

# Working Draft American National Standard

**Project**  
**T10/1826-D**

Revision 01a  
11 July 2007

---

## Information technology - SCSI / ATA Translation - 2 (SAT-2)

This is an internal working document of T10, an International Technical Committee of Accredited Standards Committee INCITS (International Committee for Information Technology Standards). As such this is not a completed standard and has not been approved. The contents may be modified by the T10 Technical Committee. The contents are actively being modified by T10. This document is made available for review and comment only.

Permission is granted to members of INCITS, its technical committees, and their associated task groups to reproduce this document for the purposes of INCITS standardization activities without further permission, provided this notice is included. All other rights are reserved. Any duplication of this document for commercial or for-profit use is strictly prohibited.

T10 Technical Editor:

Mark A. Overby  
NVIDIA Corporation  
12131 131st Ave NE, Suite 203  
Kirkland, WA  
98034  
USA

Telephone: 425-417-9412

Email: [moverby@nvidia.com](mailto:moverby@nvidia.com)

---

Reference number  
ISO/IEC XXXXX-XXX : 200x  
ANSI BSR INCITS \*\*\*-2006

## Points of Contact:

### International Committee for Information Technology Standards (INCITS) T10 Technical Committee

#### T10 Chair

John B. Lohmeyer  
LSI Logic  
4420 Arrows West Drive  
Colorado Springs, CO 80907-3444  
USA

Tel: (719) 533-7560  
Fax: (719) 533-7183  
Email: lohmeyer@t10.org

#### T10 Vice-Chair

George O. Penokie  
IBM Corporation  
MS: 49C  
3605 Highway 52 N  
Rochester, MN 55901  
USA

Tel: (507) 253-5208  
Fax: (507) 253-2880  
Email: gop@us.ibm.com

#### INCITS Secretariat

INCITS Secretariat  
1250 Eye Street, NW Suite 200  
Washington, DC 20005

Telephone: 202-737-8888  
Facsimile: 202-638-4922  
Email: incits@itic.org

#### Information Technology Industry Council

Web site: <http://www.itic.org>

**T10 Web Site**     [www.t10.org](http://www.t10.org)

**T10 Reflector**     To subscribe send e-mail to [majordomo@T10.org](mailto:majordomo@T10.org) with 'subscribe' in message body  
To unsubscribe send e-mail to [majordomo@T10.org](mailto:majordomo@T10.org) with 'unsubscribe' in message body  
Internet address for distribution via T10 reflector: [T10@T10.org](mailto:T10@T10.org)

#### Document Distribution

INCITS Online Store  
managed by Techstreet  
1327 Jones Drive  
Ann Arbor, MI 48105  
USA

Web site: <http://www.techstreet.com/incits.html>  
Telephone: (734) 302-7801 or (800) 699-9277

Global Engineering Documents, an IHS Company     Telephone: 303-792-2181 or  
15 Inverness Way East     800-854-7179  
Englewood, CO 80112-5704     Facsimile: 303-792-2192

Web site: <http://global.ihs.com>  
Telephone: (303) 397-7956 or (303) 792-2181 or (800) 854-7179

# Draft

## American National Standards for Information Systems -

### SCSI / ATA Translation - 2 (SAT-2)

Secretariat  
National Committee for Information Technology Standards

Approved mm dd yy

American National Standards Institute, Inc.

#### Abstract

This standard specifies a translation layer between SCSI and ATA protocols. This translation layer is used by storage controllers to emulate objects in a SCSI logical unit using an ATA device, providing capabilities defined by SCSI standards (e.g., the SCSI Block Commands (SBC-2) and SCSI Primary Commands (SPC-3) standards). For the purposes of this standard, ATA device capabilities are defined by ATA8-AAM, ATA8-ACS, ATA8-APT, ATA8-AST, and SATA 2.6.

# Draft

## **American National Standard**

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered and that effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he or she has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and shall in no circumstances give interpretation on any American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

**CAUTION NOTICE:** This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

**CAUTION:** The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard. As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, notice of one or more claims has been received. By publication of this standard, no position is taken with respect to the validity of this claim or of any rights in connection therewith. The known patent holder(s) has (have), however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Published by  
**American National Standards Institute**  
11 West 42nd Street, New York, NY 10036

Copyright © 2007 by American National Standards Institute  
All rights reserved.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without prior written permission of ITI, 1250 Eye Street NW, Suite 200, Washington, DC 20005.

**Printed in the United States of America**

# Draft

**Contents**

	Page
1 Scope .....	1
2 Normative References .....	4
2.1 Normative references .....	4
2.2 Approved references .....	4
2.3 References under development .....	4
2.4 Other references .....	4
3 Definitions, symbols, abbreviations, and conventions .....	5
3.1 Definitions .....	5
3.2 Symbols and abbreviations .....	10
3.3 Keywords .....	11
3.4 SAT specific terminology .....	12
3.5 Conventions .....	12
3.5.1 Overview .....	12
3.5.2 Numeric conventions .....	12
3.5.3 Bit and byte ordering .....	13
3.5.4 Notation for byte encoded character strings .....	14
3.5.5 Use of and/or .....	15
3.5.6 Notation for command descriptions .....	15
3.5.6.1 Description .....	15
3.5.7 Use of field names defined in ATA standards and specifications .....	15
4 General .....	17
5 SCSI architectural elements .....	18
5.1 Overview .....	18
5.2 Unit attention condition .....	19
5.3 Handling errors in ATA commands .....	20
5.4 ATA nexus loss .....	20
5.5 ATA hardware and software reset processing .....	20
6 Task management model .....	22
6.1 Overview .....	22
6.2 Queued commands .....	23
6.2.1 Comparison of SCSI and ATA queuing .....	23
6.2.2 Mapping of SCSI queued commands to ATA queued commands .....	24
6.2.3 Commands the SATL queues internally .....	24
6.2.4 Command queuing with multiple I_T nexuses .....	24
6.2.5 Collateral abort with queued commands .....	25
6.3 Task management functions .....	25
6.3.1 Task management functions overview .....	25
6.3.2 Aborting ATA queued commands .....	25
6.3.3 Aborting ATA non-queued commands .....	26
6.3.4 ABORT TASK .....	26
6.3.5 ABORT TASK SET .....	26
6.3.6 CLEAR ACA .....	27
6.3.7 CLEAR TASK SET .....	27
6.3.8 LOGICAL UNIT RESET .....	28
6.3.9 QUERY TASK .....	28
6.3.10 Obsolete reset task management functions .....	28
6.4 Control Byte .....	28
6.4.1 CONTROL byte overview .....	28
6.5 I_T nexus loss .....	29

7	Summary of SCSI / ATA command mappings .....	30
7.1	Translated and emulated commands .....	30
8	SCSI Primary Commands (SPC) mapping .....	32
8.1	INQUIRY command .....	32
8.1.1	INQUIRY command overview .....	32
8.1.2	Standard INQUIRY data .....	33
8.2	LOG SENSE command .....	35
8.2.1	LOG SENSE command overview .....	35
8.2.2	PC (page control) field .....	36
8.2.3	PAGE CODE field .....	36
8.3	MODE SELECT (6) command .....	37
8.3.1	MODE SELECT (6) command overview .....	37
8.3.2	MODE SELECT (6) CDB fields .....	37
8.4	MODE SELECT (10) command .....	37
8.5	MODE SENSE (6) command .....	37
8.5.1	MODE SENSE (6) command overview .....	37
8.5.2	MODE SENSE (6) CDB fields .....	38
8.6	MODE SENSE (10) command .....	38
8.7	READ BUFFER command .....	39
8.7.1	READ BUFFER command overview .....	39
8.7.2	MODE field .....	39
8.7.2.1	MODE field overview .....	39
8.7.2.2	Data mode .....	39
8.7.2.3	Descriptor mode .....	40
8.8	REQUEST SENSE command .....	40
8.8.1	REQUEST SENSE command overview .....	40
8.8.2	FORMAT UNIT in progress .....	41
8.8.3	SMART threshold exceeded condition .....	41
8.8.4	Stopped power condition .....	41
8.8.5	Unit attention condition established .....	41
8.9	SECURITY PROTOCOL IN command .....	42
8.9.1	SECURITY PROTOCOL IN command overview .....	42
8.9.1.1	SECURITY PROTOCOL field .....	42
8.9.1.2	SECURITY PROTOCOL SPECIFIC field .....	42
8.9.1.3	<b>allocation length</b> field .....	42
8.9.1.3.1	<b>allocation length</b> field translation overview .....	42
8.9.1.3.2	<b>security protocol</b> 00h - 06h .....	42
8.9.1.3.3	<b>security protocol</b> values 07h - FFh .....	43
8.10	SECURITY PROTOCOL OUT command .....	43
8.10.0.1	SECURITY PROTOCOL OUT command overview .....	43
8.10.0.2	<b>security protocol</b> field .....	43
8.10.0.3	<b>security protocol specific</b> field .....	43
8.10.0.4	<b>allocation length</b> field .....	43
8.10.0.4.1	<b>security protocol</b> field translation overview .....	44
8.10.0.4.2	<b>security protocol</b> values 00h - 06h .....	44
8.10.0.4.3	<b>security protocol</b> values 07h - FFh .....	44
8.11	SEND DIAGNOSTIC command .....	45
8.11.1	SEND DIAGNOSTIC command overview .....	45
8.11.2	SELF-TEST CODE field .....	46
8.11.3	SELFTTEST bit .....	48
8.12	TEST UNIT READY command .....	50
8.12.1	TEST UNIT READY command overview .....	50
8.12.2	TEST UNIT READY command translation .....	50
8.13	WRITE BUFFER command .....	51
8.13.1	WRITE BUFFER command overview .....	51
8.13.2	MODE field .....	51

8.13.2.1	MODE field overview .....	51
8.13.2.2	Write data mode .....	52
8.13.2.3	Download microcode mode .....	52
9	SCSI Block Commands (SBC) mapping .....	53
9.1	Translating LBA and transfer length and ATA command use constraints .....	53
9.1.1	Overview .....	53
9.1.2	Direct logical block mapping model .....	53
9.1.3	Indirect logical block mapping model .....	53
9.1.4	Selection of ATA block commands .....	53
9.2	FORMAT UNIT command .....	55
9.2.1	FORMAT UNIT command overview .....	55
9.2.2	FORMAT UNIT parameter list .....	56
9.2.3	SATL defect list header field combinations .....	56
9.2.4	FOV bit .....	56
9.2.5	DCRT bit .....	56
9.2.6	IP bit .....	57
9.3	READ commands overview .....	57
9.3.1	READ commands operation code translation .....	57
9.3.2	READ commands with FUA .....	57
9.4	READ (6) command .....	58
9.5	READ (10) command .....	58
9.6	READ (12) command .....	60
9.7	READ (16) command .....	61
9.8	READ CAPACITY (10) command .....	62
9.8.1	READ CAPACITY (10) command overview .....	62
9.8.2	READ CAPACITY (10) parameter data .....	62
9.9	READ CAPACITY (16) command .....	63
9.9.1	READ CAPACITY (16) command overview .....	63
9.9.2	READ CAPACITY (16) parameter data .....	63
9.10	REASSIGN BLOCKS command .....	64
9.10.1	REASSIGN BLOCKS command overview .....	64
9.10.2	REASSIGN BLOCKS operation code .....	64
9.11	START STOP UNIT command .....	66
9.11.1	START STOP UNIT command overview .....	66
9.11.2	Processing ending status if an error occurs .....	66
9.11.3	START STOP UNIT START bit LOEJ bit combinations .....	67
9.12	SYNCHRONIZE CACHE (10) command .....	68
9.13	SYNCHRONIZE CACHE (16) command .....	68
9.14	VERIFY (10) command .....	69
9.15	VERIFY (12) command .....	70
9.16	VERIFY (16) command .....	71
9.17	WRITE commands overview .....	72
9.17.1	WRITE commands operation code translation .....	72
9.17.2	WRITE commands with FUA .....	72
9.18	WRITE (6) command .....	73
9.19	WRITE (10) command .....	74
9.20	WRITE (12) command .....	75
9.21	WRITE (16) command .....	76
9.22	WRITE AND VERIFY commands overview .....	76
9.23	WRITE AND VERIFY (10) command .....	77
9.24	WRITE AND VERIFY (12) command .....	78
9.25	WRITE AND VERIFY (16) command .....	79
9.26	WRITE SAME (10) command .....	80
9.26.1	WRITE SAME (10) command overview .....	80
9.26.2	LBDATA bit and PBDATA bit .....	81
9.27	WRITE SAME (16) command .....	82

10 Parameters for SAT implementations .....	83
10.1 Mode parameters .....	83
10.1.1 General information .....	83
10.1.2 Commonly used SCSI mode pages overview .....	83
10.1.3 Mode parameter headers .....	84
10.1.4 Mode parameter block descriptor fields .....	85
10.1.5 Control mode page .....	85
10.1.5.1 General translation .....	85
10.1.5.2 Extended self-test completion time .....	87
10.1.6 Read-Write Error Recovery mode page .....	87
10.1.7 Caching mode page .....	88
10.1.8 Informational Exceptions Control mode page .....	90
10.1.8.1 Informational Exceptions Control mode page overview .....	90
10.1.8.2 Method of reporting informational exceptions (MRIE) .....	90
10.2 Log pages .....	91
10.2.1 Log pages overview .....	91
10.2.2 Application Client log page .....	91
10.2.2.2 LOG SELECT translation .....	92
10.2.2.3 LOG SENSE translation .....	93
10.2.3 Supported Log Pages log page .....	94
10.2.4 Self-Test Results log page .....	94
10.2.4.1 Self-Test Results log page overview .....	94
10.2.4.2 A method of determining ATA command selection for field translations .....	97
10.2.4.3 Sense key and additional sense code .....	98
10.2.5 Informational Exceptions log page .....	98
10.2.5.1 Informational Exceptions log page overview .....	98
10.2.5.2 Additional sense code and additional sense code qualifier translations .....	99
10.2.5.3 Most recent temperature reading translation .....	100
10.3 Vital product data parameters .....	100
10.3.1 Vital product data parameters overview .....	100
10.3.2 Supported VPD pages VPD page .....	100
10.3.3 Unit Serial Number VPD page .....	100
10.3.4 Device Identification VPD page .....	102
10.3.4.1 Device Identification VPD page overview .....	102
10.3.4.2 Logical unit name .....	102
10.3.4.2.1 Logical unit name overview .....	102
10.3.4.2.2 Logical unit name derived from the world wide name .....	103
10.3.4.2.3 Logical unit name derived from the model number and serial number .....	104
10.3.4.3 Examples of additional identification descriptors .....	105
10.3.4.3.1 Identification descriptors included by a SATL in an ATA host .....	105
10.3.4.3.2 Identification descriptors included by a SATL in a SAS initiator device .....	105
10.3.4.3.3 Identification descriptors included by a SATL in a SCSI to ATA protocol bridge .....	106
10.3.5 Mode Page Policy VPD page .....	107
11 Error and sense reporting .....	108
11.1 Error translation – ATA device error to SCSI error map .....	108
12 SAT-specific SCSI extensions .....	109
12.1 SAT-specific SCSI extensions overview .....	109
12.2 ATA PASS-THROUGH commands .....	109
12.2.1 ATA PASS-THROUGH commands overview .....	109
12.2.2 ATA PASS-THROUGH (12) command .....	110
12.2.3 ATA PASS-THROUGH (16) command .....	113
12.2.4 Mapping of ATA PASS-THROUGH CDB field translations .....	114
12.2.5 ATA PASS-THROUGH status return .....	115
12.2.6 ATA Return descriptor .....	116
12.3 SAT-specific mode pages .....	117

- 12.3.1 SAT-specific mode pages overview ..... 117
- 12.3.2 PATA Control mode page ..... 117
- 12.4 SAT-specific VPD pages ..... 122
  - 12.4.1 SAT-specific VPD pages overview ..... 122
  - 12.4.2 ATA Information VPD page ..... 122
    - 12.4.2.1 ATA Information VPD page overview ..... 122
    - 12.4.2.2 ATA device signature ..... 124
    - 12.4.2.3 ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data ..... 125
- Annex A INQUIRY command translation for ATAPI devices ..... 126
  - A.1 INQUIRY command overview ..... 126
  - A.2 Supported VPD Pages VPD page ..... 126
  - A.3 ATA Information VPD page ..... 126

## Tables

	Page
1 ISO and American numbering conventions examples .....	13
2 Example of ordering of bits and bytes within a multi-byte element .....	14
3 Example of ordering of bits and bytes within a multiple element .....	14
4 Format for translated command field descriptions .....	15
5 Comparison of SCSI and ATA queuing methods .....	23
6 SATL processing of ATA commands aborted by ATA collateral abort .....	25
7 Control byte fields .....	28
8 Summary of SCSI / ATA Command Mapping .....	30
9 INQUIRY CDB field translations .....	32
10 Standard INQUIRY data fields .....	33
11 LOG SENSE CDB field translations .....	35
12 PC field .....	36
13 PAGE CODE field .....	36
14 MODE SELECT (6) CDB field translations .....	37
15 MODE SENSE (6) CDB field translations .....	38
16 READ BUFFER CDB field translations .....	39
17 MODE field .....	39
18 Special Request Sense behavior reference .....	40
19 REQUEST SENSE CDB field translations .....	41
20 SECURITY PROTOCOL IN CDB field translation .....	42
21 SECURITY PROTOCOL OUT CDB field translation .....	43
22 SEND DIAGNOSTIC CDB field translations .....	45
23 SELF-TEST CODE field translation .....	46
24 SELFTEST bit .....	49
25 TEST UNIT READY CDB field translations .....	50
26 WRITE BUFFER CDB field translations .....	51
27 MODE field .....	51
28 ATA commands used for SCSI block command translations .....	54
29 FORMAT UNIT CDB field translations .....	55
30 SATL defect list header .....	56
31 SATL defect list header field combinations .....	56
32 READ (6) CDB field translations .....	58
33 READ (10) CDB field translations .....	59
34 READ (12) CDB field translations .....	60
35 READ (16) CDB field translations .....	61
36 READ CAPACITY (10) CDB field translations .....	62
37 READ CAPACITY (10) parameter data .....	62
38 READ CAPACITY(16) CDB field translations .....	63
39 READ CAPACITY (16) parameter data .....	63
40 REASSIGN BLOCKS CDB field translations .....	64
41 START/STOP UNIT CDB field translations .....	66
42 Definition of START and LOEJ bits in the START STOP UNIT CDB .....	67
43 SYNCHRONIZE CACHE (10) CDB field translations .....	68
44 SYNCHRONIZE CACHE (16) CDB field translations .....	68
45 VERIFY (10) CDB field translations .....	69
46 VERIFY (12) CDB field translations .....	70
47 VERIFY (16) CDB field translations .....	71
48 WRITE (6) CDB field translations .....	73
49 WRITE (10) CDB field translations .....	74
50 WRITE (12) CDB field translations .....	75
51 WRITE (16) CDB field translations .....	76
52 WRITE AND VERIFY (10) CDB field translations .....	77
53 WRITE AND VERIFY (12) CDB field translations .....	78
54 WRITE AND VERIFY (16) CDB field translations .....	79
55 WRITE SAME (10) CDB field translations .....	80

56 LBDATA and PBDATA fields .....	81
57 WRITE SAME (16) CDB field translations .....	82
58 Summary of SCSI / ATA mode page mapping .....	83
59 Mode parameter header (6) fields .....	84
60 Mode parameter header (10) fields .....	84
61 Mode parameter block descriptor fields .....	85
62 Control mode page fields .....	86
63 Read-Write Error Recovery mode page fields .....	87
64 Caching mode page fields .....	88
65 Informational Exceptions Control mode page fields .....	90
66 Summary of SCSI / ATA log page mapping .....	91
67 General usage application client parameter data fields .....	92
68 Parameter Storage Location .....	93
69 Supported Log Pages log page fields .....	94
70 Self-Test Results log page fields .....	94
71 Self-Test Results log parameters .....	94
72 ATA Self-test execution status values translated to SCSI sense keys and sense codes .....	98
73 Informational Exceptions log page header fields .....	98
74 Informational Exceptions general parameter data .....	99
75 ATA SMART RETURN STATUS translations .....	99
76 Summary of SCSI / ATA VPD page mapping .....	100
77 Supported VPD pages VPD page fields .....	100
79 PRODUCT SERIAL NUMBER field .....	101
78 Unit Serial Number VPD page for SAT .....	101
80 Device Identification VPD page for SAT .....	102
81 Logical unit name derived from the world wide name .....	103
82 Fields in the logical unit name .....	103
83 Logical unit name derived from the model number and serial number .....	104
84 VENDOR SPECIFIC IDENTIFIER field for logical unit name .....	104
85 Target port identifier for SAS .....	106
86 Mode Page Policy VPD page for SAT .....	107
87 Mode policy descriptor for SAT .....	107
88 Translation of ATA errors to SCSI errors .....	108
89 ATA PASS-THROUGH (12) command .....	110
90 PROTOCOL field .....	110
91 ATA PASS-THROUGH (12) command and ATA PASS-THROUGH (16) command DEVICE field .....	112
92 T_LENGTH field .....	112
93 ATA PASS-THROUGH (16) command .....	113
94 Mapping of ATA PASS-THROUGH (16) CDB fields to ATA command fields .....	114
95 EXTEND bit and T_LENGTH field .....	114
96 ATA command results .....	115
97 ATA Return descriptor .....	116
98 SCSI / ATA Translation specific mode pages .....	117
99 PATA Control mode page .....	117
100 PIO modes .....	118
101 MWDMA modes reported by MODE SENSE .....	119
102 UDMA bits requirements for changeable MODE SENSE parameters .....	120
103 UDMA for current MODE SENSE settings .....	120
104 ATA Information VPD page .....	122
105 ATA device signature .....	124
106 TRANSPORT IDENTIFIER field values .....	124
107 ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data .....	125

**Figures**

	Page
1 SCSI document relationships .....	2
2 ATA document structure .....	2
3 SCSI / ATA Translation document role .....	3
4 Example of a SATL between a SCSI application client and an ATA or ATAPI device .....	18
5 SATL contained within a SCSI to ATA protocol bridge .....	19
6 SATL contained within an ATA host .....	19
7 SATL contained in a SAS initiator device .....	19
8 REASSIGN BLOCKS command translation .....	65
9 Identification descriptors included by a SATL in an ATA host .....	105
10 Identification descriptors included by a SATL in a SAS initiator device .....	105
11 Identification descriptors included by a SATL in a SCSI to ATA protocol bridge .....	106

## Revision Information

### **R.0 SAT-2 r00 (22 February 2007)**

Created revision 0 from original SAT final document. Updated references.

