262
Table 185. Write and Verify Data Recovery Procedure Steps	265
Table 186. Verify Recovery Steps	267
Table 187. Servo Error Recovery Steps	269
Table 188. Drive Fault Recovery Steps	269
Table 189. Recommend Rewrite Errors	272
Table 190. Recommend Reassign Errors	273
Table 191. Log Only Errors	274
Table 192. Unit Error Codes with Descriptions.	284

1.0 SCSI Commands

Following is a summary of SCSI commands supported. The column "SCSI" refers to Small Computer System Interface-2 (ANSI X3.131 : 1994), Small Computer System Interface-3 Block Commands Revision 8b (ANSI NCITS.306-199x) and Small Computer System Interface-3 Primary Commands Revision 11a (ANSI x3.301-1997) standards.

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
O	5Eh	Persistent Reserve In
O	5Fh	Persistent Reserve Out
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 Results
M	17h	Release
O	57h	Release (10)*
O	A0h	Report LUNs
M	03h	Request Sense
M	16h	Reserve
O	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

* 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 meanings of these fields are 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

Table 2. Format Unit Command Descriptor Block

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	Interleave Factor							
4								
5	VU = 0			Reserved = 0			Flag	Link

The Format Unit Command performs a physical formatting of the Drive. It can perform a full format where a binary pattern of zeroes is written to every customer block or it can perform a fast format which only updates tables and writes no customer blocks. How the format function is performed depends on Mode Page 0 settings (see 1.6.2, “Page 0h - Vendor Unique Parameters”).

Note: Performance MAY NOT BE AS GOOD with the fast format operation.

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 when Initiators use the Reassign Blocks command, when defects are detected while writing customer blocks during a full format operation (see 1.6.2, “Page 0h - Vendor Unique Parameters”), and when the Automatic Reallocate function reassigns a block (see 4.6, “Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign”). 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.

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 Tables 3 and 4).

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).

Interleave Factor must be either zero or one. The target formats the file with an 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”) 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.

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”).
- 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”).
- 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”). 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”).

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 = 1	IP = 0	DSP	Immed	RSVD
2	Defect list length							
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							
Note: 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) and IP (Initialization Pattern) 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.

Note: The FCERT bit in Mode Page 0 is provided to control certification when the format command is issued with the FOV bit off.

When FOV bit is one there are eight combinations of the DPRY, DCRT, STPF, IP and DSP bits allowed. Other combinations return *Check Condition Status* with Sense Key of Illegal Request and Additional Sense Code of Invalid Field In Parameter List. The supported combinations are:

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

The DPRY (disable primary) bit set to one indicates that the Target uses portions of the medium identified as defective in the primary defect PList for Initiator addressable logical blocks.

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

The DCRT (disable certification) bit set to zero, indicates that the Target performs a certification process while executing the Format Unit Command. (This will lengthen the time it takes to complete the Format command.)

Note: Since the DCRT bit is part of the Data Out phase that follows the format command, the FCERT bit in Mode Page 0 is provided to control certification when the format command is issued with no Data Out phase (see 1.6.2, “Page 0h - Vendor Unique Parameters”). If a format command is issued with a Data Out phase then FCERT is ignored.

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 or 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 set to zero requests that the Target save all saveable Mode parameters during the format operation. If the DSP bit is set to one the Target will not save the saveable Mode 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” for more information regarding the immediate format operation. With the Immed bit set to one, the Link bit must be set to zero otherwise the command is terminated with *Check Condition Status*, Sense Key is set to Illegal Request and Additional Sense Code is set to CDB Invalid.

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>CmpLst</u>	<u>Defect List Format</u>	<u>Defect List Length</u>	<u>Command Type</u> M = Mandatory O = Optional	<u>Defect Sources</u> P = PList G = GList D = DList	<u>New GList Composition</u> E = Erase the GList K = Keep the GList A = Add the Dlist to the Glist R = Replace the Glist with the DList
<u>No Defect List</u>						
0	0	000b	N/A	M	P,G	K
<u>Block Format</u>						
1	0	000b	Zero	M	P,G	K
1	1	000b	Zero	M	P	E
<u>Bytes From Index Format</u>						
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
<u>Physical Sector Format</u>						
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 DList entries be in ascending order.

If the DList 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) 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 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 length, 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) 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 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 length, 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

Table 8. Inquiry Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 12h							
1	Reserved			Reserved = 0			CmdDt= 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.

EVPD (Enable Vital Product Data) bit set to one requests that the target 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.

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 Condition* (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.”

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

- *ASCII uCode Revision Level* (EVPD=0) can be changed when the microcode is downloaded with the `Write Buffer` command.
- *ASCII uCode Identifier* (EVPD=0 and EVPD=1, Page Code=3) 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.

BYTE 0 of the returned data on an Inquiry command is the same no matter which page(s) is(are) returned. This description is to be used for all the following page definitions.

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).

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	Reserved = 0						
2	ISO = 0		ECMA = 0			ANSI = 3		
3	RSVD = 0	TrmTsk = 0	Norm ACA = 0	RSVD = 0	Response Data Format = 2			
4	Additional Length = 159 (9Fh)							
5	Reserved = 0							
6	Reserved = 0	EncServ = 0	Reserved = 0	MultiP = 0	MChngr = 0	ACKREQ Q = 0	Addr32 = 0	Addr16
7	RelAdr = 0	WBus32 = 0	WBus16	Sync = 1	Linked = 1	TranDis = 0	CmdQue = 1	Reserved = 0
8-15	ASCII Vendor ID "IBM"							
16-31	ASCII Product Identifier							
32-35	ASCII uCode Revision Level							
36-45	ASCII Unit Serial Number							
46-55	ASCII uCode Identifier							
56-95	Reserved = 0							
96-97	Vendor Unique Reserved = 0							
98-101	ASCII Plant of Manufacture							
102-106	ASCII Date of Manufacture							
107	Reserved = 0							
108-111	ASCII FRU Count '0001'							
112-113	ASCII FRU Field Length '22'							
114-125	ASCII Assembly P/N							
126-135	ASCII Assembly EC Level							
136-147	ASCII Card Assembly Part Number							
148-157	ASCII Card Assembly EC							
158-161	ASCII Year of Mfg [YYYY]							
162-163	Reserved = 0							

If the inquiry data contained on the media is not available, the fields shown above with constant values contain those values. ASCII uCode Revision Level and ASCII uCode Identifier will be valid. Other fields may or may not hold valid values, but ASCII fields contain ASCII characters.

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 Status*. This indicates "Logical Unit Not Present" and is returned by this Drive for all logical unit numbers except 0. The RMB (removable media bit) field of 0 indicates that no removable media exists.

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 3 indicates the target supports American National Standards Institute SCSI version 3 (ANSI NCITS.306-199x).

The TrmTsk field of 0 indicates the Target does not support the TERMINATE TASK task management function as defined in the SAM.

The Normal ACA Supported bit (NormACA) of zero indicates the device server does not support setting the NACA bit to one in the Control Byte of the CDB as defined in the SAM.

A Response Data Format value of 2 indicates that the data is in the format specified in the ANSI standard (ANSI NCITS.306-199x).

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 Enclosure Services (EncServ) bit of zero indicates that this device does not contain an embedded enclosure services component.

The MultiPort (MultiP) bit of zero indicates that this device has a single port and does not implement multiport requirements.

A medium changer (MChngr) bit is only supported when the RMB bit is one.

An ACKQ/REQQ (ACKREQQ) bit of zero indicates that the device does not support a request and acknowledge data transfer handshake on a Q cable.

A wide SCSI Address 32 (Addr32) bit of zero indicates that the device does not support 32-bit wide SCSI addresses. A wide SCSI Address 32 (Addr32) bit of one indicates that the device does support 32-bit wide SCSI addresses.

A wide SCSI Address 16 (Addr16) bit of one indicates that the device supports 16-bit wide SCSI addresses.

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

A wide bus 32 (Wbus32) bit of zero indicates that the device does not support 32-bit wide data transfers.

A wide bus 16 (Wbus16) bit of one indicates that the device supports 16-bit wide data transfers. A wide bus 16 bit of zero indicates that the device does not support 16-bit wide data transfers.

Note: If the values of both the Wbus16 and Wbus32 bits are zero, the device only supports 8-bit wide data transfers.

A Synchronous transfer (Sync) bit of one indicates that the device supports synchronous data transfer.

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

A transfer disable (TranDis) bit of 0 indicates that the device does not support 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 Vendor ID is "IBM." 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 OEM customers.

The Product Identifier is a 16-character ASCII field. The first four characters are unique to product model. The balance of the field is model/customer dependent.

The Code Revision Level field indicates the level of the microcode. This is an ASCII 4-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 Alphanumeric, right aligned and the unused bytes are ASCII zero (30h).

The uCode Identifier field contains the IBM microcode identifier. The field is left aligned and the unused bytes are filled with space characters (20h).

The Plant of Manufacture field indicates the location of the manufacturing facility that made the Drive. A code of '0983' is for San Jose, California and 09RI is for Singapore.

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).

Year of Manufacture: 4 digits ASCII year.

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
0	Peripheral Qualifier			Peripheral Device Type				
1	Page Code = 00h							
2	Reserved = 0							
3	Page Length = 8							
4	First Supported Page Code = 00h							
5	Second Supported Page Code = 01h							
6	Third Supported Page Code = 03h							
7	Fourth Supported Page Code = 80h							
8	Fifth Supported Page Code = 82h							
9	Sixth Supported Page Code = 83h							
10	Seventh Supported Page Code = D1h							
11	Eighth Supported Page Code = D2h							

Bytes 0-11 are always available.

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 Codes supported by the target or logical unit.

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
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).

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 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 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 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 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 = 03

Table 12. Inquiry Data - EVPD = 1 - Page Code = 03

Byte	BIT						
	7	6	5	4	3	2	1
0	Peripheral Qualifier			Peripheral Device Type			
1	Page Code = 03h						
2	Reserved = 0						
3	Page Length = 36						
4	ASCII Length = 0						
5 - 7	Reserved = 0						
8 - 11	Load ID						
12-15	Release level/modification number						
16-19	PTF Number						
20-23	Patch Number						
24-33	ASCII uCode Identifier						
34-35	Reserved (20h)						
36-39	ASCII Servo P/N						

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 4-39). 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 a microcode Load identifier.

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 are reserved.

Bytes 8-23 of this page are used to control the level of microcode that is installed on the Drive. See 1.37, "Write Buffer."

Bytes 24-33 of this page display the drive's microcode identifier. The field is alphanumeric (ASCII), left aligned and the unused bytes are ASCII spaces (20h).

Bytes 36-39 of this page display the part number of the Servo microcode installed on the drive. This field is hex numeric ASCII (i.e., the characters will be in the set 0..9,A..F).

1.3.5 Inquiry Data Format - EVPD = 1 - Page Code = 80

Table 13. Inquiry Data - EVPD = 1 - Page Code = 80

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Peripheral 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).

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 4-19). 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 alphanumeric (ASCII), right aligned and the unused bytes are ASCII spaces (20h).

1.3.6 Inquiry Data Format - EVPD = 1 - Page Code = 82

Table 14. Inquiry Data - EVPD = 1 - Page Code = 82

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Peripheral Qualifier				Peripheral Device Type			
1	Page Code = 82h							
2	Reserved = 0							
3	Page Length = 48							
4	ASCII Length = 25							
5 - 12	First 8 bytes of Product Identifier in ASCII							
13	0							
14-21	First 8 bytes of ASCII Serial Number							
22	0							
23-28	ASCII Vendor ID 'IBM'							
29	0							
30-37	First 8 bytes of Product ID in EBCDIC							
38-45	First 8 bytes of 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).

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 Identifier is the first 8 bytes of the Product Identifier in the EVPD=0 page.

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 Identifier is the first 8 bytes of the Product Identifier in the EVPD=0 page.

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.

1.3.7 Inquiry Data Format - EVPD = 1 - Page Code = 83

Table 15. Inquiry Data - EVPD = 1 - Page Code = 83

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	Page Code = 83h							
2	Reserved = 0							
3	Page Length = 32h							
4	Reserved = 0				Code Set = 2			
5	Reserved = 0				Identifier Type = 1			
6	Reserved = 0							
7	Descriptor List Length = 22h							
8-15	ASCII Vendor ID "IBM"							
16-31	ASCII Product Identifier							
32-41	ASCII Serial Number							
42	Reserved = 0				Code Set = 1			
43					Identifier Type = 2			
44	Reserved = 0							
45	Descriptor List Length = 8							
46-48	IEEE Company Unique ID = 006094h.							
49	Plant of Manufacture Code							
50	Product Family			Number of Heads				
51-53	Serial Number (in Hex)							

Inquiry Page 0x83 (Device Identification Page) provides a way for the initiator to request a target's unique identification descriptor. This product supports two identification descriptors. Both of these descriptors are returned during the Data In Phase of the Inquiry Command. The first identification descriptor is the ASCII version. The second identification descriptor adheres to the IEEE 64-bit Global Identifier (EUI-64) Standard.

Each identification descriptor includes a standard four byte header, which includes a Code Set field, an Identifier Type field, and a Page Length. The unique Identifier data follows this header. Valid Code Set values are described in the following table.

Table 16. Code Set Values

Value	Description
0	Reserved
1	The Identifier field shall contain binary values.
2	The Identifier field shall contain ASCII graphic codes (i.e. code values 20h through 7Eh).

Valid Identifier Type values are described in the following table.

Table 17. Identifier Type Values

Value	Description
1	The first 8 bytes of the Identifier field is IBM's ASCII Vendor ID. The next 16 bytes of the Identifier field is the ASCII Product ID. The last 10 bytes of the Identifier field is the ASCII Serial Number for that unit.
2	The Identifier field contains an IEEE Extended Unique Identifier, 64-bit (EUI-64). The first 3 bytes represent the Company ID as assigned by the IEEE Registration Authority Committee. The next byte contains the Plant of Manufacture Code. The following byte is split into two fields. Bits 0-4 contain the hex equivalent to the number of heads. Bits 5-7 contain the Product Family ID. The last 3 bytes contain the hex equivalent to the Device Serial Number.

The Page Length specifies the actual number of bytes that the Identifier field occupies.

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).

The ASCII Product Identifier is the first 8 bytes of the Product Identifier in the EVPD=0 page. 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 IEEE Company ID given to IBM by the Registration Authority Committee is 006094h.

Valid Plant of Manufacture Codes include:

- San Jose, CA = 07h
- Singapore = 87h

Valid Product Family consists of a unique number assigned to each product model.

Table 18. Inquiry Data - EVPD = 1 = Page Code = D1

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	Page Code = D1h							
2	Reserved = 0							
3	Page Length = 80							
4 - 19	ASCII Media Disk Definition							
20-35	ASCII Motor Serial Number							
36-51	ASCII Flex Assembly Serial Number							
52-67	ASCII Actuator Serial Number							
68-83	ASCII Device Enclosure Serial Number							

Table 19. Inquiry Data - EPVD = 1 - Page Code = D2

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	Page Code = D2h							
2	Reserved = 0							
3	Page Length = 32							
4 - 19	ASCII Card Serial Number							
20-35	ASCII Card Assembly Part Number							

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

All ASCII fields are alphanumeric, left-aligned, and padded on the right with ASCII blanks (20h).

1.4 Log Select

Table 20. Log Select Command Descriptor Block

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 4Ch							
1	Reserved			Reserved = 0			PCR	SP
2	PC = 11b or 01b		Reserved = 0					
3	Reserved = 0							
4	Reserved = 0							
5	Reserved = 0							
6	Reserved = 0							
7	(MSB) Parameter List Length (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, while a value of 0 (except when PC=11b) indicates that the parameters should not be cleared. Parameter list length must be zero when PCR is 1.

SP bit The Save Parameters (SP) bit value of zero indicates that the page parameters not be saved. A value of 1 indicates that the page parameters that are savable be saved after they have been changed.

PC field The Page Control (PC) field defines the type of parameters to be selected. This field must be set to one of the following:

01b Current cumulative values.

11b Default cumulative values.

The PC field set to 11b (and PCR is a don't care) indicates that the Current Cumulative values are set to their default values of 0. If the PC field is set to 01b, and PCR is set to 1, the Current Cumulative values are also set to their default values.

Parameter List Length The Parameter List Length indicates the amount of data to be transferred from the initiator to the target during the DATA OUT phase. A value of zero is acceptable and is not considered an error.

Note: A specified length greater than 0x00FF will result in a *Check Condition Status* being returned. A length that results in log data being truncated will generate a *Check Condition Status*.

If any field 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 (when PCR=1 and Parameter List Length=0).

- Page 01h parameters:
 - Overrun Counter
 - Underrun Counter
- Page 02h parameters:
 - Write Error Counter
 - Write Total Recoverable Error Counter
 - Write Total Unrecoverable Error Counter
- Page 03h parameters:
 - ECC OTF Counter
 - Read Error Counter
 - Read Total Recoverable Error Counter
 - Read Total Unrecoverable Error Counter
- Page 05h parameters:
 - Verify Error Counter
 - Verify Total Posted Recoverable Error Counter
 - Verify Total Posted Unrecoverable Error Counter
- Page 06h parameters:
 - Non-Medium Error Counter
- Page 30h parameters:
 - Zero Seeks counter
 - Seeks \geq to $2/3$ counter
 - Seeks $\geq 1/3$ and $< 2/3$ counter
 - Seeks $\geq 1/6$ and $< 1/3$ counter
 - Seeks $\geq 1/12$ and $< 1/6$ counter
 - Seeks > 0 and $< 1/12$ counter
 - Overrun Counter
 - Underrun Counter
 - Device Cache Full Read Hits
 - Device Cache Partial Read Hits
 - Device Cache Write Hits
 - Device Cache Fast Writes

- Page 35h parameters:
 - Cumulative Cache Full 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.

The following tables show data format for the DATA OUT phase when the Parameter list length is not equal to zero. Only log pages 1, 2, 3, 5, and 6 are acceptable pages for log select data.

1.4.1 Log Select Page 01h

Table 21. Log Select Page 01h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 01h					
1	Reserved = 0							
2 - 3	Page Length							

Table 22. Log Select Page 01h - Parameter Code 0000h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0000h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 02h							
4 - 5	Ignored							

Table 23. Log Select Page 01h - Parameter Code 0001h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0001h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 02h							
4 - 5	Ignored							

1.4.2 Log Select Page 02h

Table 24. Log Select Page 02h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 02h					
1	Reserved = 0							
2 - 3	Page Length							

Table 25. Log Select Page 02h - Parameter Code 0002h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0002h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

Table 26. Log Select Page 02h - Parameter Code 0003h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0003h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

Table 27. Log Select Page 02h - Parameter Code 0006h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0006h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

1.4.3 Log Select Page 03h

Table 28. Log Select Page 03h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 03h					
1	Reserved = 0							
2 - 3	Page Length							

Table 29. Log Select Page 03h - Parameter Code 0000h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0000h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

Table 30. Log Select Page 03h - Parameter Code 0002h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0002h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

Table 31. Log Select Page 03h - Parameter Code 0003h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0003h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

Table 32. Log Select Page 03h - Parameter Code 0006h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0006h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

1.4.4 Log Select Page 05h

Table 33. Log Select Page 05h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 05h					
1	Reserved = 0							
2 - 3	Page Length							

Table 34. Log Select Page 05h - Parameter Code 0000h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0000h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

Table 35. Log Select Page 05h - Parameter Code 0002h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0002h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

Table 36. Log Select Page 05h - Parameter Code 0003h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0003h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

Table 37. Log Select Page 05h - Parameter Code 0006h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0006h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Ignored							

1.4.5 Log Select Page 06h

Table 38. Log Select Page 06h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0				Page Code = 06h			
1	Reserved = 0							
2 - 3	Page Length							
4 - 5	Parameter Code = 0000h							
6	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
7	Parameter Length = 04h							
8 - 11	Ignored							

1.5 Log Sense

Table 39. Log Sense Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 4Dh							
1	Reserved			RSVD = 0			PPC = 0	SP
2	PC = 11b or 01b		Page Code					
3	Reserved = 0							
4	Reserved = 0							
5	(MSB) Parameter Pointer							(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 set to 0 specifies that the Drive not save any log parameters. If set to 1, all page parameters that are savable (those pages denoted by a DS = 0 in the parameter header control byte) are saved.

Note: Vendor Unique pages 30h and 35h are exceptions to this. They contain parameters that have DS = 0, but are not savable.

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 or 11b to specify Default Cumulative values. If the PC field is set to either 00b or 10b, *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 one of the values returned by 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. A value of 0000h indicates all parameters for the select page code will be returned.

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 parameters.

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

All pages except for page 0 include 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.

The Parameter Control field, or 3rd byte of each parameter header, contains several fields.

- The Disable Update (DU) bit set to 0 (1) indicates that the drive updates (disable updates) the log parameter value to reflect all events that should be noted by that parameter. Although a setting of 1 is allowed, this value is ignored by the drive.
- The Disable Save (DS) bit is set to 1 for all non-savable parameters and is set to 0 for all savable parameters.
Note: Vendor Unique pages 30h and 35h are exceptions to this. They contain parameters that have DS = 0, but are not savable.
- The Target Save Disable (TSD) bit is set to 0 which indicates that the drive provides a target-defined method for saving log parameters.
- The Enable Threshold Comparison (ETC) bit is set to 0 which indicates that the drive does not perform comparisons between cumulative and any threshold values.
- The Threshold Met Criteria (TMC) field is not valid because this drive does not perform threshold comparisons. This field is set to 0.
- The List Binary (LBIN) bit is set to 1 for the Vendor Unique pages whose parameters are lists which indicates that the lists are in binary format, not ASCII. This bit is Reserved and set to 0 for all other pages.
- The List Parameter (LP) bit is set to 0 for parameters that are data counters. The LP bit is set to 1 for parameters that are lists.

1.5.2 Log Sense Page 00h

Table 40. Log Sense Page 00h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 0					
1	Reserved = 0							
2 - 3	Page Length = 000Bh (Number of Pages Supported)							
4	First supported page code = 00h							
5	Second supported page code = 01h							
6	Third supported page code = 02h							
7	Fourth supported page code = 03h							
8	Fifth supported page code = 05h							
9	Sixth supported page code = 06h							
10	Seventh supported page code = 30h							
11	Eighth supported page code = 32h							
12	Ninth supported page code = 33h							
13	Tenth supported page code = 35h							
14	Eleventh supported page code = 36h							

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

Requesting this page with Page Code=00h and PC field = 11b will result in a *Check Condition Status* returned with a Sense Key of Illegal Request and Additional Sense Code of Invalid Field in CDB.

1.5.3 Log Sense Page 01h

Table 41. Log Sense Page 01h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 01h					
1	Reserved = 0							
2 - 3	Page Length (<=000Ch)							

Table 42. Log Sense Page 01h - Parameter Code 0000h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0000h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 02h							
4 - 5	Buffer Under-Run Counter							

Table 43. Log Sense Page 01h - Parameter Code 0001h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0001h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 02h							
4 - 5	Buffer Over-Run Counter							

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:

- Read (6)
- Read (10)

Buffer Underrun conditions are detected during the following SCSI commands:

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

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.

1.5.4 Log Sense Page 02h

Table 44. Log Sense Page 02h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 02h					
1	Reserved = 0							
2 - 3	Page Length (<=0018h)							

Table 45. Log Sense Page 02h - Parameter Code 0002h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0002h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Write Error Counter							

Table 46. Log Sense Page 02h - Parameter Code 0003h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0003h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Write Total Recoverable Errors							

Table 47. Log Sense Page 02h - Parameter Code 0006h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0006h							
2	DU	DS = 1	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Write Total Unrecoverable Errors							

This parameter counter is incremented during failed write operations.

The statistics reported by page 02h are not lost when the Drive is powered off or a self-initiated reset occurs, on a SCSI reset or Bus Device Reset message.

1.5.5 Log Sense Page 03h

Table 48. Log Sense Page 03h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 03h					
1	Reserved = 0							
2 - 3	Page Length (<=0020h)							

Table 49. Log Sense Page 03h - Parameter Code 0000h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0000h							
2	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	ECC OTF Counter							

Table 50. Log Sense Page 03h - Parameter Code 0002h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0002h							
2	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Read Error Counter							

Table 51. Log Sense Page 03h - Parameter Code 0003h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0003h							
2	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Read Total Recoverable Errors							

Table 52. Log Sense Page 03h - Parameter Code 0006h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0006h							
2	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Read Total Unrecoverable Errors							

This Parameter Counter is incremented during failed read operations.

The ECC OTF Counter is incremented each time ECC on-the-fly hardware correction is used. The maximum value that it will reach before wrapping to zero is 0x0000FFFF. The counter resets to zero when the Drive is powered on.

The statistics reported by this page are not lost (except ECC OTF which is lost) when the Drive is powered off or a self-initiated reset occurs, on a SCSI reset or Bus Device Reset message.

1.5.6 Log Sense Page 05h

Table 53. Log Sense Page 05h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0			Page Code = 05h				
1	Reserved = 0							
2 - 3	Page Length (<=0020h)							

Table 54. Log Sense Page 05h - Parameter Code 0000h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0000h							
2	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Verify With No Delay Error Counter							

Table 55. Log Sense Page 05h - Parameter Code 0002h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0002h							
2	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Verify Error Counter							

Table 56. Log Sense Page 05h - Parameter Code 0003h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0003h							
2	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Verify Total Posted Recoverable Errors							

Table 57. Log Sense Page 05h - Parameter Code 0006h

Relative Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 1	Parameter Code = 0006h							
2	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
3	Parameter Length = 04h							
4 - 7	Verify Total Posted Unrecoverable Errors							

This parameter counter is incremented during failed verify operations.

The statistics reported by this page are not lost when the Drive is powered off or a self-initiated reset occurs, on a SCSI reset or Bus Device Reset message.

1.5.7 Log Sense Page 06h

Table 58. Log Sense Page 06h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 06h					
1	Reserved = 0							
2 - 3	Page Length = 0008h							
4 - 5	Parameter Code = 0000h							
6	DU	DS = 0	TSD=0	ETC=0	TMC=0		RSVD=0	LP=0
7	Parameter Length = 04h							
8 - 11	Non-Medium Error Counter							

This parameter counter is incremented every time a non-medium error is detected.

The statistics reported by this page are not lost when the Drive is powered off or a self-initiated reset occurs, on a SCSI reset or Bus Device Reset message.

1.5.8 Log Sense Page 30h

Table 59. 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	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
7	Parameter Length = 2Ch							
8 - 9	Zero Seeks							
10 - 11	Seeks > = to 2/3							
12 - 13	Seeks > = 1/3 and < 2/3							
14 - 15	Seeks > = 1/6 and < 1/3							
16 - 17	Seeks > = 0 and < 1/6							
18 - 19	Seeks > 0 and < 1/12							
20 - 23	Reserved = 0							
24 - 25	Overrun Counter							
26 - 27	Underrun Counter							
28 - 31	Device Cache Full Read Hits							
32 - 35	Device Cache Partial Read Hits							
36 - 39	Device Cache Write Hits							
40 - 43	Device Cache Fast Writes							
44 - 51	Reserved = 0							

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

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

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

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 Ultrastar 18ZX/9LZX Hardware Functional or Swordfish/Sawfish Hardware Functional Spec) 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:

- Read (6)
- Read (10)

Buffer Underrun conditions are detected during the following SCSI 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 additional revolutions 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 extra revolutions.

DEVICE CACHE FULL 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 Full Hits on Reads counter on Log Sense, Page 35h.

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 1.5.11 Log Sense Page 35h.

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.

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. 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. Even though the DS field equals zero, the parameters on this page are not saveable.

1.5.9 Log Sense Page 32h

Table 60. Log Sense Page 32h

Byte	BIT								
	7	6	5	4	3	2	1	0	
0	Reserved = 0		Page Code = 32h						
1	Reserved = 0								
2 3	(MSB)		Page Length = $in_0 + \dots + 1n_n$						(LSB)
Log Parameter(s)									
4 $1n_0 +$ 3	Log Parameter 0 (Length $1n_0$)								
-----	-----								
$x - 1n_n$ x	Log Parameter n (Length $2n_n$)								

1.5.9.1 Log Parameter Format - Page 32h

Table 61. 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	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=1	LP=1
3	Parameter Length = 8(m+1)							
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)							

Page 32h is a listing of physical error sites that should be reallocated. (See 4.6, “Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign” 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.

Even though the DS field equals zero, the parameters on this page are not saveable.

Requesting this page with Page Code=00h and PC field = 11b will result in a *Check Condition Status* returned with a Sense Key of Illegal Request and Additional Sense Code of Invalid Field in CDB.

1.5.10 Log Sense Page 33h

Table 62. Log Sense Page 33h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 33h					
1	Reserved = 0							
2 3	(MSB)		Page Length = $1n_0 + \dots + 1n_n$				(LSB)	
Log Parameter(s)								
4 $1n_0 +$ 3	Log Parameter 0 (Length $1n_0$)							
-----	-----							
$x - 1n_n$ x	Log Parameter n (Length $1n_n$)							

1.5.10.1 Log Parameter Format- Page 33h

Table 63. Log Parameter Format - Page 33h

Byte	BIT								
	7	6	5	4	3	2	1	0	
0 1	(MSB)		Parameter Code = 0, 1, ..., n-1, or n						(LSB)
2	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=1	LP=1	
3	Parameter Length = $4(m+1)$								
4 5 6 7	(MSB)		Logical Block Address 0				(LSB)		
-----	-----								
$4m + 4$ $4m + 5$ $4m + 6$ $4m + 7$	(MSB)		Logical Block Address m				(LSB)		

Page 33h is a listing of LBAs that should be reallocated. (See 4.6, “Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign” 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.)

Even though the DS field equals zero, the parameters on this page are not saveable.

Requesting this page with Page Code=00h and PC field = 11b will result in a *Check Condition Status* returned with a Sense Key of Illegal Request and Additional Sense Code of Invalid Field in CDB.

1.5.11 Log Sense Page 35h

Table 64. 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	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
7	Parameter Length = 24h (36)							
8 - 11	Cumulative Cache Full Hits on Reads							
12 - 15	Cumulative Cache Partial Hits on Reads							
16 - 19	Cumulative Cache Misses on Reads							
20 - 43	Reserved = 0							

Page 35h contains information about cache utilization.

The Cache Full Hits parameter is incremented when all of the requested data is in the cache. The Cache Misses parameter is incremented when the first block of the requested data is not in the cache. The Cache Partial Hits parameter is incremented when the first block of the requested data is in the cache and the last block is not.

Implementer's Note: The Initiator may use this information to "fine tune" the caching Mode parameters for the particular application it is using. (See 4.8, "Segmented Caching" 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.

Even though the DS field equals zero, the parameters on this page are not saveable.

1.5.12 Log Sense Page 36h

Table 65. Log Sense Page 36h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Reserved = 0		Page Code = 36h					
1	Reserved = 0							
2 - 3	Page Length = 0006h							
4 - 5	Parameter Code = 0							
6	DU	DS = 0	TSD=0	ETC=0	TMC=0		LBIN=0	LP=0
7	Parameter Length = 02h							
8	Reserved = 0							
9	Current Temperature Measurement							

Page 36h returns the current temperature of the disk enclosure.

The Current Temperature Measurement indicates the hexadecimal value of the drive in degrees C. If a valid reading of the temperature sensor cannot be obtained, the value returned in this field will be 0xFF. No error will be flagged.

1.6 Mode Select (6)

Table 66. Mode Select Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 15h							
1	Reserved			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 logical unit number (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 Page Format (PF) 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 Save Parameter (SP) 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)”).

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 67.

The Initiator should issue a Mode Sense command requesting all changeable values (see PC field in byte two of the CDB, Table 84, for 1.7, “Mode Sense (6)”) 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”).

Each time the Initiator receives a *Unit Attention Condition* (see 4.1.5, “Unit Attention Condition”) 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)” to determine the Target's parameters.

A Mode Select command also causes a *Unit Attention Condition* (see 4..5, “Unit Attention Condition”) 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. Not all of the Mode Select options on all pages are both changeable and saveable (also see 1.7, “Mode Sense (6)”). 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.

Table 67. Mode Select Parameter List - Header

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. **All other fields must be set to zero.** This implies that a "read-modify-write" strategy for the header bytes will not work in the case when the write protect jumper is in place causing a mode sense to return WP = 1, but requiring WP to be zero on the subsequent Mode Select.

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 enters Write Protect mode via a pin on the Option Block or by setting the Control Mode Page 0Ah SWP bit to 1. Refer to 4.14, “Options Jumper Block” for information regarding the Options Jumper Block. Refer to 1.6.10, “Page 0Ah - Control Mode Parameters” for information regarding SWP bit. The WP bit must be cleared to zero before a Mode Select is issued.

When in Write Protect mode, the following commands cause the Target to set *Check Condition Status* with sense indicating Write Protected:

- Format Unit
- Synchronize Cache
- Write (16)
- Write (10)
- Write and Verify
- Write Same
- Write Long

- Reassign Blocks

In addition, automatic rewrites and reallocates are prohibited. Recommend Rewrite/Reallocate is not affected by the Write Protect mode.

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 68. Mode Select Parameter List - Block Descriptor

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

1.6.1.2 Block Descriptor

The Block Descriptor specifies the Number of Blocks (user addressable) to format at the specified Block Length. It can be used to specify the default number of blocks or can specify a smaller number for Initiators that wish to limit the number of blocks on the Drive. The default Number of Blocks is the most blocks which can fit on the Drive at the specified Block Length.

The Initiator specifies the Number of Blocks in one of three ways:

- A value of 00000000h indicates that the number of blocks will not change from the current value if the block descriptor block length remains the same as the current block length. If the block descriptor block length is different from the current block length, the default maximum number of blocks is used.
- A value of FFFFFFFFh indicates that the default maximum number of blocks will be used.
- A value other than 0 or FFFFFFFFh specifies the number of blocks. This value must be less than or equal to the default number of blocks. If the Number of Blocks field specified is larger than can be supported then the Number of Blocks will be truncated to the maximum number allowed.

The density code for direct-access devices is 0.

A Format Unit command is required to cause these parameters to become current only if the block length parameter is different than the current block length. If the Number of Blocks is set to a reduced value and a Format Unit command is issued, the Format Unit will format the entire Drive, but the number of blocks visible to the Initiator will be limited to the specified number.

The Block Length is specified in bytes.

Note: Refer to the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec 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 3h - Format Device Parameters" for details on those values.

Table 69. 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 0h - Vendor Unique Parameters

Table 70. Mode Select Data Format Page 00h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	RSVD= 0	RSVD= 0	Page Code = 00h					
1	Page Length = 0Eh							
2	QPE	Ignored	Ignored	Ignored	Ignored	Ignored	Ignored	ARHES
3	ASDPE	Ignored	CMDAC	RPFAE	DOTF	Ignore	RRNDE	CPE
4	Ignored						DWLRO	DLRO
5	Ignored	Ignored	DSN	FRDD	DPSDP	Ignored	CAEN	Ignored
6	Ignored							
7	Ignored							
8	Ignored	ADC	QEMC	Ignored	LED Mode			
9	Temperature Threshold							
10	Command Aging Time Limit (HI)							
11	Command Aging Time Limit (LO)							
12	QPE Read Threshold							
13	Ignored							
14	DRRT	DNR	Ignore	RARRRD	FFMT	Ignored		
15	RTP	RRC	FCERT	Ignore	DRPDV	DSF	IRT	IVR

Following are parameter options for Page 0 of Mode Select. Refer to Table 70.

Byte 2 contains the following fields:

QPE: The QPE (Qualify Posted Errors) bit allows the Initiator to inhibit the reporting of recovered data errors which are recovered below a set DRP step. A QPE bit of 0 causes the Target to report all recovered data errors. A QPE bit of 1 causes the Target to report only those recovered data errors which exceed the QPE threshold. All error reporting is governed by Mode Page 1 and Mode Page 7 parameters. For example, QPE is ignored when PER is 0.

This bit applies to data errors while reading (Data Sync and ECC errors). The QPE bit has no effect on non-data errors. The user can accept the default reporting thresholds, or can specify the thresholds by using byte 12 of this page. (DRP levels are described in Appendix B.1, “Data Recovery Procedure for Data Field Error.”)

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)
- Verify
- Write and Verify

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

ARHES: The ARHES (Auto-Reallocation of a Hard Error Site) bit allows the target to auto-reallocate a site when a hard read error is encountered. An ARHES bit of 0 disables the auto-reallocate of hard error sites. An ARHES bit of 1 allows the target to auto-reallocate hard error sites that are encountered during the following commands:

- Read (6)
- Read (10)
- Verify

For all other commands the ARHES bit setting is unused and treated as 0. Once a site has been auto-reallocated, the data at the original hard error site is lost and must be restored by the initiator.

Subsequent reads to an auto-reallocated site will result in a hard error with a UEC, key, code, and qualifier of 0x01DD, 3, 80, 00, respectively, until the site has been rewritten.

If an ARHES site is encountered during a normal reassign or auto-reassign operation, it will be treated as a normal hard error site and will prohibit the operation from completing.

ARHES is only allowed if the disk drive's maximum recovery has been attempted on a site. If the drive's recovery is limited in any way through mode page settings, ARHES is disabled.

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.

CMDAC: The CMDAC bit, when enabled by setting LED Mode 0, controls how the Target defines a signal on the LED pin that indicates when the Drive is "active." See the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec for details on the electrical parameters of this LED Pin signal.

- When the CMDAC bit = 1
 - The LED pin goes to a TTL low level (LOW) for a "Command Active" condition when . . .
 1. A CDB (other than Request Sense or Inquiry) 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 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.

RPFAE: 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 1h - Error Recovery Parameters" 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 (ARRR) functions. See 4.6, "Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign."

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

Table 71. Error Reporting Modes

PER	QPE	RPFAE	RARRRD	Reporting of Recovered Errors
0	x	0	x	Recovered errors are not reported.
0	x	1	0	Only recovered errors associated with the PFA functions are reported.
0	x	1	1	Only recovered errors associated with the PFA functions are reported except that recovered errors associated with the Automatic/Recommend Rewrite/Reallocate (ARRR) function are not reported.
1	0	x	x	All recovered errors are reported.
1	1	x	x	Recovered errors that are above the threshold defined by the QPE bit are reported, plus recovered errors that are associated with the PFA (ARRR) functions. See 4.6, "Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign."

Note: See 4.7, "Predictive Failure Analysis" for details concerning PFA.

DOTF: The DOTF (Disable ECC On The Fly) bit indicates if ECC On The Fly is enabled. This bit is Status Only. If DOTF is not set then ECC On The Fly correction is enabled. If DOTF is set then ECC On The Fly correction is disabled and Software ECC correction will be used in place of ECC On The Fly correction.

RRNDE: The RRNDE (Report Recovered Non Data Errors) bit controls the reporting of recovered Non Data Errors when the PER bit is set. If RRNDE is set then recovered Non Data Errors are reported. If the RRNDE bit is not set then recovered Non Data Errors are not reported.

CPE: The CPE (concurrent processing enable) bit controls whether or not I/O processes 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”).

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”).

Concurrent Processing is enabled if CPE = 1.
Concurrent Processing is disabled if CPE = 0.

Byte 4 contains the following field:

DLRO: (Disable Latency Reduction Optimization). The file normally attempts to begin reading data before the servo has completely settled on track in an attempt to reduce the apparent read seek time, or equivalently to reduce latency (with early read some blocks will be read on arrival which normally would have required a full revolution of latency). The DLRO bit, when set, causes the servo to wait until the head has settled on track before allowing a read to begin and thus reduces the performance of the drive.

DWLRO: (Disable Write Latency Reduction Optimization). The file normally attempts to begin writing data before the servo has completely settled on track in an attempt to reduce the apparent write seek time, or equivalently to reduce latency (with early writes some blocks will be written on arrival which normally would have required a full revolution of latency). The DWLRO bit, when set, causes the servo to wait until the head has settled on track before allowing a write to begin and thus reduces the performance of the drive.

Byte 5 contains the following fields:

DSN: The Disable Target Initiated Synchronous Negotiation bit indicates whether the Drive will perform Target Initiated Synchronous Negotiation and Target Initiated Transfer Width Negotiation. DSN = 1 indicates the Drive will not perform Target Initiated Synchronous and Transfer Width Negotiation, while DSN = 0 indicates that the Drive will.

Note: The DSN bit is an indicator bit only. Changing this bit does not have any functional effect on the Drive. The actual function is controlled by an Option Jumper as described in the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec.

FRDD: A FRDD (Format and Reassign Degraded Disable) bit of 1 prevents the Drive from reporting Format or Reassign degraded on a Test Unit Ready Command, and causes media access commands (i.e., read, write) to report a media error if degraded. A FRDD bit of 0 indicates that Format or Reassign degraded is reported for the Test Unit Ready command.

DPSDP: The Data Phase Save Data Pointer bit controls whether the Drive 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.
- 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.

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 Modifier (see 1.6.10, “Page 0Ah - Control Mode Parameters”) still apply in this case. Data integrity is maintained when Restricted Reordering is in effect.

If the CAEN bit is equal to 0 and the Command Aging Time Limit (bytes 10-11) is non-zero, the Command Aging Time Limit field is treated as a Command Aging Time Out limit.

If the CAEN bit is equal to 0 and the Command Aging Time Limit (bytes 10-11) is 0, then Command Aging is not enabled.

Byte 6 and Byte 7 are ignored.

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

ADC: The ADC (Adaptive Caching) affects the Target's caching algorithm. If the ADC bit is set to one, the Target alters the Demand Read Retention Priority and Maximum Prefetch based on perceived workloads, in an attempt to maximize the number of cache hits and minimize command overhead.

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 1.6.10, “Page 0Ah - Control Mode Parameters” for a description on the how QEMC effects QErr.

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 or in the queue, 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/queued OR when the Target is in any degraded mode, the LED is high.

LED Mode = 5h (Hardware fault).

When a hardware error is detected by the target the LED goes high.

LED Mode = 6h-Fh (Reserved for future use).

Byte 9 specifies the threshold value in degrees Celsius for the thermal sensor warning message. The temperature threshold can be adjusted in the range: 6 degrees C through 65 degrees C. A value of 0 selects the default threshold which depends on which electronics card is installed on the drive. All other values above or below the adjustable range select the range limit to which they are closest.

Note: The casting temperature may be different than that reported by the sensor due to the sensor being located on the electronics card rather than on the casting.

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

Byte 12 specifies the error reporting threshold for read operations (including Write and Verify) when the QPE bit is set. A value of 0 or 0ffh indicates that the Drive default value shall be used. Any other value is interpreted as the threshold. A recovered error which requires more steps of recovery than the threshold will be reported. (Refer to the Ultrastar 18ZX/9LZX Hardware Functional Spec., section on Data Reliability.)

Byte 13 is ignored (any input value accepted).

Byte 14 contains the following fields:

DRRT: The DRRT (Disable Restore Reassign Target) bit Disables the reading and restoration of the target LBA during a Reassign Blocks command. If the DRRT bit is zero, the reassign command attempts to restore the target LBA's data. If the data cannot be restored, the target LBA is reassigned and written with a data pattern of all FF's. If the DRRT bit is one, no attempt is made to restore the target LBA.

DNR: The DNR (Disable Nested Reassign) bit disables nested reassigns. If the DNR bit is zero, nested reassigns are performed. If the DNR bit is one, Nested reassigns are not done.

RARRRD: The RARRRD (Report Automatic/Recommend Rewrite/Reallocate Disabled) bit controls reporting sense data for recovered errors associated with this function when the RPF AE bit is set. If the RARRRD bit is set then recovered errors associated with the ARRR function are not reported. If the RARRRD bit is not set then recovered errors associated with the ARRR function are reported.

FFMT: The Fast Format Enable (FFMT) bit allows formatting this drive without any writes to the customer media. All format operations are allowed including changing block sizes and manipulating defects. The file will operate normally after a fast format with the following caveat:

*Since no data is written to any customer data blocks as a result of a Fast Format operation, there is a possibility that a read attempt to any particular block (without having previously written to that block) will result in an unrecoverable data error. This will most assuredly happen if the block size is changed as every LBA will contain data of an incorrect length and, apparently, incorrect starting point. It is also possible to generate an uncorrectable data error without changing block sizes if the defect list is shortened and previously bad blocks become visible in the customer address space. Of course **ALL DATA ON THE FILE WILL BE LOST** as the result of any format operation and so any attempt to read blocks which have not been written to will result in unpredictable behavior.*

Byte 15

RTP: The RTP (Reassign Target Pad) bit determines whether the reassign target sector is padded with 00h or FFh. If the RTP bit is set to 1 then the target sector is padded with 00h. If the RTP bit is set to 0 then the target sector is padded with FFh.

RRC: The RRC (Read Retry Count) bit determines how the retry count on page 1 byte 3 is used. If the page 0 RRC bit is 0 and page 1 read retry count is 0 then no recovery is performed. If the page 0 RRC bit is 0 and page 1 read retry count is 1 then maximum recovery is performed. If the page 0 RRC bit is 1 then the page 1 read retry count is the maximum level of recovery performed.

FCERT: The FCERT (Format Certification) bit determines whether the certification step will be performed during a Format Unit command. A FCERT bit set to 0 disables certification. A FCERT bit set to 1 enables the certification step.

DRPDV: Disable Reassign Blocks Post Move Verify (DRPDV) set to zero allows the file to perform a verify operation on all blocks moved as a result of performing a reassign blocks operation. A verify failure during this time will cause the file to go *Reassign Degraded* (see 4.1.9, “Degraded Mode”). DRPDV set to one (1) will inhibit this verify operation. *Setting DRPDV to one may result in unrecoverable blocks being written which will not be reported, and the drive will not go reassign degraded.* This is generally acceptable in environments where system level redundancy can recover the lost data and it is desirable to keep the drive on line.

DSF: Discard Surface Analysis Results after Format (DSF) set to zero will allow defects discovered during the format operation to be added to the GList. A new GList will be created to hold these entries if the format is being performed without the old GList. DSF = 1 will result in no new entries being added to the GList and no new GList will be created if the file is being formatted without a GList. This may result in known bad sites becoming visible again.

IRT: The IRT (Integration Reassign Threshold) bit determines the threshold used to recommend reassignment or initiate auto-reassign. When IRT is disabled with a value set to 0, the thresholds used to determine when a sector should be reassigned and either recommended or auto-reassigned, depending upon state of the ARRE and AWRE bits, are set to values optimized for end user environments. When enabled, by setting to 1, thresholds optimized for integrating the drive into a system to minimize end-user field failures is used.

IVR: The IVR (Integration Verify Recovery) bit determines the maximum number of steps attempted during a Verify Recovery Procedure (VRP) as defined in B.1.2, “Verify Commands.” When disabled with a value set to 0, all of the steps defined in B.1.2, “Verify Commands” are attempted, if required. When enabled, by setting to 1, a maximum number of steps optimized for integrating the drive into a system to minimize end-user field failures is used.

1.6.3 Page 1h - Error Recovery Parameters

Table 72. Mode Select Data Format Page 01h

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							
5	Head Offset Count							
6	Data Strobe Offset Count							
7	Reserved = 0							
8	Ignored							
9	Reserved = 0							
10	(MSB)	Recovery Time Limit						
11		(LSB)						

The following are parameter options for Page 1 of Mode Select. Refer to Table 72.

Byte 2 contains the following fields:

AWRE: (Automatic Write Reallocation Enable) = 1, enables the automatic reallocation of defective blocks found during the verify portion of a Write and Verify Operation. This automatic reallocation will only be done on Hard Error sites. Only one hard error site may be automatically reallocated during each Write and Verify Operation. The Write and Verify Operation will fail if multiple hard errors are encountered. When AWRE = 0, the target does not perform automatic reallocation of defective blocks during Write and Verify Operations.

The AWRE bit setting is only used by the Target during Write and Verify Commands. 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 4.6, “Automatic Rewrite/Reallocate-Recommend Rewrite/Reassign” 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).

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

Note: See the description of the RPFAE bit in the section 1.6.2, "Page 0h - Vendor Unique Parameters" 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.
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.") 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.") 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).
1	0	1	0		The highest error level is reported at the end of transfer, (see B.3, "Priority of Error Reporting.") 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.") 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.”) 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.
1	1	0	1	The highest error level is reported at the end of transfer, (see B.3, “Priority of Error Reporting.”) 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.”) 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.”) 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).

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 Appendix B.1, “Data Recovery Procedure for Data Field Errors” 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. This description of Read Retry Count depends on the RRC bit (see 1.6.2, “Page 0h - Vendor Unique Parameters”) being set to zero (0). If the page 0 RRC bit is set to one (1) then the Read Retry Count is the maximum level of recovery steps performed (see Appendix B).

The **Correction Span** is ignored.

The **Head Offset Count** is ignored.

The **Data Strobe Offset Count** is ignored.

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. Due to the internal clocking inherent in this product, the time limit will have a granularity of 50 milliseconds which may change in this or future products. The Recovery Time Limit is used by the Target for 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

For all other commands, the Recovery Time Limit is unused and the target uses the default value of no time limit.

1.6.4 Page 2h - Disconnect/Reconnect Parameters

Table 73. Mode Select Data Format Page 02h

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	Ignored							
5	Ignored							
6	(MSB)	Disconnect Time Limit = 0						
7		(LSB)						
8	(MSB)	Connect Time Limit = 0						
9		(LSB)						
10	(MSB)	Maximum Burst Size						
11		(LSB)						
12	EMDP=0	Reserved = 0			DIMM	RSVD = 0	DTDC	
13	Reserved = 0							
14	Reserved = 0							
15	Reserved = 0							

Following are parameter options for Page 2 of Mode Select. Refer to Table 73.

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). Both the Read Buffer Full Ratio and the Write Buffer Empty Ratio pertain to the current active notch. For each active notch, as defined in page 0Ch, there are separate Read Buffer Full Ratios and Write Buffer Empty Ratios. When the active notch is zero, the values are applied across all notches.

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 ULTRASTAR 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec for the data rate values.)

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 **Disconnect Time Limit** is the minimum time in 100 microsecond 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 microsecond 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 **Enable Modify Data Pointers (EMDP)** bit set to 0 indicates the target will not issue the modify data pointers message.

The **Disconnect Immediate (DIMM)** bit allows the Initiator to control whether the Target is required to disconnect following the receipt of a command and prior to starting a data phase.

When the DIMM is set to 1, the Target disconnects from the SCSI Bus following the receipt of a command and prior to starting a Data In or Data Out phase.

Note: Priority commands do not disconnect from the SCSI bus (see 4.2, “Priority Commands”).

For Write (6), Write (10), Write and Verify, and Write Same commands (opcodes 0Ah, 2Ah, 2Eh, 41h), DIMM set to 0 indicates that the Target is not allowed to disconnect from the SCSI bus after receiving the 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.

For Read (6) and Read (10) commands (opcodes 08h and 28h), DIMM set to 0 prevents the Target from disconnecting from the SCSI Bus 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 disconnects until enough data has been read from the disk to satisfy the reconnection criteria as determined by the Read Buffer Full Ratio.

All other commands with DIMM set to 0 may or may not disconnect before a data phase is started.

Note: Disconnection is based on the Target's internal control algorithms, the setting of the disconnect privilege bit in the previous Identify message, and other parameters in this mode page.

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* shall 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 3h - Format Device Parameters

Table 74. Mode Select Data Format Page 03h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	RSVD = 0	RSVD = 0	Page Code = 03h					
1	Page Length = 16h (22)							
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 (or 0000h) (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							

The format device page contains parameters which specify the medium format. Sectors per Track, Interleave, Track Skew Factor, and Cylinder skew Factor fields 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. 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. This field is a function of the device Model Number.

Note: The Mode Select command accepts any value in this field but the entered value is ignored.

The **Alternate Sectors per Zone** is the average number of alternate sectors per zone of the active notch.

Note: The Mode Select command accepts any value in this field but the entered value is ignored.

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 and active notch. For more information see the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec.

A value of zero is also 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. For further information on allowed sector sizes, see the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec.

Note: The Mode Select command accepts any value in this field but the entered value is ignored.

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 and active notch as defined in the Ultrastar 18ZX/9LZX or Ultrastar 36XP/18LP Hardware Functional Spec. A value of zero is also accepted in this field.

Cylinder Skew Factor indicates the number of physical sectors between the last logical block of one cylinder and the first logical block on the next cylinder. The value is a function of the current Block Length and active notch as defined in the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec.

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 75. Mode Select Data Format Page 04h

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 (LSB)							
15								
16								
17	Reserved = 0							
18	Reserved = 0							
19	Reserved = 0							
20	(MSB) Medium Rotation Rate (LSB)							
21								
22	Reserved = 0							
23	Reserved = 0							

This page describes some physical attributes of the Drive.

Number of Cylinders defines the number of physical cylinders used for customer data storage. This field is a function of the device Model Number. See the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec for more information on the number of cylinders. 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 Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec. 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 outermost 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** 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 Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec for the maximum cylinder value.

The **Medium Rotation Rate** indicates the spindle speed in revolutions per minute. A value of zero is also accepted in this field.

1.6.7 Page 7h - Error Recovery Parameters for Verify

Table 76. Mode Select Data Format Page 07h

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								
4	Verify Correction Span = 48h								
5	Reserved = 0								
6	Reserved = 0								
7	Reserved = 0								
8	Reserved = 0								
9	Reserved = 0								
10	(MSB)	Verify Recovery Time Limit							
11		(LSB)							

Following are parameter options for Page 7 of Mode Select. Refer to Table 76.

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 4-9 are not changeable, the Mode Select Command accepts only the values indicated for bytes 4 - 9.

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 Appendix B.1.2, “Verify Commands.” Also see sections on 1.33, “Verify” and 1.36, “Write and Verify” 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” for a description of one VRP pass.) A non-zero Verify Retry Count 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.

Note: The Mode Select command accepts any value in this field but the entered value is ignored.

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 of no time limit. Due to the internal clocking inherent in this product, the time limit will have a granularity of 50 milliseconds which may change in this or future products.

1.6.8 Page 8h - Caching Parameters

Table 77. Mode Select Data Format Page 08h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	RSVD = 0	RSVD = 0	Page Code = 08h					
1	Page Length = 12h							
2	IC	ABPF	CAP	DISC	SIZE	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							
7	(LSB)							
8	(MSB) Maximum Pre-fetch							
9	(LSB)							
10	(MSB) Maximum Pre-fetch Ceiling							
11	(LSB)							
12	FSW	LBCSS	DRA	Reserved = 0				
13	Number of Cache Segments							
14	(MSB) Cache Segment Size							
15	(LSB)							
16	Reserved = 0							
17	(MSB) Non-Cache Segment Size							
18								
19	(LSB)							

Page 8 parameters are used for defining the use of the cache. See 4.8, “Segmented Caching” for a more detailed description of cache support.

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:

IC:

An IC (Initiator Control) bit of 1 specifies that the Target will use the Number of Cache Segments or Cache Segment Size fields, dependent on the Size bit, to control the cache segmentation. An Initiator Control bit of 0 specifies that the target use its own algorithm to control the cache segmentation. The target's algorithm behaves exactly as if this bit is set = 1, unless the Vendor Unique Mode Page 0 ADC bit is set = 1. When IC = 0 and ADC = 1, an adaptive caching algorithm overrides both the Number of Cache Segments and Cache Segment Size fields.

ABPF:

The ABPF (Abort Pre-Fetch) bit of 1 and DRA bit of 0 requests that the target abort the pre-fetch upon selection. The ABPF set to one takes precedence over the Minimum Pre-fetch bytes. When set to zero, with the DRA bit equal to zero, the termination of any active pre-fetch is dependent upon Caching Page bytes 4 through 11.

CAP:

The Caching Analysis Permitted (CAP) bit, when set to one, requests that the target perform caching analysis during subsequent operations. When set to zero, CAP requests that caching analysis be disabled to reduce overhead time or to prevent non pertinent operations from impacting tuning values. Even though the target supports initiators setting CAP = 0, caching related Log parameters will be maintained due to the negligible effects of updating them when CAP is set = 0.

DISC:

The Discontinuity (DISC) bit, when set to one, requests that the target continue the pre-fetch across time discontinuities, such as across cylinders and tracks, up to the limits of the space available for pre-fetch. When set to zero, the DISC requests that the pre-fetches be truncated (or wrapped) at time discontinuities. Even though the target supports initiators setting DISC = 0, prefetches across track boundaries may be executed dependent upon other prefetch Mode parameters, when DISC is set = 0.

SIZE:

The Size Enable (SIZE) bit, when set to one, indicates that the Cache Segment Size is to be used to control caching segmentation. When SIZE equals zero, the initiator requests that the Number of Cache Segments is to be used to control caching segmentation. The exception to this is when IC = 0 and the Vendor Unique Mode Page 0h ADC parameter is set = 1. In that case, the target determines cache segmentation. The value sent in the field not used to determine segmentation is ignored. When read with a Mode Sense command, both values reflect actual values, regardless of the previously selected value in the unused parameter.

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. (Also see 4.4.1, “Back to Back Writes” and 4.8.4, “Write Cache” for more information.)

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

MF:

The MF (Multiplication Factor) bit determines how the Maximum and Minimum Pre-fetch parameters are interpreted. If this bit is equal to 0, the parameters are used as is. If the bit is equal to 1, the parameters are 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 4.8, “Segmented Caching”). 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 Fh as defined below:

Value	Definition
0h	Do not distinguish between Requested Data and Other Data.
1h	Replace Requested Data before Other Data.
Fh	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 operation such as Pre-fetch, Read-Ahead, etc.

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 (see 4.8.8, “Segment Prioritization”). It may be set to 0h, 1h, or Fh. 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. A value in the range 0001h-FFFEh is the number of blocks prefetched following a read operation that will not be preempted by a subsequent command. The actual number of blocks prefetched without interruption may be decreased by other reasons such as space in the cache segment, maximum prefetch, and the end of the media. The value of FFFFh indicates that the drive self-adapts the minimum prefetch value. The adaptive minimum prefetch algorithm uses the detected workload seen by the drive to optimize throughput and response time for that workload.

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 Read Ahead just like Maximum Pre-fetch does when the MF bit is set to 0. But when MF is set to 1, it also limits Read Ahead to an absolute value, unlike Maximum Prefetch which uses a value scaled by the transfer length when MF = 1.

Byte 12 contains the following fields:

FSW:

The Force Sequential Write (FSW) bit when set to one, indicates that multiple block writes are to be transferred over the SCSI bus and written to the media in an ascending, sequential, logical block order. When the FSW bit equals zero, the target is allowed to reorder the sequence of writing addressed logical blocks in order to achieve faster command completion.

LBCSS:

The Logical Block Cache Segment Size (LBCSS) bit when set to one, indicates that the Cache Segment Size field units shall be interpreted as logical blocks. When the LBCSS bit equals zero the Cache Segment Size field units shall be interpreted as bytes.

DRA:

The Disable Read-Ahead (DRA) bit, when set to one, requests that the target not read into the buffer any logical block beyond the addressed logical block(s). When the DRA bit equals zero, the target may continue to read logical blocks into the buffer beyond the addressed logical block(s).