### **R.1 SAT-2 r01 (09 July 2007)**

Incorporated 07-148r0 as approved by the May 2007 plenary.

Incorporated 07-146r0 as approved by the May 2007 plenary.

Incorporated 06-250r1 as approved by the March 2007 plenary.

Removed reference in command summary table to READ MEDIA SERIAL number as that was removed from SAT, but the table still contained the reference.

Corrected minor typographical errors pointed out by Rob Elliott (HP)

Changes all references to direct or indirect block mapping to direct or indirect logical block mapping to match SBC-3

### **R.1a SAT-2 r02 (11 July 2007)**

Incorrectly stated in SAT-2 r01 that 06-250r1 was incorporated as approved. What was actually incorporated was 07-074r2 as approved in the May 2007 plenary.

Correctly incorporated 06-250r1 as approved by the March 2007 plenary.

## Foreword

This foreword is not part of American National Standard INCITS \*\*\*-200x.

This standard provides a common set of definitions and requirements to establish common behavior among implementations that emulate SCSI device behavior through the combined use of ATA devices and a SCSI / ATA Translation layer (SATL). The SATL may reside in a host-based driver, or it may reside in a separate component (e.g., a host bus adapter or external controller) with a separate processing unit to effect the translation. A SATL and ATA device combination may provide a functional subset of common SCSI capabilities. There is also a range of optional emulated SCSI capabilities that may be supported or not, depending on the capabilities of the SATL.

This standard defines SATL capabilities in terms of SCSI capabilities as defined by the applicable SCSI standards and working drafts, and defines the elements and use of ATA protocol to provide those SCSI capabilities and services in a consistent manner among SAT implementations that implement according to this standard.

With any technical document there may arise questions of interpretation as new products are implemented. INCITS has established procedures to issue technical opinions concerning the standards developed by INCITS. These procedures may result in SCSI Technical Information Bulletins being published by INCITS.

These Bulletins, while reflecting the opinion of the Technical Committee that developed the standard, are intended solely as supplementary information to other users of the standard. This standard, ANSI BSR INCITS \*\*\*-2006, as approved through the publication and voting procedures of the American National Standards Institute, is not altered by these bulletins. Any subsequent revision to this standard may or may not reflect the contents of these Technical Information Bulletins.

Current INCITS practice is to make Technical Information Bulletins available through:

INCITS Online Store	<a href="http://www.techstreet.com/INCITS.html">http://www.techstreet.com/INCITS.html</a>
managed by Techstreet	Telephone: 1-734-302-7801 or
1327 Jones Drive	1-800-699-9277
Ann Arbor, MI 48105	Facsimile: 1-734-302-7811

or

Global Engineering	<a href="http://global.ihs.com/">http://global.ihs.com/</a>
15 Inverness Way East	Telephone: 1-303-792-2181 or
Englewood, CO 80112-5704	1-800-854-7179
	Facsimile: 1-303-792-2192

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, National Committee for Information Technology Standards, Information Technology Institute, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by the InterNational Committee for Information Technology Standards (INCITS). Committee approval of the standard does not necessarily imply that all committee members voted for approval.

Technical Committee T10 on Lower Level Interfaces, which developed and reviewed this standard, had the following members:

John B. Lohmeyer, Chair  
George O. Penokie, Vice-Chair  
Ralph O. Weber, Secretary

<i>Organization Represented</i>	<i>Name of Representative</i>
Adaptec, Inc. ....	Chris Fore Tom Treadway (Alt)
AMCC .....	Paul von Stamwitz
Amphenol Interconnect .....	Michael Wingard Brian Miller (Alt)
Broadcom Corp. ....	Paul Griffith
Brocade Comm. Systems, Inc. ....	Robert Snively Steven Wilson (Alt)
Dallas Semiconductor .....	James A. Lott, Jr. Steve Robalino (Alt)
Dell, Inc. ....	Kevin Marks
EMC Corp. ....	Gary S. Robinson David Black (Alt) Mickey Felton (Alt)
Emulex .....	Robert H. Nixon Kenneth Hirata (Alt)
ENDL Texas .....	Ralph O. Weber I. Dal Allan (Alt)
FCI .....	Douglas Wagner
Foxconn Electronics .....	Elwood Parsons
Fujitsu.....	Mike Fitzpatrick Ben-Koon Lin (Alt)
General Dynamics .....	Nathan Hastad Tim Mackley (Alt)
Hewlett Packard Co. ....	Rob Elliott William Ham (Alt)
Hitachi Cable Manchester .....	Zane Daggett Mike Guzman (Alt) Hitoshi (Todd) Horita (Alt)
Hitachi Global Storage Tech.....	Dan Colegrove Sally Castillo (Alt)
IBM Corp. ....	George O. Penokie Kevin Butt (Alt)
Intel Corp. ....	Robert Sheffield Mark Seidel (Alt)
Iomega Corp. ....	Robert Payne
KnowledgeTek, Inc. ....	Dennis Moore
Lexar Media, Inc. ....	John Geldman Martin Furuhjelm (Alt) Pat LaVarre (Alt)
LSI Logic Corp. ....	John Lohmeyer Keith Holt (Alt)
M-Systems .....	Avraham Shimor Ariel Sobelman (Alt)
Marvell Semiconductor, Inc. ....	David Geddes Paul Wassenberg (Alt) Wei Zhou (Alt)
McDATA .....	David Peterson
Microsoft Corporation.....	Mark Benedikt Robert Griswold (Alt) David Walp (Alt)

Molex Inc.....	Jay Neer Galen Fromm (Alt) Jerry Kachlic (Alt)
Decru .....	Gideon Avida Frederick Knight (Alt)
Nvidia Corp. ....	Mark Overby
Oracle .....	James Williams William Bridge (Alt)
Panasonic Technologies, Inc .....	Terence J. Nelson Takaharu Ai (Alt)
Philips Electronics/CD Edge .....	William P. McFerrin
Pivot3, Inc. ....	Bill Galloway Thomas Grieff (Alt)
PMC-Sierra .....	Tim Symons Yuriy Greshishchev (Alt)
QLogic Corp.....	Skip Jones Craig W. Carlson (Alt)
Quantum Corp. ....	Paul Entzel Paul Suhler (Alt)
Seagate Technology .....	Gerald Houlder Alvin Cox (Alt)
Sierra Logic, Inc. ....	William Martin Narayan Ayalasomayajula (Alt)
Sun Microsystems, Inc.....	Vit Novak Charles Binford (Alt) Dwayne Edling (Alt) Erich Oetting (Alt)
Symantec .....	Roger Cummings Raymond Gilson (Alt)
Texas Instruments .....	Paul D. Aloisi Robert Kando (Alt)
TycoElectronics .....	Ashlie Fan Dan Gorenc (Alt)
Vitesse Semiconductor .....	Gregory Tabor David Allen (Alt) Jim Scott (Alt) Kevin Witt (Alt)
Western Digital.....	Curtis Stevens Mark Evans (Alt)
Xiotech Corp. ....	Jeff Williams
Xyratex.....	Rich Ramos

## Introduction

The SCSI / ATA Translation - 2 (SAT-2) standard is divided into the following clauses:

Clause 1 defines the scope of this standard.

Clause 2 enumerates the normative references that apply to this standard.

Clause 3 describes the definitions, symbols, abbreviations, and notation conventions used in this standard.

Clause 4 describes the general framework for defining elements of translation between SCSI and ATA protocol.

Clause 5 describes elements of SCSI / ATA Translation that relate to the SCSI architecture model.

Clause 6 describes the mapping of task management functions in the SATL layer.

Clause 7 provide a summary of SCSI commands mapped to ATA in this standard.

Clause 8 describes the mapping between SCSI Primary Commands and ATA protocol.

Clause 9 describes the mapping between SCSI Block Commands and ATA protocol.

Clause 10 describes the mapping of mode pages, log pages, and VPD page information to selected ATA protocol elements.

Clause 11 describes error reporting and sense data conventions for SCSI / ATA Translation.

Clause 12 describes SCSI commands and mode pages to support SCSI / ATA Translation.

Annex A describes the INQUIRY command translation for ATAPI devices.

**American National Standard for Information Systems -  
Information Technology -  
SCSI / ATA Translation****1 Scope**

The set of SCSI standards specifies the interfaces, functions, and operations necessary to ensure interoperability between conforming SCSI implementations. This standard is a functional description. Conforming implementations may employ any design technique that does not violate interoperability.

This standard defines the protocol requirements of the SCSI / ATA Translation Layer (SATL) to allow conforming SCSI / ATA translating elements to interoperate with ATA devices and SCSI application layers. The SATL covers the range of implementations that use ATA devices to emulate the behavior of SCSI devices as viewed by the SCSI application layer. The primary focus of this standard is to define SCSI / ATA Translation for an ATA device (see 3.1.8).

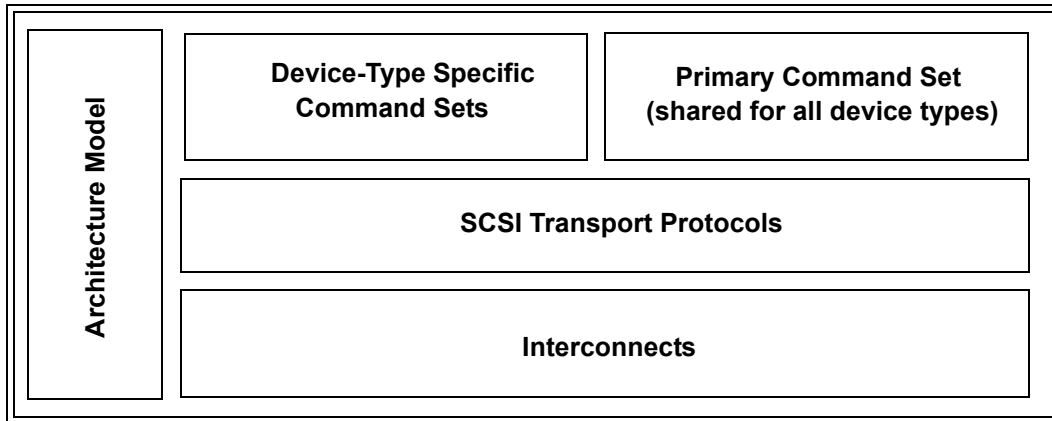
Where possible, this standard defines SCSI / ATA Translation in a manner that is consistent with the SAM-3, SPC-3, and SBC-2 standards. In some instances, the defined function of an ATA device is different from corresponding functions defined for SCSI target devices (e.g., an ATA device provides no means to abort a single ATA queued command). The translation defined in this standard, in such cases, may not be consistent with other SCSI standards. However, in such cases, this standard specifies the expected behavior, and in what manner it is inconsistent with the behavior specified in other SCSI standards.

The objective of this standard is to allow a complete set of SCSI functions while minimizing the complexity of the SATL and preserving compatibility with existing SCSI application clients.

The objectives of the SATL are:

- a) to provide host computers with device independence with respect to the ATA devices that have user storage capacity, and with respect to various implementations of the translation layer used to emulate the behavior of SCSI target devices;
- b) to define common elements representing a subset of the capabilities available in SCSI devices that apply to SCSI / ATA Translation implementations;
- c) to define common methods to manage aspects of ATA devices that do not map to previously defined elements of SCSI, with provision made for the addition of special features and functions; and
- d) to provide consistent means for discovery and control of optional SCSI features that may or may not be emulated in SCSI / ATA translator implementations. These means are provided by specifying how transport specific elements are represented in a mixed-domain topology in a manner consistent with management of devices in a SCSI domain.

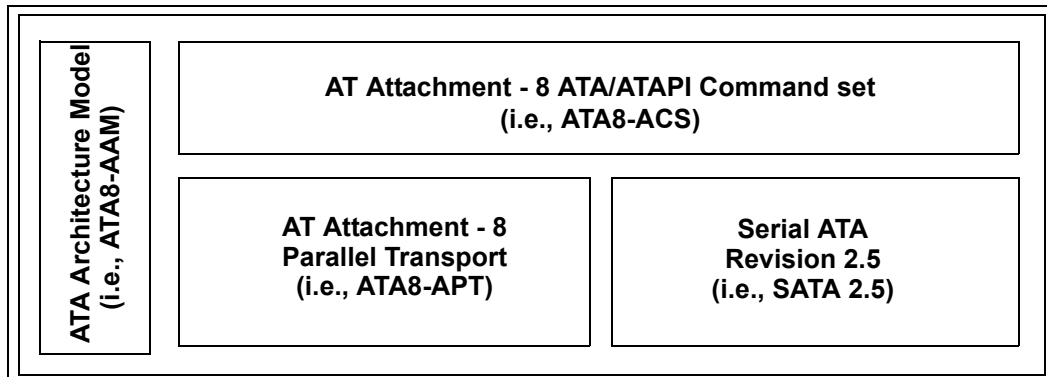
Figure 1 shows the general structure of SCSI standards. Figure 1 is not intended to imply a relationship such as a hierarchy, protocol stack, or system architecture.



**Figure 1 — SCSI document relationships**

The term SCSI is used wherever it is not necessary to distinguish between the different SCSI standards.

Figure 2 shows the relationship of the ATA8 documents to each other.



**Figure 2 — ATA document structure**

ATA-8 AST and SATA 2.6 provide similar descriptions for an ATA serial transport. Requirements of this standard depend upon the ATA serial transport as defined by SATA 2.6, but not ATA8-AST.

---



---

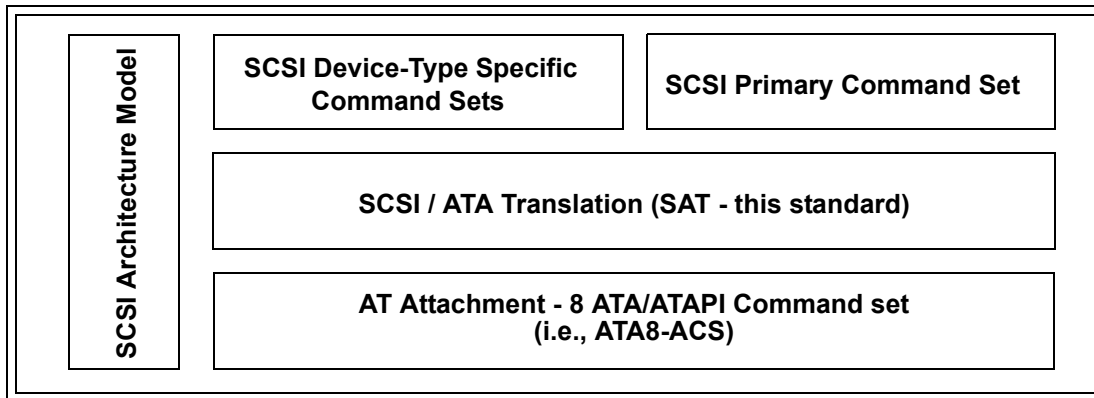
[Editor's Note 1: Globally update SATA 2.5 to SATA 2.6 and add SATA 2.6 to references](#)

---



---

Figure 3 shows the relationship of this standard to standards in both the SCSI family of standards and the ATA family of standards.



**Figure 3 — SCSI / ATA Translation document role**

This standard defines the following aspects of the translation between the SCSI application layer (see SAM-3) and ATA device protocol:

- a) the mapping of selected SCSI commands to ATA commands;
- a) the emulation of SCSI task management requests and the effects on ATA devices;
- b) the generation of responses to SCSI task management requests;
- c) the mapping the SCSI management model to ATA commands (e.g., NCQ);
- d) the mapping of ATA IDENTIFY DEVICE data and other information to standard INQUIRY data;
- e) the mapping of SCSI parameter data to ATA equivalents;
- f) the mapping mode page fields to the capabilities provided by underlying ATA devices;
- g) the mapping of ATA self-test information to log pages;
- h) the mapping of ATA IDENTIFY DEVICE data to common and protocol-specific VPD pages;
- i) the mapping of SCSI informational exceptions to ATA SMART functions;
- j) the mapping of ATA device responses to SCSI sense data reporting;
- k) the ATA PASS-THROUGH command; and
- l) other capabilities that may fit within the scope of this standard.

## 2 Normative References

### 2.1 Normative references

The following standards contain provisions that, by reference in the text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

Copies of the following documents may be obtained from ANSI: approved ANSI standards, approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), and approved and draft foreign standards (including BSI, JIS, and DIN). For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at <http://www.ansi.org>.

Additional availability contact information is provided below as needed.

### 2.2 Approved references

ANSI INCITS 397-2005, *AT Attachment with Packet Interface - 7 (ATA/ATAPI-7)* [ISO/IEC 24739-1]  
ANSI INCITS 402-2005, *SCSI Architecture Model - 3 (SAM-3)* [ISO/IEC 14776-413]  
ANSI INCITS 366-2003, *SCSI Architecture Model - 2 (SAM-2)* [ISO/IEC 14776-412]  
ANSI INCITS 408-2005, *SCSI Primary Commands - 3 (SPC-3)* [ISO/IEC 14776-453]  
ANSI INCITS 405-2005, *SCSI Block Commands - 2 (SBC-2)* [ISO/IEC 14776-322]  
ANSI INCITS 417-2006, *Serial Attached SCSI - 1.1 (SAS-1.1)* [ISO/IEC 14776-151]  
ANSI INCITS TR38-2005, *SMART Command Transport (SCT)*

### 2.3 References under development

At the time of publication, the following referenced standards were still under development. For information on the current status of the document, or regarding availability, contact the relevant standards body or other organization as indicated.

ISO/IEC 14776-861, *AT Attachment-8 Architecture Model (ATA8-AAM)* [T13/1700D]  
ISO/IEC 14776-871, *AT Attachment-8 ATA/ATAPI Command Set (ATA8-ACS)* [T13/1699D]  
ISO/IEC 14776-881, *AT Attachment-8 Parallel Transport (ATA8-APT)* [T13/1698D]  
ISO/IEC 14776-454, *SCSI Primary Commands - 4 (SPC-4)* [T10/1731-D]

NOTE 1 - For more information on the current status of the document, contact the INCITS Secretariat at 202-737-8888 (phone), 202-638-4922 (fax) or via E-mail at [incits@itic.org](mailto:incits@itic.org). To obtain copies of this document, contact Global Engineering at 15 Inverness Way, East Englewood, CO 80112-5704 at 303-792-2181 (phone), 800-854-7179 (phone), or 303-792-2192 (fax).

### 2.4 Other references

For information on the current status of the listed document(s), or regarding availability, contact the indicated organization.

*Serial ATA Revision 2.5 (SATA 2.5)*

The SATA 2.5 document may be obtained from Serial ATA International Organization (SATA IO) at [www.sata-io.org](http://www.sata-io.org).

*Mass Storage Class Bulk-Only Transport 1.0 (USB-BOT)*

The USB-BOT document may be obtained from the USB Implementers Forum, Inc. at [www.usb.org](http://www.usb.org).

## 3 Definitions, symbols, abbreviations, and conventions

### 3.1 Definitions

**3.1.1 additional sense code:** A combination of the ADDITIONAL SENSE CODE field and the ADDITIONAL SENSE CODE QUALIFIER field in the sense data (see SPC-3).