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 cache segments supported by this Drive. The number of cache segments supported by this Drive is 1-62. This parameter is valid only when the SIZE bit is zero. It is ignored when SIZE is a one.

The total cache space is divided among equally sized cache segments. (See the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec for a description of the total cache and individual cache segment sizes.)

Note: Mode Select commands that transfer Page 8h cause the entire cache to be emptied (see 4.8, “Segmented Caching”).

The **Cache Segment Size** field indicates the requested segment size in Bytes or Blocks, depending upon the value of the LBCSS bit. The Cache Segment Size field is valid only when the SIZE bit is one. It is ignored when SIZE is a zero. The drive considers this a minimum size. It may be rounded up in order to make best use of available cache space.

If the **Non-Cache Segment Size** field is greater than zero, this field advises the target how many bytes the initiator requests that the target allocate for buffer function when all other cache segments are occupied by data to be retained. If the number is at least one, caching functions in the other segments need not be impacted by cache misses to perform the SCSI buffer function. The size of the non cache portion of the buffer is fixed thus the target ignores any value in this parameter.

1.6.9 Page 09h - Peripheral Device Parameters

Table 78. Mode Select Data Format Page 09h

Byte	BIT								
	7	6	5	4	3	2	1	0	
0	RSVD=0	RSVD=0	Page Code = 09h						
1	Page Length = 0Eh								
2	(MSB) Interface Identifier = 0000h (LSB)								
3									
4	Reserved = 0								
5	Reserved = 0								
6	Reserved = 0								
7	Reserved = 0								
8	Ignored								
9	Ignored								
10	Ignored								
11	Ignored								
12	Ignored								
13	Ignored								
14	Reselection Retry Count								
15	Ignored				FAST20	FAST10	Ignored	FORCE8	

The Peripheral Device Page (Table 78) controls vendor-specific features that are not defined in ANSI specifications.

Reselection Retry Count: This byte determines the number of times the Target will attempt to reselect after a reselection timeout has occurred. A value of 0 means no Reselection retry will be attempted. When set to FFh the Target will attempt to retry Reselection forever.

FAST20 set to a one forces SDTR transfer negotiations at a transfer period of 0Ch. If set to zero the Target will negotiate at the fastest speed it is capable of.

FAST10 set to a one forces SDTR transfer negotiations at a transfer period of 19h. If set to zero the Target will negotiate at the speed determined by the FAST20 bit.

Note: The FAST10 and FAST20 bits are maximum transfer rates. If either of these bits are set, the target will not renegotiate SDTR if it is currently operating at a slower transfer rate.

FORCE8 set to a one prohibits the drive from negotiating for wide data transfers, even if the drive is capable of wide transfers. When set to zero, the drive negotiates for the widest data transfer it is capable of.

1.6.10 Page 0Ah - Control Mode Parameters

Table 79. 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 = 0Ah							
2	Reserved = 0						GLTSD = 0	RLEC = 0
3	Queue Algorithm Modifier				Reserved = 0		QErr	DQue
4	RSVD = 0	RAC = 0	Reserved = 0	SWP	RAERP = 0	UAAERP = 0	EAERP = 0	
5	Reserved = 0							
6	(MSB) Ready AEN Holdoff Period (LSB)							
7								
8	(MSB) Busy Timeout Period (LSB)							
9								
10	Reserved = 0							
11	Reserved = 0							

The Control Mode Page (Table 79) provides controls over several features. The features are tagged queuing (see 4.3.2, “Tagged Queuing”), extended contingent allegiance, asynchronous event notification, and error logging (see 4.5, “Error Logs”).

GLTSD (Global Logging Target Save Disable) bit of 0 allows the target to provide a target-defined method for saving log parameters.

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 reordering commands that are tagged with the SIMPLE QUEUE TAG message. The value of zero in this field indicates that the Drive 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 (1h) 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 (8h) specifies that no reordering be done.

A value of nine (9h) specifies that the numeric value of the Queue Tag received with a Simple Queue Tag message determines the order of execution. The larger the value of the queue tag, the earlier the command will be scheduled for execution. The Head of Queue Tag and Ordered Queue Tag messages are still honored.

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 ten (Ah) has the same effect as a value of (1h).

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 Target has returned *Check Condition Status*, continue execution in a normal manner when the pending sense data is 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 0h - Vendor Unique Parameters”) 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 Condition* will be generated for each Initiator that had an 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 is rejected with a MESSAGE REJECT message and the I/O process is executed as an untagged command.

RAC (Report A Check) bit of 0 specifies that long busy conditions (e.g. busy conditions during auto contingency allegiance) may be reported.

A **SWP (Software Write Protect)** bit of 1 specifies that the logical unit shall inhibit writing to the medium after writing all cached or buffered write data. When SWP is one, all commands requiring writes to the medium shall return *Check Condition Status* and shall set the Sense Key to DATA PROTECT and the Additional Sense Code to WRITE PROTECTED. When SWP is one the WP bit shall be set to one for subsequent MODE SENSE commands. A SWP bit of zero specifies that the logical unit may allow writing to the medium if Write Protect mode has not been selected by a pin on the Option Block. Refer to 4.14, “Options Jumper Block” for information regarding the Options Jumper Block. When the SWP bit is 0 the WP bit shall be set to 0 unless Write Protect mode has been selected by a pin on the Option Block. For a list of the commands affected by the SWP bit and details of the WP bit, refer to 1.6.1.1, “Header” section.

The **RAERP**, **UAAERP**, and **EAERP** bits enable specific events to be reported via the asynchronous event reporting 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. Since the drive never returns any asynchronous events, any value is accepted.

Busy Timeout Period specifies the maximum time in milliseconds that the Initiator allows for the Target to remain busy for unanticipated conditions which are not a routine part of commands from the initiator. This field is ignored by the drive. Thus all values are accepted.

1.6.11 Page 0Ch - Notch Parameters

Table 80. Mode Select Data Format Page 0Ch

Byte	BIT							
	7	6	5	4	3	2	1	0
0	RSVD = 0	RSVD = 0	Page Code = 0Ch					
1	Page Length = 16h							
2	ND = 1	LPN = 0	Reserved = 00h					
3	Reserved = 0							
4 5	(MSB)	Maximum number of notches						(LSB)
6 7	(MSB)	Active notch						(LSB)
8 11	(MSB)	Starting Boundary						(LSB)
12 15	(MSB)	Ending Boundary						(LSB)
16 23	(MSB)	Pages Notched = 000000000000100Ch						(LSB)

The notch page contains parameters for direct-access devices which implement a variable number of blocks per cylinder. Each section of the logical unit with a different number of blocks per cylinder is referred to as a notch. The only field that is changeable is the "Active notch" field.

"**ND**" = **1** meaning that this drive is a notched drive.

"**LPN**" = **0** meaning that the notches are based upon physical parameters of the drive (cylinder #), not logical parameters.

"**Maximum Number of Notches**" is the number of notches the drive can support. The value returned is model dependent.

"**Active Notch**" indicates which notch subsequent Mode Select/Sense command parameters pertain to.

A value of 0 is used for parameter values which apply to all notches. Values greater than 0 specify the notch number, where notch 1 is the outer most notch.

The following mode parameters are based on the current active notch:

- Page 02h
 - Read Buffer Empty Ratio.
 - Write Buffer Full Ratio.
- Page 03h
 - Alternate Sectors Per Zone.
 - Sectors Per Track.
 - Track Skew Factor.
 - Cylinder Skew Factor.

"Starting Boundary" contains the first physical location of the active notch. The first three bytes are the cylinder number and the last byte is the head. The value sent in this field is ignored.

"Ending Boundary" contains the last physical location of the active notch. The first three bytes are the cylinder number and the last byte is the head. The value sent in this field is ignored.

"Pages Notched" is a bit map of the mode page codes that indicates which pages contain parameters that may be different for different notches. The most significant bit of this field corresponds to page code 3Fh and the least significant bit corresponds to page code 00h. If a bit is one, then the corresponding mode page contains parameters that may be different for different notches. If the bit is zero, then the corresponding mode page contains parameters that are constant for all notches.

1.6.12 Page 1Ah - Power Conditions

Table 81. Mode Select Data Format Page 1Ah

Byte	BIT							
	7	6	5	4	3	2	1	0
0	RSVD = 0	RSVD = 0	Page Code = 1Ah					
1	Page Length = 0Ah							
2	Reserved = 00h							
3	Reserved = 0						Idle	Standby
4 - 7	(MSB)	Idle Condition Timer						(LSB)
8 - 11	(MSB)	Standby Condition Timer						(LSB)

All values are for the two timers and their corresponding control bits are accepted. Despite this, the drive does not alter the internal power saving algorithms based upon those values.

1.6.13 Page 1Ch - Informational Exceptions Control

Table 82. Mode Select Data Format Page 1Ch

Byte	BIT							
	7	6	5	4	3	2	1	0
0	RSVD = 0	RSVD = 0	Page Code = 1Ch					
1	Page Length = 0Ah							
2	PERF	RSVD = 0	EWASC	DExcpt	Test	RSVD = 0	LogErr	
3	Reserved = 0				MRIE			
4 - 7	(MSB) Interval Timer							(LSB)
8 - 11	(MSB) Report Count							(LSB)

The Informational exceptions control page defines the methods used by the target to control the reporting and the operations of specific informational exceptions conditions. This page shall only apply to informational exceptions that report an Additional Sense Code of FAILURE PREDICTION THRESHOLD EXCEEDED or WARNING to the initiator.

Informational exception conditions occur as the result of vendor specific events within the target. An informational exception condition may occur asynchronous to any commands issued by an initiator.

A PERFORMANCE bit (PERF) of zero indicates that informational exception operations that are the cause of delays are acceptable. A performance bit of one indicates that the target shall not cause delays while doing informational exception operations. A performance bit set to one may cause the target to disable some or all informational exception operations, thereby limiting the reporting of informational exception conditions.

An Enable Warning ASC bit (EWASC) of zero indicates that PFA warning messages will not be reported. An EWASC bit of 1 allows PFA warning messages to be reported.

A Disable Exception control bit (DExcpt) of zero indicates that informational exception operations shall be enabled. The reporting of informational exception conditions when the DExcpt bit is set to zero is determined from the Method of Reporting Informational Exceptions field.

Note: Disable Exception control (DExcpt) bit of zero indicates the failure prediction threshold exceeded reporting shall be enabled. The method for reporting the failure prediction threshold exceeded when the DExcpt bit is set to zero is determined from the Method of Reporting Informational Exceptions field. A DExcpt bit of one indicates the target shall disable reporting of the failure prediction threshold exceeded. The Method of Reporting Informational Exceptions field is ignored when DExcpt is set to one and EWASC is set to zero.

An Enable Warning ASC bit (EWASC) of zero indicates the target shall disable reporting of the warning. The Method of Reporting Informational Exceptions field is ignored when DExcpt is set to one and EWASC is set to zero. An EWASC bit of one indicates warning reporting shall be enabled. The method for reporting the warning when the EWASC bit is set to one is determined from the Method of Reporting Informational Exceptions field.

A Test bit (Test) of one instructs the target to generate an informational exception test report at each interval time (if DExcpt = 0). The condition will be reported with an Additional Sense Code of FFh. A Test bit of zero instructs the target to stop generating informational exception test reports.

The log errors bit (LogErr) of either one or zero indicates the target shall log informational exception conditions.

The Method of Reporting Informational Exceptions field (MRIE) specifies the method that should be used by the target to report informational exception conditions (see Table 83.)

The Interval Timer field indicates the period in 100 millisecond increments for reporting that an information exception condition (a PFA event is an example of an informational exception condition) has occurred. The file will not report informational exception conditions more frequently than the time specified by the Interval Timer field and as soon as possible after the timer interval has elapsed. After the informational exception condition has been reported the interval timer shall be restarted. The default setting for the Interval Timer field is FFFFFFFFh. When this default is set, the interval timer is disabled and causes the file to behave as though the interval timer were equal to 0. The interval timer will not be maintained across power cycles.

The Report Count field indicates the number of times to report an informational exception condition. A value of zero in the Report Count field indicates there is no limit on the number of times the device server reports an informational exception condition. The internal report count will not be maintained across power cycles.

Table 83. Method of Reporting Informational Exception Fields

MRIE	Description
0h	No reporting of informational exception conditions.
2h	Generate <i>Unit Attention</i> : This method instructs the target to report informational exception conditions by returning a <i>Check Condition Status</i> on any command. The Sense Key shall be set to <i>Unit Attention</i> and the Additional Sense Code shall indicate the cause of the informational exception condition. The command that has the <i>Check Condition Status</i> shall not be executed before the informational exception condition is reported.
3h	Conditionally generate recovered error: This method instructs the target to report informational exception conditions, dependent on the value of the PER bit of the error recovery parameters mode page, by returning a <i>Check Condition Status</i> on successful Read or Write commands. The Sense Key shall be set to RECOVERED ERROR and the Additional Sense Code shall indicate the cause of the informational exception condition.
4h	Unconditionally generate recovered error: This method instructs the target to report informational exception conditions, regardless of the value of the PER bit of the error recovery parameters mode page, by returning a <i>Check Condition Status</i> on successful Read or Write commands. The Sense Key shall be set to RECOVERED ERROR and the Additional Sense Code shall indicate the cause of the informational exception condition.
6h	Only report informational exception condition on request: This method instructs the target to preserve the informational exception(s) information. To find out about informational exception conditions the initiator polls the target by issuing an unsolicited REQUEST SENSE command. The Sense Key shall be set to NO SENSE and the Additional Sense Code shall indicate the cause of the informational exception condition.
Fh	Conditionally generate recovered error: This method instructs the target to report informational exception conditions, dependent on the values of the PER bit of the error recovery parameters mode page and the RPFAE, RARRRD bits of the vendor unique parameters mode page, by returning a <i>Check Condition Status</i> on successful Read or Write commands. The Sense Key shall be set to RECOVERED ERROR and the Additional Sense Code shall indicate the cause of the informational exception condition. For information on RPFAE, RARRRD bits see 1.6.2, “Page 0h - Vendor Unique Parameters.”

1.7 Mode Sense (6)

Table 84. 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.

If the DBD (Disable Block Descriptor) bit is zero, the target will return the Block Descriptor. If the DBD bit is set to 1, the target will not return the Block Descriptor.

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

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" 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 table depicts the format of the Mode Sense data available from the Target.

Table 85. Mode Sense Data Format - Header

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							

Table 86. Mode Sense Data Format - Block Descriptor

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

See 1.6.1.1, “Header” section 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 4.8, “Segmented Caching” 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 specifies the number of blocks currently available to the Initiator. It is a function of the Number of Blocks specified by the Mode Sense command and the current block length. Refer to the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec for more information about capacity.

If the Number of Blocks is FFFFFFFFh then the number of blocks is greater than FFFFFFFEh. In this case the Read Capacity command may be used to determine the actual number of blocks for the device. The Block Length field is dependent on how the media is currently formatted.

1.7.2 Mode Sense Data Format (of All Pages)

Table 87. Mode Sense Data Format Page 00h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	PS = 1	RSVD=0	Page Code = 0					
1	Page Length = 0Eh							
2	QPE	Ignored	Ignored	Ignored	Ignored	Ignored	Ignored	ARHES
3	ASDPE	Ignored	CMDAC	RPFAE	DOTF	Ignored	RRNDE	CPE
4	Ignored						DWLRO	DLRO
5	Ignored	Ignored	DSN	FRDD	DPSDP	Ignored	CAEN	Ignored
6	Ignored							
7	Ignored							
8	Ignored	ADC	QEMC	Ignored	LED Mode			
9	Temperature Threshold							
10	Command Aging Time Limit (HI)							
11	Command Aging Time Limit (LO)							
12	QPE-2 for read operations							
13	Ignored							
14	DRRT	DNR	Ignored	RARRRD	FFMT	Ignore		
15	RTP	RRC	FCERT	Ignored	DRPDV	DSF	IRT	IVR

Table 88. Mode Sense Data Format 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 = 60h								
5	Head Offset Count = 0								
6	Data Strobe Offset Count = 0								
7	Reserved = 0								
8	Ignored								
9	Reserved = 0								
10	(MSB)	Recovery Time Limit							
11		(LSB)							

Table 89. Mode Sense Data Format Page 02h

Byte	eeBIT							
	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	Ignored							
5	Ignored							
6	(MSB)	Disconnect Time Limit = 0						(LSB)
7								
8	(MSB)	Connect Time Limit = 0						(LSB)
9								
10	(MSB)	Maximum Burst Size						(LSB)
11								
12	EMDP	Ignored			DIMM	RSVD = 0	DTDC	
13	Reserved = 0							
14	Reserved = 0							
15	Reserved = 0							

Table 90. Mode Sense Data Format Page 03h

Byte	BIT							
	7	6	5	4	3	2	1	0
0	PS = 1	RSVD=0	Page Code = 03h					
1	Page Length = 16h							
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							
13	(LSB)							
14	(MSB) Interleave = 0001h							
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							

Table 91. Mode Sense Data Format 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) Number of Cylinders (LSB)							
3								
4								
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 (LSB)							
15								
16								
17	Reserved = 0							
18	Reserved = 0							
19	Reserved = 0							
20	(MSB) Medium Rotation Rate (LSB)							
21								
22	Reserved = 0							
23	Reserved = 0							

Table 92. Mode Sense Data Format 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								
4	Verify Correction Span = 48h								
5	Reserved = 0								
6	Reserved = 0								
7	Reserved = 0								
8	Reserved = 0								
9	Reserved = 0								
10	(MSB) Verify Recovery Time Limit								
11									(LSB)

Table 93. Mode Sense Data Format 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	IC	ABPF	CAP	DISC	SIZE	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															
7									(LSB)							
8	(MSB) Maximum Pre-fetch															
9									(LSB)							
10	(MSB) Maximum Pre-fetch Ceiling															
11									(LSB)							
12	FSW	LBCSS	DRA	Reserved = 0												
13	Number of Cache Segments															
14	(MSB) Cache Segment Size															
15									(LSB)							
16	Reserved = 0															
17	(MSB) Non-Cache Segment Size															
18									(LSB)							
19																

Table 94. Mode Sense Data Format Page 09h

Byte	BIT								
	7	6	5	4	3	2	1	0	
0	PS = 1	RSVD = 0	Page Code = 09h						
1	Page Length = 0Eh								
2	(MSB) Interface Identifier = 0000h (LSB)								
3									
4	Reserved = 0								
5	Reserved = 0								
6	Reserved = 0								
7	Reserved = 0								
8	Ignored								
9	Ignored								
10	Ignored								
11	Ignored								
12	Ignored								
13	Ignored								
14	Reselection Retry Count								
15	Ignored				FAST20	FAST10	Ignored	FORCE8	

Table 95. Mode Sense Data Format Page 0Ah

Byte	BIT								
	7	6	5	4	3	2	1	0	
0	PS = 1	RSVD = 0	Page Code = 0Ah						
1	Page Length = 0Ah								
2	Reserved = 0						GLTSD	RLEC = 0	
3	Queue Algorithm Modifier				Reserved = 0		QErr	DQue	
4	RSVD = 0	RAC = 0	Reserved = 0		SWP	RAERP = 0	UAAERP = 0	EAERP = 0	
5	Reserved = 0								
6	(MSB) Ready AEN Holdoff Period (LSB)								
7									
8	(MSB) Busy Timeout Period (LSB)								
9									
10	Reserved = 0								
11	Reserved = 0								

Table 96. Mode Sense Data Format Page 0Ch

Byte	BIT								
	7	6	5	4	3	2	1	0	
0	PS = 1	RSVD = 0	Page Code = 0Ch						
1	Page Length = 16h								
2	ND = 1	LPN = 0	Reserved = 00h						
3	Reserved = 0								
4 5	(MSB)	Maximum number of notches						(LSB)	
6 7	(MSB)	Active notch						(LSB)	
8 11	(MSB)	Starting Boundary						(LSB)	
12 15	(MSB)	Ending Boundary						(LSB)	
16 23	(MSB)	Pages Notched = 000000000000100Ch						(LSB)	

Table 97. Mode Sense Data Format Page 1Ah

Byte	BIT							
	7	6	5	4	3	2	1	0
0	PS = 1	RSVD = 0	Page Code = 1Ah					
1	Page Length = 1Ah							
2	Reserved = 00h							
3	Reserved = 0						Idle	Standby
4 - 7	(MSB)	Idle Condition Time						(LSB)
8 - 11	(MSB)	Standby Condition Timer						(LSB)

Table 98. Mode Sense Data Format Page 1Ch

Byte	BIT							
	7	6	5	4	3	2	1	0
0	PS = 1	RSVD = 0	Page Code = 1Ch					
1	Page Length = 0Ah							
2	PERF	RSVD = 0	EWASC	DExcpt	Test	RSVD = 0	LogErr	
3	Reserved = 0				MRIE			
4 - 7	(MSB)	Interval Timer						(LSB)
8 - 11	(MSB)	Report Count						(LSB)

The Parameters Savable (PS) bit is set to 1 for pages 0, 1, 2, 4, 7, 8, 9, 0Ah, 0Ch, 1Ah, and 1Ch to indicate that some of the parameters of those pages can be saved.

The Parameters Savable (PS) bit is set to 1 for page 3 to indicate that parameters may be changed; however, they cannot 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)”).

1.8 Persistent Reserve In

Table 99. Persistent Reserve In command

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 5Eh							
1	Reserved			Service Action				
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)	Allocation Length						(LSB)
8								
9	Control							

The Persistent Reserve In command is used to obtain information about persistent reservations and reservation keys that are present within a target. This command is used in conjunction with the Persistent Reserve Out command.

The actual length of the Persistent Reserve In parameter data is available in the parameter data. The Allocation Length field in the CDB indicates how much space has been reserved for the returned parameter list.

If the length is not sufficient to contain the entire parameter list, the first portion of the list shall be returned. This shall not be considered an error. If the remainder of the list is required, the initiator should send a new

Persistent Reserve In command with an Allocation Length field large enough to contain the entire list.

PERSISTENT RESERVE IN Service Actions

The Service Action codes for the Persistent Reserve In command are defined below.

Table 100. Persistent Reserve In Service Action codes

Code	Name	Description
00h	READ KEYS	Reads all registered Reservation Keys
01h	READ RESERVATION	Reads the current persistent reservation
02h-1Fh	Reserved	Reserved

Read Keys

The Read Keys service action requests that the target return a parameter list containing a header and a list of each currently registered initiator's reservation key. If multiple initiators have registered with the same key, then that key value shall be listed multiple times, once for each such registration.

Read Reservation

The Read Reservation service action requests that the target return a parameter list containing a header and the persistent reservation, if any, that is present in the target.

PERSISTENT RESERVE IN parameter data for Read Keys

The format for the parameter data provided in response to a Persistent Reserve In command with the Read Keys service action is shown below.

Table 101. Persistent Reserve In parameter data for Read Keys

Byte	BIT							
	7	6	5	4	3	2	1	0
0 3	Generation (MSB) (LSB)							
4 7	Additional Length (n-7) (MSB) (LSB)							
	Reservation Key List							
8 15	Reservation Key (first) (MSB) (LSB)							
	• • •							
n-7 n	Reservation Key (last) (MSB) (LSB)							

The Generation field contains a 32-bit counter that the target shall increment every time a Persistent Reserve Out command requests a Register, a Clear, a Preempt, or a Preempt And Abort service action. The counter shall not be incremented by a Persistent Reserve In command, by a Persistent Reserve Out command that performs a Reserve or Release service action, or by a Persistent Reserve Out command that is not performed due to an error or reservation conflict. Regardless of the APTPL value the generation value shall be set to 0 as part of the power on reset process.

The Additional Length field contains a count of the number of bytes in the Reservation key list. If the allocation length specified by the Persistent Reserve In command is not sufficient to contain the entire parameter list, then only the bytes from 0 to the maximum allowed allocation length shall be sent to the initiator. The incremental remaining bytes shall be truncated, although the Additional Length field shall still contain the actual number of bytes in the reservation key list without consideration of any truncation resulting from an insufficient allocation length. This shall not be considered an error.

The Reservation Key list contains the 8-byte reservation keys for all initiators that have registered through all ports with the target.

PERSISTENT RESERVE IN parameter data for Read Reservation

The format for the parameter data provided in response to a Persistent Reserve In command with the Read Reservation service action is shown below.

Table 102. Persistent Reserve In parameter data for Read Reservation

Byte	BIT							
	7	6	5	4	3	2	1	0
0 3	(MSB) Generation (LSB)							
4 7	(MSB) Additional Length (n-7) (LSB)							
8 n	(MSB) Reservation Descriptors (LSB)							

The Generation field shall be as defined for the Persistent Reserve In Read Keys parameter data. The Additional Length field contains a count of the number of bytes to follow in the Reservation Descriptor(s).

If the allocation length specified by the Persistent Reserve In command is not sufficient to contain the entire parameter list, then only the bytes from 0 to the maximum allowed allocation length shall be sent to the initiator. The remaining bytes shall be truncated, although the Additional Length field shall still contain the actual number of bytes of the Reservation Descriptor(s) and shall not be affected by the truncation. This shall not be considered an error.

The format of the Reservation Descriptors is defined in the **Persistent Reserve In Reservation Descriptor** table (Table 103). There shall be a Reservation Descriptor for the persistent reservation, if any, present in the target having a persistent reservation.

Table 103. Persistent Reserve In Reservation Descriptor

Byte	BIT							
	7	6	5	4	3	2	1	0
0 7	(MSB) Reservation Key (LSB)							
8 11	(MSB) Scope-specific Address (LSB)							
12	Reserved							
13	Scope				Type			
14	Reserved							
15	Reserved							

The Reservation Descriptor for each reservation shall contain the Reservation Key under which the persistent reservation is held. The type and scope of each persistent reservation as present in the Persistent Reserve Out command that created the persistent reservation shall be returned.

If the scope is a logical unit reservation the Scope-Specific Address shall be set to zero.

Persistent Reservations Scope

The value in the Scope field shall indicate whether a persistent reservation applies to an entire logical unit or to an element. The values in the Scope field are defined below.

Table 104. Persistent Reservation SCOPE Codes

Code	Name	Description
0h	LU	Persistent reservation applies to the full logical unit
1h		Obsolete
2h	Element	Persistent reservation applies to the specified element
3h-Fh	Reserved	Reserved

LU Scope

A Scope field value of LU shall indicate that the persistent reservation applies to the entire logical unit. The LU scope shall be implemented by all device servers that implement Persistent Reserve Out.

Persistent Reservations Type

The value in the Type field shall specify the characteristics of the persistent reservation being established for all data blocks within the element or within the logical unit. The **Persistent Reservation Type Codes** table (Table 105), defines the characteristics of the different type values. For each persistent reservation type, the table lists the code value and describes the required target support. The description of required target support is divided into two paragraphs. The first paragraph defines the required handling for read operations. The second paragraph defines the required handling for write operations.

Table 105. Persistent Reservation Type Codes

Code	Name	Description
0h		Obsolete
1h	Write Exclusive	Reads Shared: Any application client on any initiator may execute tasks that perform transfers from the storage medium or cache of the target to the initiator. Writes Exclusive: Any task from any initiator other than the initiator holding the persistent reservation that requests a transfer from the initiator to the storage medium or cache of the target shall result in a reservation conflict.
2h		Obsolete
3h	Exclusive Access	Reads Exclusive: Any task from any initiator other than the initiator holding the persistent reservation that requests a transfer from the storage medium or cache of the target to the initiator shall result in a reservation conflict.
4h		Obsolete
5h	Write Exclusive Registrants Only	Reads Shared: Any initiator may execute tasks that request transfers from the storage medium or cache of the target to the initiator. Writes Exclusive: A task that requests a transfer to the storage medium or cache of the target from an initiator that is not currently registered with the target shall result in a reservation conflict.
6h	Exclusive Access Registrants Only	Reads Exclusive: A task that requests a transfer from the storage medium or cache of the target to an initiator that has not previously requested a REGISTER service

		<p>action with the target shall result in a reservation conflict.</p> <p>Writes Exclusive: A task that requests a transfer to the storage medium or cache of the target from an initiator that is not currently registered with the target shall result in a reservation conflict.</p>
7h-Fh	Reserved	

1.9 Persistent Reserve Out

Table 106. Persistent Reserve Out command

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 5Fh							
1	Reserved			Service Action				
2	Scope				Type			
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)	Parameter List Length (18h)						(LSB)
8								
9	Control							

The Persistent Reserve Out command is used to request service actions that reserve a logical target for the exclusive or shared use of a particular initiator. The command uses other service actions to manage and remove such reservations. The command shall be used in conjunction with the Persistent Reserve In command and shall not be used with the Reserve and Release commands.

Initiators requesting Persistent Reserve Out service actions are identified by a reservation key provided by the initiator. An initiator may use the Persistent Reserve In command to identify which initiators are holding a persistent reservation and use the Persistent Reserve Out command to preempt that reservation if required.

If a Persistent Reserve Out command is attempted, but there are insufficient target resources to complete the operation, the target shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to Insufficient Registration Resources.

The Persistent Reserve Out command contains fields that specify a persistent reservation service action, the intended scope of the persistent reservation, and the restrictions caused by the persistent reservation. If a Scope field specifies a scope that is not implemented, the target shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and additional sense data shall be set to INVALID FIELD IN CDB.

Fields contained in the Persistent Reserve Out parameter list specify the information required to perform a particular persistent reservation service action.

The parameter list shall be 24 bytes in length and the Parameter list length field shall contain 24 (18h). If the

Parameter list length is not 24, the target shall return a CHECK C ONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to PARAMETER LIST LENGTH ERROR.

PERSISTENT RESERVE OUT Service Actions

The Persistent Reserve Out command service actions are defined below.

Table 107. Persistent Reserve Out service action codes

Code	Name	Description
00h	REGISTER	Registers a reservation key with the target.
01h	RESERVE	Creates a persistent reservation having a specified scope and type.
02h	RELEASE	Releases the selected reservation for the requesting initiator.
03h	CLEAR	Clears all reservation keys and all persistent reservations.
04h	PREEMPT	Preempts persistent reservations from another initiator.
05h	PREEMPT & ABORT	Preempts persistent reservations from another initiator and aborts the task set for the preempted initiator.
06h-1Fh	Reserved	

PERSISTENT RESERVE OUT parameter list

The parameter list required to perform the Persistent Reserve Out command is defined in the **Persistent Reserve Out parameter list** table (Table 108). All fields shall be sent on all Persistent Reserve Out commands, even if a particular field is not required for the specified Service Action and scope values.

Table 108. Persistent Reserve Out parameter list

Byte	BIT							
	7	6	5	4	3	2	1	0
0 7	(MSB) Reservation Key							(LSB)
8 15	(MSB) Service Action Reservation Key							(LSB)
16 19	(MSB) Scope-specific Address							(LSB)
20	Reserved							APTPL
21	Reserved							
21	Reserved							
21	Reserved							

The Reservation Key field contains an 8-byte value provided by the initiator to the target to identify the initiator that is the source of the Persistent Reserve Out command. The target shall verify that the Reservation Key field in a Persistent Reserve Out command matches the registered reservation key for the initiator from which the task was received except for the Register service action for an unregistered initiator which shall have a reservation key value of zero. If a Persistent Reserve Out command specifies a Reservation Key field other than the reservation key registered for the initiator, the

target shall return a Reservation Conflict status. The reservation key of the initiator shall be verified to be correct regardless of the service action and scope values.

The Service Action Reservation Key field contains information needed for the following service actions: the Register, Preempt, and Preempt and Abort service actions. For the Register service action, the Service Action Reservation Key field contains the new reservation key to be registered. For the Preempt, and Preempt and Abort service actions, the Service Action Reservation Key field contains the reservation key of the persistent reservations that are being preempted. The service action reservation key is ignored for all other service actions.

The Activate Persist Through Power Loss (APTPL) bit shall be valid only for the Register service action. In all other cases, the APTPL bit shall be ignored.

If the last valid APTPL bit value received by the target is zero, the loss of power in the target shall release the persistent reservation for all logical units and remove all reservation keys. If the last valid APTPL bit value received by the target is one, the target shall retain any persistent reservation that may be present and the reservation keys for all registered initiators even if power is lost and later returned.

1.10 Pre-Fetch

Table 109. Pre-Fetch Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 34h							
1	Reserved			Reserved = 0			Immed	RelAdr
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 pre-fetch may be terminated upon receipt of another command, (see 4.8.3, “Read Ahead” 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.

4. 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.

1.11 Read (6)

Table 110. Read (6) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 08h							
1	Reserved			(MSB) Logical Block Address				
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 Appendix B.1, “Data Recovery Procedure for Data Field Errors”). ERPs are controlled by the error recovery parameters (see 1.6.3, “Page 1h - Error Recovery Parameters”).

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 greater than 2,097,152. At 512 byte blocks, this only allows access to 1.048 Gigabytes.

1.12 Read (10)

Table 111. Read (10) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 28h							
1	Reserved			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 - Caching Parameters” and “Segmented Caching.”

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, no data is transferred. The CDB is validated and protocol checked and, if no problems are found, *Good Status* is returned, immediately. This condition is not considered an error.

1.13 Read Buffer

Table 112. Read Buffer Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 3Ch							
1	Reserved			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 those 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 and must be an integer

multiple of four. An offset greater than or equal to the buffer capacity or one that is not an integer multiple of four 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 field 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.

1.14 Read Capacity

Table 113. Read Capacity Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 25h							
1	Reserved				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 114. 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								

1.15 Read Defect Data (10)

Table 115. Read Defect Data Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 37h							
1	Reserved				Reserved = 0			
2	Reserved = 0			PList	GList	Defect List Format		
3-6	Reserved = 0							
7	Allocation Length							
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.

Table 116. Read Defect Data - Header

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

Table 117. Read Defect Data - Defect Descriptor(s)

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. To read defects beyond this number, use the Read Defect Data (12) command, which allows a maximum allocation length of FFFFFFFFh.

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

Table 118. 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 Sector (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:
 $(\text{sector number} * \text{user bytes/sector}) + (1/2 * \text{user bytes/sector})$

Table 119. 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.

1.16 Read Defect Data (12)

Table 120. Read Defect Data Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = B7h							
1	Reserved			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 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 an eight byte header, followed by zero or more defect descriptors.

Table 121. Read Defect Data - Header

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

Table 122. Read Defect Data - Defect Descriptor

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

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 eight 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 eight. 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 123. 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 Sector (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:
 $(\text{sector number} * \text{user bytes/sector}) + (1/2 * \text{user bytes/sector})$

Table 124. Defect Descriptor - Physical Sector Format

Byte	BIT							
	7	6	5	4	3	2	1	0
0	(MSB) Cylinder Number of Defects (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.

1.17 Read Long

Table 125. Read Long Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 3Eh							
1	Reserved			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 Data CRC 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 in the Sense Data 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:

$$\text{transfer length} = \text{logical block size} + (32 * n),$$

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 neither read from nor retained in the cache.

1.18 Reassign Blocks

Table 126. Reassign Blocks Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 07h							
1	Reserved			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 Logical Blocks called target LBAs to alternate physical sectors called spares. Excess capacity is allocated during format such that a limited number of LBA's may be reassigned to alternate physical sectors while maintaining all LBA's on the drive in sequential order.

The Logical Block Addresses are transferred to the Target during the Data Out phase. The physical sector corresponding to the LBA in the Reassign Blocks Data Descriptor is reassigned. Additional physical sectors residing in the affected data area may also be reassigned if nested reassignment is enabled.

The Mode Page 0h DNR (Disable Nested Reassignment) bit determines if nested reassignment is enabled. For more information on the DNR bit see 1.6.2, "Page 0h - Vendor Unique Parameters." If nested reassignment is disabled then only the target LBA is reassigned. If Nested Reassignment is enabled, a test is made to determine if additional physical sectors in the affected area need reassignment. If a sector is determined to need reassignment and its data is recovered, that sector will be reassigned along with the target LBA. The sectors determined to need reassignment are called nested defects. Up to 3 nested defects can be reassigned per target LBA.

All data is preserved during a reassign command except for the target LBA data. The Mode Page 0h DRRT (Disable Restore Reassign Target) bit determines if the reassign blocks command will attempt to recover the target LBA data. For more information on the DRRT bit, see 1.6.2, "Page 0h - Vendor Unique Parameters." If the drive cannot recover the data, the target block data being reassigned will be filled with FFh or 00h depending on the setting of the Mode Page 0h RTP (Reassign Target Pad) bit. For more information on the RTP bit, see 1.6.2, "Page 0h - Vendor Unique Parameters." If the target cannot recover the data at the target LBA then the Initiator will have to restore the data after the Reassign Blocks command completes successfully.

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 4.17, "Bring-Up Sequence" for more information.

If the reassignment begins to move data and is interrupted or fails to complete successfully, the Target enters a degraded mode of operation. In this mode data can be read but Writing to the drive is prohibited. For more information on the degraded mode see 4.1.9, "Degraded Mode."

Upon successful completion of this command, the location of the physical sectors reassigned during the command are added to the GList. The reassigned sectors are marked defective and cannot be accessed again until after a format operation discards the GList.

The 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 127. 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			
3					(LSB)			

Table 128. Format of Reassign Blocks Data - Descriptor(s)

Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 3	Reassign Blocks Data Descriptor 0							
4n - 4n + 3	Reassign Blocks Data Descriptor n							

The Reassign Blocks defect list contains a 4-byte header followed by the Reassign Blocks Data Descriptors.

The Defect List Length in the Header is a multiple of 4 (the length of a defect descriptor), where 16 is its maximum value. The Reassign Blocks command requests the reassignment of up to four logical blocks per command. Therefore up to four defect descriptors are allowed. Each defect descriptor specifies a 4 byte logical block address. When more than one defect descriptor is used, the logical block addresses specified shall be in ascending order.

If the Reassign Blocks command was unable to complete successfully, the command will terminate with *Check Condition Status*. The LBA of the first defect descriptor not reassigned will be returned in the Command Specific Information field of the Sense Data. If the first defect descriptor not reassigned cannot be determined, this field will be set to all FFh. For more information on command specific information, see Appendix A, "SCSI Sense Data Format."

If the Reassign Blocks command fails due to an unrecoverable read error in an LBA between the reassign target and the spare sector the LBA of the unrecovered block will be returned in the Information field of the Sense Data and the valid bit will be set to one. If the drive is not left in reassign degraded mode, the initiator should insert the new LBA into the defect descriptor and reissue the Reassign Blocks command. For more details on the Information Field, see Appendix A, "SCSI Sense Data Format."

1.19 Receive Diagnostic Results

Table 129. Receive Diagnostic Results Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 1Ch							
1	Reserved			Reserved = 0				PCV = 0
2	Page Code = 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 Diagnostic command be sent to the initiator.

The Page Code Valid (PCV) bit of 0 indicates that the most recent Send Diagnostic command shall define the data returned by the Receive Diagnostic Results command.

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.

1.19.1 Supported Diagnostic Pages - Receive Diagnostic

Table 130. Supported Diagnostic Pages - Receive Diagnostic

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Page Code = 00h							
1	Reserved = 0							
2	(MSB Page Length = 3 (LSB))							
3								
4	First Supported Page Code = 00h							
5	Second Supported Page Code = 40h							
6	Third Supported Page Code = A0h							

The supported diagnostic page returns a list of supported pages in ascending order.

1.19.2 Receive Diagnostic 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 Diagnostic command translate address page.

Table 131. Translate Address Page - Receive Diagnostic

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Page Code = 40h							
1	Reserved = 0							
2 3	(MSB) Page Length (LSB)							
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 Diagnostic 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 Diagnostic 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 none 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. 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 within the address to be translated.

1.19.3 Receive Diagnostic Page A0h

The Receive Diagnostic Device LED Control Page A0h is returned as a result of the Send Diagnostic Device LED Control Page A0h. For a description of the parameters in this page, see 1.19.3, “Send Diagnostic Page A0h.”

Table 132. Device LED Control Page - Receive Diagnostic

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Page Code = A0h							
1	Reserved = 0							
2	(MSB) Page Length = 0002h (LSB)							
3								
4	Reserved = 0							
5	Reserved = 0							SBDL

1.20 Release (6)

Table 133. Release (6) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 17h							
1	Reserved			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 1.24, “Reserve (6)” and 4.1.10, “Command Processing While Reserved”).

1.21 Release (10)

Table 134. Release (10) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 57h							
1	Reserved			3rdPty	Reserved		LongID = 0	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.

A Long ID (LongID) bit of 0 indicates that Long IDs are not supported.

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 an 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 1.25, “Reserve (10) and 4.1.10, “Command Processing While Reserved”).

1.22 Report LUNs

Table 135. Report LUNs Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = A0h							
1	Reserved = 0							
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 Report LUNs command requests that the target return the known Logical Unit Numbers (LUN) to the initiator. The Report LUNs command should always be available and is unaffected by any reservations.

The Allocation Length must be at least 16 bytes. If the Allocation Length is less than 16 bytes, the target will return *Check Condition Status* with Sense Key of Illegal Request and Additional Sense Code of Invalid Field in CDB. If the Allocation Length is not sufficient to contain the Logical Unit Number values for all configured logical units, the target shall report as many logical unit number values as will fit in the specified Allocation Length. This is not considered an error.

The Report LUNs command will send the LUN List in the subsequent Data Out Phase. The format of the LUN List is shown in the following table.

Table 136. LUN Reporting Parameter List Format

Byte	BIT							
	7	6	5	4	3	2	1	0
0 - 3	(MSB) LUN List Length = 8 (LSB)							
4	Reserved = 0							
5	Reserved = 0							
6	Reserved = 0							
7	Reserved = 0							
8 - 15	(MSB) LUN = 0 (LSB)							

The LUN list length shall contain the length in bytes of the LUN list that is available to be transferred. This product only supports one LUN. Therefore, the LUN list length must be set to 8. The only supported LUN is zero.

1.23 Request Sense

Table 137. Request Sense Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 03h							
1	Reserved				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”). 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.”

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.”

1.24 Reserve (6)

Table 138. Reserve (6) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 16h							
1	Reserved			3rdPty	3rd Part ID			Ext = 0
2	Reserved 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 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”).

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*.

1.25 Reserve (10)

Table 139. Reserve (10) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 56h							
1	Reserved			3rdPty	Reserved		LongID = 0	Ext = 0
2	Reservation Identification							
3	Third Party Device ID							
4	Reserved = 0							
5	Reserved = 0							
6	Reserved = 0							
7	Extent List Length = 0							
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.

A Long ID (LongID) bit of 0 indicates that long IDs are not supported.

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 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”).

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*.

1.26 Rezero Unit

Table 140. Rezero Unit Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 01h							
1	Reserved				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 the cylinder containing Logical Block Address 0.

1.27 Seek (6)

Table 141. Seek (6) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 0Bh							
1	Reserved			(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 greater than the value returned by the Read Capacity command, the Drive returns a *Check Condition Status* with a Sense Key of Illegal Request and an Additional Sense Code of Invalid Field in CDB.

1.28 Seek (10)

Table 142. Seek (10) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 2Bh							
1	Reserved			Reserved = 0				
2	(MSB) Logical Block Address							
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 greater than the value returned by the Read Capacity command, the Drive returns a *Check Condition Status* with a Sense Key of Illegal Request and an Additional Sense Code of Invalid Field in CDB.

1.29 Send Diagnostic

Table 143. Send Diagnostic Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 1Dh							
1	Reserved			PF	RSVD = 0	SelfTest	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.

The PF (Page Format) Bit set to one specifies that the Send Diagnostic parameters conform to the page structure as specified in the ANSI SCSI Standard.

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.

- ROM checksum Test
- Data Buffer Test
- Seek Test
- Head Offset Test
- Read Test (all heads)
- Write Test (all heads)
- Cyclic Redundancy Code (CRC) Test
- Error Correction Code (ECC) 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”).

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 DevOfI and UnitOfI bits are ignored because they are not applicable for this device.

1.29.1 Send Diagnostic Page 00h

Table 144. Supported Diagnostics Pages - Send Diagnostic

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Page Code = 00h							
1	Reserved = 0							
2	(MSB) Page Length = 00h (LSB)							
3								

The supported diagnostics page allows the initiator to request that a list of supported diagnostic pages be returned on the next Receive Diagnostic Results command.

The page code field is 00h.

The page length field is 00h. If the page length field is not 0, the command will terminate 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.

1.29.2 Send Diagnostic 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 Diagnostic Results 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.

Table 145. Translate Address Page - Send Diagnostic

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 address format followed by the address to be translated.

The Page Code field is 40h.

The Page Length field is 000Ah and specifies the length of the parameter bytes which follow.

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. 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.

1.29.3 Send Diagnostic Page A0h

The Device LED Control Page A0h allows the initiator to turn on or off the device LED light.

Table 146. Device LED Control Page - Send Diagnostic

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Page Code = A0h							
1	Reserved = 0							
2	(MSB) Page Length = 0002h (LSB)							
3								
4	Reserved = 0							
5	Reserved = 0							SBDL

The page begins with a four-byte page header which specifies the page code and length. A SBDL (Set Bad Device Light) bit of one turns the device LED on, zero turns device LED off. If the target detects an error in the passed parameter data it shall 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.

1.30 Start/Stop Unit

Table 147. Start/Stop Unit Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 1Bh							
1	Reserved			Reserved = 0				Immed
2	Reserved = 0							
3	Reserved = 0							
4	Power Conditions = 0				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” 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”).
- 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”).

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”).

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 4.17, “Bring-Up Sequence”). 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.

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.

1.31 Synchronize Cache

Table 148. Synchronize Cache Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 35h							
1	Reserved			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.

1.32 Test Unit Ready

Table 149. Test Unit Ready Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 00h							
1	Reserved			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 (provided the LUN is not Busy or Reserved), then this command returns *Good Status*. 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 Appendix B.3, “Priority of Error Reporting”).

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.

1.33 Verify

Table 150. Verify Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 2Fh							
1	Reserved			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 Mismatch Sense Key.

DRP actions for this command are controlled by Mode parameters in 1.6.7, “Page 7h - Error Recovery Parameters for Verify.” The actual DRP steps are defined in Appendix B.1.2, “Verify Commands.” Mismatch 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 ensures 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.

1.34 Write (6)

Table 151. Write (6) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 0Ah							
1	Reserved			(MSB)	Logical Block Address			
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 Error Recovery Procedures (ERP) (see Appendix B.3, “Data Recovery Procedure for Data Field Errors”). ERPs are controlled by the error recovery parameters (see 1.6.3, “Page 1h - Error Recovery Parameters”).

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.

1.35 Write (10)

Table 152. Write (10) Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 2Ah							
1	Reserved			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 Force Unit Access (FUA) 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 Disable Page Out (DPO) 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 - Caching Parameters” and “Segmented Caching.”

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 zero, no data is transferred. The CDB is validated and protocol checked and, if no problems are found, *Good Status* is returned, immediately.

1.36 Write and Verify

Table 153. Write and Verify Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 2Eh							
1	Reserved			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 a 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.” The actual DRP steps are defined in Appendix B.1.2, “Verify Commands.” 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 ensures 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 - Caching Parameters” and 4.8, “Segmented Caching.”

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.

1.37 Write Buffer

Table 154. Write Buffer Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 3Bh							
1	Reserved			Reserved = 0		Mode		
2	Buffer ID							
3	(MSB) (LSB) Buffer Offset							
4								
5								
6	(MSB) (LSB) Parameter List Length							
7								
8								
9	VU = 0		Reserved = 0			Flag	Link	

The Write Buffer command is used to test the Target buffer and download microcode to the Target. The values allowed in the Mode field are:

- 000b - Combined Header and Data Mode
- 001b - Vendor Unique Mode - Write Memory
- 010b - Write Data
- 101b - Download and Save

If any modes other than those 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 four-byte header consists of all reserved bytes. Receipt of a non-zero byte in the header results in *Check Condition Status* with Sense Key of Illegal Request and Additional Sense Code of Invalid Field in Parameter List.

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 a 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.

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.

- **Mode = 001b (Vendor Unique - Write Memory):**

In this mode, 16 bytes of data are inserted as a new entry in the Media Error Log.

To add 16 bytes to the new Media Error Log Entry, the Buffer ID must be set to 0xF6.

The Buffer Offset must be zero. Receipt of a non-zero Buffer Offset results in *Check Condition Status* with a Sense Key of Illegal Request and Additional Sense Code of Invalid Field in CDB.

The Parameter List Length must be set to 16 (decimal). Exactly 16 bytes of data will be inserted in the new Media Error Log Entry. Receipt of an invalid Parameter List Length results in *Check Condition Status* with a Sense Key of Illegal Request and Additional Sense Code of ***Parameter List Length Error***.

- **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 a 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 offset also must be an integer multiple of four. An offset greater than or equal to the buffer capacity or not an integer multiple of four results in *Check Condition Status* with a Sense Key of Illegal Request and Additional Sense Code of Invalid Field in CDB.

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 a 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 = 101b (Download and Save):**

The microcode is downloaded to the control storage of the Target.

The downloaded microcode is saved in non-volatile memory. The original microcode is lost. Non-volatile memory includes electrically programmable memory.

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 sends 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 not incremented by one), the Target returns *Check Condition Status* with a 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 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 a Sense Key of 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.

The command will alter the contents of electrical memory. The target need not be allowed to disconnect before the electrical memory is altered. It is imperative that the Target not be interrupted during this electrical memory altering operation. The Target does not respond to SCSI reset and selection during this operation. The Initiator should take steps to prevent power cycles during execution of this command and should allow for a command time-out for this command as recommended in the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec.

When electrical memory is altered, an internally generated reset occurs as part of the command. 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”). The Additional Sense Code **Power On Reset**, is reported in the sense data. All queued commands are cleared. Mode Select conditions are restored to their last saved values. The Target preserves Wide Data Transfer and Synchronous Data Transfer agreements only for the Initiator that issued the Write Buffer command.

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.

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”). The Additional Sense Code **Microcode has been changed**, is reported in the sense data.

The first 19 bytes of the microcode data set 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 4.8, “Segmented Caching.”

1.38 Write Long

Table 155. Write Long Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 3Fh							
1	Reserved			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 DATA CRC and ECC bytes. The ECC field depends solely on the data and DATA CRC bytes and is unaffected by sync 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.

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 in the Sense Data 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: transfer length = logical block size + (32 * 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.

1.39 Write Same

Table 156. Write Same Command Descriptor Block

Byte	BIT							
	7	6	5	4	3	2	1	0
0	Operation Code = 41h							
1	Reserved				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 157.

Table 157. SCSI Status Byte

Byte	BIT							
	7	6	5	4	3	2	1	0
Status	Reserved = 0		Status Code					RSVD

Table 158. 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
ACA Active	R	R	1	1	0	0	0	R

Note: 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 1.10, “Pre-Fetch”). 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” 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 for a different Initiator (refer to 1.24, “Reserve (6)” and 1.20, “Release (6)”), 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” 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 target’s 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” 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.

30h ACA Active status

Auto Contingent Allegiance Active Status is not supported.

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.

Table 159. Supported One-Byte Messages

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
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:

Table 160. Supported Two-Byte Messages

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:

Table 161. Supported Multiple-Byte Messages

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 a 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 0h-Vendor Unique Parameters”), 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)”).

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”).

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 3.1.17, “Identify (80h or C0h)”).

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”). 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” and 3.6, “SCSI Bus Related Error Handling Protocol”).

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”). 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” and 3.6, “SCSI Bus Related Error Handling Protocol” 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 “Message State/Action Table” 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 3.3, “Attention Condition”). 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,” 3.2, “Supported Message Functions,” and 3.6, “SCSI Bus Related Error Handling Protocol”).

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. A *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.” The target goes to BUS FREE following receipt of a Clear Queue message. Pending sense data is preserved for all initiators.

3.1.15 Queue Tag Messages (20h, 21h, 22h)

Table 162. Queue Tag Messages

Byte	Value	Description
0	20h 21h 22h	Queue Tag message
1	XXh	Queue Tag

Queue Tag messages consist of:

- Simple Queue Tag Message
- Head of Queue Tag Message
- Ordered 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 responds as outlined in 4.1.3, “Incorrect Initiator Connection.” A Queue Tag becomes available for reassignment when the I/O process ends.

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 an I_T_L nexus is revived for the I/O process (untagged command).

3.1.15.1 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.

3.1.15.2 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.

3.1.15.3 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.

3.1.16 Ignore Wide Residue (23h)

Table 163. Ignore Wide Residue Message Format

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.

3.1.17 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.
- Bits 5-0** 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.”

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”). 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.

3.1.18 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.

Table 164. Extended Message Format

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.

3.1.19 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, Self Initiated Reset, or Transceiver Mode changed).

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.

The implied synchronous agreement remains in effect until the Target is Reset (Power-on Reset, SCSI Bus “hard” Reset, Bus Device Reset message, Self Initiated Reset, or Transceiver Mode changed), a new

synchronous agreement is negotiated, or the Target receives a Wide Data Transfer Request message. If a Reset occurs or a Wide Data Transfer Request is negotiated, the Target goes to asynchronous mode.

3.1.19.1 Synchronous Data Transfer Request Message Definition

Table 165. 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 Factor (See Table 166).
4	X	REQ/ACK offset.

Table 166. Synchronous Data Transfer Period Factor

Value	Description
0Ah	Transfer period equals 25 nSec
0Bh	Transfer period equals 30.3 nSec
0Ch	Transfer period equals 50 nSec
0Dh-FFh	Transfer period equals transfer period factor times 4 nSec

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 25 nSec to 375 nSec in 25 nSec increments. The Target uses the “fast-40 SCSI data transfer” timings if a synchronous agreement results in a target transfer period of less than 50 nSec. The Target uses the “fast-20 SCSI data transfer” timings if a synchronous agreement results in a target transfer period greater than or equal to 50 nSec and less than 100 nSec. The Target uses the “fast SCSI data transfer” timings if a synchronous agreement results in a target transfer period greater than or equal to 100 nSec and less than 200 nSec. The Drive does not support a target transfer period of 25 nSec with an established 8 bit transfer width.

The Target’s response and initiated values are based on Mode Page 09h, (see 1.6.9, “Page 09h-Peripheral Device Parameters”), FAST 20 and FAST 10 bits being set to zero. Other settings of the FAST 20 and FAST 10 bits will limit the Target’s response.

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. The Drive does not support REQ/ACK offset of 1 with a target transfer period of 25 nSec.

3.1.19.2 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:

Table 167. Initiator Request/Target Response

Initiator Request	Target Response	Target Transfer Period	Maximum Burst Rate
0 <= Mi <= 10	Mt = 10	25 nSec	40.00 MT/s
11 <= Mi <= 12	Mt = 12	50 nSec	20.00 MT/s
13 <= Mi <= 18	Mt = Mi	75 nSec	13.34 MT/s
19 <= Mi <= 25	Mt = Mi	100 nSec	10.00 MT/s
26 <= Mi <= 31	Mt = Mi	125 nSec	8.00 MT/s
32 <= Mi <= 37	Mt = Mi	150 nSec	6.67 MT/s
38 <= Mi <= 43	Mt = Mi	175 nSec	5.71 MT/s
44 <= Mi <= 50	Mt = Mi	200 nSec	5.00 MT/s
51 <= Mi <= 56	Mt = Mi	225 nSec	4.44 MT/s
57 <= Mi <= 62	Mt = Mi	250 nSec	4.00 MT/s
63 <= Mi <= 68	Mt = Mi	275 nSec	3.64 MT/s
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 <= 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, 40.00 MT/s on a 16 bit transfer width is an effective transfer rate of 80.00 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 372 nSec ($M > 93$), the Target responds 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 equal to 25 nSec ($M = 10$) with a REQ/ACK offset value of 1, the Target responds with a transfer period of 50 nSec ($M = 12$) with a REQ/ACK offset of 1.

If the Initiator requests a transfer period equal to 25 nSec ($M = 10$), but has not established a 16 bit transfer width agreement, the Target responds with a transfer period of 50 nSec ($M = 12$).

If the Initiator is running in Single Ended mode and requests a transfer period equal to 25 nSec ($M = 10$), the Target responds with a transfer period of 50 nSec ($M = 12$).

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"). For any other message, the Target completes negotiation and goes to synchronous mode.

3.1.19.3 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. The transfer period is equal to 25 nSec ($M = 10$) when running LVD mode with an established 16 bit transfer width. The transfer period is equal to 50 nSec ($M = 12$) when running Single Ended mode or when running LVD mode with an established 8 bit transfer width. The REQ/ACK offset is 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"). 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's corresponding transfer period as shown in the following table:

Table 168. Target Response to Initiator's Transfer Period

Initiator's Response	Target Transfer Period	Maximum Burst Rate
$0 \leq M_i \leq 9$	Send Message Reject (Asynchronous mode)	N/A
$10 \leq M_i \leq 10$	25 nSec	40.00 MT/s
$11 \leq M_i \leq 12$	50 nSec	20.00 MT/s
$13 \leq M_i \leq 18$	75 nSec	13.34 MT/s
$19 \leq M_i \leq 25$	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 ≤ Mi ≤ 75	300 nSec	3.33 MT/s
76 ≤ Mi ≤ 81	325 nSec	3.08 MT/s
82 ≤ Mi ≤ 87	350 nSec	2.86 MT/s
88 ≤ Mi ≤ 93	375 nSec	2.67 MT/s
94 ≤ Mi ≤ 255	Send Message Reject (Asynchronous mode)	N/A

Note: When operating in Single Ended mode, if the corresponding transfer period received from the Initiator indicates a transfer period that is less than 50 nSec ($M < 12$), the Target sends a Message Reject message to the Initiator to indicate asynchronous mode.

Note: If the corresponding transfer period received from the Initiator indicates a transfer period that is equal to 25 nSec ($M = 10$) and a REQ/ACK offset value of 1, 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”).

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” 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”).

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 3.1.19.2, “Synchronous Negotiation Started by the Initiator” and 3.1.19.3, “Synchronous Negotiation Started by the Target”.

Note: Target initiated synchronous negotiation is selectable via the Option Jumper Block. See 4.14, “Options Jumper Block” for more information.

3.1.20 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, Self Initiated Reset, or Transceiver Mode changed).

Target-initiated negotiation occurs either immediately following the Command phase or immediately following the first reconnection. 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 resets the synchronous agreement to asynchronous mode.

The Target’s width agreement is based on Mode Page 09h, (see 1.6.9, “Page 09h-Peripheral Device Parameters”), FORCE 8 bit being set to zero. FORCE 8 being set will prohibit wide transfers.

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, Self Initiated Reset, or Transceiver Mode changed), or a new data transfer width agreement is negotiated. If a Reset occurs, the Target goes to eight bit mode.

3.1.20.1 Wide Data Transfer Request Message Definition

Table 169. 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). Values of E greater than 01h are reserved.

3.1.20.2 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:

Table 170. Initiator Request/Target Response

Initiator Request	Target Response	Target Data Transfer Width
$E_i = 0$	$E_t = 0$	8 Bit Data Transfers
$E_i > 0$	$E_t = 1$	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"). For any other message, the Target completes negotiation and goes to the negotiated data transfer width.