**3.1.2 allocation length:** A value in the ALLOCATION LENGTH field of a command descriptor block that specifies the maximum number of bytes that an application client has allocated in the Data-In Buffer, and that is used to limit the maximum amount of variable length data (e.g., mode data, log data, diagnostic data) returned to an application client (see SPC-3).

**3.1.3 application client:** An object that is the source of SCSI commands (see SAM-3).

**3.1.4 AT Attachment (ATA):** A family of standards that define the attachment of storage devices to hosts (see ATA8-AAM, ATA8-ACS, ATA8-APT, and SATA 2.5).

**3.1.5 AT Attachment Packet Interface (ATAPI):** The elements of the ATA standards that define the PACKET Command feature set, which provides the capability to encapsulate SCSI and other types of commands and pass them through an ATA transport.

**3.1.6 ATA abort retry:** A policy implemented by a SATL whereby the SATL retries ATA commands aborted by ATA collateral abort (see 3.1.7) once.

**3.1.7 ATA collateral abort:** An ATA command that is aborted as a side effect of a different command being aborted when an ATA device is processing queued commands (i.e., NCQ or TCQ).

**3.1.8 ATA device:** A device that is compliant with the ATA standards, implements the General feature set, and does not implement the PACKET Command feature set (see ATA8-ACS).

**3.1.9 ATA device capacity:** The capacity of an ATA device, in bytes, is the ATA logical sector size (see 3.1.15) times one more than the ATA maximum LBA (see 3.1.16).

**3.1.10 ATA domain:** An I/O subsystem that is made up of one ATA host, a service delivery subsystem, and one or more ATA devices or ATAPI devices (see ATA8-AAM).

**3.1.11 ATA flush command:** One of the following ATA commands: FLUSH CACHE or FLUSH CACHE EXT (see ATA8-ACS).

**3.1.12 ATA hardware reset:** The routines performed by the ATA device server and the ATA device port in an ATA device after a hardware reset event occurs (see ATA8-AAM). The hardware reset routines performed by the ATA device include the actions performed by the ATA device for an ATA software reset (see 3.1.22 and ATA-AAM), and the actions defined in ATA8-ACS and the applicable ATA transport standards (see 5.5).

**3.1.13 ATA host:** An object that originates requests to be processed by an ATA device or an ATAPI device.

**3.1.14 ATA LBA:** A logical block address (see 3.1.47) used to reference a logical sector in an ATA device (see ATA8-ACS).

**3.1.15 ATA logical sector size:** The size of an ATA logical sector in bytes, calculated as two times the value returned in ATA IDENTIFY DEVICE data words 118:117 (see ATA8-ACS) if the ATA device returns a value of one in ATA IDENTIFY DEVICE data word 106 bit 12 (i.e., the ATA device supports the Long Logical Sector feature set). The size of an ATA Logical Sector is 512 bytes if the ATA device returns a value of zero in ATA IDENTIFY DEVICE data word 106 bit 12 (i.e., the ATA device does not support the Long Logical Sector feature set).

NOTE 2 - The Logical Sector Size indicated by an ATA device is represented in words; therefore, the number of bytes in an ATA device logical sector is two times the value indicated in the Logical Sector Size.

**3.1.16 ATA maximum LBA:** The maximum user LBA for 48-bit address feature set returned in ATA IDENTIFY DEVICE data words (103:100) minus one if the ATA device returns a value of one in ATA IDENTIFY DEVICE data word 86 bit 10 (i.e., the ATA device supports the 48-bit address feature set), or the total number of user addressable sectors returned in ATA IDENTIFY DEVICE data words (61:60) minus one if the ATA device returns a value of zero in ATA IDENTIFY DEVICE data word 86 bit 10 (i.e., the ATA device does not support the 48-bit address feature set) (see ATA8-ACS),

**3.1.17 ATA nexus loss event:** A transport-specific event where an ATA host port is no longer in communication with an ATA device port (see ATA8-AAM, see 5.4).

**3.1.18 ATA non-queued command:** An ATA command that is not an ATA queued command (see 3.1.19).

**3.1.19 ATA queued command:** One of the following ATA commands: READ DMA QUEUED, READ DMA QUEUED EXT, WRITE DMA QUEUED, WRITE DMA QUEUED EXT, or WRITE DMA QUEUED FUA EXT (see ATA8-ACS); or READ FPDMA QUEUED or WRITE FPDMA QUEUED command (see SATA 2.5).

**3.1.20 ATA read command:** One of the following ATA commands: READ DMA, READ DMA EXT, READ DMA QUEUED, READ DMA QUEUED EXT, READ MULTIPLE, READ MULTIPLE EXT, READ SECTOR(S), or READ SECTOR(S) EXT (see ATA8-ACS); or READ FPDMA QUEUED (see SATA 2.5).

**3.1.21 ATA Sector Count:** A count of ATA logical sectors to transfer or process, represented by the Count field in an ATA command (see ATA8-ACS).

**3.1.22 ATA software reset:** A reset that is triggered by a task management function request (see ATA8-AAM, see 5.5).

**3.1.23 ATA verify command:** One of the following ATA commands: READ VERIFY SECTOR(S) or ATA READ VERIFY SECTOR(S) EXT (see ATA8-ACS).

**3.1.24 ATA volatile settings:** ATA device settings affecting the way an ATA device responds to ATA commands that are configurable using ATA commands (e.g., ATA SET FEATURES or ATA SET MAX EXT), and that are set by the SATL to correspond to SCSI mode parameters, log parameters, or INQUIRY data.

**3.1.25 ATA write command:** One of the following ATA commands: WRITE DMA, WRITE DMA EXT, WRITE DMA FUA EXT, WRITE DMA QUEUED, WRITE DMA QUEUED EXT, WRITE DMA QUEUED FUA EXT, WRITE MULTIPLE, WRITE MULTIPLE EXT, WRITE MULTIPLE FUA EXT, WRITE SECTOR(S), or WRITE SECTOR(S) EXT (see ATA8-ACS); or WRITE FPDMA QUEUED (see SATA 2.5).

**3.1.26 ATAPI device:** A device that is compliant with the ATA standards and implements the PACKET command feature set (see ATA8-ACS).

**3.1.27 auto-contingent allegiance (ACA):** The task set condition established following the return of a CHECK CONDITION status when the NACA bit is set to one in the CONTROL byte (see SAM-3).

**3.1.28 autosense:** Sense data that is returned in the same I\_T\_L\_Q nexus transaction as the CHECK CONDITION status (see SAM-3). The alternative to autosense (i.e., use of a REQUEST SENSE command) is defined in SAM-2.

NOTE 3 - Autosense is defined in SAM-2. SAM-3 specifies what SAM-2 defines as autosense as the only valid way of returning SENSE data, but does not refer to it as autosense.

**3.1.29 big-endian:** A format for storage or transmission of binary data in which the most significant byte appears first. In a multi-byte value, the byte containing the most significant bit is stored in the lowest memory address and transmitted first and the byte containing the least significant bit is stored in the highest memory

address and transmitted last (e.g., for the value 0080h, the byte containing 00h is stored in the lowest memory address and the byte containing 80h is stored in the highest memory address).

**3.1.30 byte:** A sequence of eight contiguous bits considered as a unit.

**3.1.31 command descriptor block (CDB):** A structure used to communicate a command from a SCSI application client to a SCSI device server.

**3.1.32 device server:** An object within the logical unit that processes SCSI tasks according to the rules for task management (see SAM-3).

**3.1.33 direct logical block mapping:** A SATL implementation that maps logical blocks on a logical unit one-for-one with ATA logical sectors on an ATA device, where the LBA of a logical block has the same value as the LBA of the corresponding ATA logical sector and the number of bytes in a logical block equals the number of bytes in an ATA logical sector (see 9.1.2).

**3.1.34 domain:** A SCSI domain (see SAM-3) or an ATA domain (see ATA8-AAM).

**3.1.35 DRQ data block:** A unit of data words associated with available status when using either the PIO data-in command protocol or the PIO data-out command protocol (see ATA8-ACS)

**3.1.36 field:** A group of one or more contiguous bits.

**3.1.37 first-party direct memory access (FPDMA):** A method by which a device accesses host memory (see SATA 2.5).

**3.1.38 indirect logical block mapping:** A SATL implementation that does not follow the constraints of direct logical block mapping (see 3.1.33 and 9.1.3).

**3.1.39 I\_T nexus:** A nexus between a SCSI initiator port and a SCSI target port (see SAM-3).

**3.1.40 I\_T\_L nexus:** A nexus between a SCSI initiator port, a SCSI target port, and a logical unit (see SAM-3).

**3.1.41 I\_T\_L\_Q nexus:** A nexus between a SCSI initiator port, a SCSI target port, a logical unit, and a task (see SAM-3).

**3.1.42 least significant bit (LSB):** In a binary code, the bit or bit position with the smallest numerical weighting in a group of bits that, when taken as a whole, represent a numerical value (e.g., in the number 0001b, the bit that is set to one).

**3.1.43 link reset:** Performing the link reset sequence.

**3.1.44 link reset sequence:** A phy reset sequence (see SATA 2.5).

**3.1.45 little-endian:** A format for storage or transmission of binary data in which the least significant byte appears first. In a multi-byte value, the byte containing the least significant bit is stored in the lowest memory address and transmitted first and the byte containing the most significant bit is stored in the highest memory address and transmitted last (e.g., for the value 0080h, the byte containing 80h is stored in the lowest memory address and the byte containing 00h is stored in the highest memory address).

**3.1.46 logical block:** A set of user data words accessed and referenced as a unit.

**3.1.47 logical block address (LBA):** The value used to reference a logical block.

**3.1.48 logical unit:** An externally addressable entity within a SCSI target device. See SAM-3 for a detailed definition of a logical unit.

**3.1.49 logical unit capacity:** The capacity of a logical unit in bytes calculated as length in bytes of each logical block times one more than the LBA of the last logical block on the logical unit.

**3.1.50 logical unit number (LUN):** An identifier for a logical unit.

**3.1.51 logical unit reset event:** An event that triggers a logical unit reset (see SAM-3).

**3.1.52 logical unit reset:** A condition resulting from a hard reset condition or a logical unit reset event in which the logical unit performs the logical unit reset operations described in SAM-3, SPC-3, and this standard.

**3.1.53 medium:** The material on which data is stored (e.g., a magnetic disk).

**3.1.54 most significant bit (MSB):** In a binary code, the bit or bit position with the largest numerical weighting in a group of bits that, when taken as a whole, represent a numerical value (e.g., in the number 1000b, the bit that is set to one).

**3.1.55 native command queuing (NCQ):** A method by which a SATA device that does not implement the PACKET Command feature set may maintain and order the processing of up to 32 outstanding commands (see SATA 2.5).

**3.1.56 nexus:** A relationship between a SCSI initiator port and a SCSI target port that may extend to a logical unit and a task (see SAM-3).

**3.1.57 non-queued command:** An ATA non-queued command (see 3.1.18).

**3.1.58 object:** An architectural abstraction or container that encapsulates data types, services, or other objects that are related in some way.

**3.1.59 Parallel ATA (PATA):** A parallel transport protocol (see ATA8-APT).

**3.1.60 PATA bus:** All of the conductors and connectors required to attain signal line continuity between every driver, receiver, and terminator for each signal between one PATA host and one or two PATA devices (see ATA8-APT).

**3.1.61 PATA device:** An ATA device or ATAPI device that uses the PATA transport protocol (see ATA8-APT).

**3.1.62 PATA host:** An ATA host that uses the PATA transport protocol (see ATA8-APT).

**3.1.63 power on:** Power being applied.

**3.1.64 queue:** The arrangement of tasks within a task set usually according to the temporal order that they were created.

**3.1.65 queued command:** An ATA queued command (see 3.1.19), or a SCSI command received by the SATL from an application client for an emulated logical unit while the emulated logical unit is processing another SCSI command (see SAM-3).

**3.1.66 reset event:** A transport protocol specific event that results in a hard reset condition (see SAM-3) or a hardware reset (see ATA8-AAM).

**3.1.67 SAS address:** A worldwide unique name or identifier assigned to a SAS initiator port, SAS target port, expander device, SAS initiator device, or SAS target device (see SAS-1.1).

**3.1.68 SAS initiator device:** A device containing SSP, STP, and/or SMP initiator ports in a SAS domain (see SAS-1.1).

**3.1.69 SAS initiator port:** An SSP initiator port, STP initiator port, and/or SMP initiator port in a SAS domain (see SAS-1.1).

**3.1.70 SATA device:** An ATA device or ATAPI device that uses the Serial ATA transport protocol (see SATA 2.5).

**3.1.71 SATA host:** An ATA host that implements the Serial ATA transport protocol (see SATA 2.5).

**3.1.72 SCSI / ATA Translation (SAT):** A standard defining the translation of SCSI commands and responses to ATA commands and responses (i.e., this standard).

**3.1.73 SCSI / ATA Translation layer (SATL):** The functional layer defined in this standard that uses an ATA device to emulate objects in a SCSI logical unit, including the device server, task manager, and task set (see SAM-3).

**3.1.74 SCSI device:** A device that contains one or more SCSI ports that are connected to a service delivery subsystem and supports a SCSI application protocol.

**3.1.75 SCSI hard reset:** A condition resulting from a power on condition or a reset event in which the SCSI device performs the hard reset operations described in SAM-3, SPC-3, and the appropriate command and transport standards.

**3.1.76 SCSI initiator port:** A SCSI initiator device object that acts as the connection between application clients and a service delivery subsystem through which requests and responses are routed (see SAM-3).

**3.1.77 SCSI read command:** A READ (6), READ (10), READ (12), or READ (16) command (see SBC-2).

**3.1.78 SCSI synchronize cache command:** A SYNCHRONIZE CACHE(10), or SYNCHRONIZE CACHE (16) command (see SBC-2).

**3.1.79 SCSI target port:** A SCSI target device object that contains a task router and acts as the connection between device servers and task managers and a service delivery subsystem through which requests and responses are routed (see SAM-3).

**3.1.80 SCSI verify command:** A VERIFY (10), VERIFY (12), or VERIFY (16) command (see SBC-2).

**3.1.81 SCSI write command:** A WRITE (6), WRITE (10), WRITE (12), or WRITE (16) command (see SBC-2).

**3.1.82 SCSI write and verify command:** A WRITE AND VERIFY (10), WRITE AND VERIFY (12), or WRITE AND VERIFY (16) command (see SBC-2).

**3.1.83 Serial ATA (SATA):** A serial transport protocol that serves as an ATA service delivery subsystem (see SATA 2.5).

**3.1.84 Serial ATA Tunneled Protocol (STP):** The protocol used by STP initiator ports to communicate with STP target ports in a SAS domain (see SAS 1.1)

**3.1.85 Serial Attached SCSI (SAS):** A set of protocols and the interconnect defined by SAS-1.1.

**3.1.86 service delivery subsystem:** That part of a SCSI I/O system that transmits service requests to a logical unit or SCSI target device and returns logical unit or SCSI target device responses to a SCSI initiator device (see SAM-3), or that part of an ATA I/O system that connects an ATA host port and one or more ATA/ATAPI device ports and is a single path for the transfer of requests and responses between a host and one or more devices (see ATA8-AAM).

**3.1.87 service response:** The device service response or SCSI transport protocol specific service response returned to an application client by the SATL on completion of a SCSI transport protocol service request (see SAM-3).

**3.1.88 STP initiator port:** A SAS initiator device object in a SAS domain that interfaces to a service delivery subsystem with STP (see SAS-1.1).

**3.1.89 STP target port:** A SAS target device object in a SAS domain that interfaces to a service delivery subsystem with STP (see SAS-1.1).

**3.1.90 STP/SATA bridge:** An expander device object containing an STP target port, a SATA host port, and the functions required to forward information between the STP target port and SATA host port to enable STP initiator ports in a SAS domain to communicate with SATA devices in an ATA domain (see SAS-1.1).

**3.1.91 task:** An object within the logical unit representing the work associated with a command (see SAM-3).

**3.1.92 task management function:** A task manager service capable of being requested by an application client to affect the processing of one or more tasks (see SAM-3).

**3.1.93 task set:** A group of tasks within a device server whose interaction is dependent on the task management and auto-contingent allegiance rules (see SAM-3).

**3.1.94 tagged command queuing (TCQ):** A method that makes use of the ATA Tagged Command Queuing feature set, by which an ATA device may maintain and order the processing of up to 32 outstanding commands, identifying the context of each outstanding command with a unique tag (see ATA8-ACS).

**3.1.95 Transport Protocol-Specific Information Unit (TPSIU):** A transport-specific information unit used to transport information between initiator ports and target ports that may contain additional information needed by a service delivery subsystem to effect the requested information unit transfers (e.g., the Command Block Wrapper defined in USB-BOT).

**3.1.96 word:** A sequence of two contiguous bytes considered as a unit.

## 3.2 Symbols and abbreviations

≠ or NE	not equal
≤ or LE	less than or equal to
±	plus or minus
≈	approximately
x	multiply
+	add
-	subtract
< or LT	less than
= or EQ	equal
> or GT	greater than
≥ or GE	greater than or equal to
ACA	auto-contingent allegiance (see 3.1.27)
ATA	AT Attachment (see 3.1.4)
ATAPI	AT Attachment Packet Interface (see 3.1.4)
CDB	Command Descriptor Block (see 3.1.31)
FIS	Frame Information Structure (see SATA 2.5)
FPDMA	First-party direct memory access (see 3.1.37)
FUA	Force Unit Access
LBA	Logical Block Address (see 3.1.47)
LSB	Least significant bit (see 3.1.42)
LUN	Logical unit number (see 3.1.50)
MSB	Most significant bit (see 3.1.54)

n/a	not applicable
NCQ	Native command queuing (see 3.1.55)
PATA	Parallel ATA (see 3.1.59)
SAS	Serial Attached SCSI (see 3.1.85)
SAT	SCSI / ATA Translation (see 3.1.72)
SATA	Serial ATA (see 3.1.83)
SATA 2.5	Serial ATA 2.5 specification (see 2.4)
SATL	SCSI / ATA Translation Layer (see 3.1.73)
SAM-2	SCSI Architecture Model-2 standard (see 2.2)
SAM-3	SCSI Architecture Model-3 standard (see 2.2)
SCSI	Small Computer System Interface family of standards
SCT	Smart Command Transport standard (see 2.3)
SPC-3	SCSI Primary Commands-3 standard (see 2.2)
STP	Serial ATA Tunneled Protocol (see 3.1.84)
SW	software
TCQ	Tagged command queuing (see 3.1.94)
TPSIU	Transport Protocol-Specific Information Unit (see 3.1.95)
VPD	Vital Product Data (see SPC-3)

### 3.3 Keywords

**3.3.1 expected:** A keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented.

**3.3.2 invalid:** A keyword used to describe an illegal or unsupported bit, byte, word, field or code value. Receipt of an invalid bit, byte, word, field or code value shall be reported as an error.

**3.3.3 mandatory:** A keyword indicating an item that is required to be implemented as defined in this standard.

**3.3.4 may:** A keyword that indicates flexibility of choice with no implied preference.

**3.3.5 may not:** Keywords that indicates flexibility of choice with no implied preference.

**3.3.6 obsolete:** A keyword indicating that an item was defined in a previous version of a standard but has been removed from the most recent version of that standard.

**3.3.7 optional:** A keyword that describes features that are not required to be implemented by this standard. However, if any optional feature defined by this standards is implemented, it shall be implemented as defined in this standard.

**3.3.8 reserved:** A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients are not required to check reserved bits, bytes, words or fields for zero values. Receipt of reserved code values in defined fields shall be reported as an error.

**3.3.9 shall:** A keyword indicating a mandatory requirement (equivalent to “is required”). Designers are required to implement all such requirements to ensure interoperability with other products that conform to this standard.

**3.3.10 should:** A keyword indicating flexibility of choice with a preferred alternative (equivalent to “it is strongly recommended”).

**3.3.11 vendor specific:** A keyword indicating specification of the referenced item is determined by the SCSI device vendor.

### 3.4 SAT specific terminology

**3.4.1 emulated:** A term designating that the SATL is required to implement functions in addition to or in place of functions supported by an ATA device to provide a defined SCSI capability.

**3.4.2 unspecified:** A term designating that this version of this standard does not specify a translation for a SCSI field. A translation for an unspecified field may be specified by future versions of this standard. Translation of fields marked unspecified shall not conflict with other standards in the set of SCSI standards.

### 3.5 Conventions

#### 3.5.1 Overview

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in 3.1 or in the text where they first appear. Names of commands, statuses, sense keys, and additional sense codes are in all uppercase (e.g., REQUEST SENSE). Lowercase is used for words having the normal English meaning.

If there is more than one CDB length for a particular command (e.g., MODE SENSE (6) and MODE SENSE (10)) and the name of the command is used in a sentence without any CDB length descriptor (e.g., MODE SENSE), then the condition specified in the sentence applies to all CDB lengths for that command.