3.1.20.3 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. If the FORCE8 bit is set, (see 1.6.9, "Page 09h - Peripheral Device Parameters"), the transfer width exponent equals 0 ($E = 0$). If the FORCE8 bit is not set, the transfer width exponent equals 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"). 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's corresponding transfer width exponent as shown in the following table:

Table 171. 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").

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" 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 3.6.2, "Message Out Phase Retry").

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 3.1.20.2, “Transfer Width Negotiation Started by the Initiator” and 3.1.20.3, “Transfer Width Negotiation Started by the Target”.

Note: Target initiated negotiation of wide transfers is selectable via the Option Jumper Block. See 4.14, “Options Jumper Block” 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 and Figure 2 for an example of the desired time to assert ATN during an information transfer phase.

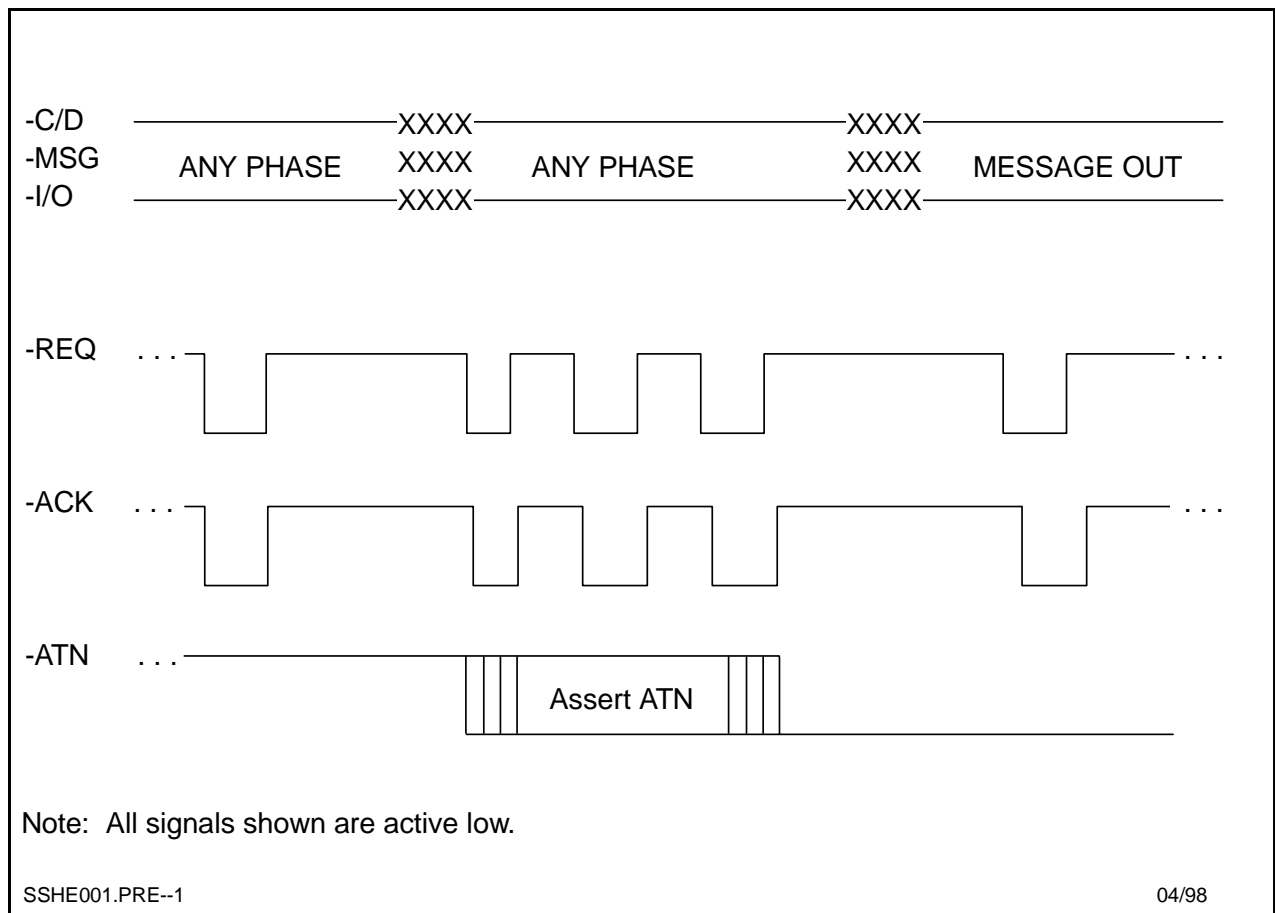


Figure 1. Attention Condition during Asynchronous Information Transfer phase

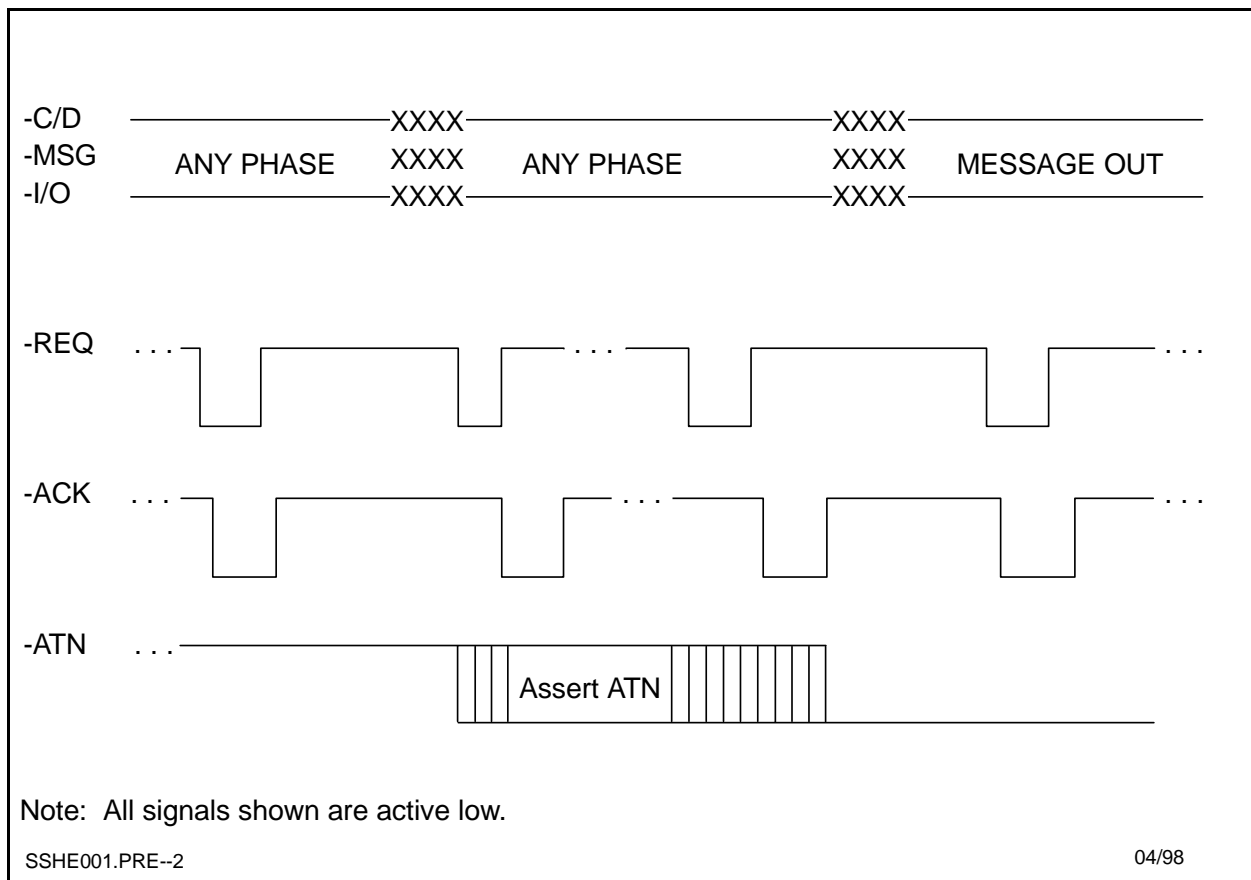


Figure 2. Attention Condition during Synchronous Data Transfer phase

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 may or may not be on a logical block boundary. The Initiator must continue REQ/ACK handshakes (asynchronous transfer) or ACK pulses until the Target enters Message Out phase. The logical block in which ATN is asserted may or may not be written to the disk.
- If ATN occurs during a parameter or sense data phase, the Target detects the Attention Condition within 256 bytes after the assertion of ATN. The Initiator must continue REQ/ACK handshakes (asynchronous transfer) or ACK pulses until the Target enters Message Out phase.
- 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”).

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” 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,” 3.6.5, “Status Phase Retry,” and 3.6.3, “Message In Phase Retry” 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” 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 logical block with the parity error 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” 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 172. 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 state.
- ACTION 1B & ACTION 2B are both done in response to Input 2 if this is the second error for this state.

The following Message Action Legend defines the meaning of each Target Action entry in the Message State/Action Table.

Table 173. 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
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
SVQ	=	Establish ITLQ Nexus and Save TAG if immediately after ID MSG, else message reject
SW	=	Send Wide Data Transfer Request message

Table 174. 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)
SCSI State Number	1	2	3	4	5	6	7	8	9
Valid Identify (80H, C0H)	SVL SVD (2)	SVD (24)	SVD (24)	SVD (24)	SVD (24)	SVD (24)	SVD (24)	SVD (24)	SVD (24)
Invalid Identify (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)
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)	SS CHK (19)	ND CD NP	ND NP	ID (8)	SQ (9)
								SS BF	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	RP (5)	SDP (6)	DM (7)	ID (8)	QT (9)
					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	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)

Table 174. Message State/Action Table (Part 1, cont.)

SCSI Phase or State Interrupted	Message In Sync Data Xfer Request (01h,01h) (Initiator Started)	Message In Sync Data Xfer Request (01h,01h) (Target Started)	Message In Wide Data Xfer Request (01h,03h) (Initiator 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	1	2	3	4	5	6	7	8	9
Target Detected SCSI Parity Error During Command	na	na	na	RP (5)	na	na	na	na	na
				SS CHK (19)					
Target Detected SCSI Parity Error During Message Out (response to	RMO (1)	RMO (2)	RMO (3)	RMO (4)	RMO (5)	RMO (6)	RMO (7)	RMO (8)	RMO (9)
	PS BF	SS1 BF	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF	SS BF

Table 175. Message State/Action Table (Part 2)

SCSI Phase or State Interrupted	Message In Sync Data Xferc Request (01h, 01h) (Initiator Started)	Message In Sync Data Xfer Request (01h, 01h) (Initiator Started)	Message In Wide Data Xfer Request (01h, 03h) (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, C0H)	SVD (24)	AT SVD (24)	SVD (24)	DW SVD SM (11)	SVD SS CHK (19)	SVD SS CHK (19)	SVD (24)	SVD (24)	SVD (24)
Invalid Identify (Different LUN)	SS BF	AT SS BF	SS BF	DW 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)
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)
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)
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)
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)
Abort (06H)	PS BF	AT PS BF	PS BF	DW 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)	IW (18)
									SS BF
No Operation (NOP) (08H)	NP	AT NP	NP	DW SM (11)	SS CHK (19)	SS CHK (19)	NP	NP	NP
Message Parity Error (09H)	SM (10)	SM (11)	SW (12)	SW (13)	SS BF	SS BF	SS BF	SS BF	IW (18)
	AT SS BF	AT SS BF	DW SS BF	DW SS BF					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	AT PS BF	PS BF	DW 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
Any Unsup-	MR (23)	AT MR (23)	MR (23)	DW MR (23)	MR SS CHK (19)	MR SS CHK (19)	MR (23)	MR (23)	MR (23)

ported or
Illegal
Message

--	--	--	--	--	--	--	--	--	--

Table 175. Message State/Action Table (Part 2, cont.)

SCSI Phase or State Interrupted	Message In Sync Data Xfer Request (01h, 01h) (Initiator Started)	Message In Sync Data Xfer Request (01h, 01h) (Initiator Started)	Message In Wide Data Xfer Request (01h, 03h) (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
Target Detected SCSI Parity Error During Command	na	na	na	na	na	na	na	na	na
Target Detected SCSI Parity Error During Message Out (response to ATN)	RMO (10)	RMO (11)	RMO (12)	RMO (13)	RMO (14)	RMO (15)	RMO (16)	RMO (17)	RMO (18)
	AT SS BF	AT SS BF	DW SS BF	DW SS BF	SS BF	SS BF	SS BF	SS BF	SS BF

Table 176. 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)	ATN Asserted Between Phases
SCSI State Number	19	20	21	22	23	24
Valid Identify (80H, C0H)	SVD (24)	SVD (24)	SVD (24)	SVD (24)	SVD (24)	SVD (24)
Invalid Identify (Different LUN)	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)
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)
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)
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)
Initiator Detected Error (05H)	RP (20)	SS BF	PS BF	SS BF	SS CHK (19)	SS CHK (19)
	SS BF					
Abort (06H)	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	MR(23)
No Operation (NOP) (08H)	NP	NP	NP	NP	NP	NP
Message Parity Error (09H)	SS BF	RP (20)	CC (21)	LCC (22)	MR (23)	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
Abort Tag Message (0Dh)	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
Any Unsupported or Illegal Message	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
Target Detected SCSI Parity Error During Message Out (response to ATN)	RMO (19)	RP(20)	RMO (21)	RMO (22)	RMO (23)	RMO (24)
	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 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"), 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 with the exception of the messages listed in 4.1.3, "Incorrect Initiator Connection."
- 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, "Support Messages"). 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.
Note: It is not considered an error if the message received during the Message Out phase is an Abort, Abort Tag, Clear Queue, or Bus Device Reset.
- 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. 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.”
- 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 phase 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 or Aborted Command/Internal Target Failure (0Bh/44h) for any other data phase interruption.

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, or Aborted Command/Internal Target Failure (0Bh/44h) for any other data phase interruption.

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”).
2. *Check Condition Status* for Incorrect Initiator Connection (see 4.1.3, “Incorrect Initiator Connection”).
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”).
4. *Check Condition Status* for *Unit Attention Condition* (see 4.1.5, “Unit Attention Condition”).
5. *Check Condition Status* during start-up or format operations (see 4.1.6, “Command Processing During Start-up and Format Operations”).
6. *Check Condition Status* for Internal Error condition (see 4.1.7, “Internal Error Condition”).
7. *Check Condition Status* for Deferred Error condition (see 4.1.8, “Deferred Error Condition”).
8. *Check Condition Status* for Degraded Mode condition (see 4.1.9, “Degraded Mode”).
9. *Reservation Conflict Status* (see 4.1.10, “Command Processing While Reserved”).
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 Identify message with an invalid LUN, and continues to Command phase to receive the CDB from the Initiator.

The Target's response to an invalid LUN varies with the command, as follows:

- | | |
|----------------------|---|
| Inquiry | 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"). |
| Request Sense | Execute the command, return the sense data with the Sense Key set to Illegal Request and the Additional Sense Code and Qualifier set to Logical Unit Not Supported , and return <i>Good Status</i> (see also 1.23, "Request Sense"). |
| All Others | 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.
- 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 in order to send an *Abort*, *Abort Tag*, *Clear Queue*, or *Bus Device Reset* message during the same MESSAGE OUT phase as the *Identify* message for any of the above conditions.

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”).

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” 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”).
 - The conditions to execute concurrently are met (see 4.4, “Concurrent Command Processing”).
- *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” 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”).
- The command is added to the command queue for an I_T_L nexus if the following conditions exist:
 - No Queue Tag message was received during the connection which established the I/O process.
 - Disconnection is allowed for the current I/O process.
 - 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.
 - 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 the following conditions exist:
 - A Queue Tag message was received during the connection which established the I/O process.
 - Tagged Queuing is enabled (DQue = 0).
 - An I/O process (either active or queued) exists at the Target.
 - Disconnection is allowed for the current I/O process.
 - 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.

- The command is not linked to a previous command.
- *Queue Full Status* is returned if one of the following conditions exist:
 - The command would otherwise be queued (according to the rules described above) but the command queue is full and all slots are utilized.
 - 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.
 - Tagged Queuing is enabled (DQue = 0) and a Format Unit command was previously queued but has not yet begun execution.
 - Tagged Queuing is enabled (DQue = 0), the Target is in a Degraded Mode (see 4.1.9, “Degraded Mode”), and a Start Unit command was previously queued but has not yet begun execution.
- *Busy Status* is returned if one of the following conditions exist:
 - Tagged Queuing is disabled (DQue = 1) and a Format Unit command was previously queued but has not yet begun execution.
 - Tagged Queuing is disabled (DQue = 1), the Target is in a Degraded Mode (see 4.1.9, “Degraded Mode”), and a Start Unit command was previously queued but has not yet begun execution.
 - 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” 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, Self Initiated reset, or Transceiver Mode changed. In all of these cases, a *Unit Attention Condition* is generated for each Initiator. For Power-On reset, SCSI Bus reset, Bus Device Reset message, and Self Initiated reset, the Additional Sense Code and Additional Sense Code Qualifier reported is ***Power On, Reset, or Bus Device Reset*** occurred. For Transceiver Mode changed the Additional Sense Code and Additional Sense code Qualifier reported is ***Transceiver Changed to SE*** or ***Transceiver Changed to LVD***

- 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***.

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:

Inquiry	Execute the command, return <i>Good Status</i> , and preserve the <i>Unit Attention Condition</i> .
Request Sense	Execute the command, return any pending sense data, return <i>Good Status</i> , and preserve the <i>Unit Attention Condition</i> . If there is not any pending sense data, the sense data associated with the highest priority <i>Unit Attention Condition</i> is returned and the highest priority <i>Unit Attention Condition</i> is cleared for this Initiator.
All Others	Do not execute the command, return a <i>Check Condition Status</i> , and clear the highest priority <i>Unit Attention Condition</i> for this Initiator. If the following command from this Initiator is Request Sense, then the sense data associated with highest priority <i>Unit Attention Condition</i> 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:

Inquiry	Execute the command, return <i>Good Status</i> , and preserve any lower priority <i>Unit Attention Condition(s)</i> for that Initiator (sense data for the highest priority <i>Unit Attention Condition</i> is lost).
Request Sense	Execute the command, return a Sense Key of <i>Unit Attention</i> and Additional Sense Code and Qualifier that corresponds to the highest priority <i>Unit Attention Condition</i> , return <i>Good Status</i> , and preserve any lower priority <i>Unit Attention Condition(s)</i> .
All Others	Execute the command unless prevented by another higher priority status, (see 4.1.1, "Priority of SCSI Status Byte Reporting" for more information) or another lower priority <i>Unit Attention Condition</i> , and return the appropriate status for the command. (Sense data for the highest priority <i>Unit Attention Condition</i> 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)

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 operation and the Target is in a Degraded Mode (see 4.1.9, “Degraded Mode”) or is executing a format operation, the Target's response varies with the command as follows:

Inquiry: Execute the command and return *Good Status*.

Request Sense: The Additional Sense Code Qualifier that is returned is dependent on the type of I/O processes that are active:
 For the Start/Stop Unit command and the Auto-start operation, the qualifier returned is *logical unit is in the process of becoming ready*. For the Format Unit command, the qualifier returned is *Logical Unit Not Ready*, Format in Progress, and the Sense Key specific bytes are set to return the progress indication.

Start/Stop Unit: If the start-up is the active operation as a result of a Start Unit Immediate command, and the received Start Unit is an Immediate command, the drive returns *Good Status*.
 Otherwise, if the start-up operation is active and another Start 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 the command is added to the command queue.
 Otherwise, the drive does not execute the command, it returns *Check Condition Status*. The Sense data generated is described in Request Sense above.

All Others: Do not execute the command and return *Check Condition Status*. 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:

- Inquiry:** Execute the command, return *Good Status*, and do not clear the Internal Error condition.
- Request Sense:** Execute the command, return the sense data generated by the Internal Error condition, return *Good Status*, and clear the Internal Error condition for that Initiator.
- All Others:** Do not execute the command, return *Check Condition Status*, 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:

Inquiry: Execute the command, return *Good Status*, and do not clear the Deferred Error condition.

Request Sense: Execute the command, return the Sense data generated by the Deferred Error condition, return *Good Status*, and clear the Deferred Error condition for that Initiator.

All Others: Do not execute the command, return *Check Condition Status*, 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.

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.
- Failure of a Send Diagnostic self-test or a start-up sequence. (*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 reads and/or writes various data records in the Reserved Area of the Drive during the start-up sequence or while executing certain SCSI commands. Failures to read or write any of these data records result in entering this Degraded Mode.
- 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. 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***.
4. 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***.
5. 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***.
6. 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/Write Buffer, Request Sense, Start/Stop Unit, Reserve, Release, Mode Sense, Mode Select(SP=0) commands		
Degraded Mode	Description	
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> • <i>Power on Self Test Degraded</i> 	Executes.	
	Success:	Reports <i>Good Status</i> .
	Failure:	Generates <i>Check Condition</i>
<ul style="list-style-type: none"> • <i>Format Degraded</i> • <i>Reassign Degraded</i> 		

Reassign Blocks command		
Degraded Mode	Description	
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> • <i>Power on Self Test Degraded</i> • <i>Format Degraded</i> 	Does not execute. Generates <i>Check Condition</i> .	
<i>Reassign Degraded</i>	Executes: Note: The LBA specified by the command must match the original LBA that was being reassigned in order to resume the Reassign Blocks operation. Otherwise, <i>Check Condition Status</i> is returned and the command is not executed.	
	Success:	Reports <i>Good Status</i> . Clears <i>Reassign Degraded</i> mode.
	Failure:	Generates <i>Check Condition</i> . <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"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> • <i>Power on Self Test Degraded</i> • <i>Format Degraded</i> • <i>Reassign Degraded</i> 	Does not execute. Generates <i>Check Condition</i> .	

Test Unit Ready command		
Degraded Mode	Description	
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> • <i>Power on Self Test Degraded</i> 	Executes. Generates <i>Check Condition</i> .	
<ul style="list-style-type: none"> • <i>Format Degraded</i> • <i>Reassign Degraded</i> (FRDD bit set to 0, see “Page 0h - Vendor Unique Parameters.”) 	Executes. Generates <i>Check Condition</i> .	
<ul style="list-style-type: none"> • <i>Format Degraded</i> • <i>Reassign Degraded</i> (FRDD bit set to 1, see “Page 0h - Vendor Unique Parameters.”) 	Executes. Reports <i>Good Status</i> or status other than Degraded Mode.	
	Success:	Reports <i>Good Status</i> .
	Failure:	Generates <i>Check Condition</i> .

Format Unit command		
Degraded Mode	Description	
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> • <i>Power on Self Test Degraded</i> 	Does not execute. Generates <i>Check Condition</i> .	
<i>Format Degraded</i>	Executes.	
	Success:	Reports <i>Good Status</i> . Clears <i>Format Degraded</i> . Clears <i>Reassign Degraded</i> if it exists internally.
	Failure:	Enters <i>Format Degraded</i> mode. Generates <i>Check Condition</i> .
<i>Reassign Degraded</i>	Executes.	
	Success:	Reports <i>Good Status</i> . Clears <i>Reassign Degraded</i> mode.
	Failure:	Enters <i>Format Degraded</i> mode. Generates <i>Check Condition</i> .

Send Diagnostic command	
Degraded Mode	Description
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> 	Does not execute. Generates <i>Check Condition</i> .
<i>Power on Self Test Degraded</i>	Executes.
	Success: Reports <i>Good Status</i> . Clears <i>Power on Self Test Degraded</i> mode.
	Failure: Enters <i>Power on Self Test Degraded</i> mode. Generates <i>Check Condition</i> .
<i>Format Degraded</i>	Executes.
	Success: Reports <i>Check Condition</i> with <i>Format Degraded</i> mode.
	Failure: Enters <i>Power on Self Test Degraded</i> mode. Generates <i>Check Condition</i> .
<i>Reassign Degraded</i>	Executes.
	Success: Reports <i>Good Status</i> . Clears <i>Reassign Degraded</i> mode.
	Failure: Generates <i>Check Condition</i> . 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.

Read, Read Extended, Read Capacity with PMI=1, Read Long, or Verify commands	
Degraded Mode	Description
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> 	Does not execute. Generates <i>Check Condition</i> .
<i>Power on Self Test Degraded</i>	Executes.
	Success: Reports <i>Good Status</i> .
	Failure: Generates <i>Check Condition</i> .
<i>Format Degraded</i>	Does not execute. Generates <i>Check Condition</i> with <i>Format Degraded</i> .
<i>Reassign Degraded</i>	Executes.
	Success: Reports <i>Good Status</i> .
	Failure: Generates <i>Check Condition</i> .

Mode Select with Save Parameters bit = 1	
Degraded Mode	Description
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> • <i>Power on Self Test Degraded</i> 	Does not execute. Generates <i>Check Condition</i> .
<ul style="list-style-type: none"> • <i>Format Degraded</i> • <i>Reassign Degraded</i> 	Executes.
	Success: Reports <i>Good Status</i> .
	Failure: Generates <i>Check Condition</i> .

All Other commands	
Degraded Mode	Description
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> 	Does not execute. Generates <i>Check Condition</i> .
<ul style="list-style-type: none"> • <i>Power on Self Test Degraded</i> • <i>Format Degraded</i> • <i>Reassign Degraded</i> 	Executes.
	Success: Reports <i>Good Status</i> .
	Failure: Generates <i>Check Condition</i> .

Degraded Mode affects the operation of the automatic rewrite/reassignment function and various internal Target idle time functions as shown in the following tables:

Automatic Rewrite/Reassignment function	
Degraded Mode	Description
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> • <i>Power on Self Test Degraded</i> • <i>Format Degraded</i> • <i>Reassign Degraded</i> 	Does not execute.

Update of Performance Counter Data in Reserved Area and Error Logging function	
Degraded Mode	Description
<ul style="list-style-type: none"> • <i>Motor Stop Degraded</i> • <i>Configuration Sector Degraded</i> • <i>Power on Self Test Degraded</i> 	Does not execute.
<ul style="list-style-type: none"> • <i>Format Degraded</i> • <i>Reassign Degraded</i> 	Executes.

Disk Sweep function	
Degraded Mode	Description
<i>Motor Stop Degraded</i>	Does not execute.
<ul style="list-style-type: none"> • <i>Configuration Sector Degraded</i> • <i>Power on Self Test Degraded</i> • <i>Format Degraded</i> • <i>Reassign Degraded</i> 	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” 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 Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec).

When this occurs, the Target executes the start-up sequence (see 4.17, “Bring-Up Sequence”). The Target attempts to clear the following Degraded Mode conditions:

- *Motor Stop Degraded*
 - *Configuration Sector 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. 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*
 - *Power on Self Test Degraded*
 - *Reassign Degraded*
- The Target is reset by a SCSI Bus reset or Bus Device Reset message, and we are not *Motor Stop Degraded* but another Degraded Mode Condition exists.

When this occurs, the Target executes the start-up sequence. The Target attempts to clear the following Degraded Mode conditions:

- *Configuration Sector 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*
- *Power on Self Test Degraded*
- *Reassign Degraded*

- A Start Unit command is executed while ***Motor Stop Degraded***, and the Target has previously completed 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 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” 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”).

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” 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” for more information.)
- Read, Write, Prefetch, Write and Verify, Write Same, and Verify Commands which cause an Invalid Field in CDB error.

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” for the circumstances in which I/O processes can be removed from the queue without executing.)

4.3.1 Untagged Queuing

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 enabled.

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 executed next, in the order received, or allow the Target to determine the order of execution. The target reordering of commands is controlled by the Mode

Select Page 0Ah parameter Queue Algorithm Modifier. See 1.6.10, “Page 0Ah - Control Mode Parameters” for more information. Commands received with *Head of Queue* tags are executed Last-In First-Out (LIFO). (See 3.1.15, “Queue Tag Messages (20h, 21h, 22h)” 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.”

The Target can queue up to 64 I/O processes. The value of the Disable Queuing (DQue) bit determines how many commands an initiator may queue.

When Tagged Queuing (DQue = 1) is disabled, each Initiator may queue 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 is Powered down.

The remaining queue slots are shared among all Initiators. The exact number of shared slots depends 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 processes in the queue. The number of I/O processes a particular Initiator can queue varies between 2 and 64 I/O processes depending on the number of Initiators that have sent an 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 64 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.

4.3.4 Rules for Queuing Commands

For a description of the rules for queuing commands, see “Command Processing During Execution of Active I/O Process.”

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” 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,” 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.”
- 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” 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 nonpriority 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”) are always allowed to execute concurrently regardless of the number of active I/O processes:
 - Inquiry
 - Request Sense
- Back-to-Back Write commands when Concurrent Processing Enable (CPE) bit is 1 (See 1.6.2, “Page 0h - Vendor Unique Parameters”).

When an I/O process ends in *Check Condition Status*, other active I/O processes from the same initiator to 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”).

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. This means that once the Data Out phase for the first write command has completed, the Data Out phase for the next command can be started, prior to status being sent for the first command. It is possible to have completed several Data phases for multiple write commands prior to having sent status for any of them when executing with CPE set to 1.

During Concurrent Processing, status is always returned for each command in the same order the data phases occurred. If one of the write commands returns *Check Condition Status*, the SCSI operations for the other active write command(s) are suspended until the pending sense data is cleared.

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 - Caching Parameters” and 4.8.4, “Write Cache” for more information). Execution of the second write command begins after the first write command returns status, possibly before the first write command has completed writing data to the media.

If an unrecoverable error occurs with Write Caching Enabled, WCE=1, the data following the error LBA that has been read into the cache will not be written to the media, even though *Good Status* may have been sent for the data. In this case, *Deferred Sense* will be built for all initiators.