The names of fields are in small uppercase (e.g., ALLOCATION LENGTH). When a field name is a concatenation of acronyms, uppercase letter may be used for readability (e.g., NORMACA). Normal case is used when the contents of a field are being discussed. Fields containing only one bit are usually referred to as the NAME bit instead of the NAME field.

If a conflict arises between text, tables, or figures, the order of precedence to resolve the conflicts is text; then tables; and finally figures. Not all tables or figures are fully described in the text. Tables show data format and values. Notes do not constitute any requirements for implementors.

#### 3.5.2 Numeric conventions

- | A binary number is represented in this standard by any sequence of digits **consisting** ~~comprised~~ of only the Western-Arab numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included in binary number representations to increase readability or delineate field boundaries (e.g., 0 0101 1010b or 0\_0101\_1010b).
- | A hexadecimal number is represented in this standard by any sequence of digits **consisting** ~~comprised~~ of only the Western-Arab numerals 0 through 9 and/or the upper-case English letters A through F immediately followed by a lower-case h (e.g., FA23h). Underscores or spaces may be included in hexadecimal number representations to increase readability or delineate field boundaries (e.g., B FD8C FA23h or B\_FD8C\_FA23h).
- | A decimal number is represented in this standard by any sequence of digits **consisting** ~~comprised~~ of only the Western-Arab numerals 0 through 9 not immediately followed by a lower-case b or lower-case h (e.g., 25).

When the value of the bit or field is not relevant, x or xx appears in place of a specific value.

This standard uses the ISO convention for representing decimal numbers (e.g., the thousands and higher multiples are separated by a space and a comma is used as the decimal point). Table 1 shows some examples of decimal numbers represented using the ISO and American conventions.

**Table 1 — ISO and American numbering conventions examples**

ISO	American
0,6	0.6
3,141 592 65	3.14159265
1 000	1,000
1 323 462,95	1,323,462.95

Lists sequenced by letters (e.g., a-red, b-blue, c-green) show no ordering relationship between the listed items. Numbered lists (e.g., 1-red, 2-blue, 3-green) show a ordering relationship between the listed items.

### 3.5.3 Bit and byte ordering

In this standard, data structures may be defined by a table. A table defines a complete ordering of elements (i.e., bits, bytes, fields, and dwords) within the structure. The ordering of elements within a table does not in itself constrain the order of storage or transmission of the data structure, but in combination with other normative text in this standard, may constrain the order of storage or transmission of the structure.

In a table, any element that is presented in a row above another element in a lower row is more significant than the lower element, and any element presented to the left of another element in the same row is more significant than the element to the right.

If a table shows bit numbering (see table 2), the least significant bit (LSB) is numbered 0 and each more significant bit has the next greater number than the immediately less significant bit. If a table shows numbering of bytes or characters (see table 3), the most significant byte or character is represented at the lowest number and each less significant byte or character has the next greater number than the immediately more significant byte.

In a field in a table consisting of more than one bit that contains a single value (e.g., a number), the least significant bit (LSB) is shown on the right and the most significant bit (MSB) is shown on the left (e.g., in a byte, bit 7 is the MSB and is shown on the left, bit 0 is the LSB and is shown on the right). The MSB and LSB are not labeled if the field consists of eight or fewer bits. The MSB and LSB are labeled if the field consists of more than eight bits and has no internal structure defined.

In a field in a table consisting of more than one byte that contains multiple fields each with their own values (e.g., a descriptor), there is no MSB and LSB of the field itself and thus there are no MSB and LSB labels. Each individual field has an MSB and LSB, but they are not labeled.

In a field containing a text string (e.g., ASCII or UTF-8), only the MSB of the first character and the LSB of the last character are labeled.

Multiple byte fields are represented with only two rows, with the non-sequentially increasing byte number denoting the presence of additional bytes.

A data dword consists of 32 bits. Table 2 shows a data dword containing a single value, where the MSB is on the upper left in bit 31 and the LSB is on the lower right in bit 0.

**Table 2 — Example of ordering of bits and bytes within a multi-byte element**

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB) Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)

Note 1 - The Bit x labels in the individual table cells are for reference only and should not appear within tables the use this element format.

Table 3 shows a data dword containing four one-byte fields, where byte 0 (the first byte) is on the left and byte 3 (the fourth byte) is on the right. Each byte has an MSB on the left and an LSB on the right.

**Table 3 — Example of ordering of bits and bytes within a multiple element**

Bit Byte	7	6	5	4	3	2	1	0
0	First byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
1	Second byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
2	Third byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
3	Fourth byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)

Note 1 - The Bit x labels in the individual table cells and the xx byte labels in the individual bytes are for reference only and should not appear within tables the use these element formats. In this example the MSB and LSB labels are for reference only, however, they may appear in mult-byte fields as described in this subclause.

**3.5.4 Notation for byte encoded character strings**

When this standard requires one or more bytes to contain specific encoded character, the specific characters are enclosed in single quotation marks. The single quotation marks identify the start and end of the characters that are required to be encoded but are not themselves to be encoded. The characters that are to be encoded are shown in exactly the case that is to be encoded.

An ASCII space character (i.e., 20h) may be represented in a string by the character '↵' (e.g., 'SCSI↵device').

The encoded characters and the single quotation marks that enclose them are preceded by text that specifies the character encoding methodology and the number of characters required to be encoded.

The encoded characters and the single quotation marks that enclose them are preceded by text that specifies the character encoding methodology and the number of characters required to be encoded.

EXAMPLE - Using the notation described in this subclause, stating that eleven ASCII characters 'SCSI device' are to be encoded would be the same writing out the following sequence of byte values: 53h 43h 53h 49h 20h 64h 65h 76h 69h 63h 65h.

### 3.5.5 Use of and/or

The text 'x and/or y' means the same as:

- a) x;
- b) y; or
- c) x and y;

where x and y are defined terms, descriptive elements, steps in a procedure, or other phrases used to define requirements in this standard.

### 3.5.6 Notation for command descriptions

#### 3.5.6.1 Description

The description of each command begins with a subclause describing the general method applied in translating the SCSI command to the corresponding ATA command(s), as well as any constraints and special considerations that may apply to the translation applied.

The subclause describing the general translation method for each command contains a table formatted like table 4 with two columns as follows:

- a) the first column lists each of the fields in the SCSI CDB (see SPC-3 and SBC-2); and
- b) the second column is either a brief description of the corresponding ATA protocol element(s) used to implement the identified SCSI field, or a reference to a subsequent subclause containing a more lengthy description of the method of emulation or implementation.

**Table 4 — Format for translated command field descriptions**

Field	Description or reference
IMPLEMENTED OR EMULATED FIELD	A brief identification of the corresponding ATA protocol element, or a paragraph reference if there are special considerations that need to be applied in the use of the corresponding ATA protocol element that require a separate paragraph of description.
SUMMARY EMULATED FIELD	Summary field with more detailed structure.
UNSPECIFIED FIELD	Unspecified (see 3.4.2)

Tables listing fields in mode pages have an additional column that defines whether the field is changeable or not.

### 3.5.7 Use of field names defined in ATA standards and specifications

This standard discusses fields and values defined in other standards and specifications, in particular the ATA8-APT, ATA8-ACS, ATA8-AAM, and SCT standards developed by T13, and the SATA 2.5 specification. Such fields and values discussed in this standard are shown using the same notation conventions used in the standards where those fields and values are defined.

When this standard uses terms defined in T13 ATA standards or the SATA-2.5 specification, the following conventions apply:

- a) The names of abbreviations, commands, and acronyms used as signal names are in all uppercase (e.g., IDENTIFY DEVICE). Fields containing only one bit are usually referred to as the “name” bit instead of the “name” field.
- b) Names of device registers, fields in data structures, and other defined terms begin with an upper-case letter (e.g., LBA Mid register) and may be represented in mixed-case (e.g. PhyRdy).
- c) The expression “word n” or “bit n” shall be interpreted as indicating the content of word n or bit n.
- d) Bit names are shown in all uppercase letters.
- e) Bit (n:m) denotes a set of bits, for example, bits (7:0).

## 4 General

This standard defines a translation layer (i.e., the SATL) that provides a method for a SCSI application layer (see SAM-3) to access Serial ATA or Parallel ATA devices by representing ATA devices as SCSI direct-access block devices.

Implementations of SCSI / ATA Translation may provide varying levels of SCSI functionality.

EXAMPLE 1 - The SATL may provide a level of SCSI emulation that is indistinguishable from native SCSI devices in terms of reported capabilities. Such SATL implementations need little guidance from this standard to effect interoperability since other SCSI protocol standards define all that is required to establish interoperability.

EXAMPLE 2 - A SCSI / ATA Translation implementation may implement a subset of SCSI, have limited or no capability to maintain persistent information about the characteristics or state of the emulated SCSI device, have limited capability to manage device state information that carries forward from one command to the next, and maintain little or no capability to coordinate between multiple commands outstanding at a time. The characteristics and behavior of the underlying ATA devices in these minimal implementations of the SATL are expected to be more visible to the SCSI application clients.

This standard provides a set of definitions, conventions, and guidelines for:

- a) the consistent reporting by the SATL of capabilities of emulated SCSI devices; and
- b) the consistent identification of the attached devices by the application clients.

These provisions allow application clients to observe consistent behavior whether or not the application clients recognize the presence of a SATL in a system.

By defining expected behavior in terms of the SCSI commands issued, corresponding activity in the ATA domain, and expected SCSI responses based on the results of activity in the ATA domain, this standard eliminates:

- a) incompatibility between legacy SCSI / ATA Translation implementations; and
- b) SCSI application client /device interdependence.

This standard refers to behaviors for SCSI devices defined in SBC-2 and SPC-3. Unless otherwise specified, any behaviors that are optional in SBC-2 or SPC-3 are optional for devices implementing SCSI / ATA Translation. Any optional behaviors referred to in this standard and implemented by the SATL shall be implemented as described in this standard.

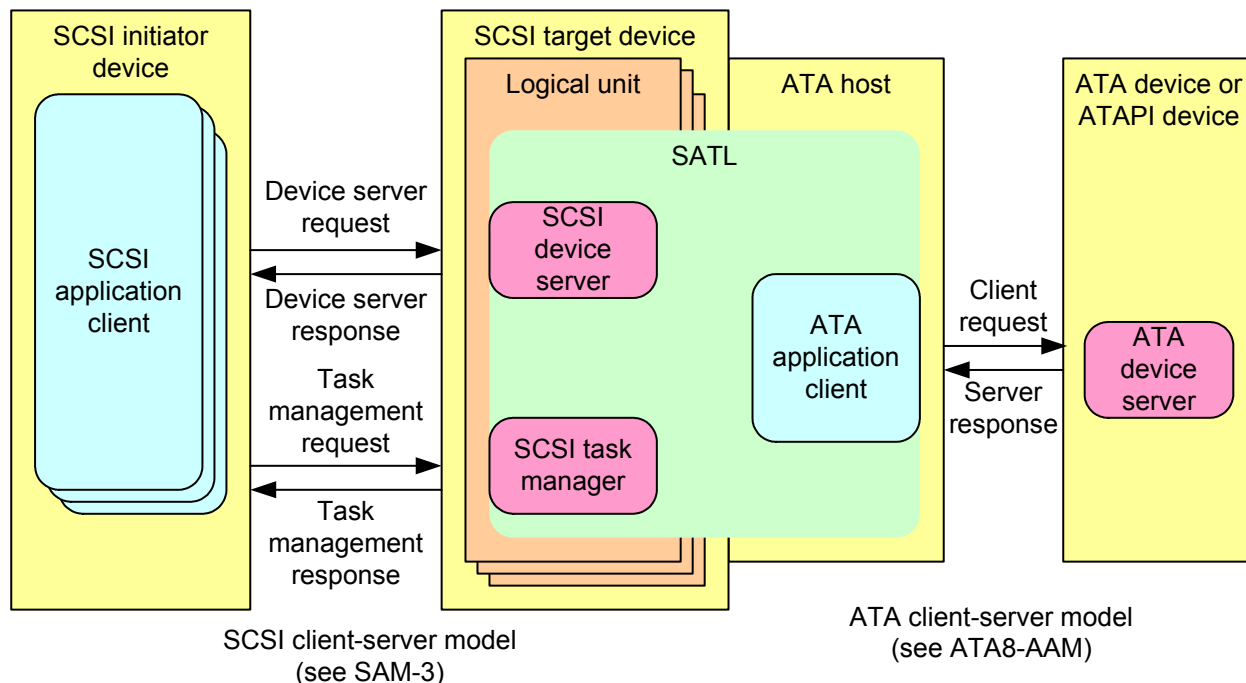
If the SATL receives a SCSI request specifying any value in any field of the CDB that the SATL does not support, unless otherwise specified in the description of the command, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB (see SPC-3).

If the SATL receives a SCSI request specifying any value in any field of the parameter data that the SATL does not support, unless otherwise specified in the description of the parameter, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST (see SPC-3).

## 5 SCSI architectural elements

### 5.1 Overview

Clause 5 defines SCSI / ATA translation elements that impact the representation of the storage domains defined in SAM-3 and ATA8-AAM. Figure 4 shows a SATL providing a communication path between a SCSI application client and an ATA device.



**Figure 4 — Example of a SATL between a SCSI application client and an ATA or ATAPI device**

The SATL provides the communication path between a SCSI application client and an ATA device or ATAPI device by:

- emulating a SCSI logical unit;
- integrating an ATA host; and
- providing the translation elements to link them together.

This standard defines SCSI / ATA translation using SCSI and ATA command sets. This standard does not define the mapping of transport capabilities as defined at the SCSI transport protocol layer and the ATA protocol interconnect layer.

An implementation utilizing a SATL may include a SCSI transport. A SATL may appear in different configurations:

**EXAMPLE 1** - Figure 5 shows a SATL contained within a SCSI to ATA protocol bridge, where the ATA device is being accessed by an ATA host port, and the SATL is being accessed with a SCSI target port using a SCSI transport protocol (e.g, FCP-3 over Fibre Channel).

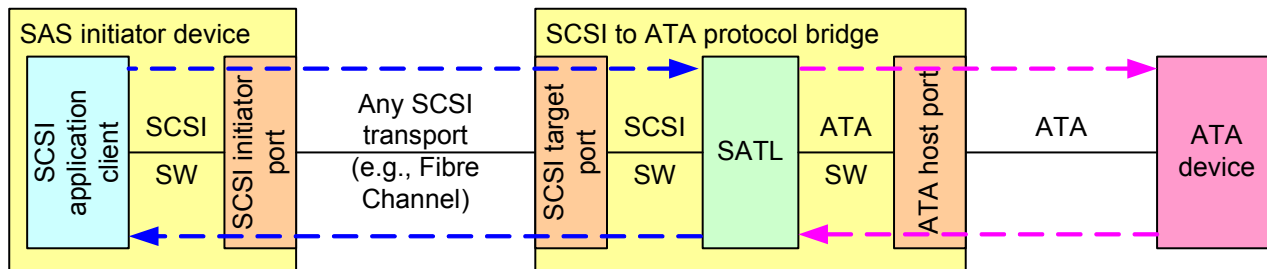


Figure 5 — SATL contained within a SCSI to ATA protocol bridge

EXAMPLE 2 - Figure 6 shows an ATA Host Bus Adapter (HBA) directly connected to an ATA device. The SATL provides SCSI transport protocol layer services to a SCSI application client in accordance with SAM-3.

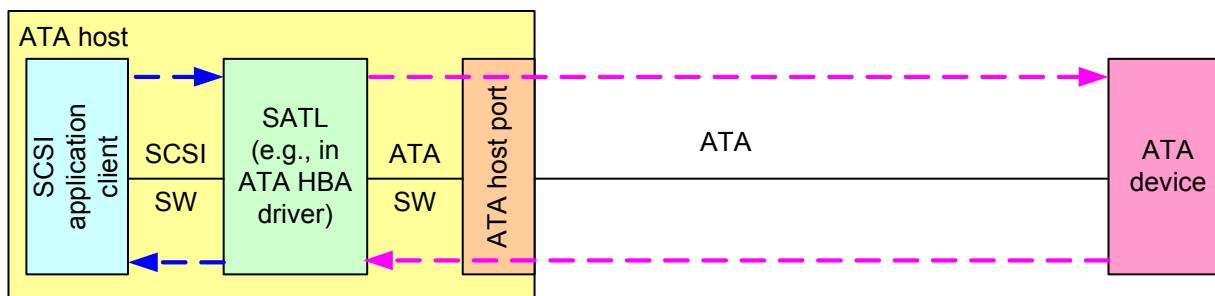


Figure 6 — SATL contained within an ATA host

EXAMPLE 3 - Figure 7 shows an ATA device accessed by a SAS STP initiator port (see SAS-1.1) through a SAS interconnect. The SAS initiator device includes a SATL to provide the SCSI transport protocol layer services to the application client in accordance with SAM-3.

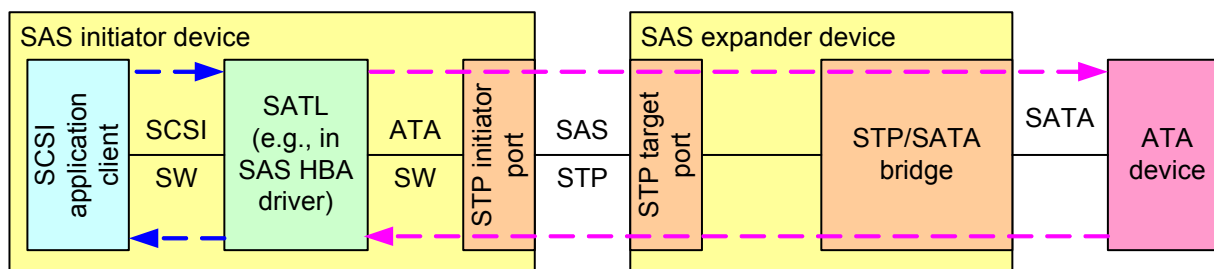


Figure 7 — SATL contained in a SAS initiator device

## 5.2 Unit attention condition

The SATL shall report events affecting the state of the emulated SCSI device to the SCSI application clients by emulating unit attention conditions (see SAM-3).

A SATL that detects a link reset for a Serial ATA device or initiates any reset of an ATA device shall establish a unit attention condition on behalf of the logical unit corresponding to the ATA device with the sense key set to UNIT ATTENTION and the additional sense code set to POWER ON, RESET, or BUS DEVICE RESET

OCCURRED for the SCSI initiator port associated with each I\_T nexus. The method a SATL uses to detect a link reset on the Serial ATA link is vendor specific.

The SATL shall report unit attention conditions, in accordance with SAM-3, regardless of whether the condition results from accessing an ATA device or a condition internal to the SATL.

### 5.3 Handling errors in ATA commands

When a SCSI command is translated into one or more ATA commands and one of the ATA commands completes with an error, the SATL shall terminate processing of the SCSI command and report the error as described in clause 11.

When interpreting data from an ATA command, the SATL shall use the data only if no error was reported for the command. In addition:

- a) when interpreting IDENTIFY DEVICE data, if the Integrity word contains the Signature value defined in ATA8-ACS (i.e., word 255 bits 7:0), then SATL shall use the data only if the Checksum is correct;
- b) when interpreting SMART READ DATA data for the Summary SMART error log (i.e., log address 01h), the Comprehensive SMART error log (i.e., log address 02h), the SMART self-test log (i.e., log address 06h), or the Selective self-test log (i.e., log address 09h) (see ATA8-ACS), the SATL shall use the data only if the data structure checksum (i.e., byte 511) is correct; and
- c) when interpreting READ LOG EXT data for the Extended Comprehensive SMART error log (i.e., log address 03h) or Extended SMART self-test log (i.e., log address 07h) (see ATA8-ACS), the SATL shall use the data only if the data structure checksum (i.e., byte 511) is correct.

### 5.4 ATA nexus loss

An ATA nexus loss event occurs when the SATL loses communication with the ATA device. If an ATA nexus loss event occurs:

- a) the SATL shall terminate all commands being processed for the corresponding logical unit; and
- b) the SATL shall establish a unit attention condition for each I\_T nexus with the additional sense code set to:
  - A) if the SATL is able to determine that the ATA device is no longer physically present, REPORTED LUNS DATA HAS CHANGED or DRIVE NOT PRESENT;
  - B) if the SATL is unable to determine if the ATA device is physically present or not, INQUIRY DATA HAS CHANGED; or
  - C) if the SATL is able to determine that the ATA device is present, INTERNAL TARGET FAILURE.

NOTE 4 - The method by which the SATL determines physical presence or absence of the ATA device is outside the scope of this standard (e.g., using cold presence detect, (see SATA 2.5), or a change in the ELEMENT STATUS CODE field in the Device or Array Device element (see SES-2).

NOTE 5 - SAM-3 and SPC-3 define how the SATL processes subsequent commands when the logical unit is no longer available (i.e., incorrect logical unit selection).

If the ATA nexus is restored or the SATL detects a power-on condition for an ATA device, the SATL shall perform the processing described in 5.5 for those events.

### 5.5 ATA hardware and software reset processing

An ATA hardware reset may be caused either by the SATL or by the ATA device. If an ATA hardware reset or an ATA software reset occurs except as part of processing a SCSI task management function (see 6.3), then the SATL shall:

- a) terminate processing of all tasks for each logical unit affected by the reset;
- b) restore the ATA volatile settings (see 3.1.24) of the ATA device (e.g., by sending an ATA SET FEATURES command) to values consistent with the saved values of mode pages if savable mode pages are supported and available, or default values if savable mode pages are not supported or are not available; and

- c) establish a unit attention condition for each I\_T\_L nexus with the additional sense code set to POWER ON, RESET, OR BUS DEVICE RESET OCCURRED.

## **6 Task management model**

### **6.1 Overview**

A SATL may support the full task management model or the basic task management model as well as specific features of the task management model (e.g. SIMPLE and ORDERED task attributes) depending on the task management and queueing capabilities of the SATL and whether the SATL supports SATA native command queueing (NCQ) or the ATA Tagged Command Queuing (TCQ) feature set.

## 6.2 Queued commands

### 6.2.1 Comparison of SCSI and ATA queuing

Some differences between SCSI and ATA queuing methods are shown in table 5.

**Table 5 — Comparison of SCSI and ATA queuing methods**

Feature	SCSI	NCQ	TCQ
Ordering	Specified by task attributes (e.g. SIMPLE, ORDERED) associated with each command.	Always at the discretion of the device.	Always at the discretion of the device.
Queue depth	Indeterminate	Fixed at 1 to 32 commands as reported in the ATA IDENTIFY DEVICE data word 75.	Fixed at 1 to 32 commands as reported in the ATA IDENTIFY DEVICE data word 75.
Queue full reporting	TASK SET FULL status	n/a	n/a
Queue full management	Device manages and indicates via TASK SET FULL status.	ATA host managed.	ATA host managed.
Queued commands	Queuing is applicable to all commands.	Limited to READ FPDMA QUEUED and WRITE FPDMA QUEUED commands.	Limited to READ DMA QUEUED, READ DMA QUEUED EXT, WRITE DMA QUEUED, WRITE DMA QUEUED EXT and WRITE DMA QUEUED FUA EXT commands, or a NOP command with a non-zero subcommand code.
Handling of non-queued commands received while one or more queued commands are being processed	n/a; all commands are queued commands.	Receipt of any command other than a READ FPDMA QUEUED command or a WRITE FPDMA QUEUED command is an error.	Receipt of any command other than a NOP command with a non-zero subcommand code, a SERVICE command, a READ DMA QUEUED command, a READ DMA QUEUED EXT command, a WRITE DMA QUEUED command, a WRITE DMA QUEUED EXT command, or a WRITE DMA QUEUED FUA EXT command is an error.
Error handling	Controlled with mode parameters.	Any error aborts all queued commands.	Any error aborts all queued commands.

### 6.2.2 Mapping of SCSI queued commands to ATA queued commands

A SATL that translates SCSI tagged commands to an ATA device using NCQ or TCQ, whether or not the SATL also queues commands internally, shall either:

- a) indicate support for the basic task management model in standard INQUIRY data (i.e., the BQUE bit is set to one and CMDQUE bit is set to zero), and follow the rules for the basic task management model (see SAM-3); or
- b) indicate support for the full task management model in standard INQUIRY data (i.e., the BQUE bit is set to zero and CMDQUE bit is set to one), and set the QERR (i.e., queue error management) field of the Control mode page (see 10.1.5) as follows:
  - A) a value of 01b if the SATL does not reissue ATA queued commands aborted by the ATA device due to an error condition on any one of the ATA queued commands; or
  - B) a value other than 01b if the SATL reissues all other ATA queued commands (i.e., except the one in error) aborted by the ATA device due to an error condition on any one of the ATA queued commands.

For each SCSI tagged command that the SATL translates to ATA queued commands (see 3.1.19), the SATL shall allocate an available tag value (e.g., for NCQ, the value corresponding to the position of a bit set to zero in the SActive register). The SATL shall maintain a mapping between allocated ATA queued command tags and the corresponding SCSI task tags.

The SATL shall use the maximum queue depth supported by the ATA device (i.e., indicated by IDENTIFY DEVICE data word 75), and may either:

- a) return a status of TASK SET FULL in response to a SCSI command issued to the corresponding emulated SCSI logical unit when the ATA device represented has the maximum number of ATA queued commands outstanding; or
- b) queue the command internally and return TASK SET FULL status when the SATL exhausts internal queueing resources.

### 6.2.3 Commands the SATL queues internally

If the translation of a SCSI command requires the SATL to send a non-queued command to the ATA device, then the SATL shall not send the non-queued command to the ATA device until all commands outstanding in the ATA device have returned command complete (i.e., with or without error).

If the ATA device corresponding to a logical unit has not returned command complete for all ATA commands the SATL has previously sent to the ATA device, and the SATL receives a SCSI command that requires the SATL to send a non-queued command to the ATA device, the SATL shall:

- a) suspend processing of the SCSI command, maintain the SCSI command in an internal queue, and resume processing when the ATA device returns command complete for all ATA commands the SATL has previously sent to the ATA device;
- b) return TASK SET FULL status for the SCSI command; or
- c) return BUSY status for the SCSI command.

The SATL shall perform task management in accordance with the task management model (see SAM-3) indicated in standard INQUIRY data and the Control mode page (see SPC-3).

### 6.2.4 Command queuing with multiple I\_T nexuses

In some configurations the SATL may receive SCSI requests from multiple I\_T nexuses. If the SATL receives SCSI requests from multiple I\_T nexuses (e.g., the configuration shown in figure 5), as specified in SAM-3, the task tags maintained in the SATL mapping of task tags to NCQ tags or TCQ tags shall be qualified by the I\_T nexus from which the command was received. When translating from an NCQ tag or TCQ tag to the corresponding SCSI task tag, the SATL shall determine the correct I\_T nexus using the qualification information associated with the SCSI task tag. The SATL may return TASK SET FULL even if the ATA device has available NCQ tags or TCQ tags in order to maintain tags available for other I\_T nexuses.

### 6.2.5 Collateral abort with queued commands

Error conditions with outstanding commands to an ATA device terminate all outstanding ATA commands being processed by the ATA device. An ATA host determines the status and error for each outstanding ATA queued command affected by the error condition and which ATA command(s) caused the error(s) (see ATA8-ACS or SATA 2.5). The SATL shall process aborted ATA commands as shown in table 6.

**Table 6 — SATL processing of ATA commands aborted by ATA collateral abort**

Association between the aborted ATA command and the ATA command that caused the error		Value of the QERR field set in the Control mode page	Method applied by the SATL for processing the aborted ATA command
I_T_L_Q nexus	I_T nexus		
same		00b	The SATL shall terminate the affected I_T_L_Q nexus with CHECK CONDITION status with the sense key and the additional sense code set according to the reported ATA error as described in clause 11.
		01b	
different	same	01b	The SATL shall terminate the affected I_T_L_Q nexus, but the SATL shall neither return status for the I_T_L_Q nexus affected by the aborted ATA command, nor retry the aborted ATA command.
		00b	The SATL shall reissue the ATA command and continue processing the corresponding I_T_L_Q nexus.
different		00b	The SATL shall terminate the affected I_T_L_Q nexus and establish a unit attention condition (see SAM-3) for the affected initiator port with the additional sense code set to COMMANDS CLEARED BY ANOTHER INITIATOR.
		01b	

## 6.3 Task management functions

### 6.3.1 Task management functions overview

6.3 describes the translation of SCSI task management functions to ATA equivalents.

### 6.3.2 Aborting ATA queued commands

Some task management functions processed by the SATL may result in ATA commands aborted by ATA collateral abort (see 3.1.7) affecting an I\_T\_L\_Q nexus other than the I\_T\_L\_Q nexus(es) specified in the task management function request. The subclause defining the translation for each task management function defines how the SATL processes the I\_T\_L\_Q nexuses affected by the task management function.

Processing some task management functions requires the SATL to abort one or more ATA commands being processed by an ATA device.

The SATL shall abort an ATA queued command being processed by an ATA device by sending an ATA CHECK POWER MODE command to the ATA device.

NOTE 6 - The ATA CHECK POWER MODE command is used to abort ATA queued commands because it is an ATA non-queued command that does not transfer data. The ATA CHECK POWER MODE command does not affect ATA volatile settings.

### 6.3.3 Aborting ATA non-queued commands

To abort an ATA non-queued command the SATL shall:

- a) issue an ATA software reset to the ATA device; and
- b) restore ATA volatile settings (see 3.1.24) to values consistent with current mode parameter settings.

### 6.3.4 ABORT TASK

The service request for the ABORT TASK task management function is (see SAM-3):

Service Response = ABORT TASK (IN (I\_T\_L\_Q nexus)).

If no ATA commands associated with the I\_T\_L\_Q nexus specified in the ABORT TASK task management function are outstanding to the ATA device, then the SATL shall abort the task for the specified I\_T\_L\_Q nexus from the SATL internal context and respond to the ABORT TASK task management function with a service response of FUNCTION COMPLETE (see SAM-3).

If the ATA device is processing one or more ATA commands that are related to the specified I\_T\_L\_Q nexus, then the SATL shall either:

- a) allow the ATA command(s) to complete as follows:
  - 1) wait until the ATA device returns command complete for the ATA command(s);
  - 2) if the completed ATA command completes processing of the specified I\_T\_L\_Q nexus, then return completion status for the I\_T\_L\_Q nexus; and
  - 3) return a service response of FUNCTION COMPLETE for the ABORT TASK task management function regardless of whether or not completion status was returned for the I\_T\_L\_Q nexus;
- or
- b) abort the ATA command(s) (see 6.3.2) for the specified I\_T\_L\_Q nexus and respond to the ABORT TASK task management function with a service response of FUNCTION COMPLETE.

If aborting the ATA commands related to the specified I\_T\_L\_Q nexus results in one or more other ATA commands being aborted by ATA collateral abort (see 3.1.7), then the SATL shall:

- a) if the SATL supports ATA abort retry (see 3.1.6), then re-issue all ATA commands aborted by ATA collateral abort (see 3.1.7) and continue processing the affected I\_T\_L\_Q nexuses; or
- b) if the SATL does not support ATA abort retry, then for each I\_T nexus affected by an ATA command aborted by ATA collateral abort:
  - 1) terminate all but one of the SCSI tasks without returning a function result; and
  - 2) complete processing of the remaining SCSI task by returning CHECK CONDITION status with the sense key set to UNIT ATTENTION and additional sense code set to COMMANDS CLEARED BY DEVICE SERVER.

### 6.3.5 ABORT TASK SET

The service request for the ABORT TASK SET task management function (see SAM-3) is:

Service Response = ABORT TASK SET (IN (I\_T\_L nexus)).

If the ATA device is not processing ATA commands for SCSI tasks associated with the specified I\_T\_L nexus, then the SATL shall abort all tasks for the specified I\_T\_L nexus from the SATL internal context and respond to the ABORT TASK SET task management function with a service response of FUNCTION COMPLETE.

If the ATA device is processing any ATA commands related to the specified I\_T\_L nexus, then the SATL shall either:

- a) allow the ATA command(s) to complete as follows:
  - 1) wait until the ATA device returns command complete for the ATA command(s);
  - 2) if the completed ATA command completes processing a SCSI task in the task set, return completion status for the SCSI task; and
  - 3) after all ATA commands return completion status, return a service response of FUNCTION COMPLETE for the ABORT TASK SET task management function;

or

- b) abort outstanding ATA command(s) (see 6.3.2) for the specified I\_T\_L nexus, and respond to the ABORT TASK SET task management function with a service response of FUNCTION COMPLETE.

If aborting ATA commands for the specified I\_T\_L nexus results in ATA commands aborted by ATA collateral abort (see 3.1.7) that are related to processing SCSI commands in an I\_T\_L nexus other than the specified I\_T\_L nexus, then:

- a) if the SATL supports ATA abort retry (see 3.1.6), then the SATL shall re-issue all ATA commands aborted by ATA collateral abort and continue processing of the affected I\_T\_L\_Q nexuses; or
- b) if the SATL does not support ATA abort retry, then for each I\_T\_L nexus other than the specified I\_T\_L nexus that had one or more SCSI tasks affected due to ATA commands aborted by ATA collateral abort, the SATL shall abort all commands for each affected I\_T\_L nexus and establish a UNIT ATTENTION condition with the additional sense code set to COMMANDS CLEARED BY ANOTHER INITIATOR.

NOTE 7 - A SATL that does not support ATA abort retry (see 3.1.6) is not able to comply with the SAM-3 requirement that ABORT TASK SET not abort tasks other than those in the specified I\_T\_L nexus.

### 6.3.6 CLEAR ACA

The service request for the CLEAR ACA task management function (see SAM-3) is:

Service Response = CLEAR ACA (IN (I\_T\_L nexus)).

The SATL shall process the CLEAR ACA task management function as defined in SAM-3.

NOTE 8 - The SATL responds to a CLEAR ATA task management function with a service response of FUNCTION REJECTED, as the SATL reports a NORMACA bit set to zero in standard INQUIRY data (see 8.1.2).

### 6.3.7 CLEAR TASK SET

The service request for the CLEAR TASK SET task management function (see SAM-3) is:

Service Response = CLEAR TASK SET (IN (I\_T\_L nexus)).

If the SATL indicates support for the full task management model (see 6.2.2), then the SATL shall process the CLEAR TASK SET task management function in accordance with a single task set that includes SCSI tasks for all I\_T\_L nexuses (i.e., the TST field in the Control mode page is set to 000h, see 10.1.5).

If the ATA device is processing any ATA commands, then the SATL shall:

- a) abort all outstanding ATA command(s);
- b) abort all tasks in the task set; and
- c) respond to the CLEAR TASK SET task management function with a service response of FUNCTION COMPLETE.

If the SATL aborts tasks in the task set for an I\_T\_L nexus other than the specified I\_T\_L nexus, then for each other I\_T\_L nexus, the SATL shall establish a unit attention condition with the additional sense code set to COMMANDS CLEARED BY ANOTHER INITIATOR.

### 6.3.8 LOGICAL UNIT RESET

The service request for the LOGICAL UNIT RESET task management function (see SAM-3) is:

Service Response = LOGICAL UNIT RESET (IN (I\_T\_L nexus)).

The SATL shall:

- 1) reset the ATA device as follows:
  - 1) optionally issue an ATA software reset (see 3.1.22) to the ATA device; and
  - 2) if the ATA software reset is not successful or not issued, then issue an ATA hardware reset (see 3.1.12) to the ATA device;

NOTE 9 - It is vendor-specific how the SATL determines if the ATA software reset is successful.

- 2) abort all tasks in the task set from the SATL internal context;
- 3) restore ATA volatile settings (see 3.1.24) to values consistent with the emulation of saved or default values of mode parameters, log parameters, and INQUIRY data (see SPC-3); and
- 4) return a service response of FUNCTION COMPLETE for the LOGICAL UNIT RESET task management function.

NOTE 10 - If more than one PATA device is present on a PATA bus, issuing an ATA software reset causes both devices to be reset.

### 6.3.9 QUERY TASK

The service request for the QUERY TASK task management function (see SAM-3) is:

Service Response = QUERY TASK (IN (I\_T\_L\_Q nexus)).

If the SATL supports the QUERY TASK task management function, the SATL shall return a service response of FUNCTION SUCCEEDED if the specified I\_T\_L\_Q nexus is in the task set, or the SATL shall return a service response of FUNCTION COMPLETE if the specified I\_T\_L\_Q nexus is not in the task set.

If the SATL does not support the QUERY TASK task management function the SATL shall return a service response of FUNCTION REJECTED.

### 6.3.10 Obsolete reset task management functions

The obsolete TARGET RESET task management function is sometimes used by a SCSI application client to cause a hard reset (i.e., similar to a power-on condition) for each logical unit of a specified target device. The SATL may process the TARGET RESET task management function by issuing an ATA hardware reset (see 3.1.12) to the ATA device(s) associated with the target device.

## 6.4 Control Byte

### 6.4.1 CONTROL byte overview

Table 7 describes SATL handling of the CDB CONTROL byte. See SAM-3 for CONTROL byte details.

**Table 7 — Control byte fields**

Field	Description
Vendor specific	The SATL may use this field for vendor-specific purposes.
NACA	If set to one, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
LINK	If set to one, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

## 6.5 I\_T nexus loss

The SATL may detect an I\_T nexus loss event (see SAM-3). If the SATL detects an I\_T nexus loss event the SATL handles the I\_T nexus loss event differently depending on whether the SATL provides multiple I\_T nexus access to the emulated SCSI logical unit.

If the SATL does not provide multiple I\_T nexus access to the emulated SCSI logical unit, the SATL shall handle the I\_T nexus loss as follows:

- 1) abort any outstanding ATA command(s) (see 6.3.2);
- 2) delete all tasks in the task set from the SATL internal context; and
- 3) establish a unit attention condition for each affected I\_T nexus with the additional sense code set to I\_T NEXUS LOSS OCCURRED.

If the SATL provides multiple I\_T nexus access to the emulated SCSI logical unit, the SATL shall handle the I\_T nexus loss as follows:

- 1) allow any outstanding ATA command(s) for each I\_T nexus that is not lost to complete;
- 2) abort any remaining ATA command(s) (see 6.3.2 and 6.3.3);
- 3) delete all tasks in the task set from the SATL internal context for tasks associated with the I\_T nexus that the I\_T nexus loss event occurred; and
- 4) establish a unit attention with the additional sense code set to I\_T NEXUS LOSS OCCURRED for the SCSI initiator port associated with the I\_T nexus that was lost.

## 7 Summary of SCSI / ATA command mappings

### 7.1 Translated and emulated commands

In the event of a discrepancy between the contents of this clause and the description of individual commands, description of individual commands shall apply.

Clause 7, clause 8, and clause 9 describe the SCSI to ATA command mapping for ATA devices. Translation for ATAPI devices is described in Annex Annex A.

The SATL shall not issue more than one ATA command to the ATA device representing the logical unit with the exception of ATA queued commands (see 3.1.19). The SATL shall queue received SCSI commands as necessary to enforce this.

Table 8 lists the SCSI / ATA command mappings defined in this standard. A SATL may implement commands defined in SPC-3 and SBC-2, but not listed in table 8. Translation of commands not listed in table 8 is vendor-specific.

**Table 8 — Summary of SCSI / ATA Command Mapping** (part 1 of 2)

SCSI command	ATA command(s)	Reference
ATA PASS-THROUGH (12)	Any	12.2.2
ATA PASS-THROUGH (16)		12.2.3
FORMAT UNIT	READ VERIFY SECTORS, READ VERIFY SECTORS EXT, WRITE SECTORS, WRITE SECTORS EXT	9.2
INQUIRY	IDENTIFY DEVICE	8.1
LOG SENSE	Log page dependent (see 10.2)	8.2
MODE SELECT (6)	Mode page dependent (see 10.1)	8.3
MODE SELECT (10)		8.4
MODE SENSE (6)		8.5
MODE SENSE (10)		8.6
READ (6)	See 9.1	9.3
READ (10)		9.5
READ (12)		9.6
READ (16)		9.7
READ BUFFER	READ BUFFER	8.7
READ CAPACITY (10)	IDENTIFY DEVICE	9.8
READ CAPACITY (16)		9.9
<del>READ MEDIA SERIAL NUMBER</del>	<del>IDENTIFY DEVICE</del>	<del>8.8</del>
REASSIGN BLOCKS	READ VERIFY SECTOR(S), READ VERIFY SECTOR(S) EXT, WRITE DMA, WRITE DMA EXT, WRITE DMA FUA EXT, WRITE DMA QUEUED, WRITE DMA QUEUED EXT, WRITE DMA QUEUED FUA EXT, and WRITE FPDMA QUEUED	9.10
REPORT LUNS	n/a	SPC-3
REQUEST SENSE	SMART RETURN STATUS	8.8
SEND DIAGNOSTIC	<del>EXECUTE DEVICE DIAGNOSTIC</del> <del>SMART EXECUTE OFF-LINE IMMEDIATE</del>	8.9
START STOP UNIT	FLUSH CACHE, FLUSH CACHE EXT, STANDBY, READ VERIFY SECTOR(S), or MEDIA EJECT	9.11

Table 8 — Summary of SCSI / ATA Command Mapping (part 2 of 2)

SCSI command	ATA command(s)	Reference
SYNCHRONIZE CACHE (10)	FLUSH CACHE or	9.12
SYNCHRONIZE CACHE (16)	FLUSH CACHE EXT	9.13
TEST UNIT READY	CHECK POWER MODE	8.12
VERIFY (10)	See 9.1	9.14
VERIFY (12)		9.15
VERIFY (16)		9.16
WRITE (6)	See 9.1	9.18
WRITE (10)		9.19
WRITE (12)		9.20
WRITE (16)		9.21
WRITE AND VERIFY (10)	See 9.1	9.23
WRITE AND VERIFY (12)		9.24
WRITE AND VERIFY (16)		9.25
WRITE BUFFER	WRITE BUFFER or DOWNLOAD MICROCODE	8.13
WRITE SAME (10)	See 9.1	9.26
WRITE SAME (16)		9.27

## 8 SCSI Primary Commands (SPC) mapping

### 8.1 INQUIRY command

#### 8.1.1 INQUIRY command overview

The INQUIRY command requests general information about a logical unit and target device. The INQUIRY command and selected vital product data pages shall be emulated using information from the ATA IDENTIFY DEVICE command and other information (see 8.1.2). Table 9 describes the emulation of fields in the INQUIRY CDB.