4.5 Error Logs

The target maintains error logs in RAM and in the reserved area for use in failure analysis and Automatic/Recommend Reassignment. When an error occurs, it is entered into the logs maintained in RAM and is analyzed for Automatic/Recommend Reassignment. Under some circumstances 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.

4.6 Automatic Rewrite/Reallocate - Recommend Rewrite/Reassign (ARRR)

The target supports Auto and Recommend Reallocate for READ, WRITE and VERIFY, and VERIFY. (See 1.6, “Mode Select” 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 (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.

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 following table 177 lists all of the possible Sense Key/Code/Qualifier combinations that are valid for Auto Reallocate/Recommend Reassign.

It is important to note that Auto Reallocate/Recommend Reassign is affected by the settings of the PER, QPE, and RPFAE, and RARRRD bits. Refer to Table 71 for the various combination of settings for these bits and corresponding behavior.

The sense data combinations associated with auto/recommend rewrite/reallocate are listed below.

Table 177. Auto/Recommend Rewrite/Reallocate Sense Data Combinations.

Auto/Recommend Rewrite/Reallocate Sense data combinations. Sense data combinations for automatic or recommend rewrite or reallocate.			
Key	Code	Qual.	Description
3	11	04	Auto reallocate failed due to unrecovered read error.
3	11	0B	Unrecovered read error - Recommend Reassign.
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.

4.7 Predictive Failure Analysis (PFA)

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 RPFAE descriptions in 1.6.2, "Page 0h - Vendor Unique Parameters" and in 1.6.13, "Page 1Ch - Informational Exceptions Control."

4.8 Segmented Caching

4.8.1 Overview

Segmented Caching divides the data buffer into several smaller buffer segments. Each buffer segment is used as a separate command 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.

4.8.2 Options Supported

- Log Sense Vendor Unique Page 35h: Cache Hit/Miss
 - Cumulative Cache Full Hits on Reads
 - Cumulative Cache Partial Hits on Reads
 - Cumulative Cache Misses on Reads
- Mode Select Page 0: Vendor Unique Parameters
 - ADC: Adaptive Caching
- Mode Select Page 8: Caching Parameters
 - IC: Initiator Control
 - ABPF: Abort Pre-Fetch
 - CAP: Caching Analysis Permitted
 - DISC: Discontinuity
 - SIZE: Size Enable
 - RCD: Read Cache Disable
 - MF: Multiplication Factor
 - WCE: Write Cache Enable
 - Demand Read Retention Priority
 - Write Retention Priority
 - Disable Pre-Fetch Transfer Length
 - Minimum Pre-Fetch
 - Maximum Pre-Fetch Ceiling
 - FSW: Force Sequential Writes
 - LBCSS: Logical Block Cache Size
 - DRA: Disable Read-Ahead
 - Number of Cache Segments
 - Cache Segment Size
 - Non-Cache Segment Size
- Pre-Fetch Command
- Read-Ahead for Read commands
- Write Retention for Write and Write And 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 And Verify Commands

See 1.5.11, "Log Sense Page 35h", 1.6.8, "Page 8h - Caching Parameters", 1.12, "Read (10)", and 1.35, "Write (10)" for more details on the caching options.

4.8.3 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 - Caching Parameters"). 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 DRA (Disable Read-Ahead) 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.

The number of Read-Ahead blocks is 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 and whether or not the ADC bit is set = 1.
- 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.
- Whether or not a new command received while Read-Ahead is executing causes the Read-Ahead to be aborted depends upon how much data has already been read ahead and the value of the Minimum Pre-fetch and ABPF parameters. If the amount of data prefetched (or read ahead) does not meet or exceed the Minimum Pre-Fetch parameter value then the Read-Ahead is not aborted. If the amount prefetched does exceed that value, then the Read-Ahead may be aborted. (See 1.6.8, "Caching Parameters" for more information about Minimum Pre-fetch.) If ABPF = 1, this takes precedence over

the Minimum Prefetch value and Read-Ahead is aborted when a new command is received regardless of whether the Minimum Prefetch value was met or not.

- If ADC (Adaptive Caching) bit in the Mode Select Page 0 is on, the number of Read-Ahead blocks may be reduced if the drive is receiving non-sequential commands.

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), regardless of the state of the Mode Page 8 DISC parameter, 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"), the Target:
 1. Continues the current Read-Ahead operation.
 2. Executes the priority command.
 - If the command is a Read 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, assuming Minimum Pre-fetch requirements are met or the Mode Page 8 ABPF bit is set.
 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 4.8.6, "Segment Selection" 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 (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, assuming Minimum Pre-fetch requirements are met or the Mode Page 8 ABPF bit is set.

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 4.8.6, "Segment Selection" 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:
 - 1) 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.
 - 2) 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 And Verify operation.
 - 3) 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:
 - 1) Transfers the requested data from the data buffer segment.
 - 2) Does not access the media and does not start a Read-Ahead operation.
- If the command is a Read whose starting LBA is in the data buffer segment for the current Read-Ahead operation and FUA is off, the Target:
 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, assuming Minimum Pre-fetch requirements are met or the Mode Page 8 ABPF bit is set.
 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 4.8.6, "Segment Selection" 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"), 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 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 4.8.6, "Segment Selection" 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 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 4.8.6, "Segment Selection" 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:
 - 1) 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.
 - 2) 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 And Verify operation.
 - 3) 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:
 - 1) Transfers the requested data from the data buffer segment.
 - 2) 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 4.8.6, "Segment Selection" for more information.)

4.8.4 Write Cache

When the drive's Mode Parameter $WCE = 0$, a write-through caching algorithm is employed for data transferred as a result of a write command. This means that a Command Complete message is not returned from the Target to the Initiator until the requested data is successfully stored on the magnetic media. And the data is cached following the completion of the write command. This data resides in a cache segment such that subsequent read commands that access the same LBAs can get cache hits if that data is not overwritten with

other LBAs from intervening requests. See the following sections on segment selection and data retention/prioritization for more details.

When $WCE = 1$, a write-back caching algorithm is employed for data transferred as a result of a write command. This means that command complete is returned from the Target to the Initiator before requested data is successfully stored on the magnetic media. The data is cached following the completion of the write command for use in possible cache hits for subsequent read commands if $RCD = 0$.

When both WCE and $FSW = 1$, the drive will not write addressed logical blocks to the magnetic media in an order different than it received those blocks over the SCSI bus. If $WCE = 1$ and $FSW = 0$, the drive may alter the order in which it writes those blocks to the media such that it is different than the order in which it received them over the SCSI bus.

See 4.4.1, "Back to Back Writes" for more particular usage information.

4.8.5 Synchronize Cache on SCSI Reset

Write data stored in the cache as a result of a write command, when $WCE = 1$, that has not been written to the media when a SCSI Reset is received is synchronized/destaged as part of the Reset processing. Thus coherence between what a system believes the drive has written to the magnetic media and what it actually has written is maintained across SCSI Resets.

4.8.6 Segment Selection

A data buffer segment must be selected before any operation that reads or writes the media can be started.

1. If the command is one of the following:

- Read
- Pre-Fetch
- Verify
- Write

the Cache is scanned to see if the start of the Requested Data is already in any of the segments. If it is, that segment is selected. If it is not in the cache, then the following Priority/Least Recently Used (LRU) algorithm is used to select the segment:

- 1) An Empty segment is chosen.
- 2) If there are no Empty segments, the oldest Low Priority segment is chosen.
- 3) If there are no Low Priority segments, the oldest Middle Priority segment is chosen.
- 4) 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.

2. If the command is one of the following:

- Mode Select with the Save Parameter bit on
- Read Long
- Verify with the BytChk bit on
- Write Long
- Write Same
- Write and Verify

the previously described Priority/LRU algorithm is used to select the segment.

4.8.7 Data Discard

If the command is one of the following:

- Verify with the BytChk bit on
- Write
- Write And Verify
- Write Long
- Write Same

the Cache is scanned to see if the Requested Data is also located in unselected cache segments. All data within the range of the Requested Data located in segments that are not selected is invalidated, or "discarded" so that it cannot be used for cache hits in subsequent operations.

4.8.8 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 with the Save Parameter bit on
 - Read Long
 - Verify with the BytChk bit on
 - Write Long
 - Write Same
2. If the command is one of the following, all segments are marked empty:
 - Format Unit
 - Mode Select to Page 8 or Vendor Unique Page 0
 - Reassign Block
 - Send Diagnostic
 - Write Buffer
 - Read Defect Data (10)
 - Read Defect Data (12)
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:
 - 1) The DPO bit is on, the segment is assigned Lowest Priority.
 - 2) 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:
 - 1) The DPO bit is on, the segment is assigned low Priority.
 - 2) 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:
- 1) The DPO bit is on, the segment is assigned Lowest Priority.
 - 2) 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

4.9 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 up to the number of times specified in Reselection Retry Count (see 1.6.9, "Page 09h - Peripheral Device Parameters"). If the target exhausts the retries and fails to reselect the initiator, the target generates sense data with a Sense Key of Aborted Command and an Additional Sense Code of Select/Reselect failure (45h) and sets *Check Condition Status*. If the target is successful in reselecting the initiator before it exhausts the retries, command execution is resumed and no sense data is generated.

4.10 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.

4.11 Nonarbitrating 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.

4.12 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.

4.13 Multiple Initiator Systems

This section describes how the target supports Multiple initiator systems.

4.13.1 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.

4.13.2 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" for more information.

4.13.3 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.

4.14 Options Jumper Block

The Drive provides an Options Jumper Block which allows the drive to be customized using jumpers instead of software settings. Refer to the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec for detailed information on the location and function of these jumpers. Functions listed below are controlled via the options jumper block.

- Auto Start
- LED output
- Write Protect
- Disabling Target Initiated Synchronous Negotiation
- Disabling Target Initiated Wide Transfer Negotiation
- Disabling Unit Attentions
- Disabling SCSI Parity
- Auto Start Delay

4.15 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 0h-Vendor Unique Parameters" for information regarding the CMDAC and LED Mode fields. The "Jumper Settings" section of the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec shows the location of the LED pin.

4.16 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.

4.16.1 Reset Sources

There are five 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	This is the reset generated by the SCSI Bus Device Reset Message (0Ch).

message

Transceiver Mode changed This is a reset generated when the Differential Sense line changes state.

4.16.2 Reset Actions

The action taken by the Drive following a reset is dependent on the source of the reset.

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 4.17, "Bring-Up Sequence") to put the Drive in a ready state

SCSI Bus reset, SCSI Bus Device Reset message, and Transceiver Mode changed

These three 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.

4.17 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 "Test Unit Ready" 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 4.18.1, "BATS1 Tests")
2. Enable SCSI Bus
(The Target can only respond to SCSI commands after the power-up sequence is complete.)

Note: Refer to the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec 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.

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 4.19, "Reserved Area"). Set ***Configuration Sector Degraded*** if the configuration record cannot be read.
3. Electronics/Disk Enclosure (DE) Compatibility Test.
4. Read the remaining configuration information from the Reserved Area.
5. Perform BATS2 testing (see 4.18.2, "BATS2 Tests.")
6. Check for an interrupted Format Unit operation.
7. Resume a checkpointed Reassign Blocks command or automatic reallocation (see 4.1.9.4, "Existing Degraded Mode.")

4.18 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.

4.18.1 BATS1 Tests

BATS1 consists of the following tests:

- Microprocessor Functional Test
- Controller Hardware Test
- ROM Checksum Test
- Microprocessor RAM Test
- Test enough of the Data Buffer to install the IP Direct Access DRAM and the SCSI Sequencer. Fails if not enough good DRAM.

4.18.2 BATS2 Tests

BATS2 consists of the following tests:

- Data Buffer Test
- Seek Test
- Head Offset Test
- Read Test (all heads)
- Write Test (all heads)
- Cyclic Redundancy Code (CRC) Test
- Error Correction Code (ECC) Test
- Data Cyclical Redundancy Code (CRC) Test

4.19 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 failure analysis.
- Rewrite/Reallocation
- 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

Appendix A. SCSI Sense Data Format

Following is the format of data returned by the Target in response to the Request Sense command.

Table 178. 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 - 29	Physical Error Record (See Description)							
30 - 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 29) 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 29) 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 29) 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 29) 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 29) 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 29) 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 or FFFFFFFFh if the first LBA cannot be determined. Refer to 1.18, "Reassign Blocks" for more information.

If an *Auto-Reallocation*, this field contains the LBA that was not reassigned.

Byte 12 Additional Sense Code.

Byte 13 Additional Sense Code Qualifier.

Note: In Table 179, 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 179, the Key, Code, and Qualifier fields are all hex values. (i.e., Sense Key E is Eh, Sense Code 0C is 0Ch, etc.)

Table 179. Valid Sense Key, Code, Qualifier Combinations Used by the Drive.

Key	Code	Qual	Description	Initiator Recovery
<i>Sense Key = No Sense</i>				
0	00	00	<i>No Additional Sense Information</i> (00 00) No error.	None
<i>Sense Key = Recovered Error</i>				
1	01	00	<i>No Index/Sector Signal</i> (01 C2) Fake and Extra Index. (01 C9) Write with No Sector Pulses.	Appendix C.4.3, "Recovered Drive Error"
1	02	00	<i>No Seek Complete</i> (01 C0) Too many missing Servo IDs detected by the Controller or Channel Hardware (02 53) Servo error - Gray Code window exceed 9 tracks (Position). (02 54) Servo error - Gray Code window exceed 3 tracks (Track Follow). (02 65) Servo error - Three consecutive bad SIDs (seeking). (02 66) Servo error - Three consecutive bad SIDs (half track). (02 67) Servo error - Seek timeout. (02 8D) Servo error - ASIC/SP Sector Count mismatch.	Appendix C.4.3, "Recovered Drive Error"
1	03	00	<i>Peripheral Device Write Fault</i> (01 6B) Arm Electronics Not Ready. (01 C1) Arm Electronics error. (01 C4) Sector overrun error. (01 C5) Interface Processor write inhibit error. (01 C7) Microjog Write Inhibit. (01 CD) IP Retract Error. (02 1B) External Write Inhibit.	Appendix C.4.3, "Recovered Drive Error"
1	09	00	<i>Track Following Error</i> (02 60) Servo error - Excessive Runout - Recalibrate requested from Servo. (02 61) Servo error - Timed out waiting for PES interrupt. (02 62) Servo error - Settle timeout. (02 63) Servo error - Three consecutive bad SIDs (track follow). (02 64) Servo error - Three consecutive bad SIDs (settle).	Appendix C.4.3, "Recovered Drive Error"
1	0B	01	<i>Temperature Warning Error</i>	None

			(02 2F) Specified File Temperature Exceeded.	
1	16	00	Data Synchronization Mark Error (01 D1) No Data Field sync byte found. (01 D5) Data Sync error detected while outside of the write band. (01 DB) Erroneous Sync Byte Found.	Appendix C.3.3, "Data Error Logging"
1	16	01	Data Synchronization Mark Error - Data Rewritten (01 D1) No Data Field sync byte found, Data Rewritten (01 D5) No Data Field sync byte found, Data Rewritten (01 DB) Erroneous Sync Byte Found, Data Rewritten.	Appendix C.3.3, "Data Error Logging"
1	16	02	Data Synchronization Mark Error - Recommend Rewrite (01 D1) No Data Field sync byte found, Recommend Rewrite (01 D5) No Data Field sync byte found, Recommend Rewrite (01 DB) Erroneous Sync Byte Found, Recommend Rewrite.	Appendix C.3.1, "Rewrite an LBA"
1	16	03	Data Synchronization Mark Error - Data Auto-Reallocated (01 D1) No Data Field sync byte found, Data Auto Reallocated (01 D5) No Data Field sync byte found, Data Auto Reallocated (01 DB) Erroneous Sync Byte Found, Data Auto Reallocated.	Appendix C.3.3, "Data Error Logging"
1	16	04	Data Synchronization Mark Error - Recommend Reassignment (01 D1) No Data Field sync byte found, Recommend Reassignment (01 D5) No Data Field sync byte found, Recommend Reassignment (01 DB) Erroneous Sync Byte Found, Recommend Reassignment.	Appendix C.3.2, "Reassign a Physical Sector"
1	17	01	Recovered Data with Retries (01 D4) ECC check corrected without using ECC correction. (01 D8) ECC Error Detected while outside of write band corrected without ECC.	Appendix C.3.3, "Data Error Logging"
1	17	02	Recovered Data with Positive Head Offset (01 D9) Data recovered using positive offsets.	Appendix C.3.3, "Data Error Logging"
1	17	03	Recovered Data with Negative Head Offset (01 DA) Data recovered using negative offsets.	Appendix C.3.3, "Data Error Logging"
1	17	06	Recovered Data Without ECC - Data Auto-Reallocated (01 D4) Recovered data without ECC, Auto Reallocated. (01 D8) Recovered data without ECC, Auto Reallocated. (01 D9) Recovered data without ECC, Auto Reallocated. (01 DA) Recovered data without ECC, Auto Reallocated. (01 DC) Recovered data without ECC, Auto Reallocated.	Appendix C.3.3, "Data Error Logging"
1	17	07	Recovered Data Without ECC - Recommend Reassignment (01 D4) Recovered data without ECC, Recommend Reassignment. (01 D8) Recovered data without ECC, Recommend Reassignment. (01 D9) Recovered data without ECC, Recommend Reassignment. (01 DA) Recovered data without ECC, Recommend Reassignment.	Appendix C.3.2, "Reassign a Physical Sector"
1	17	08	Recovered Data Without ECC - Recommend Rewrite (01 D4) Recovered data without ECC, Recommend Rewrite. (01 D8) Recovered data without ECC, Recommend Rewrite. (01 D9) Recovered data without ECC, Recommend Rewrite. (01 DA) Recovered data without ECC, Recommend Rewrite.	Appendix C.3.1, "Rewrite an LBA"
1	17	09	Recovered Data Without ECC - Data Rewritten (01 D4) Recovered data without ECC, Data Rewritten. (01 D8) Recovered data without ECC, Data Rewritten. (01 D9) Recovered data without ECC, Data Rewritten.	Appendix C.3.3, "Data Error Logging"

			(01 DA) Recovered data without ECC, Data Rewritten.	
1	18	01	Recovered Data with Error Correction and Retries Applied (01 D3) Data correction applied to Drive data for a Data ECC check. (01 D7) ECC Error Detected while outside of write band corrected with ECC.	Appendix C.3.3, "Data Error Logging"
1	18	02	Recovered Data - Data Auto-Reallocated (01 D3) Recovered data with ECC, Auto Reallocated. (01 D7) Recovered data with ECC, Auto Reallocated.	Appendix C.3.3, "Data Error Logging"
1	18	05	Recovered Data - Recommend Reassignment (01 D3) Recovered data with ECC, Recommend Reassignment (01 D7) Recovered data with ECC, Recommend Reassignment	Appendix C.3.2, "Reassign a Physical Sector"
1	18	06	Recovered Data With ECC - Recommend Rewrite (01 D3) Recovered data with ECC, Recommend Rewrite. (01 D7) Recovered data with ECC, Recommend Rewrite.	Appendix C.3.1, "Rewrite an LBA"
1	18	07	Recovered Data With ECC - Data Rewritten (01 D3) Recovered data with ECC, Data Rewritten. (01 D7) Recovered data with ECC, Data Rewritten.	Appendix C.3.3, "Data Error Logging"
1	1C	01	Primary Defect List Not Found (01 06) Requested P List does not match returned list format (READ DEFECT DATA only)	Appendix C.4.13, "Defect List Recovery"
1	1C	02	Grown Defect List Not Found (01 07) Requested G List does not match returned list format. (READ DEFECT DATA only)	Appendix C.4.13, "Defect List Recovery"
1	1F	00	Partial Defect List Transferred (01 0A) Defect list longer than 64k, 64k of data returned. (READ DEFECT DATA only)	Appendix C.4.13, "Defect List Recovery"
1	44	00	Internal Target Failure (01 5D) No Sector Found caused by hardware fault or software. (01 69) Servo/Fileside Interlock Head Compare Error. (01 86) A servo command is already active. (01 87) Interface Processor detected Servo Sanity Error (01 A0) Error reading NOID Tables (01 C6) Read Write Ready Dropped Error. (01 C8) Interrupt Occurred with no interrupt bits set. (01 CB) Motor Speed Error. (02 16) Servo ID overrun Error (02 18) Arm Electronics (AE) Idle Error (02 19) Interface Processor Ready Timeout Error (02 34) NOID Sequence Error. (02 4C) SID Estimator Error. (02 55) Servo error - KF Adjustment too large. (02 57) Servo error - AE serial port read-back miscompare	Appendix C.4.3, "Recovered Drive Error"
1	44	00	Internal Target Failure (02 58) Servo error - AE select error (wrong AE selected) (02 59) Servo error - Detected AE serial port sequence error (02 5A) Servo error - AE serial port timeout (stuck busy) (02 5B) Servo error - Head control port not taking effect (2ms timeout) (02 5C) Servo error - Spindle serial port timeout (stuck busy) (02 5D) Servo error - VCM not enabled in Voltage Mode (02 69) Servo error - Command not accepted while actuator in retract. (02 6C) Servo error - Head number out of range. (02 6E) Servo error - Invalid command. (02 6F) Servo error - Estimator Position Out Of Range.	Appendix C.4.3, "Recovered Drive Error"

			<p>(02 70) Servo error - Offset out of range. (02 71) Servo error - Unable to orient sector after breaking free of latch. (02 72) Servo error - Unable to break free of latch or no SID detected. (02 73) Servo error - Unable to break free of latch or no SID detected, while in internal recovery mode. (02 74) Servo error - Unable to achieve track following after breaking free of latch. (02 75) Servo error - Not track following when recalibrate clean-up was entered.</p>	
1	44	00	<p>Internal Target Failure (02 77) Servo error - Unable to achieve track follow after breaking free of latch, while in servo internal recovery mode. (02 7F) Servo error - AGC limit exceeded. (02 80) Servo error - Command rejected because the motor is not spinning or motor is not in speed tolerance. (02 81) Servo error - Seek timeout during KT Calibration. (02 83) Servo error - Seek timeout during AGC calibration. (02 84) Servo error - Command rejected - Invalid Recalibrate command qualifier. (02 85) Servo error - AGC saturated during calibration. (02 8F) Servo error - Real velocity exceeded - 2.75m/s four consecutive times.</p>	Appendix C.4.3, "Recovered Drive Error"
1	5D	00	<p>Predictive Failure Analysis Threshold Reached on Recovered Error (01 63) Media Problem, Recommend Device Replacement (01 64) Hardware Problem, Recommend Device Replacement (02 20) Channel Noise Problem, Recommend Device Replacement (02 21) Channel Asymmetry Problem, Recommend Device Replacement (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</p>	Appendix C.4.16, "Predictive Failure Analysis"
1	5D	FF	<p>Predictive Failure Analysis Threshold Reached for Test Error. (02 2A) Test PFA Error. Generated when Test Bit in Mode Page 0x1C is set.</p>	None
Sense Key = <i>Not Ready</i>				
2	04	00	<p>Logical Unit Not Ready Cause Not Reportable (01 1B) Motor Start Failed due to Timer 1 being disabled (01 5A) Motor is Stuck, Cannot be started. (01 84) Motor timeout 3 tries limit. (01 85) Motor gross timeout during presettle. (01 8B) Motor gross timeout during postsettle.</p>	Appendix C.4.4, "Drive Not Ready"
2	04	01	<p>Logical Unit Is In The Process of Becoming Ready (01 02) Unavailable while Start/Stop Unit Command active. (01 03) Unavailable while Bring-up active.</p>	Appendix C.4.1, "Drive Busy"
2	04	02	<p>Logical Unit Not Ready, initializing command required (01 01) Degraded Mode - Motor not running.</p>	Appendix C.4.6, "Degraded Mode"
2	04	04	<p>Logical Unit Not Ready, Format in Progress (01 04) Unavailable while Format active.</p>	Appendix C.4.1, "Drive Busy"
2	31	00	<p>Medium Format Corrupted Reassign Failed (01 11) Degraded Mode - Reassign Blocks unsuccessful after pushdown started.</p>	Appendix C.4.6, "Degraded Mode"

2	31	01	Format Command Failed (01 12) Degraded Mode - Format unsuccessful. (01 17) NOID Table build failed during a Format Unit command	Appendix C.4.6, "Degraded Mode"
2	40	80	Diagnostic Failure (01 22) Degraded Mode - Bringup not successful.	Appendix C.4.6, "Degraded Mode"
2	40	B0	Diagnostic Failure (01 96) BATS#2 Error - Head Offset Test failure. (01 98) Self Initiated Reset - Cause Unknown. (01 99) Self Initiated Reset - SCSI Controller Chip Reset unsuccessful. (01 9A) Self Initiated Reset - IP direct access redundancy error. (01 9B) Self Initiated Reset - Zero Divide Error. (01 9C) Self Initiated Reset - Control Store Address Fault. (01 9D) Self Initiated Reset - Unused Op Code. (01 9F) Self Initiated Reset - Invalid Queue Operation.	Appendix C.4.12, "Self Initiated Reset"
2	4C	00	Logical Unit Failed Self-Configuration (01 13) Degraded Mode - Configuration not loaded.	Appendix C.4.6, "Degraded Mode"
Sense Key = Medium Error				
3	11	00	Unrecovered Read Error (01 D2) Data ECC Check (01 D6) Data ECC Check detected while outside of the write band.	Appendix C.3.3, "Data Error Logging"
3	11	04	Unrecovered Read Error-Auto-Reallocation Failed (01 D1) Recovered Read Error, Auto Reallocate failed because of unreadable data. (01 D3) Recovered Read Error, Auto Reallocate failed because of unreadable data. (01 D4) Recovered Read Error, Auto Reallocate failed because of unreadable data. (01 D5) Recovered Read Error, Auto Reallocate failed because of unreadable data. (01 D7) Recovered Read Error, Auto Reallocate failed because of unreadable data. (01 D8) Recovered Read Error, Auto Reallocate failed because of unreadable data. (01 D9) Recovered Read Error, Auto Reallocate failed because of unreadable data. (01 DA) Recovered Read Error, Auto Reallocate failed because of unreadable data. (01 DB) Recovered Read Error, Auto Reallocate failed because of unreadable data.	Appendix C.3.4, "Reassign Blocks Recovered"
3	11	0B	Unrecovered Read Error-Recommend Reassignment (01 D2) Unrecovered Read Error, Recommend Reassignment (01 D6) Unrecovered Read Error, Recommend Reassignment	Appendix C.3.2, "Reassign a Physical Sector"
3	16	00	Data Synchronization Mark Error (01 D1) No Data Field sync byte found. (01 D5) Data Sync error detected while outside of the write band. (01 DB) Erroneous Sync Byte Found.	Appendix C.3.3, "Data Error Logging"
3	16	04	Data Synchronization Mark Error - Recommend Reassignment (01 D1) No Data Field sync byte found, Recommend Reassignment (01 D5) No Data Field sync byte found, Recommend Reassignment (01 DB) Erroneous Sync Byte Found, Recommend Reassignment.	Appendix C.3.2, "Reassign a Physical Sector"
3	19	02	Defect List Error in Primary List. (01 65) Error in Primary Defect list (READ DEFECT DATA only)	Appendix C.4.7,

				"Reserved Area Hard Error"
3	19	03	Defect List Error in Grown List. (01 66) Error in Grown Defect list (READ DEFECT DATA only)	Appendix C.4.7, "Reserved Area Hard Error"
3	31	00	Medium Format Corrupted Reassign Failed (01 11) Degraded Mode - Reassign Blocks unsuccessful after pushdown started. (0x xx) Unrecovered Read Error of Customer Data during Reassign after pushdown started.	Appendix C.3.4, "Reassign Blocks Recovery"
3	31	01	Format Failed (01 12) Degraded Mode - Format unsuccessful. (01 17) NOID Table build failed during a Format Unit command	Appendix C.4.6, "Degraded Mode"
3	80	00	Data Auto-Reallocated-Data Awaiting Restoration (01 DD) Site awaiting data restoration. This site was auto reallocated because of a Hard Error.	None
Sense Key = <i>Hardware Error</i>				
4	01	00	No Index/Sector Signal (01 C2) Fake and Extra Index. (01 C9) Write with No Sector Pulses. (02 1C) Error in Target Sector Generation logic.	Appendix C.4.2, "Unrecovered Drive Error"
4	02	00	No Seek Complete (01 8A) Servo processor did not finish command in time. (01 C0) Too many missing Servo IDs detected by the Controller or Channel Hardware. (02 52) Servo error-Power dissipation too high. (02 53) Servo error-Gray Code window exceed 9 tracks (Position). (02 54) Servo error-Gray Code window exceed 3 tracks (Track Follow). (02 65) Servo error-Three consecutive bad SIDs (seeking). (02 66) Servo error-Three consecutive bad SIDs (half track). (02 67) Servo error-Seek Timeout. (02 6A) Servo error-Three consecutive invalid Gray codes. (02 6B) Servo error-Estimator error saturated. (02 6D) Servo error-Gray code cylinder mismatch. (02 8D) Servo error-ASIC/SP Sector Count mismatch.	Appendix C.4.2, "Unrecovered Drive Error"
4	03	00	Peripheral Device Write Fault (01 6B) Arm electronics not ready. (01 C1) Arm electronics error. (01 C4) Sector overrun error. (01 C5) Interface Processor write inhibit error. (01 C7) Microjog Write Inhibit. (01 CD) IP Retract Error. (02 1B) External Write Inhibit.	Appendix C.4.2, "Unrecovered Drive Error"
4	09	00	Track Following Error (02 60) Servo error - Excessive Runout - Recalibrate requested from Servo. (02 61) Servo error - Timed out waiting for PES interrupt. (02 62) Servo error - Settle timeout. (02 63) Servo error - Three consecutive bad SIDs (track follow). (02 64) Servo error - Three consecutive bad SIDs (settle).	Appendix C.4.2, "Unrecovered Drive Error"
4	11	00	Unrecovered Read Error in Reserved Area (01 D2) Data ECC Check (Reserved Area)	Appendix C.4.7,