**Table 9 — INQUIRY CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 12h. The SATL shall issue an ATA IDENTIFY DEVICE command to the ATA device.
EVPD	The SATL shall implement this field as defined in SPC-3 (see 10.3).
PAGE CODE <sup>a</sup>	The SATL: <ul style="list-style-type: none"> <li>a) shall support the Supported VPD Pages VPD page (00h) (see 10.3.2);</li> <li>b) may support the Unit Serial Number VPD page (80h) (see 10.3.3);</li> <li>c) shall support the Device Identification VPD page (83h) (see 10.3.4);</li> <li>d) should support the Mode Page Policy VPD page (87h) (see 10.3.5) and</li> <li>e) shall support the ATA Information VPD page (89h) (see 12.4.2).</li> </ul>
ALLOCATION LENGTH	The SATL shall implement this field as defined in SPC-3.
CONTROL	6.4
<sup>a</sup> VPD page translations are defined in 10.3	

## 8.1.2 Standard INQUIRY data

Table 10 describes the standard INQUIRY data fields supported by the SATL.

**Table 10 — Standard INQUIRY data fields** (part 1 of 3)

Field	Description or reference
PERIPHERAL QUALIFIER	The SATL shall set this field to 000b to indicate that the peripheral device is currently connected to this logical unit. <sup>a</sup>
PERIPHERAL DEVICE TYPE	The SATL shall set this field to 00h to indicate that the peripheral device is a direct access block device. <sup>a</sup>
RMB	The SATL shall set this bit to the value of bit 7 of the general configuration word of the ATA IDENTIFY DEVICE data retrieved from the ATA device.
VERSION	The VERSION field indicates the version of SPC to which the SATL complies (see SPC-3) (e.g., 05h for SPC-3).
NORMACA	The SATL shall set this bit to zero to indicate the SATL does not support the NACA bit in the CONTROL byte (see 6.4).
HiSUP	Unspecified (see 3.4.2)
RESPONSE DATA FORMAT	The SATL shall set this field to 2h.
ADDITIONAL LENGTH	The SATL shall set this field to the length of the INQUIRY data that follows.
SCCS	Unspecified (see 3.4.2)
ACC	Unspecified (see 3.4.2)
TPGS	Unspecified (see 3.4.2)
3PC	Unspecified (see 3.4.2)
PROTECT	Unspecified (see 3.4.2)
BQUE	Unspecified (see 3.4.2)
ENC SERV	Unspecified (see 3.4.2)
MULTIP	Unspecified (see 3.4.2)
MCHNGR	The SATL shall set this bit to zero to indicate the peripheral device is not attached to a medium transport element.
ADDR16	Unspecified (see 3.4.2)
WBUS16	Unspecified (see 3.4.2)
<p><sup>a</sup> If the INQUIRY command is issued to an incorrect logical unit the SATL shall set the PERIPHERAL QUALIFIER field to 011b and shall set the PERIPHERAL DEVICE TYPE field to 1Fh.</p> <p><sup>b</sup> See 3.5.4.</p> <p><sup>c</sup> The full ATA IDENTIFY DEVICE data Model number field contents and the Firmware Revision field contents are returned in the ATA Information VPD page (see 12.4.2).</p> <p><sup>d</sup> The encoding used by the SPC-3 standard for INQUIRY version descriptors and the encoding used by the ATA8-ACS standard for IDENTIFY DEVICE major and minor version numbers differ. The two standards may not define values for the same revisions.</p>	

Table 10 — Standard INQUIRY data fields (part 2 of 3)

Field	Description or reference
SYNC	Unspecified (see 3.4.2)
CMDQUE	Unspecified (see 3.4.2)
T10 VENDOR IDENTIFICATION	The SATL shall set the T10 VENDOR IDENTIFICATION field to 'ATA-----' <sup>b</sup> .
PRODUCT IDENTIFICATION <sup>c</sup>	<p>The SATL shall set the PRODUCT IDENTIFICATION field to a representation of the first 16 bytes of the ATA IDENTIFY DEVICE data Model number field, where each pair of bytes are swapped to create a valid ASCII string format:</p> <ol style="list-style-type: none"> <li>1) byte 0 contains ATA IDENTIFY DEVICE word 27 bits 15:8 (i.e., byte 1);</li> <li>2) byte 1 contains ATA IDENTIFY DEVICE word 27 bits 7:0 (i.e., byte 0);</li> <li>3) byte 2 contains ATA IDENTIFY DEVICE word 28 bits 15:8 (i.e., byte 3);</li> <li>4) byte 3 contains ATA IDENTIFY DEVICE word 28 bits 7:0 (i.e., byte 2);</li> <li>...</li> <li>15) byte 14 contains ATA IDENTIFY DEVICE word 34 bits 15:8 (i.e., byte 15); and</li> <li>16) byte 15 contains ATA IDENTIFY DEVICE word 34 bits 7:0 (i.e., byte 14).</li> </ol>
PRODUCT REVISION LEVEL <sup>c</sup>	<p>The SATL shall set the PRODUCT REVISION LEVEL field to a four byte ASCII character representation of the ATA IDENTIFY DEVICE data Firmware revision field. Each pair of bytes are swapped to create a valid ASCII string format. Since the ATA IDENTIFY DEVICE data Firmware revision field contains eight ASCII characters and the standard INQUIRY data PRODUCT REVISION LEVEL field is four ASCII characters, the SATL shall select four of the eight ASCII characters from the ATA IDENTIFY DEVICE data Firmware revision field to return in the PRODUCT REVISION LEVEL field as follows:</p> <ol style="list-style-type: none"> <li>a) If the ATA IDENTIFY DEVICE data words 26:25 are set to four ASCII spaces (i.e., 20202020h), then the four ASCII characters selected shall contain: <ol style="list-style-type: none"> <li>1) byte 0 contains ATA IDENTIFY DEVICE data word 23 bits 15:8 (i.e., byte 1);</li> <li>2) byte 1 contains ATA IDENTIFY DEVICE data word 23 bits 7:0 (i.e., byte 0);</li> <li>3) byte 2 contains ATA IDENTIFY DEVICE data word 24 bits 15:8 (i.e., byte 3); and</li> <li>4) byte 3 contains ATA IDENTIFY DEVICE data word 24 bits 7:0 (i.e., byte 2);</li> </ol> <p style="text-align: center;">or</p> </li> <li>b) If the ATA IDENTIFY DEVICE data words 26:25 are not set to four ASCII spaces (i.e., 20202020h), then the four ASCII characters selected shall contain: <ol style="list-style-type: none"> <li>1) byte 0 contains ATA IDENTIFY DEVICE data word 25 bits 15:8 (i.e., byte 5);</li> <li>2) byte 1 contains ATA IDENTIFY DEVICE data word 25 bits 7:0 (i.e., byte 4);</li> <li>3) byte 2 contains ATA IDENTIFY DEVICE data word 26 bits 15:8 (i.e., byte 7); and</li> <li>4) byte 3 contains ATA IDENTIFY DEVICE data word 26 bits 7:0 (i.e., byte 6).</li> </ol> </li> </ol>
CLOCKING	Unspecified (see 3.4.2)
QAS	Unspecified (see 3.4.2)
IUS	Unspecified (see 3.4.2)
<p><sup>a</sup> If the INQUIRY command is issued to an incorrect logical unit the SATL shall set the PERIPHERAL QUALIFIER field to 011b and shall set the PERIPHERAL DEVICE TYPE field to 1Fh.</p> <p><sup>b</sup> See 3.5.4.</p> <p><sup>c</sup> The full ATA IDENTIFY DEVICE data Model number field contents and the Firmware Revision field contents are returned in the ATA Information VPD page (see 12.4.2).</p> <p><sup>d</sup> The encoding used by the SPC-3 standard for INQUIRY version descriptors and the encoding used by the ATA8-ACS standard for IDENTIFY DEVICE major and minor version numbers differ. The two standards may not define values for the same revisions.</p>	

**Table 10 — Standard INQUIRY data fields** (part 3 of 3)

Field	Description or reference
VERSION DESCRIPTOR 1 to VERSION DESCRIPTOR 8	The SATL shall include version descriptors (see SPC-3) for: <ul style="list-style-type: none"> <li>a) the SCSI Architecture Model standard (e.g., SAM-3);</li> <li>b) this standard;</li> <li>c) the SCSI Primary Commands standard (e.g., SPC-3);</li> <li>d) the SCSI Block Commands standard (e.g., SBC-2);</li> <li>e) if the SATL receives SCSI commands through a SCSI target port (see figure 5 in 5.1), the version of the transport protocol to which the SCSI target port was designed;</li> <li>f) if the SATL sends ATA commands through a SAS STP initiator port (see figure 7 in 5.1), the version of SAS (e.g., SAS-1.1) to which the SAS STP initiator port was designed; and</li> <li>g) the version(s) of ATA standards (e.g., ATA8-ACS and ATA8-AAM) to which the ATA device claims compliance in the ATA IDENTIFY DEVICE data word 80 (i.e., Major version number) and ATA IDENTIFY DEVICE data word 81 (i.e., Minor version number), and words 222 and 223.<sup>d</sup></li> </ul>
Vendor specific parameters	Unspecified (see 3.4.2)
<p><sup>a</sup> If the INQUIRY command is issued to an incorrect logical unit the SATL shall set the PERIPHERAL QUALIFIER field to 011b and shall set the PERIPHERAL DEVICE TYPE field to 1Fh.</p> <p><sup>b</sup> See 3.5.4.</p> <p><sup>c</sup> The full ATA IDENTIFY DEVICE data Model number field contents and the Firmware Revision field contents are returned in the ATA Information VPD page (see 12.4.2).</p> <p><sup>d</sup> The encoding used by the SPC-3 standard for INQUIRY version descriptors and the encoding used by the ATA8-ACS standard for IDENTIFY DEVICE major and minor version numbers differ. The two standards may not define values for the same revisions.</p>	

## 8.2 LOG SENSE command

### 8.2.1 LOG SENSE command overview

The LOG SENSE command provides a means for the application client to retrieve statistical or other operational information maintained by the SCSI target device about the SCSI target device or its logical units. Table 11 shows the translation for fields specified in the LOG SENSE CDB.

**Table 11 — LOG SENSE CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 4Dh. The SATL shall implement support for this field by returning the log page data for the particular page requested.
PPC	Unspecified (see 3.4.2)
SP	Unspecified (see 3.4.2)
PC	8.2.2
PAGE CODE	8.2.3
PARAMETER POINTER	Unspecified (see 3.4.2)
ALLOCATION LENGTH	The SATL shall implement support for this field as defined in SPC-3.
CONTROL	6.4

### 8.2.2 PC (page control) field

The SATL shall implement this field as defined in SPC-3. The SATL interpretation and support of the page control values is shown in table 12.

**Table 12 — PC field**

Code	Description
00b	Threshold values: unspecified (see 3.4.2)
01b	Cumulative values: supported
10b	Default threshold values: unspecified (see 3.4.2)
11b	Default cumulative values: unspecified (see 3.4.2)

### 8.2.3 PAGE CODE field

The SATL shall support this field as defined in SPC-3. The SATL emulation for support of the PAGE CODE field is provided in table 13.

**Table 13 — PAGE CODE field**

Code	Description
00h	Supported Log Pages log page: The SATL shall implement this page by returning a list of supported log pages (see 10.2.3).
10h	Self-Test Results log page: The SATL shall determine if the ATA SMART self-test is supported from the ATA IDENTIFY DEVICE data word 84 bit 1. If the ATA SMART self-test is not supported (i.e., word 84 bit 1 is set to zero) the SATL shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB. If the ATA SMART self-test is supported (i.e., word 84 bit 1 is set to one) the SATL shall return the translated Self-Test Results log page to the application client (see 10.2.4).
2Fh	Informational Exceptions log page: The SATL shall determine if the ATA SMART feature set is supported from the ATA IDENTIFY DEVICE data word 82 bit 0. If the ATA SMART feature set is not supported (i.e., word 82 bit 0 is set to zero) the SATL shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB. If the ATA SMART feature set is supported (i.e., word 82 bit 0 is set to one) the SATL shall determine if the ATA SMART feature set is enabled or disabled from the ATA IDENTIFY DEVICE data word 85 bit 0. If the ATA SMART feature set is disabled (i.e., word 85 bit 0 is set to zero) the SATL shall return a CHECK CONDITION status with the sense key set to ABORTED COMMAND and additional sense code set to ATA DEVICE FEATURE NOT ENABLED. If the ATA SMART feature set is enabled (i.e., word 85 bit 0 is set to one) the SATL shall return the translated Informational Exceptions log page to the application client (see 10.2.5.1).
All others	Unspecified (see 3.4.2)

## 8.3 MODE SELECT (6) command

### 8.3.1 MODE SELECT (6) command overview

The MODE SELECT(6) command (see SPC-3) provides a means for an application client to specify medium, logical unit, or peripheral device parameters to a device server in the SATL. Device servers that implement the MODE SELECT (6) command shall also implement the MODE SENSE (6) command. Application clients should issue a MODE SENSE (6) command prior to each MODE SELECT (6) command to determine supported mode pages, page lengths, and other parameters.

The Mode Page Policy VPD page should be implemented (see 10.3.5). After a logical unit reset, the SATL shall set all mode page values to saved or default values. See clause 10 for supported mode pages.

### 8.3.2 MODE SELECT (6) CDB fields

The SATL shall support MODE SELECT (6) CDB fields as shown in table 14.

**Table 14 — MODE SELECT (6) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 15h. The SATL shall modify logical unit, or peripheral device parameters for supported mode pages and parameters as specified in mode pages received from the application client. Some operational parameters in individual pages are provided via ATA. See clause 10 for specific requirements.
SP	Unspecified (see 3.4.2)
PF	If this bit is set to zero (i.e., specifies that mode pages are vendor specific), then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. The SATL shall support this bit being set to one (i.e., specifies that all mode page formats correspond to SPC-3 and SBC-2 mode page formats).
PARAMETER LIST LENGTH	This field should be set to the length of the mode parameter list to be transferred from the application client.
CONTROL	6.4

## 8.4 MODE SELECT (10) command

The MODE SELECT (10) command (see SPC-3) provides a means for an application client to set parameters in the device server in a SATL. It is a complementary command to the MODE SENSE(10) command.

The SATL shall implement the MODE SELECT (10) command using the translation described in 8.3. Device servers that implement the MODE SELECT (10) command shall also implement the MODE SENSE (10) command. See 10.1 for supported mode pages.

## 8.5 MODE SENSE (6) command

### 8.5.1 MODE SENSE (6) command overview

The MODE SENSE (6) command (see SPC-3) provides a means for a device server in a SATL to report parameters to an application client. It is a complementary command to the MODE SELECT(6) command. Device servers that implement the MODE SENSE (6) command shall also implement the MODE SELECT(6) command. See 10.1 for supported mode pages.

### 8.5.2 MODE SENSE (6) CDB fields

The SATL shall support MODE SENSE (6) CDB fields as shown in table 15.

**Table 15 — MODE SENSE (6) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 1Ah. The SATL shall return the requested mode pages to the application client. Some operational parameters in individual pages are gathered by issuing ATA commands (see 10.1).
DBD	A DBD bit set to zero specifies that zero or more block descriptors may be returned in MODE SENSE data. The SATL shall support only the mode parameter block descriptor format for direct-access block devices.
PC	Current values (i.e., the PC field is set to 00b) shall be supported. Reporting changeable, saveable, and default values is unspecified (see 3.4.2).
PAGE CODE	This field specifies the particular mode page requested (see 10.1). If the SATL does not support the specified mode page, the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
SUB PAGE CODE	This field specifies the sub page code within the page code specified by PAGE CODE field that is requested by the application client (see 10.1).
ALLOCATION LENGTH	The SATL shall implement this field as defined in SPC-3 (see 3.1.2).
CONTROL	6.4

### 8.6 MODE SENSE (10) command

The MODE SENSE (10) command (see SPC-3) provides a means for a device server in a SATL to report parameters to an application client. It is a complementary command to the MODE SELECT(10) command.

The SATL shall implement the MODE SENSE (10) command using the translation described in 8.5. Device servers that implement the MODE SENSE (10) command shall also implement the MODE SELECT(10) command. See 10.1 for supported mode pages.

## 8.7 READ BUFFER command

### 8.7.1 READ BUFFER command overview

The READ BUFFER command (see SPC-3) is used in conjunction with the WRITE BUFFER command as a diagnostic function for testing memory in the SCSI device and the integrity of a service delivery subsystem. This command shall not alter the medium. Table 16 shows the translation for fields specified in the CDB for the READ BUFFER command.

**Table 16 — READ BUFFER CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 3Ch. The SATL shall issue the ATA READ BUFFER command to the ATA device.
MODE	8.7.2
BUFFER ID	If the the BUFFER ID field is set to 00h then the SATL shall return information describing or data read from the sector buffer in the ATA device, depending on the value in the MODE field (see 8.7.2). If the the BUFFER ID field is set to a value other than 00h then the translation is unspecified (see 3.4.2), and the SATL shall process the READ BUFFER command as defined in SPC-3.
BUFFER OFFSET	The meaning of this field depends on the contents of the MODE field (see 8.7.2).
ALLOCATION LENGTH	The meaning of this field depends on the contents of the MODE field (see 8.7.2).
CONTROL	6.4

The logical sector buffer in a ATA device shall be used to emulate the READ BUFFER command, so the size of the buffer is limited to 512 bytes for data buffer and echo buffers.

### 8.7.2 MODE field

#### 8.7.2.1 MODE field overview

Table 17 describes values of the MODE field that the SATL shall support.

**Table 17 — MODE field**

Code	Description or reference
02h (i.e., Data)	If BUFFER ID field is set to 00h, then the translation shall be to the ATA READ BUFFER command (see 8.7.2.2). Otherwise, the translation is unspecified (see 3.4.2).
03h (i.e., Descriptor)	8.7.2.3
All others	Unspecified (see 3.4.2)

#### 8.7.2.2 Data mode

If the BUFFER ID field is set to 00h, the BUFFER OFFSET field is set to 00h, and the ALLOCATION LENGTH field is set to 512, then the SATL shall return 512 bytes of data.

If the BUFFER ID field is set to 00h, the BUFFER OFFSET field is set to 00h, and the ALLOCATION LENGTH field is set to a value other than 512, then the SATL shall either:

- a) return the the lesser of 512 bytes of data or the number of bytes specified in the ALLOCATION LENGTH field from the buffer in the ATA device by sending an ATA READ BUFFER command to the ATA device; or
- b) terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

If the BUFFER ID field is set to 00h and the BUFFER OFFSET field is set to a value other than 00h then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

The SATL may support a value other than 00h in the BUFFER ID field. If the SATL supports a value other than 00h in the BUFFER ID field the implementation shall be as defined in SPC-3.

A WRITE BUFFER command may be sent to the same buffer ID before it is read with the READ BUFFER command.

### 8.7.2.3 Descriptor mode

If the ALLOCATION LENGTH field is set to less than four, the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

If the ALLOCATION LENGTH field is set to four or greater, the SATL shall return four bytes of data describing the requested buffer, including the OFFSET BOUNDARY field and the BUFFER CAPACITY field.

If the BUFFER ID field is set to zero then the SATL shall return values in the OFFSET BOUNDARY field and the BUFFER CAPACITY field describing the buffer in the ATA device that is accessible using the ATA READ BUFFER command and the ATA WRITE BUFFER command by returning a value of FFh in the OFFSET BOUNDARY field (i.e., zero is the only supported value in the BUFFER OFFSET field) and a value of 200h (i.e., 512 bytes) in the BUFFER CAPACITY field.

The SATL may support a value other than zero in the BUFFER ID field. If the SATL supports a value other than zero in the BUFFER ID field the implementation shall be as defined in SPC-3.

## 8.8 REQUEST SENSE command

### 8.8.1 REQUEST SENSE command overview

The REQUEST SENSE command requests any available sense data to be returned to the application client.

If the SCSI transport protocol for the SATL supports autosense (see 3.1.28), the SATL shall return sense data using autosense. Otherwise, the SATL shall return sense data in response to the REQUEST SENSE command (see SAM-2).

The SATL shall determine if there is sense data to return to the application client. If the SATL has no sense data to return, then the SATL shall complete the REQUEST SENSE command with GOOD status with the sense key set to NO SENSE and the additional sense code set to NO ADDITIONAL SENSE DATA (see SPC-3). Table 18 lists examples of conditions where the SATL has sense data to return.

**Table 18 — Special Request Sense behavior reference**

<b>Emulated device state</b>	<b>Reference</b>
Status other than GOOD to return	SPC-3
FORMAT UNIT in progress	8.8.2
SMART threshold exceeded condition	8.8.3
Stopped power condition (i.e., ATA device in standby power management state)	8.8.4
Unit attention condition established	8.8.5

Table 19 shows the fields in the REQUEST SENSE CDB.

**Table 19 — REQUEST SENSE CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 03h. The SATL shall return any available sense data to the application client.
DESC <sup>a</sup>	Unspecified (see 3.4.2)
ALLOCATION LENGTH	Unspecified (see 3.4.2)
CONTROL	6.4
<sup>a</sup> If the SATL supports the ATA PASS-THROUGH command (see 12.2), then the SATL shall support returning descriptor format sense data (i.e., specified by the DESC bit set to one).	

### 8.8.2 FORMAT UNIT in progress

If the SATL is processing a FORMAT UNIT command and the SATL receives a REQUEST SENSE command, the SATL shall return GOOD status with the sense key set to NOT READY with the additional sense code set to LOGICAL UNIT NOT READY, FORMAT IN PROGRESS. The sense key specific bytes shall be set to progress indication as defined in SBC-2 and SPC-3.

### 8.8.3 SMART threshold exceeded condition

If:

- a) the ATA device has the SMART feature set enabled (i.e., IDENTIFY DEVICE data word 85 bit 0 is set to one);
- b) the MRIE field in the Informational Exceptions Control mode page is set to 6h (see 10.1.8.2);
- c) the DEXCPT bit in the Informational Exceptions Control mode page is set to zero; and
- d) the most recent ATA SMART RETURN STATUS command to the ATA device indicates that the error threshold has been exceeded;

then the SATL shall:

- a) return parameter data containing sense data with the sense key set to NO SENSE with the additional sense code set to HARDWARE IMPENDING FAILURE GENERAL HARD DRIVE FAILURE; and
- b) complete the REQUEST SENSE command with GOOD status.

### 8.8.4 Stopped power condition

If the emulated logical unit is in the stopped power condition (i.e., the ATA device is in the Standby power management state) and there is no sense data to return for a previously returned CHECK CONDITION status, then the SATL shall:

- 1) return parameter data containing sense data with the sense key set to NO SENSE with the additional sense code set to NO ADDITIONAL SENSE DATA; and
- 2) complete the REQUEST SENSE command with GOOD status.

Sense data returned for a previously returned CHECK CONDITION status resulting from a media access command or a TEST UNIT READY command received when the logical unit is in the stopped power condition is described in 8.12 (i.e., the TEST UNIT READY command) and 9.11 (i.e., the START STOP UNIT command).

### 8.8.5 Unit attention condition established

The SATL shall:

- 1) return parameter data containing sense data describing the unit attention condition (see SPC-3); and
- 2) complete the REQUEST SENSE command with GOOD status.

## 8.9 SECURITY PROTOCOL IN command

### 8.9.1 SECURITY PROTOCOL IN command overview

The SECURITY PROTOCOL IN command provides a means for the application client to retrieve security information from a SCSI target device. Table 20 shows the translation for fields specified in the SECURITY PROTOCOL IN CDB.

**Table 20 — SECURITY PROTOCOL IN CDB field translation**

Field	Description or Reference
OPERATION CODE	Set to A2h. The SATL shall issue the ATA TRUSTED RECEIVE command or the ATA TRUSTED RECEIVE DMA command to the ATA device.
SECURITY PROTOCOL	8.9.1.1
SECURITY PROTOCOL SPECIFIC	8.9.1.2
INC_512	8.9.1.3
ALLOCATION LENGTH	8.9.1.3
CONTROL	6.4

#### 8.9.1.1 SECURITY PROTOCOL field

The SECURITY PROTOCOL field shall be copied to the ATA Security\_Protocol field.

#### 8.9.1.2 SECURITY PROTOCOL SPECIFIC field

The SECURITY PROTOCOL SPECIFIC field shall be copied to the ATA SP\_Specific field.

#### 8.9.1.3 ALLOCATION LENGTH field

##### 8.9.1.3.1 ALLOCATION LENGTH field translation overview

---



---

[Editor's Note 4: This translation does not address what happens if an application client specifies a zero allocation length. If it were to do so, under the current translation, it would cause a protocol violation on the ATA side as zero length transfers are not permitted for PIO or DMA proto](#)

---



---

The translation of ALLOCATION LENGTH varies based on the value of SECURITY PROTOCOL.

---



---

[Editor's Note 5: I think the following clause should probably be made into a table instead of the text as it was voted into the standard.](#)

---



---

##### 8.9.1.3.2 SECURITY PROTOCOL 00h - 06h

If INC\_512 is set to one:

- a) If ALLOCATION LENGTH is greater than FFFFh, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB;
- b) Otherwise, the ATA Transfer\_Length field shall be set to ALLOCATION LENGTH (15:0). After completion of the ATA TRUSTED RECEIVE or ATA TRUSTED RECEIVE DMA command, the data shall be transferred to the SCSI application client.

If INC\_512 is set to zero:

- a) If ALLOCATION\_LENGTH is greater than 1FF\_FE0h, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB;
- b) otherwise, the ATA Transfer\_Length field shall be translated from bytes to a number of padded 512-byte units from the result of the following calculation:

$$\text{ATA Transfer\_Length}(15:0) = ( (\text{ALLOCATION\_LENGTH} + 511) / 512 )$$

After successful completion of the ATA TRUSTED RECEIVE or ATA TRUSTED RECEIVE DMA command, the data shall be transferred to the SCSI application client up to the specified ALLOCATION\_LENGTH number of bytes.

**8.9.1.3.3 SECURITY PROTOCOL values 07h - FFh**

The translation of this field is unspecified (see 3.4.2).

**8.10 SECURITY PROTOCOL OUT command**

**8.10.0.1 SECURITY PROTOCOL OUT command overview**

The SECURITY PROTOCOL OUT command provides a means for the application client to send security information to a SCSI target device. Table 21 shows the translation for fields specified in the SECURITY PROTOCOL OUT CDB.

**Table 21 — SECURITY PROTOCOL OUT CDB field translation**

Field	Description or Reference
OPERATION CODE	Set to A2h. The SATL shall issue the ATA TRUSTED RECEIVE command or the ATA TRUSTED RECEIVE DMA command to the ATA device.
SECURITY PROTOCOL	8.9.1.1
SECURITY PROTOCOL SPECIFIC	8.9.1.2
INC_512	8.9.1.3
ALLOCATION_LENGTH	8.9.1.3
CONTROL	6.4

**8.10.0.2 SECURITY PROTOCOL field**

The SECURITY PROTOCOL field shall be copied to the ATA Security\_Protocol field.

**8.10.0.3 SECURITY PROTOCOL SPECIFIC field**

The SECURITY PROTOCOL SPECIFIC field shall be copied to the ATA SP\_Specific field.

**8.10.0.4 ALLOCATION\_LENGTH field**

---



---

Editor's Note 6: This translation does not address what happens if an application client specifies a zero allocation length. If it were to do so, under the current translation, it would cause a protocol violation on the ATA side as zero length transfers are not permitted for PIO or DMA protocols.

---



---

**8.10.0.4.1 SECURITY PROTOCOL field translation overview**

The translation of ALLOCATION LENGTH varies based on the value of SECURITY PROTOCOL.

**8.10.0.4.2 SECURITY PROTOCOL values 00h - 06h**

If the INC\_512 is set to one:

- a) If TRANSFER LENGTH is greater than FFFFh, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB;
- b) Otherwise, the ATA Transfer\_Length field shall be set to ALLOCATION LENGTH (15:0). The ATA TRUSTED SEND or ATA TRUSTED SEND DMA command shall transfer the data.

If the INC\_512 bit is set to zero:

- a) If TRANSFER LENGTH is greater than 1FF\_FE00h, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB;
- b) Otherwise, the ATA Transfer\_Length field shall be translated from bytes to a number of padded 512-byte units from the result of the following calculation:

$$\text{ATA Transfer\_Length}(15:0) = ( (\text{TRANSFER LENGTH} + 511) / 512 )$$

The final data block may be padded (see SPC-4). The ATA TRUSTED SEND or ATA TRUSTED SEND DMA command shall transfer the padded data for ATA Transfer\_Length number of data blocks.

**8.10.0.4.3 SECURITY PROTOCOL values 07h - FFh**

The translation of this field is unspecified (see 3.4.2).

## 8.11 SEND DIAGNOSTIC command

### 8.11.1 SEND DIAGNOSTIC command overview

The SEND DIAGNOSTIC command provides a mechanism for an application client to request diagnostic operations to be performed on the target device, logical unit, or both. The SATL shall implement the default self-test feature (see SPC-3). Table 22 shows the translation for fields specified in the SEND DIAGNOSTIC CDB.

**Table 22 — SEND DIAGNOSTIC CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 1Dh. See 8.11.2.
SELF-TEST CODE	8.11.2 and 8.11.3.
PF	Unspecified (see 3.4.2)
SELFTEST	8.11.3
DEVOFFL	If the DEVOFFL bit is set to zero, then the SATL shall process the command as specified in SPC-3. If the DEVOFFL bit is set to one, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
UNITOFFL	If the UNITOFFL bit is set to zero, then the SATL shall process the command as specified in SPC-3. If the UNITOFFL bit is set to one, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
PARAMETER LIST LENGTH	If the PARAMETER LIST LENGTH field is set to zero, then the SATL shall process the command as specified in SPC-3. If the PARAMETER LIST LENGTH field is not set to zero, then the SATL shall terminate the command with a CHECK CONDITION status with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB.
CONTROL	6.4

### 8.11.2 SELF-TEST CODE field

The SATL shall determine if the value in the SELF-TEST CODE field is valid depending on the value of the SELFTEST bit and what is reported by the ATA device with respect to the ATA SMART EXECUTE OFF-LINE IMMEDIATE command (see 8.11.3).

If the value of the SELF-TEST CODE field is valid, then the SATL shall process the command as described in table 23.

**Table 23 — SELF-TEST CODE field translation (part 1 of 2)**

Code	Name of test	Description of test
000b	Default self-test	Used when the SELFTEST bit is set to one.
001b	Background short self-test	The SATL shall perform the following: <ol style="list-style-type: none"> <li>1) return status for the SEND DIAGNOSTIC command as soon as the CDB has been validated and initialize the Self-Test Results log page (see 10.2.4 and SPC-3); and</li> <li>2) issue an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the LBA Low register set to 1 (i.e., Execute SMART Short self-test routine immediately in off-line mode) to the ATA device.</li> </ol>
010b	Background extended self-test	The SATL shall perform the following: <ol style="list-style-type: none"> <li>1) return status for the SEND DIAGNOSTIC command as soon as the CDB has been validated and initialize the Self-Test Results log page (see 10.2.4 and SPC-3); and</li> <li>2) issue an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the LBA Low register set to 2 (i.e., Execute SMART Extended self-test routine immediately in off-line mode) to the ATA device.</li> </ol>
011b	Reserved	
100b	Abort background self-test	If a previous SEND DIAGNOSTIC command specified a background self-test function and that self-test has not completed (see SPC-3), then the SATL shall issue an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the LBA Low register set to 127 (i.e., Abort off-line mode self-test routine) to the ATA device. If the ATA SMART EXECUTE OFF-LINE IMMEDIATE command completes without error, the SATL shall return GOOD status. If the ATA command completes with an error the SATL shall respond as defined in SPC-3.
101b	Foreground short self-test	The SATL shall issue an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the LBA Low register set to 129 (i.e., Execute SMART Short self-test routine immediately in captive mode) to the ATA device. If the ATA SMART EXECUTE OFF-LINE IMMEDIATE command completes without error, the SATL shall update the Self-Test Results log page prior to returning GOOD status. If the ATA command completes with an error the SATL shall first update the Self-Test Results log page (i.e., if supported, see SPC-3), and terminate the command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to LOGICAL UNIT FAILED SELF-TEST.

Table 23 — SELF-TEST CODE field translation (part 2 of 2)

Code	Name of test	Description of test
110b	Foreground extended self-test	The SATL shall issue an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the LBA Low register set to 130 (i.e., Execute SMART Extended self-test routine immediately in captive mode) to the ATA device. If the ATA SMART EXECUTE OFF-LINE IMMEDIATE command completes without error, the SATL shall update the Self-Test Results log page prior to returning GOOD status. If the ATA command completes with an error, the SATL shall first update the Self-Test Results log page (i.e., if supported, see SPC-3), and then terminate the command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and additional sense code set to LOGICAL UNIT FAILED SELF-TEST.
111b	Reserved	

### 8.11.3 SELFTEST bit

The SATL shall translate the SELFTEST bit according to whether or not the ATA device supports and has

enabled the ATA SMART EXECUTE OFF-LINE IMMEDIATE command as shown in table 24.

**Table 24 — SELFTEST bit**

Code	ATA SMART EXECUTE OFF-LINE IMMEDIATE command <sup>a</sup>		SATL emulation
	supported	enabled	
0	no	n/a	The SATL shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
	yes	no	The SATL shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to ATA DEVICE FEATURE NOT ENABLED.
		yes	The SELF-TEST CODE field is valid, and the SATL shall process the SEND DIAGNOSTIC command according to the value specified in the SELF-TEST CODE field as defined in 8.11.2.
1	no	n/a	The SATL shall send three ATA verify commands (see 3.1.23) to the ATA device with the Sector Count set to one and the LBA set to: <ul style="list-style-type: none"> <li>a) zero;</li> <li>b) the maximum user-addressable LBA; and</li> <li>c) an arbitrary number between zero and the maximum user-addressable LBA.</li> </ul>
	yes	no	If any of the three ATA verify commands ends with an error, then the SATL shall terminate the SEND DIAGNOSTIC command with a CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to LOGICAL UNIT FAILED SELF-TEST. If all three ATA verify commands complete without error <sup>b</sup> , then the SATL shall return GOOD status.
	yes		The SATL shall issue an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the LBA Low register set to 129 (i.e., Execute SMART Short self-test routine immediately in captive mode) to the ATA device. If the ATA EXECUTE OFF-LINE IMMEDIATE command completes without error, the SATL shall return GOOD status. If the ATA EXECUTE OFF-LINE IMMEDIATE command completes with an error, the SATL shall terminate the SEND DIAGNOSTIC command with a CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to LOGICAL UNIT FAILED SELF-TEST.
<sup>a</sup> The SATL shall determine if the ATA SMART EXECUTE OFF-LINE IMMEDIATE command is supported and enabled based on the ATA IDENTIFY DEVICE data word 84 bit 1, and word 85 bit 0 (see ATA8-ACS).			
<sup>b</sup> The SATL may retry any of the three ATA Verify commands if an ATA Verify command fails on the first attempt, and the retried command may specify an alternate LBA. If the retried command completes without error, the SATL may consider the ATA Verify command as having completed without error.			

## 8.12 TEST UNIT READY command

### 8.12.1 TEST UNIT READY command overview

The TEST UNIT READY command is used to determine whether the device is ready (see table 25).

**Table 25 — TEST UNIT READY CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 09h. See 8.12.2.
CONTROL	6.4

### 8.12.2 TEST UNIT READY command translation

The SATL processes the TEST UNIT READY command as follows:

- 1) If any condition exists that prevents the SATL from issuing commands to the ATA device, the SATL should terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY with the additional sense code of LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE;
- 2) If the device is in the stopped state as the result of processing a START STOP UNIT command (see 9.11), then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY and the additional sense code of LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED;
- 3) If the ATA device is performing a self-test in the foreground mode, the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to NOT READY, and the additional sense code set to LOGICAL UNIT NOT READY, SELF-TEST IN PROGRESS;
- 4) If the SATL is processing a FORMAT UNIT command for the emulated device (see 9.2), then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY and the additional sense code set to LOGICAL UNIT NOT READY, FORMAT IN PROGRESS;
- 5) If the ATA device supports the Removable Media feature set (i.e., ATA IDENTIFY DEVICE data word 82 bit 2 is set to one), then the SATL shall issue an ATA GET MEDIA STATUS command to the ATA device. If the ATA device completes the command with the NM bit set to one in the Error register, then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY and the additional sense code of MEDIUM NOT PRESENT; and
- 6) If the ATA device completed the most recent ATA command with the DF bit set to one in the Status register, then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code of LOGICAL UNIT FAILURE.

If none of the conditions defined in items 1 through 6 exist, then the SATL shall issue an ATA CHECK POWER MODE command to the ATA device, and:

- a) If the ATA CHECK POWER MODE command completes with an error, then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY, and the additional sense code set to LOGICAL UNIT DOES NOT RESPOND TO SELECTION; or
- b) If the ATA CHECK POWER MODE command completes without error, then the SATL shall complete the TEST UNIT READY command with GOOD status.

## 8.13 WRITE BUFFER command

### 8.13.1 WRITE BUFFER command overview

The WRITE BUFFER command (see SPC-3) is used in conjunction with the READ BUFFER command as a diagnostic function for testing logical unit memory in the SCSI target device and the integrity of a service delivery subsystem. An additional mode is provided for downloading and saving microcode.

Table 26 shows the translation for fields specified in the WRITE BUFFER CDB.

**Table 26 — WRITE BUFFER CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 3Bh. The SATL shall: a) issue an ATA WRITE BUFFER command to the ATA device; b) issue an ATA DOWNLOAD MICROCODE command to the ATA device; or c) emulate the specified function (i.e., if supported); depending on the values in the BUFFER ID field and MODE field (see 8.13.2.1).
MODE	8.13.2.1
BUFFER ID	If the the BUFFER ID field is set to 00h then the SATL shall transfer data to the buffer in the ATA device, download microcode to the ATA device, or emulate the specified WRITE BUFFER function, depending on the value set in the MODE field (see 8.13.2). If the the BUFFER ID field is set to a value other than 00h then the translation is unspecified (see 3.4.2), and the SATL shall process the WRITE BUFFER command as defined in SPC-3.
BUFFER OFFSET	The meaning of this field depends on the contents of the MODE field (see 8.13.2.1).
PARAMETER LIST LENGTH	The meaning of this field depends on the contents of the MODE field (see 8.13.2.1).
CONTROL	6.4

### 8.13.2 MODE field

#### 8.13.2.1 MODE field overview

The MODE field specifies the function to be performed by the SATL. If the MODE field is set to 02h, the SATL shall issue an ATA WRITE BUFFER command to the ATA device. If the MODE field is set to 05h the SATL shall issue a DOWNLOAD MICROCODE command to the ATA device as specified in table 27.

**Table 27 — MODE field**

Code	Description or reference
02h (i.e., Write data)	Translated to ATA WRITE BUFFER command (see 8.13.2.2).
05h (i.e., Download microcode and save)	Translated to the ATA DOWNLOAD MICROCODE command. The features register shall be set to 07h indicating downloaded microcode is saved for immediate and future use (see 8.13.2.3).
All others	Unspecified (see 3.4.2)

### 8.13.2.2 Write data mode

If:

- a) the BUFFER ID field is set to 00h;
- b) the BUFFER OFFSET field is set to 00h; and
- c) the PARAMETER LIST LENGTH field is set to 512;

then the SATL shall write the specified number of bytes to the buffer in the ATA device by sending an ATA WRITE BUFFER command to the ATA device.

If the BUFFER ID FIELD is set to 00h and either:

- a) the BUFFER OFFSET field is set to a value other than 00h; or
- b) the PARAMETER LIST LENGTH field is set to a value other than 512;

then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

The SATL may support a value other than 00h in the BUFFER ID field. If the SATL supports a value other than 00h in the BUFFER ID field the implementation shall be as defined in SPC-3.

### 8.13.2.3 Download microcode mode

In this mode, data transferred to the SATL from the application client is transmitted to the ATA device using the ATA DOWNLOAD MICROCODE command.

The SATL shall issue an ATA DOWNLOAD MICROCODE command to the ATA device when it receives a WRITE BUFFER command with the MODE field set to 05h. The SATL shall transfer the microcode image or control information from the application client to the ATA device, and then complete the WRITE BUFFER command with GOOD status. The SATL shall check if the ATA DOWNLOAD MICROCODE command completed with an error. If the ATA DOWNLOAD MICROCODE command completed with an error, the SATL shall establish a unit attention condition and return a deferred error (see SPC-3 and clause 11).

After the ATA device reinitializes successfully, running the new microcode image, the SATL shall establish a unit attention condition (see SAM-3) for the initiator port associated with all I\_T nexuses except the I\_T nexus on which the set of WRITE BUFFER commands was received, with the additional sense code set to MICROCODE HAS BEEN CHANGED.