				"Reserved Area Hard Error"
4	16	00	Data Synchronization Mark Error in Reserved Area (01 D1) No Data Field sync byte found. (Reserved Area) (01 DB) Erroneous Sync Byte Found. (Reserved Area)	Appendix C.4.7, "Reserved Area Hard Error"
4	19	02	Defect List Error in Primary List. (01 5F) Error in Primary Defect list.	Appendix C.4.7, "Reserved Area Hard Error"
4	19	03	Defect List Error in Grown List. (01 81) Error in Grown Defect list (used by Format Unit and Reassign Block commands).	Appendix C.4.7, "Reserved Area Hard Error"
4	31	00	Medium Format Corrupted Reassign Failed (0x xx) Unrecovered Hardware or Reserved Area data error during Reassign after pushdown started.	Appendix C.3.4, "Reassign Blocks Recovery"
4	32	00	No Defect Spare Location Available (01 56) GLIST full - cannot add more entries. (01 57) The defects per track limit was exceeded during a Format or Reassign operation. (01 5E) No spare sectors remaining.	Appendix C.4.5, "No Defect Spare"
4	40	80	Diagnostic Failure (01 50) Microcode Check Sum error detected during ROS Test. (01 89) Reserved area sector valid check failed. (01 8C) Configuration Sector valid check failed. (01 8D) Configuration Sector uploaded but Check Sum error. (01 8E) Reserved area sector version check failed.	Appendix C.4.6, "Degraded Mode"
4	40	85	Diagnostic Failure (01 20) Microcode Check Sum error detected during download of Microcode.	Appendix C.4.15, "Microcode Error"
4	40	90	Diagnostic Failure (01 95) BATS#2 Error - Seek test failure. (01 96) BATS#2 Error - Head Offset Test failure. (02 90) BATS#2 Error (Split Field Table checksum error) (02 91) BATS#2 Error (Split Field Table address error) (02 92) BATS#2 Error (Split Field Table stuck bit error)	Appendix C.4.6, "Degraded Mode"
4	40	A0	Diagnostic Failure (01 93) BATS#2 Error - Read/Write test failure. (01 94) BATS#2 Error - ECC/CRC test failure. (01 97) BATS#2 Error - Hardware XOR Test Failure. (02 93) BATS#2 Error (CRC test failure, mode 1). (02 94) BATS#2 Error (CRC test failure, mode 2). (02 95) BATS#2 Error (CRC test failure, mode 3).	Appendix C.4.6, "Degraded Mode"
4	40	B0	Diagnostic Failure (01 90) Self Initiated Reset - Invalid Input. (01 96) BATS#2 Error - Head Offset Test failure. (01 98) Self Initiated Reset - Cause Unknown. (01 99) Self Initiated Reset - SCSI Controller Chip Reset unsuccessful. (01 9A) Self Initiated Reset - IP direct access redundancy error. (01 9B) Self Initiated Reset - Zero Divide Error. (01 9C) Self Initiated Reset - Control Store Address Fault (01 9D) Self Initiated Reset - Unused Op Code. (01 9F) Self Initiated Reset - Invalid Queue Operation. (01 A1) Controller "Ready" Timeout Error .	Appendix C.4.12, "Self Initiated Reset"

4	40	D0	<p>Diagnostic Failure</p> <p>(01 24) Mismatch between the Servo Processor ROS and DE. (01 2F) Mismatch between the Interface Processor ROS and the DE.</p>	Appendix C.4.11, "Components Mismatch"
4	44	00	<p>Internal Target Failure Defect List Error prevented one or more defects from being used in a Format Unit command or from being reported</p> <p>(01 09) LBA to PBA Conversion Timeout Error (01 0C) NOID Table integrity error detected. (01 0F) Error detected while reading the NOID Tables from disk. (01 10) Too few valid GEM measurements available to perform a GEM Predictive Failure Analysis. (01 16) ROS Microcode Download failed. (01 21) Mismatch between the Interface Processor ROS and Servo Processor ROS. (01 23) Failure to load Servo Microcode into RAM. (01 2C) SCSI Controller Chip internal parity error. (01 3F) SCSI Controller Chip detected an CRC error during read. (01 5B) Too many nested defects to perform nested reassign. (01 5C) Reassign could not find the target LBA. (01 5D) No Sector Found caused by hardware fault or software. (01 69) Servo/Fileside Interlock Head Compare Error. (01 7B) SCSI interrupt invalid. (Bus Free)</p>	Appendix C.4.2, "Unrecovered Drive Error"
4	44	00	<p>Internal Target Failure</p> <p>(01 80) SP interrupt on but SP Status Valid bit is off. (01 86) A servo command is already active. (01 87) Interface Processor detected Servo Sanity Error (01 8F) Buffer too small to do a requested function. (01 91) Interface Processor could not lock the SCSI Sequencer (01 A0) Error reading NOID Tables (01 C3) SP lost. (01 C6) Read Write Ready Dropped Error. (01 C8) Interrupt Occurred with no interrupt bits set. (01 CB) Motor Speed Error. (01 F2) Buffer Controller Chip Error - Invalid interrupt error. (01 F8) Buffer Controller Chip Error - Channel parity error on read. (01 FE) Buffer Controller Chip Error - Drive Pointer updated incorrectly (02 00) Disk Manager Error - Detected an ECC On The Fly overrun (02 01) Buffer Controller Chip Error - Pipeline already full (02 02) Buffer Controller Chip Error - FIFO overrun/underun</p>	Appendix C.4.2, "Unrecovered Drive Error"
4	44	00	<p>Internal Target Failure</p> <p>(02 03) Disk Manager Chip detected an CRC error during write. (02 05) Disk Manager Chip detected a sector pulse with current split loaded. (02 06) Disk Manager Chip detected a sector pulse in customer data during read or write. (02 07) Disk Manager Chip Detected an XOR error. (02 16) Servo ID overrun Error (02 18) Arm Electronics (AE) Idle Error (02 19) Interface Processor Ready Timeout Error (02 1E) Start Pipeline while in the Very Busy state. (02 1F) Parity Error detected in the Skip Sector FIFO. (02 34) NOID Sequence Error. (02 36) NOID Table Redundancy Error. (02 37) DAC Optimization Error. (02 40) Skip Sector FIFO overflow error. (02 41) Data Sector ID Counter error. (02 42) Skip Sector FIFO Load Check. (02 43) SID Counter error.</p>	Appendix C.4.2, "Unrecovered Drive Error"

4	44	00	<p>Internal Target Failure</p> <p>(02 44) Split Field Table CRC error. (02 45) PRSI Write error. (02 46) Bus Control logic timeout. (02 4A) Error detected loading the Split Field Table. (02 4B) IP Register write integrity error. (02 4C) SID Estimator Error. (02 55) Servo error - KF Adjustment too large. (02 57) Servo error - AE serial port read-back miscompare (02 58) Servo error - AE select error (wrong AE selected) (02 59) Servo error - Detected AE serial port sequence error (02 5A) Servo error - AE serial port timeout (stuck busy) (02 5B) Servo error - Head control port not taking effect (2ms timeout) (02 5C) Servo error - Spindle serial port timeout (stuck busy) (02 5D) Servo error - VCM not enabled in Voltage Mode (02 5F) Servo error - EEZ latch at speed. (02 68) Servo error - Target track out of range.</p>	Appendix C.4.2, "Unrecovered Drive Error"
4	44	00	<p>Internal Target Failure</p> <p>(02 69) Servo error - Command not accepted while actuator in retract. (02 6C) Servo error - Head number out of range. (02 6E) Servo error - Invalid command. (02 6F) Servo error - Estimator Position Out Of Range. (02 70) Servo error - Offset out of range. (02 71) Servo error - Unable to orient sector after breaking free of latch. (02 72) Servo error - Unable to break free of latch or no SID detected. (02 73) Servo error - Unable to break free of latch or no SID detected, while in internal recovery mod (02 74) Servo error - Unable to achieve track following after breaking free of latch. (02 75) Servo error - Not track following when recalibrate clean-up was entered. (02 76) Servo error - Seek timeout on clean-up seek to cyl 0. (02 77) Servo error - Unable to achieve track follow after breaking free of latch, while in servo internal recovery mode. (02 7E) Servo error - AGC not calibrated. (02 7F) Servo error - AGC limit exceeded. (02 80) Servo error - Command rejected, while motor not spinning.</p>	Appendix C.4.2, "Unrecovered Drive Error"
4	44	00	<p>Internal Target Failure</p> <p>(02 81) Servo error - Seek timeout during KT Calibration. (02 82) Servo error - Command not accepted, while Servo RAM not loaded. (02 83) Servo error - Seek timeout during AGC calibration. (02 84) Servo error - Command rejected - Invalid Recalibrate command qualifier. (02 85) Servo error - AGC saturated during calibration. (02 8C) Servo error - Loss of Sync, 6 of 8 missing SIDs (02 8F) Servo error - Real velocity exceeded - 2.75m/s four consecutive times. (8x xx) Invalid UEC - x xx is Invalid UEC.</p>	Appendix C.4.2, "Unrecovered Drive Error"
Sense Key = <i>Illegal Request</i>				
5	1A	00	<p>Parameter List Length Error</p> <p>(01 35) Command parameter list length error.</p>	Appendix C.4.8, "Interface Protocol"
5	20	00	<p>Invalid Command Operation Code</p> <p>(01 30) Invalid Operation code.</p>	Appendix C.4.8, "Interface Protocol"

5	21	00	Logical Block Address out of Range (01 31) Invalid LBA.	Appendix C.4.8, "Interface Protocol"
5	24	00	Invalid Field in CDB (01 32) CDB Invalid. (01 37) Data length error on Read Long or Write Long. (01 3A) Invalid Buffer ID in Write Buffer Command.	Appendix C.4.8, "Interface Protocol"
5	25	00	Logical Unit Not Supported (01 33) Invalid LUN.	Appendix C.4.8, "Interface Protocol"
5	26	00	Invalid Field in Parameter List (01 34) Command parameter data invalid. (01 36) Microcode and Load ID mismatch during Write Buffer Command. (01 38) Invalid Field in Parameter Data - See Field Pointer Value. (01 39) Invalid LBA in Reassign Command when Reassign degraded. (01 3B) Microcode and Servo Processor ROS mismatch during Write Buffer Command. (01 3C) Microcode and DE mismatch during Write Buffer Command.	Appendix C.4.8, "Interface Protocol"
5	55	00	System Buffer Full (02 08) Number of buffer segments too small to perform requested command.	Appendix C.4.2, "Unrecovered Drive Error"
Sense Key = Unit Attention				
6	28	00	Not Ready To Ready Transition, (Medium may have changed) (01 40) Unit Attention - Not Ready to Ready Transition (Format Completed)	Appendix C.4.10.1, "Not Ready to Ready Transition"
6	29	01	Unit Attention - POR Occurred (01 41) Unit Attention - POR Occurred.	None
6	29	02	Unit Attention - SCSI Bus Reset Occurred (01 41) Unit Attention - SCSI Bus Reset Occurred.	None
6	29	03	Unit Attention - Bus Device Reset Occurred (01 41) Unit Attention - Bus Device Reset Occurred.	None
6	29	04	Unit Attention - Self Initiated Reset Occurred (01 45) Unit Attention - Self Initiated Reset.	None
6	29	05	Transceiver Changed to SE (01 4A) Unit Attention - Transceiver Changed to SE.	None
6	29	06	Transceiver Changed to LVD (01 4B) Unit Attention - Transceiver Changed to LVD.	
6	2A	01	Mode Parameters Changed (01 42) Unit Attention - Mode Select Parameters have changed.	Appendix C.4.10.3, "Mode Parameters Changed"
6	2A	02	Log Parameters Changed (01 49) Unit Attention - Log Parameters Changed.	Appendix C.4.10.6, "Log Select Parameters Changed"
6	2F	00	Commands Cleared by Another Initiator (01 44) Unit Attention - Command cleared by another initiator.	Appendix C.4.10.5, "Commands"

				Cleared by Another Initiator"
6	3F	01	Microcode has been changed (01 43) Unit Attention - Write Buffer.	Appendix C.4.10.4 "Microcode has Changed"
6	3F	03	Inquiry Parameters Changed (01 46) Unit Attention - Inquiry Parameters have changed.	
Sense Key = Data Protect				
7	27	00	Write Protected (01 1C) Command not allowed while in Write Protect Mode.	1.6.2 "Page 0h-Vendor Unique Parameters"
Sense Key = Aborted Command				
B	1B	00	Synchronous Data Transfer Error (01 05) Synchronous transfer error - Extra pulses on synchronous transfer.	Appendix C.4.9, "Aborted Command"
B	25	00	Logical Unit Not Supported (01 71) Different LUN addressed (Identify message) from first selected. (Bus Free)	Appendix C.4.9, "Aborted Command"
B	43	00	Message Error (01 72) Inappropriate Message Reject message received. (Bus Free) (01 77) Attention dropped too late. (Bus Free) (01 78) Message parity error received when no message sent by Target. (Bus Free) (01 79) Unexpected or Invalid Message Error.	Appendix C.4.9, "Aborted Command"
B	44	00	Internal Target Failure (01 2D) Cannot resume the operation (Data transfer).	Appendix C.4.9, "Aborted Command"
B	45	00	Select or Reselect Failure (01 73) Reselection timeout. (Bus Free)	Appendix C.4.9, "Aborted Command"
B	47	00	SCSI Parity Error (01 61) Unrecovered SCSI parity error detected by Target during a command or data phase. (01 74) Unrecovered SCSI parity error detected by the Target during a MESSAGE OUT phase. (Bus Free) (01 7A) Unrecovered SCSI parity error detected by the Initiator (Message Parity Error Message). (Bus Free .)	Appendix C.4.9, "Aborted Command"
B	48	00	Initiator Detected Error Message Received (01 60) Initiator Detected Error for other than STATUS or linked COMMAND COMPLETE phase. (01 75) Initiator Detected Error message for STATUS or Linked COMMAND COMPLETE phase. (Bus Free).	Appendix C.4.9, "Aborted Command"
B	49	00	Invalid Message Error (01 76) Invalid message or attention dropped before all bytes of an extended message are transferred. (Bus Free).	Appendix C.4.9, "Aborted Command"
B	4E	00	Overlapped Commands Attempted (01 62) Invalid Initiator Connection.	Appendix C.4.9, "Aborted Command"

Sense Key = <i>Miscompare</i>				
E	1D	00	<i>Miscompare During Verify Operation</i> (01 92) Miscompare during byte by byte verify.	Appendix C.4.14, "Miscompare Recovery"

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.

Bytes 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 179. These fields identify the actual number of retries used in attempting to recover from the error condition.

Table 180. 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 181. These fields define an indication of progress in completing the *Format Unit* command.

Table 181. 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 **Not Ready** 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 182. These fields designate the bytes and bits in error in the *Mode Select* command parameter bytes.

Table 182. Field Pointer Bytes

Byte	BIT							
	7	6	5	4	3	2	1	0
15	SKSV = 1	C/D	RSVD = 0		BPV	Bit Pointer		
16	Field Pointer							
17								

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.

Byte 18 - 19 Reserved

Bytes 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

Bytes 24 - 29 Physical Error Record

ILI = 1 - This field contains zeros.

ILI = 0 - These bytes contain the physical location of the error as cylinder, head and sector. Bytes 24 and 25 are Cylinder high and Cylinder low respectively. Byte 26 is the head number. Bytes 28 and 29 are sector high and sector low respectively.

Byte 27 is retained for compatibility with previous members of the product family and will contain the sector number for sectors 0 through 254 and will be set to FFh for sector numbers greater than 254 or if the sector number is undetermined.

If the head is undetermined, its value is set to FFh. If the Cylinder value is undetermined,

bytes 24 and 25 are set to FFFFh. If the sector number is undetermined bytes 27, 28, and 29 are all set to 0FFh.

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 through 29 will all be set to FFh.

Bytes 30 - 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 and Data ECC 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 = Increase Bias Current in Read Head by 1 unit, -1 = Reduce Bias Current in Read Head by 1 unit.
- TO - Track Offset movement of the Read Head slightly off center. The percentage indicates the fraction of a track width. The sign indicates the direction of movement, with + being toward the inner diameter of the disk surface.
- ECC Burst - Amount of ECC correction applied.
- 9R - Software equivalent to 9 byte ECC correction.
- 12B - 12 Byte Correction. In this mode a 12 byte random error can be corrected.
- 24M - 24 Byte Mixed Mode. Multi-Byte ECC correction on adjacent and random error bytes where the Thermal Asperity has caused an error in less than 24 bytes. Additional random errors in the interleave can also be corrected. The number of random bytes that can be corrected depends on the number of error bytes in the burst caused by the Thermal Asperity.
- 20M - 20 Byte Mixed Mode. In this mode, a 16 byte error (or less) caused by a Thermal Asperity can be corrected. An additional 4 byte random error can also be corrected.
- 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.
- HSC - Head State Change. A short write is done in the noncustomer data area to change the residual head state characteristics.
- VCO Calibrate - Calibrates the Voltage Controlled Oscillator in the Channel Hardware.
- A/D Calibrate - Calibrates the Analog to Digital Convertor in the Channel Hardware.
- GEM - Generalized Error Measurement.
- Read Gate Adjust - Used to attempt to recover from a Thermal Asperity in the Data Sync Field. Read Gate activation can be adjusted early or late by the amount specified.
- SAF - Signal Adaptive Filter. This filter is located in the arm electronics and is used to reduce noise on the read signal.
- VGA - Variable Gain Amplifier. This filter is located in the Channel Hardware and is used to reduce noise on the read signal.
- VM - Viterbi Margin. A parameter that the Channel Hardware measures to determine how closely the measured signal compares to the desired signal.
- UHR - Unstable Head Recovery. Adjust read bias to increase signal magnitude.
- BTA - Burnish TA. Attempt to reduce the effects of TA.
- LBTA - Long Burnish TA. More extensive variation of BTA.

B.1.1 Read Command Data Recovery Procedure Steps

Table 183. Read Recovery Steps

DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
0	Initial Read	0	9R	
1	Reread	0	9R	
2	+9% TO	0	9R	Trunc & Fake Bit Sync On. SAF on with shorter recovery.
3	-9% TO	0	9R	
4	Reread	0	9R	
5	Reread	0	9R	
6	HSC/Rd	0	9R	Count Data Sync Errors. If Channel Circuitry detects a TA on any of these steps, immediately jump to the TA Branch of the Read Recovery Table (Step 7).
7	+6% TO	0	9R	
8	-6% TO	0	9R	
9	+9% TO	0	9R	
10	-9% TO	0	9R	
11	+12% TO	0	9R	
12	-12% TO	0	9R	
13	+15% TO	0	9R	
14	-15% TO	0	9R	
15	+18% TO	0	9R	
16	-18% TO	0	9R	
17	Reread	+1	9R	Cond. UHR.
18	Reread	0	9R	Adjust VGA input impedance to 113 Ohms.
19	+1 EQU	0	9R	Adjust Read Gate 2 Bytes Late.
20	-1 EQU	0	9R	Adjust Read Gate 2 Bytes Early.
21	Reread	-2	9R	Adjust VGA input impedance to 113 Ohms.
22	Reread	0	9R	Adjust BTA/VGA input impedance to 113 Ohms. Adjust Read Gate 4 Bytes Late.
23	BTA/Reread	0	9R	Adjust Read Gate 4 Bytes Early. Adjust BTA/VGA input impedance to 113 Ohms.
24	+9% TO	0	9R	Adjust BTA/VGA input impedance to 113 Ohms.
25	-9% TO	0	12B	Adjust BTA/VGA input impedance to 113 Ohms.
26	+21% TO	0	12B	
27	-21% TO	0	12B	

28	Reread	0	12B	Adjust Read Gate 6 Bytes Late. Adjust BTA/VGA input impedance to 113 Ohms.
29	Reread	0	12B	Adjust Read Gate 8 Bytes Late. Adjust BTA/VGA input impedance to 113 Ohms.
30	Reread	0	12B	Adjust Read Gate 10 Bytes Late. Adjust BTA/VGA input impedance to 113 Ohms.
31	Reread	-4	12B	
32	Reread	0	12B	Perform Channel A/D Calibrate.
33	+21% TO	0	12B	
34	-21% TO	0	12B	
35	Reread	0	12B	Perform conditional VCO calibrate. If able to read test sector, then skip VCO.
36	Reread	0	24M	
37	Reread	0	24M	Disable Channel's gain and timing adjustments during bytes containing VM errors.
38	Reread	0	24M	Disable Channel's gain and timing adjustments 4 bytes sooner than bytes containing VM errors.
39	+21% TO	0	24M	
40	-21% TO	0	24M	
41	Reread	-2	24M	Adjust VGA input impedance to 113 Ohms.
42	Reread	0	24M	Adjust VGA input impedance to 113 Ohms. Adjust Read Gate 6 Bytes Late.
43	Reread	0	24M	Adjust VGA input impedance to 113 Ohms. Adjust Read Gate 8 Bytes Late.
44	Reread	-4	24M	
45	Reread	0	20M	
46	+18% TO	0	20M	
47	-18% TO	0	20M	
48	LBTA/Reread	0	20M	
49	UHR/Reread	0	20M	

Table 184. TA Branch of Read Recovery

DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
7	Reread	0	12B	
8	+6% TO	0	12B	
9	-6% TO	0	12B	
10	+9% TO	0	12B	
11	-9% TO	0	12B	
12	+12% TO	0	12B	
13	-12% TO	0	12B	
14	+15% TO	0	12B	
15	-15% TO	0	12B	
16	+18% TO	0	12B	
17	-18% TO	0	12B	
18	BTA/Reread	-2	12B	Adjust BTA/VGA input impedance to 113 Ohms.
19	BTA/Reread	0	12B	Adjust BTA/VGA input impedance to 113 Ohms. Adjust Read Gate 2 Bytes Late.
20	BTA/Reread	0	12B	Adjust VGA input impedance to 113 Ohms. Adjust Read Gate 2 Bytes Early.
21	BTA/-1 EQU	0	12B	
22	Reread	0	12B	Adjust VGA input impedance to 113 Ohms. Adjust Read Gate 4 Bytes Late.
23	Reread	0	12B	Adjust VGA input impedance to 113 Ohms. Adjust Read Gate 4 Bytes Early.
24	+18% TO	0	12B	Adjust VGA input impedance to 113 Ohms.
25	-18% TO	0	12B	Adjust VGA input impedance to 113 Ohms.
26	Reread	0	12B	Perform A/D calibrate.
27	Reread	0	12B	Perform Conditional VCO Calibrate. (If able to read test sector, then skip VCO Calibrate.)
28	+21% TO	0	12B	Adjust VGA input impedance to 113 Ohms.
29	-21% TO	0	12B	Adjust VGA input impedance to 113 Ohms.
30	Reread	0	24M	Adjust VGA input impedance to 113 Ohms.
31	Reread	0	24M	Disable Channel's Gain and Timing adjustments during bytes containing VM errors.
32	Reread	0	24M	Disable Channel's Gain and Timing adjustments 4 bytes sooner than bytes containing VM errors.
33	+18% TO	0	24M	
34	-18% TO	0	24M	

35	HSC/Reread	0	24M	
36	Reread	0	24M	
37	+18% TO	0	24M	Adjust VGA input impedance to 113 Ohms.
38	-18% TO	0	24M	Adjust VGA input impedance to 113 Ohms.
39	Reread	-2	24M	Adjust VGA input impedance to 113 Ohms.
40	-1 EQU	0	24M	
41	Reread	-4	24M	Adjust VGA input impedance to 113 Ohms.
42	Reread	0	20M	
43	+15% TO	0	20M	Adjust VGA input impedance to 113 Ohms.
44	-15% TO	0	20M	Adjust VGA input impedance to 113 Ohms.
45	Reread	-2	20M	Adjust VGA input impedance to 113 Ohms.
46	Reread	-4	20M	Adjust VGA input impedance to 113 Ohms.
47	-1 EQU	0	20M	
48	+1 EQU	0	20M	
49	Reread	0	20M	

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 verify operation that fails is considered a hard error if the byte by byte option is in effect and a miscompare error, confirmed to be non-ECC is detected. The Sense Key reported for this error is 0Eh (Miscompare Error during Verify byte-by-byte Operation).
2. In all other cases, a verify operation that fails 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.

B.1.2.1 Write and Verify

Table 185. Write and Verify Data Recovery Procedure Steps

DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
0	Initial Read	0	9R	
1	Reread	0	9R	
2	+9% TO	0	9R	Trunc & Fake Bit Sync On. SAF on with shorter recovery.
3	-9% TO	0	9R	
4	Reread	0	9R	
5	Reread	0	9R	
6	HSC/Rd	0	9R	Count Data Sync Errors. If Channel Circuitry detects a TA on any of these steps, immediately jump to the TA Branch of the Read Recovery Table (Step 7).
7	+6% TO	0	9R	
8	-6% TO	0	9R	
9	+9% TO	0	9R	
10	-9% TO	0	9R	
11	+12% TO	0	9R	
12	-12% TO	0	9R	
13	+15 TO	0	9R	
14	-15 TO	0	9R	
15	+18% TO	0	9R	
16	-18% TO	0	9R	
17	Reread	+1	9R	Cond. UHR.
18	Reread	0	9R	Adjust VGA input impedance to 113 Ohms.
19	+1 EQU	0	9R	Adjust Read Gate 2 Bytes Late.
20	-1 EQU	0	9R	Adjust Read Gate 2 Bytes Early.
21	Reread	-2	9R	Adjust VGA input impedance to 113 Ohms.
22	Reread	0	9R	Adjust BTA/VGA input impedance to 113 Ohms. Adjust Read Gate 4 Bytes Late.
23	BTA/Reread	0	9R	Adjust Read Gate 4 Bytes Early. Adjust BTA/VGA input impedance to 113 Ohms.
24	+9% TO	0	9R	Adjust BTA/VGA input impedance to 113 Ohms.
25	-9% TO	0	12B	Adjust BTA/VGA input impedance to 113 Ohms.
26	+21% TO	0	12B	
27	-21% TO	0	12B	

28	Reread	0	12B	Adjust Read Gate 6 Bytes Late. Adjust BTA/VGA input impedance to 113 Ohms.
29	Reread	0	12B	Adjust Read Gate 8 Bytes Late. Adjust BTA/VGA input impedance to 113 Ohms.
30	Reread	0	12B	Adjust Read Gate 10 Bytes Late. Adjust BTA/VGA input impedance to 113 Ohms.
31	Reread	-4	12B	
32	Reread	0	12B	Perform Channel A/D Calibrate.
33	+21% TO	0	12B	
34	-21% TO	0	12B	
35	Reread	0	12B	Perform conditional VCO calibrate. If able to read test sector, then skip VCO.

B.1.2.2 Verify Recovery

Table 186. Verify Recovery Steps

DRP Steps	Action	Read Bias	ECC Burst	Miscellaneous
0	Initial Read	0	9R	
1	Reread	0	9R	
2	+9% TO	0	9R	Trunc & Fake Bit Sync On. SAF on with shorter recovery.
3	-9% TO	0	9R	
4	Reread	0	9R	
5	Reread	0	9R	
6	HSC/Rd	0	9R	Count Data Sync Errors. If Channel Circuitry detects a TA on any of these steps, immediately jump to the TA Branch of the Read Recovery Table (Step 7).
7	+6% TO	0	9R	
8	-6% TO	0	9R	
9	+9% TO	0	9R	
10	-9% TO	0	9R	
11	+12% TO	0	9R	
12	-12% TO	0	9R	
13	+15 TO	0	9R	
14	-15 TO	0	9R	
15	+18% TO	0	9R	
16	-18% TO	0	9R	
17	Reread	+1	9R	Cond. UHR.
18	Reread	0	9R	Adjust VGA input impedance to 113 Ohms.
19	+1 EQU	0	9R	Adjust Read Gate 2 Bytes Late.
20	-1 EQU	0	9R	Adjust Read Gate 2 Bytes Early.
21	Reread	-2	9R	Adjust VGA input impedance to 113 Ohms.
22	Reread	0	9R	Adjust BTA/VGA input impedance to 113 Ohms. Adjust Read Gate 4 Bytes Late.
23	BTA/Reread	0	9R	Adjust Read Gate 4 Bytes Early. Adjust BTA/VGA input impedance to 113 Ohms.
24	+9% TO	0	9R	Adjust BTA/VGA input impedance to 113 Ohms.
25	-9% TO	0	12B	Adjust BTA/VGA input impedance to 113 Ohms.
26	+21% TO	0	12B	
27	-21% TO	0	12B	

28	Reread	0	12B	Adjust Read Gate 6 Bytes Late. Adjust BTA/VGA input impedance to 113 Ohms.
29	Reread	0	12B	Adjust Read Gate 8 Bytes Late. Adjust BTA/VGA input impedance to 113 Ohms.
30	Reread	0	12B	Adjust Read Gate 10 Bytes Late. Adjust BTA/VGA input impedance to 113 Ohms.
31	Reread	-4	12B	
32	Reread	0	12B	Perform Channel A/D Calibrate.
33	+21% TO	0	12B	
34	-21% TO	0	12B	
35	Reread	0	12B	Perform conditional VCO calibrate. If able to read test sector, then skip VCO.

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 3 classes:

- Drive Fault
- Position (Servo)
- Spindle Motor

B.2.1 Position Error (Servo Error Recovery)

Table 187. Servo Error Recovery Steps

Step	Action	Read Bias	Miscellaneous
0	Initial Servo Operation	0	
1-3	Retry Servo Operation	0	Recalibrate, Seek to last good location prior to each step.

B.2.2 Drive Fault

Table 188. Drive Fault Recovery Steps

Step	Action	Read Bias	Miscellaneous
0	Initial Read/Write	0	
1-9	Reread/Rewrite	0	Recalibrate, Seek to last good location prior to each step

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.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 ERP

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.

Nondata errors are those errors that do not have a direct relationship with transferring data to and from the media. Nondata 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

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 through 6)
- Command specific information (Sense bytes 8 through 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 through 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 bit ARRE (See 1.6, "Mode Select") set active allows the Drive to rewrite or relocate recovered read data errors. Nonrecovered data errors or the ARRE bit being 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)*. 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 189, the Key, Code, and Qualifier fields are all hex values (i.e., Sense Key 1 is 1h, Sense Code 17 is 17h, etc).

Table 189. 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 190, the Key, Code, and Qualifier fields are all hex values (i.e., Sense Key 1 is 1h, Sense Code 17 is 17h, etc).

Table 190. Recommend Reassign Errors

Sense data combinations for recommend reassignment.			
Key	Code	Qual	Description
1	16	04	Sync Byte Error - Recommend Reassignment.
1	17	07	Recovered data without ECC - Recommend Reassignment.
1	18	05	Recovered data with ECC - Recommend Reassignment.
3	11	0B	Unrecovered read error - 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 LBA's from the LBA being reassigned to the location of the closest spare in either direction changes. All candidates for reassignment are listed in Page 33 of Log Sense Data. Refer to 1.5, "Log Sense."

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 191, the Key, Code, and Qualifier fields are all hex values (i.e., Sense Key 1 is 1h, Sense Code 17 is 17h, etc).

Table 191. Log Only Errors

Sense data combinations for automatic rewrite or reallocate or that requires no system action except log.			
Key	Code	Qual.	Description
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	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 bit ARRE is active, the Drive will automatically reallocate LBA's determined to be defective. For those LBA's where the error is unrecoverable or the initiator elects to not have the Drive automatically reallocate LBA's, the Drive will recommend reassignment of the LBA.

Recovery from a failed reassignment is to:

- Update the defect descriptor to remove the LBA's that have been successfully reassigned and then retry the *Reassign Blocks* command. The LBA contained in the Command Specific Information field of the Sense Data is the LBA in the first defect descriptor that was not reassigned because of the failure. If the command failed because of an unrecoverable read error other than those specified in the defect descriptor, add this LBA to the defect descriptor and retry the command. Refer to 1.18, "Reassign Blocks" for additional information.
- 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."

C.4 Nondata Error Recovery Procedure

The Drive will follow a logical recovery procedure for nondata 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 Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec for time-out limits. **This is not an error condition.** The initiator can test for completion of the operation by issuing *Test Unit Ready (00)* (or media access) commands.

- If the *Test Unit Ready (00)* (or media access) command completes with *Check Condition Status* then issue a *Request Sense (03)*
 - 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, then continue re-issuing the *Test Unit Ready (00)* and *Request Sense* commands for the duration of a media access timeout or until the drive returns *Good Status*.
If the drive has been busy for longer than the limit specified in the Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec, then service the drive using the service guidelines recommended in "Drive Service Strategy," otherwise return to normal processing.
- 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:
 - i. *Good Status* or recovered Sense Key, follow the recovery procedure in C.4.3, "Recovered Drive Error."
 - ii. 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:
 - i) *Good Status*, follow the recovery procedure in C.4.3, "Recovered Drive Error."
 - ii) Recovered sense or Hardware error sense, then service the drive using the service guideline recommended in C.1, "Drive Service Strategy."

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 Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec.
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."
3. Retry the failing command.
4. If the failing command completes with:
 1. *Good Status*, log the error as recovered.
 2. 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."

C.4.5 No Defect Spare

Three conditions can exist that will cause this error:

1. When the *Reassign Blocks (07)* command is issued and there are no spares available for the Drive to use for the relocation requested.
2. When the Glist is full and the sector to be reassigned cannot be added.
3. During a format operation, there was not enough space available to fulfill the spare requirement (Dlist is too large).

Service the Drive following the C.1, "Drive Service Strategy."

C.4.6 Degraded Mode

Refer to 4.1.9, "Reassign Blocks" for the definition of this state. There are four (4) 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 Ultrastar 18ZX/9LZX Hardware Functional or Ultrastar 36XP/18LP Hardware Functional Spec), service the drive using the service guideline recommended in C.1, "Drive Service Strategy."
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)*.

- 3 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)* is either:
 - 1) Execute the *Send Diagnostic (1D)* command, or
 - 2) Power cycle the drive.
- If the failure repeats, service the drive using the service guideline recommended in "Drive Service Strategy."

Recovery for a failed power up sequence is either:

- 1) Issue a *Start Unit (1B)* command, or
- 2) Power cycle the drive.

If the failure repeats, service the drive using the service guideline recommended in "Drive Service Strategy."

- 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)*. Service the drive using the recommended service guidelines specified in C.1, "Drive Service Strategy" if the power cycle or the *Send Diagnostic (1D)* command fail to complete successfully.

- 4 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"). 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)* 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."

If the above defined recovery procedures fail to clear the degraded mode condition, the Drive should be replaced. Follow the procedure C.1, "Drive Service Strategy" 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" 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
 - i. *Good Status*, log the error as recovered
 - ii. *Check Condition Status* with sense data for an Illegal Request, verify there is no outside cause (e.g., power supply) for the failure
 - iii. Other, follow the recommendations for the error condition reported.
 - 1) Retry the failing command.
 - 2) If this retry of the failing command completes with:
 - i) *Good Status*, log the error as recovered
 - ii) *Check Condition Status* with sense data for an Illegal Request, service the drive using the service guideline recommended in C.1, "Drive Service Strategy."
 - iii) Other, follow the recommendations for the error condition reported.

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 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:
 - i. *Good Status*, log the error as recovered.
 - ii. 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:
 - i. *Good Status*, log the error as recovered.
 - ii. Abort command sense, then service the drive using the service guideline recommended in C.1, "Drive Service Strategy."
- 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)*.

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)*.

C.4.10.4 Microcode Has Changed

Write Buffer (3B) 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 currently being executed. 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 Log Select Parameters Changed

A Log Select (4C) command successfully completed. This means that the Log Select command cleared statistical information successfully (See Log Select command). *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."
2. Load the proper level of microcode using the *Write Buffer (3B)* with the download and save option.

To load new microcode, the initiator should:

- i. Issue an *Inquiry (12)* for EVPD page 3. EVPD page 3 of the inquiry data contains the valid LID.

- ii. Issue a *Write Buffer (3B)* 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.
 - iii. 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."
3. Execute the *Send Diagnostic (1D)* with the self test bit active.
 4. Check the send diagnostic end status. If the status is
 - GOOD, Return to normal processing.
 - *Check Condition Status*, issue a *Request Sense (03)* 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."

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."
2. Replace the electronics with the correct part number electronics.
3. If bring-up complete status is:
 - GOOD, Return to normal processing.
 - *Check Condition Status*, issue a *Request Sense (03)* 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."

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)*.
 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."
 - 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 "Drive Service Strategy."
- 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 following the service guidelines in C.1, "Drive Service Strategy."

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)* 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."
 - 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 reliably write or read the media. Service the drive using the procedures specified in C.1, "Drive Service Strategy."

C.4.15 Microcode Error

The microcode 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)* retry the *Write Buffer (3B)*. 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."

If the check sum error occurred during normal processing, the initiator may attempt to load microcode before deciding to service the drive using the service guidelines recommended in C.1, "Drive Service Strategy."

To load new microcode, the initiator should:

1. Issue an *Inquiry (12)* 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."
- *Check Condition Status* 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."
- *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."

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." These codes are used internally by IBM and may change without notice.

Table 192. 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 Command active.
01 03	Unavailable while Bring-up active.
01 04	Unavailable while Format active.
01 05	Synchronous transfer error - Extra pulses on synchronous transfer.
01 06	Requested P List does not match returned list format (READ DEFECT DATA only).
01 07	Requested G List does not match returned list format. (READ DEFECT DATA only).
01 08	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.
01 09	LBA to PBA Conversion Timeout Error.
01 0A	Defect list longer than 64k, 64k of data returned. (READ DEFECT DATA only).
01 0C	NOID Table integrity error detected.
01 0F	Error detected while reading the NOID Tables from disk.
01 10	Too few valid GEM measurements available to perform a GEM Predictive Failure Analysis.
01 11	Degraded Mode - Reassign Blocks unsuccessful after pushdown started.
01 12	Degraded Mode - Format unsuccessful.
01 13	Degraded Mode - Configuration not loaded.
01 16	ROS Microcode Download failed.
01 17	NOID Table build failed during a Format Unit command.
01 1B	Motor Start Failed due to Timer 1 being disabled.
01 1C	Command not allowed while in Write Protect Mode.
01 20	Microcode Check Sum error detected during download of Microcode.
01 21	Mismatch between the Interface Processor ROS and Servo Processor ROS.
01 22	Degraded Mode - Bringup not successful.
01 23	Failure to load Servo Microcode into RAM.

01 24	Mismatch between the Servo Processor ROS and DE.
01 2C	SCSI Controller Chip internal parity error.
01 2D	Cannot resume the operation (Data transfer).
01 2F	Mismatch between the Interface Processor ROS and the DE.
01 30	Invalid Operation 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	Microcode and Load ID 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	Microcode and Servo Processor ROS mismatch during Write Buffer Command.
01 3C	Microcode and DE mismatch during Write Buffer Command.
01 3F	SCSI Controller Chip detected a CRC error during read.
01 40	Unit Attention - Not Ready to Ready Transition (Format Completed).
01 41	Unit Attention - POR Occurred.
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 46	Unit Attention - Inquiry Parameters have changed.
01 49	Unit Attention - Log Parameters Changed.
01 4A	Unit Attention - Transceiver Changed to SE.
01 4B	Unit Attention - Transceiver Changed to LVD.
01 50	Microcode Check Sum error detected during ROS Test.
01 56	GLIST full - cannot add more entries.
01 57	The defects per track limit was exceeded during a Format or Reassign operation.
01 5A	Motor is Stuck, Cannot be started.
01 5B	Too many nested defects to perform nested reassign.
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 (READ DEFECT DATA only).
01 66	Error in Grown Defect list (READ DEFECT DATA only).
01 69	Servo/Fileside Interlock Head Compare Error.
01 6B	Arm Electronics Not Ready.
01 70	Abort Message Received (Bus Free).
01 71	Different LUN addressed (Identify message) from first selected. (Bus Free).
01 72	Inappropriate Message Reject message received. (Bus Free).
01 73	Reselection timeout. (Bus Free).
01 74	Unrecovered SCSI parity error detected by the Target during a MESSAGE OUT phase (Bus Free).
01 75	Initiator Detected Error message for STATUS or Linked COMMAND COMPLETE phase (Bus Free).
01 76	Invalid message or attention dropped before all bytes of an extended message are transferred (Bus Free).
01 77	Attention dropped too late (Bus Free).
01 78	Message parity error received when no message sent by Target (Bus Free).
01 79	Unexpected or Invalid Message Error.
01 7A	Unrecovered SCSI parity error detected by the Initiator (Message Parity Error Message) (Bus Free).
01 7B	SCSI interrupt invalid (Bus Free).
01 7C	Abort Tag Message Received (Bus Free).
01 7D	Clear Queue Message Received (Bus Free).
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 84	Motor timeout 3 tries limit.
01 85	Motor gross timeout during presettle.
01 86	A servo command is already active.
01 87	Interface Processor 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 gross timeout during postsettle.
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 Initiated Reset - Invalid Input.
01 91	Interface Processor could not lock the SCSI Sequencer.
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.
01 97	BATS#2 Error - Hardware XOR Test Failure.
01 98	Self Initiated Reset - Cause Unknown.
01 99	Self Initiated Reset - SCSI Controller Chip Reset unsuccessful.
01 9A	Self Initiated Reset - Buffer Controller Chip Reset unsuccessful.
01 9B	Self Initiated Reset - Zero Divide Error.
01 9C	Self Initiated Reset - Control Store Address Fault.
01 9D	Self Initiated Reset - Unused Op Code.
01 9F	Self Initiated Reset - Invalid Queue Operation.
01 A0	Error reading NOID Tables.
01 A1	Controller "Ready" Timeout Error.
01 C0	Too many missing Servo IDs detected by the Controller or 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 C6	Read Write Ready Dropped Error.
01 C7	Microjog Write Inhibit.
01 C8	Interrupt Occurred with no interrupt bits set.
01 C9	Write with No Sector Pulses.
01 CB	Motor Speed Error.
01 CD	IP Retract Error.

01 D1	No Data Field sync byte found.
01 D2	Data ECC Check.
01 D3	Data correction applied to Drive data for a Data ECC check..
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	Erroneous Sync Byte Found.
01 DC	Recovered data without ECC, Auto Reallocated.
01 DD	Site awaiting data restoration. This site was auto reallocated because of a Hard Error.
01 F2	Buffer Controller Chip Error - Invalid interrupt error.
01 F8	Buffer Controller Chip Error - Channel parity error on read.
01 FE	Buffer Controller Chip Error - Drive Pointer updated incorrectly.
02 00	Disk Manager Error - Detected an ECC On The Fly overrun.
02 01	Buffer Controller Chip Error - Pipeline already full.
02 02	Buffer Controller Chip Error - FIFO overrun/underun.
02 03	Disk Manager Chip detected a CRC error during write.
02 05	Disk Manager Chip detected a sector pulse with current split loaded.
02 06	Disk Manager Chip detected a sector pulse in customer data during read or write.
02 07	Disk Manager Chip Detected an XOR error.
02 08	Number of buffer segments too small to perform requested command.
02 0F	Servo Diagnostic Transfer Timeout
02 16	Servo ID overrun Error.
02 18	Arm Electronics (AE) Idle Error.
02 19	Interface Processor Ready Timeout Error.
02 1B	External Write Inhibit.
02 1C	Error in Target Sector Generation logic.
02 1E	Start Pipeline while in the Very Busy state.
02 1F	Parity Error detected in the Skip Sector FIFO.
02 20	Channel Noise Problem, Recommend Device Replacement.
02 21	Channel Asymmetry Problem, Recommend Device Replacement.
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 2A	Test PFA Error. Generated when Test Bit in Mode Page 0x1C is set.
02 2F	Specified File Temperature Exceeded.
02 34	NOID Sequence Error.
02 36	NOID Table Redundancy Error.
02 37	DAC Optimization Error.
02 40	Skip Sector FIFO overflow error.
02 41	Data Sector ID Counter error.
02 42	Skip Sector FIFO Load Check.
02 43	SID Counter error.
02 44	Split Field Table CRC error.
02 45	PRSI Write error.
02 46	Bus Control logic timeout.
02 4A	Error detected loading the Split Field Table.
02 4B	IP Register write integrity error.
02 4C	SID Estimator Error.
02 52	Servo error - Power dissipation too high.
02 53	Servo error - Gray Code window exceed 9 tracks (Position).
02 54	Servo error - Gray Code window exceed 3 tracks (Track Follow).
02 55	Servo error - KF Adjustment too large.
02 57	Servo error - AE serial port read-back miscompare.
02 58	Servo error - AE select error (wrong AE selected).
02 59	Servo error - Detected AE serial port sequence error.
02 5A	Servo error - AE serial port timeout (stuck busy).
02 5B	Servo error - Head control port not taking effect (2ms timeout).
02 5C	Servo error - Spindle serial port timeout (stuck busy).
02 5D	Servo error - VCM not enabled in Voltage Mode.
02 5F	Servo error - EEZ Latch at speed.
02 60	Servo error - Excessive Runout - Recalibrate requested from Servo.
02 61	Servo error - Timed out waiting for PES interrupt.
02 62	Servo error - Settle timeout.
02 63	Servo error - Three consecutive bad SIDs (track follow).

02 64	Servo error - Three consecutive bad SIDs (settle).
02 65	Servo error - Three consecutive bad SIDs (seeking).
02 66	Servo error - Three consecutive bad SIDs (half track).
02 67	Servo error - Seek timeout.
02 68	Servo error - Target track out of range.
02 69	Servo error - Command not accepted while actuator in retract.
02 6A	Servo error - Three consecutive invalid Gray codes.
02 6B	Servo error - Estimator error saturated.
02 6C	Servo error - Head number out of range.
02 6D	Servo error - Gray code cylinder mismatch.
02 6E	Servo error - Invalid command.
02 6F	Servo error - Estimator Position Out Of Range.
02 70	Servo error - Offset out of range.
02 71	Servo error - Unable to orient sector after breaking free of latch.
02 72	Servo error - Unable to break free of latch or no SID detected.
02 73	Servo error - Unable to break free of latch or no SID detected, while in internal recovery mode.
02 74	Servo error - Unable to achieve track following after breaking free of latch.
02 75	Servo error - Not track following when recalibrate clean-up was entered.
02 76	Servo error - Seek timeout on clean-up seek to cyl 0.
02 77	Servo error - Unable to achieve track follow after breaking free of latch, while in servo internal recovery mode.
02 7E	Servo error - AGC not calibrated.
02 7F	Servo error - AGC limit exceeded.
02 80	Servo error - Command rejected because the motor is not spinning or motor is not in speed tolerance.
02 81	Servo error - Seek timeout during KT Calibration.
02 82	Servo error - Command not accepted, while Servo RAM not loaded.
02 83	Servo error - Seek timeout during AGC calibration.
02 84	Servo error - Command rejected - Invalid Recalibrate command qualifier.
02 85	Servo error - AGC saturated during calibration.
02 8C	Servo error - Loss of Sync, 6 of 8 missing SIDs.
02 8D	Servo error - ASIC/SP Sector Count mismatch.
02 8F	Servo error - Real velocity exceeded - 2.75m/s four consecutive times.
02 90	BATS#2 Error (Split Field Table checksum error)
02 91	BATS#2 Error (Split Field Table address error).

02 92	BATS#2 Error (Split Field Table stuck bit error).
02 93	BATS#2 Error (CRC test failure, mode 1).
02 94	BATS#2 Error (CRC test failure, mode 2).
02 95	BATS#2 Error (CRC test failure, mode 3).
0x xx	Unrecovered error during Reassign after pushdown has started. This error can be associated with a number of UEC's
4x xx	Thermal Asperity Detected during error.
8x xx	Invalid UEC - x xx is Invalid UEC.

Index

A

abort, 170
abort pre-fetch (ABPF), 87
aborted command, 278
active notch, 94
adaptive caching (AC), 68
additional save data pointer enable (ASDPE), 65
allocation length field, 45
alternate sectors per zone, 80
alternate tracks per logical unit, 81
alternate tracks per zone, 81
ASCII, 5
attention condition, 183
automatic read reallocation enable (ARRE), 71
automatic rewrite/reallocate - recommend
rewrite/reassign (ARRR), 224
automatic write reallocation enable (AWRE), 71
auto-reallocation of a hard error site (ARHES),
65

B

back-to-back writes, 222, 223
basic assurance tests (BATS), 240
block descriptor, 62
bring-up sequence, 238
buffer overrun, 54
buffer underrun, 54
busy timeout period, 93

C

cache segment size, 89
caching analysis permitted (CAP), 87
caching parameters, 86
choosing buffer ratios, 78
CMDAC bit, 65
command aging enable (CAEN), 68
command phase retry, 196
command processing during execution of active
I/O process, 201
Command Processing During Start-up and
Format Operations, 205
command processing while reserved, 217
command queuing, 219
commands, 17
 format unit, 19
 inquiry, 25

HMSCSO16.LWP

log select, 38
log sense, 45
mode select, 60
mode sense, 100
persistent reserve in, 112
persistent reserve out, 117
pre-fetch, 120
read, 122, 123
read buffer, 124
read capacity, 126
read defect data, 127, 130
reassign blocks, 134
receive diagnostic results, 136
release (10), 140
release (6), 139
report LUNs, 141
request sense, 143
reserve (10), 145
reserve (6), 144
rezero unit, 147
seek (10), 148
seek (6), 148
send diagnostic, 149
start/stop, 153
synchronize cache, 154
test unit ready, 155
verify, 156
write (10), 158
write (6), 157
write buffer, 161
write long, 164
write same, 165
commands cleared by another initiator, 280
components mismatch, 280
concurrent command processing, 222
concurrent processing enable (CPE), 67
connect time limit, 78
control mode parameters, 91
correction span, 76
current temperature measurement, 59
cylinder skew factor, 81

D

data bytes per physical sector, 81
data discard, 233
data error logging, 274
data in phase retry, 197
data out phase retry, 197

data phase save data pointer (DPSDP), 67
data recovery procedure, 271
data recovery procedure (DRP), 76
data recovery procedure for data field errors, 259
data strobe offset count, 76
data transfer disconnect control (DTDC), 79
defect list recovery, 282
deferred error condition, 207
definitions for tables, 259
degraded mode, 276
demand read retention priority, 88
device cache fast writes, 55
device cache partial read hits, 55
device cache write hits, 55
disable correction (DCR), 73
disable ECC on the fly (DOTF), 66
disable latency reduction optimization (DLRO), 67
disable nested reassign (DNR), 69
disable pre-fetch transfer length, 88
disable queuing (DQue), 92
disable read-ahead (DRA), 89
disable reassign blocks post move verify (DRPDV), 69
disable restore reassign target (DRRT), 69
disable transfer on error (DTE), 73
disable write latency reduction optimization (DWLRO), 67
discard surface analysis results after format (DSF), 70
disconnect, 169
disconnect immediate (DIMM), 79
disconnect time limit, 78
disconnect/reconnect parameters, 77
discontinuity (DISC) bit, 87
DList, 19
drive busy, 275
drive fault, 269
drive fault recovery steps, 269
drive not ready, 276
drive service strategy, 270
drive step rate, 83
DRP table, 261

E

ECC on-the-fly hardware, 51
effects of errors on command queuing, 220
enable early recovery (EER), 72

enable modify data pointers (EMDP), 79
enable vital product data (EVPD), 25
ending boundary, 95
error logs, 224
error recovery for nondata errors, 269
error recovery parameters, 71
error recovery parameters for verify, 84
error recovery procedures (ERP), 269, 270
extended messages, 174

F

fast format enable (FFMT), 69
fast10, 90
fast20, 90
field replaceable unit code, 256
force sequential write (FSW), 88
force8, 90
format and reassign degraded disable (FRDD), 67
format certification (FCERT), 69
format device parameters, 80
format unit command, 19

G

GList, 19
global logging target save disable (GLTSD), 91

H

head offset count, 76
hex values, 5

I

identify message in retry, 197
immed bit, 22
incorrect initiator connection, 200
informational exceptions control, 97
initiator control (IC), 86
initiator data transfer mode parameters, 236
initiator detected error, 170
initiator error recovery procedures, 270
initiator mode select/mode sense parameters, 236
initiator sense data, 236
inquiry command, 25
integration reassign threshold (IRT), 70
integration verify recovery (IVR), 70
interface protocol, 278
interleave factor, 19
internal error condition, 205

invalid LUN processing, 199

L

landing zone cylinder, 83

LED pin, 65, 237

log only errors, 274

log page parameters, 46

log parameter format, 56, 57

log parameters, 46

log select command, 38

log select parameters changed, 280

log sense command, 45

logical block cache segment size (LBCSS), 89

logical unit number (LUN), 60

LUN, 60

M

maximum burst size, 78

maximum number of notches, 94

maximum pre-fetch, 88

maximum pre-fetch ceiling, 88

medium rotation rate, 83

message action legend, 187

message in phase retry, 195

message out phase retry, 195

message reject, 170

message state/action table, 186, 187, 188, 189, 193

messages

 abort tag, 171

 bus device reset, 171

 clear queue, 172

 command complete, 169

 extended, 174

 head of queue tag, 173

 identify, 173

 ignore wide residue, 173

 linked command complete, 171

 linked command complete with flag, 171

 message parity error, 171

 no operation, 171

 ordered queue tag, 173

 queue tag messages, 172

 simple queue tag, 173, 219

 supported messages functions, 182

microcode error, 283

microcode has changed, 280

minimum pre-fetch, 88

miscompare recovery, 282

mode parameters changed, 280

mode select command, 60

mode sense command, 100

mode sense data format (of all pages), 104

mode sense data format (of header & block descriptor), 102

modes of operation, 73

 DCR, 73

 DTE, 73

 PER, 73

 TB, 73

multiple initiator systems, 236

multiplication factor (MF), 87

N

no defect spare, 276

nonarbitrating systems, 235

non-cache segment size, 89

nondata error recovery procedures, 275

not ready to ready transition, 280

notch parameters, 94

notched drive (ND), 94

number of cache segments, 89

number of cylinders, 82

number of heads, 82

O

options jumper block, 237

options supported, 227

overrun counter, 55

P

page code field, 45

page control (PC) field, 45

page format (PF), 60

page header, 46

pages notched, 95

parameter pointer control (PPC), 45

parameter pointer field, 45

PC field, 45

PC meaning, 101

peripheral device parameters, 90

persistent reserve in command, 112

persistent reserve out command, 117

PList, 19

position error, 269

post error (PER), 72

- power conditions, 96
- PPC bit, 45
- predictive failure analysis, 283
- predictive failure analysis (PFA), 226
- pre-fetch command, 120
- priority commands, 218
- priority of error reporting, 269
- priority of SCSI status byte reporting, 199
- protocol, 199

Q

- qualify posted errors (QPE), 64
- queue algorithm modifier, 91
- queue error management (QErr), 92
- queue full, 220
- queuing, 219

R

- read buffer command, 124
- read buffer full ratio, 77
- read cache disable (RCD), 87
- read capacity command, 126
- read command (10), 123
- read command (6), 122
- read command data recovery procedure steps, 260
- read continuous (RC), 72
- read defect data (10), 127
- read defect data (12), 130
- read defect data command, 130
- read error counter, 51
- read long command, 133
- read recovery steps, 261
- read retry count (RRC), 69
- read-ahead function, 228
- ready AEN holdoff period, 92
- reassign a physical sector, 273
- reassign blocks command, 134
- reassign blocks recovery, 274
- reassign target pad (RTP), 69
- receive diagnostic results command, 136
- recommend rewrite errors, 272
- recommendations for system error log, 271
- recommended initiator error recovery procedures, 270
- reconnecting to a read command, 78
- reconnecting to a write command, 78
- recovered drive error, 276

- recovery procedures, 259, 264
- recovery time limit, 76
- release command (10), 140
- release command (6), 139
- report a check (RAC), 92
- report automatic/recommend rewrite/reallocate disabled (RARRRD), 69
- report log exception condition, 91
- report LUNs command, 141
- report predictive failure analysis error (RPFAE), 66
- report recovered non-data errors (RRNDE), 66
- request sense command, 143
- reselection retry count, 90
- reselection timeout, 235
- reserve command (10), 145
- reserve command (6), 144
- reserved area, 240
- reserved area hard error, 277
- reset, 280
- rewrite errors, 272
- rewrite on LBA, 272
- rezero unit command, 147
- rigid disk drive geometry parameters, 82
- rules for queuing commands, 220

S

- save data pointer, 169
- save parameter (SP), 45, 60
- saveable pages, 60
- SCSI bus parity error detection, 185
- SCSI bus related error handling protocol, 194
- SCSI message system, 168
- SCSI protocol, 199
- SCSI sense data format, 242
- SCSI status byte, 166
- sectors per track, 81
- seek command (10), 148
- seek command (6), 148
- seek counter, 54
- seeks, 54
- segment prioritization, 233
- segment selection, 232
- segmented caching, 227
- selection without ATN, 235
- self initiated reset, 281
- send diagnostic command, 149
- sense data description, 243

sense-key specific field, 256
servo error recovery, 269
simple queue tag message in phase retry, 197
single initiator selection, 235
size enable (SIZE), 87
Small Computer System Interface (SCSI), 4
software write protect (SWP), 92
SP bit, 45
spindle motor error, 269
SRP table, 269
start/stop unit command, 153
starting boundary, 95
starting cylinder-reduced write current, 83
starting cylinder-write precompensation, 82
status phase retry, 196
statuses
 auto-contingent allegiance active, 167
 busy, 166
 check condition, 166
 condition met, 166
 good, 166
 intermediate/condition met, 167
 intermediate/good, 167
 queue full, 167
 reservation conflict, 167
summary of SCSI commands, 17
supported message functions, 182
supported messages, 168
synchronize cache command, 154
synchronize cache on SCSI reset, 232
system error log, 271

T

TA branch of read recovery, 264
TA DRP table, 264
tagged queuing, 219
termination of I/O processes, 221
test unit ready command, 155
tests, basic assurance (BATS), 240
track skew factor, 81
tracks per zone, 80
transfer block (TB), 72

U

UEC list, 284, 285, 286, 287, 288, 289, 290
underrun counter, 55
unexpected bus free phase error condition, 194
unit attention, 280

unit error code, 257
unit error codes, 284, 285, 286, 287, 288, 289,
291
unrecovered drive error, 275
untagged queuing, 219

V

valid sense key, code qualifier combinations used
by the drive, 245
vendor unique parameters, 64
verify command, 156, 282
verify correction span, 85
verify recovery time limit, 85
verify retry count, 85
vital product data (VPD), 25

W

write and verify command, 159, 282
write buffer command, 161
write buffer full ratio, 77
write cache, 231
write cache enable (WCE), 87
write command (10), 158
write command (6), 157
write long command, 164
write retention priority, 88
write same command, 165

Z

zero seeks, 54