## 9 SCSI Block Commands (SBC) mapping

### 9.1 Translating LBA and transfer length and ATA command use constraints

#### 9.1.1 Overview

A SATL may implement a direct logical block mapping of ATA logical sectors to SCSI logical blocks (see 9.1.2), or the SATL may implement indirect logical block mapping translation (see 9.1.3).

#### 9.1.2 Direct logical block mapping model

If the SATL implements direct logical block mapping (see 3.1.33), the logical block size indicated by the BLOCK LENGTH IN BYTES field in the READ CAPACITY data (see 9.8.2 and 9.9.2) shall equal the ATA logical sector Size (see 3.1.15). The ATA LBA of an ATA logical sector shall equal the logical block address of the corresponding SCSI logical block.

#### 9.1.3 Indirect logical block mapping model

If the SATL implements indirect block mapping (see 3.1.38), the constraints of the direct logical block mapping model do not apply. The logical block size indicated by the BLOCK LENGTH IN BYTES field in the READ CAPACITY data (see 9.8.2 and 9.9.2) may not equal the ATA logical sector size (see 3.1.15) (e.g., SCSI logical block size of 520 bytes with an ATA Logical Sector Size of 512 bytes). The SATL translates between the SCSI LOGICAL BLOCK ADDRESS field and the ATA LBA in a vendor-specific manner. The result of a logical block address translated in one direction and then translated in the reverse direction shall yield the original logical block address.

#### 9.1.4 Selection of ATA block commands

The ATA commands the SATL may use to implement the functions specified by SCSI block commands depend upon:

- a) the value of the LOGICAL BLOCK ADDRESS and TRANSFER LENGTH fields specified in the SCSI CDB; and
- b) the capabilities of the ATA device and the ATA host within the SATL.

Table 28 relates selection conditions to allowable ATA commands used to implement SCSI block storage data transfer commands. ATA commands listed in the Allowed ATA commands column shall not be used in the emulation of a SCSI block command if the prerequisite conditions listed in Selection Prerequisites columns are not met (i.e., the word 'yes' in a Selection Prerequisites column means the prerequisite shall be met before the SATL may use an ATA command listed in that row, and the word 'no' indicates the prerequisite need not be met for the SATL to use the ATA command listed).

Table 28 — ATA commands used for SCSI block command translations

Selection Prerequisites <sup>a</sup>					Allowed ATA commands
Highest ATA logical sector accessed	ATA feature sets required to be supported and enabled <sup>d</sup>				
Required that the logical sector address is < 2 <sup>28</sup> <sup>b</sup>	48-bit Address <sup>b</sup>	DMA <sup>c</sup>	Overlap	SATA 2.5 NCQ	
no	no	no	no	no	FLUSH CACHE
yes <sup>e</sup>	no	no	no	no	READ MULTIPLE READ SECTOR(S) READ VERIFY SECTOR(S) WRITE MULTIPLE WRITE SECTOR(S)
yes <sup>e</sup>	no	yes	no	no	READ DMA WRITE DMA
yes <sup>e</sup>	no	yes	yes	no	READ DMA QUEUED WRITE DMA QUEUED
no	yes	yes	no	no	FLUSH CACHE EXT READ DMA EXT WRITE DMA EXT WRITE DMA FUA EXT
no	yes	yes	yes	n/a	READ DMA QUEUED EXT WRITE DMA QUEUED EXT WRITE DMA QUEUED FUA EXT
no	yes	no	no	no	READ MULTIPLE EXT READ SECTOR(S) EXT READ VERIFY SECTOR(S) EXT WRITE MULTIPLE EXT WRITE MULTIPLE FUA EXT WRITE SECTOR(S) EXT
no	no	no	no	yes	READ FPDMA QUEUED WRITE FPDMA QUEUED

<sup>a</sup> If the SATL implements the direct mapping model (see 9.1.2) between ATA logical sectors and SCSI logical blocks, then this represents the last logical block transferred. If the SATL implements the indirect logical block mapping model, then this constraint is vendor-specific.

<sup>b</sup> If the ATA device supports neither the 48-bit Address feature set (i.e., ATA IDENTIFY DEVICE data word 83 bit 10 is set to zero) nor NCQ (see SATA 2.5) and the LBA of the logical sector is greater than (2<sup>28</sup>-1), the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

<sup>c</sup> The DMA prerequisite requires both the ATA host in the SATL and the ATA device to have the same DMA transfer mode enabled (i.e., ATA IDENTIFY DEVICE data word 49 bit 8 is set to one and at least one DMA mode is enabled in the ATA IDENTIFY DEVICE data word 63 or word 88).

<sup>d</sup> See ATA8-ACS.

<sup>e</sup> The SATL may transfer the number of logical blocks requested in the TRANSFER LENGTH field of the SCSI CDB by sending multiple ATA commands, each time incrementing the ATA LBA by the ATA Sector Count transferred.

The SATL may use the ATA commands listed in table 28 in the translation of SCSI read commands (see 3.1.77), SCSI write commands (see 3.1.81), SCSI write and verify commands (see 3.1.82), SCSI verify commands (see 3.1.80), and SCSI synchronize cache commands (see 3.1.78) if the prerequisites defined for the command as shown in table 28 are satisfied. The translations for specific SCSI block commands in clause 9 further constrain the use of the available ATA commands in implementing the translation.

The SATL emulation of the READ (6) command and the WRITE (6) command in which the TRANSFER LENGTH field is set to zero, shall translate the transfer length to 256, and issue ATA commands that operate on the ATA logical sectors corresponding to the specified 256 SCSI logical blocks.

In all other cases, the SATL shall transfer or operate on the ATA logical sectors corresponding to the number of logical blocks specified by the SCSI command.

## 9.2 FORMAT UNIT command

### 9.2.1 FORMAT UNIT command overview

The FORMAT UNIT command verifies that all logical block addresses accessible to SCSI application clients are formatted and ready for data transfers. Table 29 shows the translation for fields specified in the FORMAT UNIT CDB.

**Table 29 — FORMAT UNIT CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 04h. If no defect list header is provided or a defect list header is provided with the DCRT bit set to one, then the SATL shall return GOOD status without issuing any commands to the ATA device. If the SATL supports certification of media and a defect list header is provided with the DCRT bit set to zero, then the SATL shall certify the media as described in 9.2.5
DEFECT LIST FORMAT	If the DEFECT LIST FORMAT field is the mandatory format (000b) or the vendor specific format (110b) the defect list length shall be zero (see SBC-2). If the DEFECT LIST FORMAT field is any other value the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
CMPLIST	If a complete list is specified (i.e., the FMTDATA bit is set to one, and the CMPLIST bit is set to one) then the SATL shall terminate the command with a CHECK CONDITION status with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB.
FMTDATA	If set to zero no data shall be transferred from the application client. If set to one the FORMAT UNIT parameter list shall be transferred from the application client. The SATL may accept a FORMAT UNIT parameter list (see 9.2.2). The SATL shall ignore any defect list descriptors and any other fields provided in the FORMAT UNIT parameter list.
LONGLIST	Unspecified (see 3.4.2)
FMTPINFO	Unspecified (see 3.4.2)
RTO_REQ	If the RTO_REQ bit is set to one and the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall terminate the command with CHECK CONDITION status with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB. If the SATL implements indirect logical block mapping, the translation for this field is unspecified (see 3.4.2).
CONTROL	6.4

The SATL shall process commands received during the processing of the FORMAT UNIT command as specified in SBC-2.

### 9.2.2 FORMAT UNIT parameter list

If the FORMAT command CDB specifies a FMTDATA bit of one, the SATL shall accept a FORMAT UNIT parameter list consisting of a short or long defect list header and may accept an initialization pattern descriptor. The SATL shall ignore any defect descriptors provided. Table 30 defines the SATL handling of fields in the FORMAT UNIT defect list header.

**Table 30 — SATL defect list header**

Field	Description or reference
FOV	9.2.3 and 9.2.4
DPRY	The SATL shall ignore this field.
DCRT	9.2.3 and 9.2.5
STPF	Unspecified (see 3.4.2)
IP	9.2.3 and 9.2.6
IMMED	9.2.3
DEFECT LIST LENGTH	The SATL shall ignore any defect descriptors provided.

### 9.2.3 SATL defect list header field combinations

Table 31 describes the actions the SATL takes depending on the values set in the IMMED bit, the FOV bit, the DCRT bit, and the IP bit.

**Table 31 — SATL defect list header field combinations**

IMMED	FOV	DCRT	IP	Description of SATL processing
1	n/a	n/a	n/a	The SATL may complete the FORMAT UNIT command immediately with GOOD status.
n/a	0	n/a	n/a	
n/a	1	1	0	
0	1	0	0	If the SATL does not support media certification, then the SATL may terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD PARAMETER LIST. Otherwise, the SATL shall issue the required ATA read commands and ATA write commands to certify and initialize the media as specified by DCRT bit and IP bit, and shall then return GOOD status if no unrecoverable write errors occur.
		0	1	
		1	1	

### 9.2.4 FOV bit

The FOV bit may be set to one to include an initialization pattern descriptor and no defect descriptors, otherwise the SATL may terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD PARAMETER LIST.

### 9.2.5 DCRT bit

If the DCRT bit is set to zero and media certification is supported by the SATL, then the SATL shall issue ATA verify commands (see 3.1.23) to access all the logical sectors on the medium of the ATA device that ~~comprise every logical block emulated by the SATL~~ [the SATL uses to emulate logical blocks accessible by the application client](#). For every unrecoverable read error that is encountered, the SATL shall issue an ATA write command (see 3.1.25) to the defective logical sector to attempt to cause logical sector reallocation. The data written shall be the data pattern specified by the initialization pattern descriptor, if any, or vendor-specific. After writing the affected logical sector, the SATL shall again issue an ATA verify command to the same logical sector to verify the alternate logical sector is not defective. The process (e.g., verify, write, verify, write, etc.)

shall repeat until the logical sector is verified successfully or the disk returns a fatal error other than an unrecoverable read error (e.g., device fault). See 5.3 for a description of error handling for multiple ATA command sequences.

### 9.2.6 IP bit

If the SATL supports an IP bit value of one and the IP bit is set to one, the SATL shall process the command as follows:

- a) If the ATA device supports the SCT LBA Segment Access (see SCT) command and the value of the INITIALIZATION PATTERN LENGTH field in the initialization pattern descriptor is four, and the value of the IP MODIFIER field in the initialization pattern descriptor is zero, then the SATL should issue an SCT LBA Segment Access command to the ATA device with the Function Code field set to 0001b (i.e., Repeat Write Pattern), with the Start field and the Count field set to initialize the area of the media accessible by the application client, and with the Pattern field set to the value of the INITIALIZATION PATTERN field from the FORMAT command initialization pattern descriptor; and
- b) if the SCT LBA Segment Access command is not used to write the initialization pattern, then the SATL shall write the specified pattern by issuing ATA write commands (see 3.1.25 and 9.1) to the ATA device.

If the IP bit is set to zero, then the SATL shall return GOOD status.

NOTE 11 - The SATL should reverse the order of the bytes between the Pattern field in the SCT LBA Segment Access command and the value stored in the INITIALIZATION PATTERN field in the FORMAT command initialization pattern descriptor to adjust for the translation from little-endian to big-endian byte ordering.

## 9.3 READ commands overview

### 9.3.1 READ commands operation code translation

This subclause applies to the translation of READ(6), READ(10), READ(12), and READ(16) commands.

The SATL shall issue ATA read commands (see 3.1.20) in accordance with the constraints specified in 9.1 to cause the ATA device to transfer the logical blocks specified in the SCSI read command (see 3.1.77).

If the SATL returns a CHECK CONDITION status with a sense key set to a value other than ILLEGAL REQUEST while processing the command, the SATL may transfer a vendor-specific amount of data before terminating the command.

### 9.3.2 READ commands with FUA

If the SATL does not support FUA and the FUA bit is set to one, the SATL shall terminate the READ (10), READ (12) or READ (16) command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

The SATL shall process a SCSI read command with the FUA bit set to one as follows:

- a) If the ATA device supports NCQ (i.e., ATA IDENTIFY DEVICE data word 76 bit 8 is set to one) the SATL shall issue a READ FPDMA QUEUED command (see SATA 2.5) with the FUA bit in the Device register set to one;
- b) If the ATA device supports the TCQ (see ATA8-ACS) and there are outstanding ATA queued commands, then the SATL shall:
  - 1) wait until all ATA queued commands have completed;
  - 2) if the write cache is enabled (ATA8-ACS) on the ATA device, issue an ATA verify command (see 3.1.23); and,
  - 3) issue an ATA read command as specified in 9.3.1;
 or
- c) If the ATA device supports neither NCQ nor TCQ, or there are no outstanding ATA queued commands, then the SATL shall:
  - 1) if the write cache is enabled on the ATA device, issue an ATA verify command (see 3.1.23); and,
  - 2) issue an ATA read command as specified in 9.3.1.

## 9.4 READ (6) command

The READ (6) command is used to request the device to transfer logical blocks of user data to the application client (see SBC-2). Table 32 shows the translation for fields specified in the READ (6) CDB.

**Table 32 — READ (6) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 08h. See 9.3.1
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA read command (see 3.1.20) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
TRANSFER LENGTH <sup>a</sup>	The transfer length shall be used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count in the ATA read command (see 3.1.20) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2).
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies to transfer 256 logical blocks to the application client (see SBC-2).	

## 9.5 READ (10) command

The SATL shall process the READ (10) command the same as the READ (6) command (see 9.3.1). with the

additional fields in the CDB implemented as described in the table 33 and 9.3.2.

**Table 33 — READ (10) CDB field translations**

<b>Field</b>	<b>Description or reference</b>
OPERATION CODE	Set to 28h. See 9.3.1 and 9.3.2.
RDPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
FUA	9.3.2
FUA_NV	The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2. Note 1 - Some application clients may expect the device server to return CHECK CONDITION status if the FUA-NV bit is set to one and the Extended INQUIRY Data VPD page is not supported.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA read command (see 3.1.20) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH <sup>a</sup>	The transfer length is used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count in the ATA read command (see 3.1.20) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall issue as many ATA read commands as needed to satisfy the transfer length specified by the READ (10) command.
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies that a data transfer shall not take place.	

## 9.6 READ (12) command

The SATL shall process the READ (12) command the same as the READ (10) command (see 9.3.1), with the fields in the CDB implemented as described in table 34 and 9.3.2.

**Table 34 — READ (12) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to A8h. See 9.3.1 and 9.3.2.
RDPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
FUA	9.3.2
FUA_NV	The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2. Note 1 - Some application clients may expect the device server to return CHECK CONDITION status if the FUA-NV bit is set to one and the Extended INQUIRY Data VPD page is not supported.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA read command (see 3.1.20) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH <sup>a</sup>	The transfer length is used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count in the ATA read command (see 3.1.20) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall issue as many ATA read commands as needed to satisfy the transfer length specified by the READ (12) command.
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies that a data transfer shall not take place.	

### 9.7 READ (16) command

The SATL shall process the READ (16) command the same as the READ (10) command (see 9.3.1), with the fields in the CDB implemented as described in table 35 and 9.3.2.

**Table 35 — READ (16) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 88h. See 9.3.1 and 9.3.2.
RDPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
FUA	9.3.2
FUA_NV	The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2. Note 1 - Some application clients may expect the device server to return CHECK CONDITION status if the FUA-NV bit is set to one and the Extended INQUIRY Data VPD page is not supported.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA read command (see 3.1.20) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH <sup>a</sup>	The transfer length is used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count in the ATA read command (see 3.1.20) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall issue as many ATA read commands as needed to satisfy the transfer length specified by the READ (16) command.
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies that a data transfer shall not take place.	

## 9.8 READ CAPACITY (10) command

### 9.8.1 READ CAPACITY (10) command overview

The READ CAPACITY (10) command (see SBC-2) requests that the device server transfer eight bytes of parameter data describing the capacity and medium format of the direct-access block device to the application client. Table 36 shows the translation for fields specified in the READ CAPACITY (10) CDB.

**Table 36 — READ CAPACITY (10) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 25h. The SATL shall use ATA IDENTIFY DEVICE data to compute the ATA device's maximum user addressable medium capacity of the ATA device.
LOGICAL BLOCK ADDRESS	If the LOGICAL BLOCK ADDRESS field is not set to zero the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
PMI	If the PMI bit is not set to zero the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
CONTROL	6.4

### 9.8.2 READ CAPACITY (10) parameter data

The SATL shall return READ CAPACITY (10) parameter data as defined by SBC-2. Table 37 describes the translation of fields in the READ CAPACITY (10) parameter data.

**Table 37 — READ CAPACITY (10) parameter data**

Field	Description or reference
RETURNED LOGICAL BLOCK ADDRESS <sup>a</sup>	If the number of logical blocks exceeds the maximum value that is able to be specified in the RETURNED LOGICAL BLOCK ADDRESS field, the device server shall set the RETURNED LOGICAL BLOCK ADDRESS field to FFFFFFFFh. The application client should then issue a READ CAPACITY (16) command (see 9.9) to retrieve the READ CAPACITY (16) parameter data. Otherwise, this field is unspecified (see 3.4.2).
BLOCK LENGTH IN BYTES <sup>a</sup>	If the SATL implements direct logical block mapping (see 3.1.33) then the BLOCK LENGTH IN BYTES field shall contain the ATA logical sector size (see 3.1.15). Otherwise the BLOCK LENGTH IN BYTES field is unspecified (see 3.4.2).
<sup>a</sup> The values reported in the RETURNED LOGICAL BLOCK ADDRESS field and the BLOCK LENGTH IN BYTES field shall be such that the logical unit capacity (see 3.1.49) is less than or equal to the ATA device capacity (see 3.1.9).	

## 9.9 READ CAPACITY (16) command

### 9.9.1 READ CAPACITY (16) command overview

The READ CAPACITY (16) command (see SBC-2) requests that the device server transfer eight bytes of parameter data describing the capacity and medium format of the direct-access block device to the application client. Table 38 shows the translation for fields specified in the READ CAPACITY (16) CDB.

**Table 38 — READ CAPACITY(16) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 9E/10h. The SATL shall use ATA IDENTIFY DEVICE data to compute the maximum user addressable medium capacity of the ATA device.
LOGICAL BLOCK ADDRESS	If the LOGICAL BLOCK ADDRESS field is not set to zero the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
ALLOCATION LENGTH	Unspecified (see 3.4.2)
PMI	If the PMI bit is not set to zero the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
CONTROL	6.4

### 9.9.2 READ CAPACITY (16) parameter data

The SATL shall return READ CAPACITY (16) parameter data as defined by SBC-2. Table 39 describes the translation of fields in the READ CAPACITY (16) parameter data.

**Table 39 — READ CAPACITY (16) parameter data**

Field	Description or reference
RETURNED LOGICAL BLOCK ADDRESS <sup>a</sup>	The maximum value that shall be returned in the RETURNED LOGICAL BLOCK ADDRESS field is FFFFFFFF_FFFFFFFEh. Otherwise, this field is unspecified (see 3.4.2).
BLOCK LENGTH IN BYTES <sup>a</sup>	If the SATL implements direct logical block mapping (see 3.1.33) then the BLOCK LENGTH IN BYTES field shall contain the ATA logical sector size (see 3.1.15). Otherwise the BLOCK LENGTH IN BYTES field is unspecified (see 3.4.2).
RTO_EN	Unspecified (see 3.4.2)
PROT_EN	Unspecified (see 3.4.2)
<sup>a</sup> The values reported in the RETURNED LOGICAL BLOCK ADDRESS field and the BLOCK LENGTH IN BYTES field shall be such that the logical unit capacity (see 3.1.49) is less than or equal to the ATA device capacity (see 3.1.9).	

## 9.10 REASSIGN BLOCKS command

### 9.10.1 REASSIGN BLOCKS command overview

The REASSIGN BLOCKS command requests that the SATL reassign defective logical blocks (see SBC-2). ATA devices do not support or have a direct translation for the REASSIGN BLOCKS command. The SATL shall emulate the REASSIGN BLOCKS command as defined in table 40.

**Table 40 — REASSIGN BLOCKS CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 07h. See 9.10.2.
LONGLBA	See SBC-2
LONGLIST	See SBC-2
CONTROL	6.4

The REASSIGN BLOCKS command parameter list transferred from the application client contains the LBAs of logical blocks to be reassigned.

If the SATL implements direct logical block mapping (see 9.1.2), then the values set by the SATL in the ATA LBA of the ATA verify command(s) and ATA write command(s) shall equal the value(s) of the LBAs in the parameter list. Otherwise, the mapping is unspecified (see 3.4.2).

The SATL shall support the LONGLBA bit and the LONGLIST bit (see SBC-2).

### 9.10.2 REASSIGN BLOCKS operation code

The SATL shall accept a parameter list specifying LBAs of logical blocks to be reassigned (see SBC-2).

The SATL shall process each ATA LBA corresponding to LBAs specified in the parameter list as shown in figure 8.

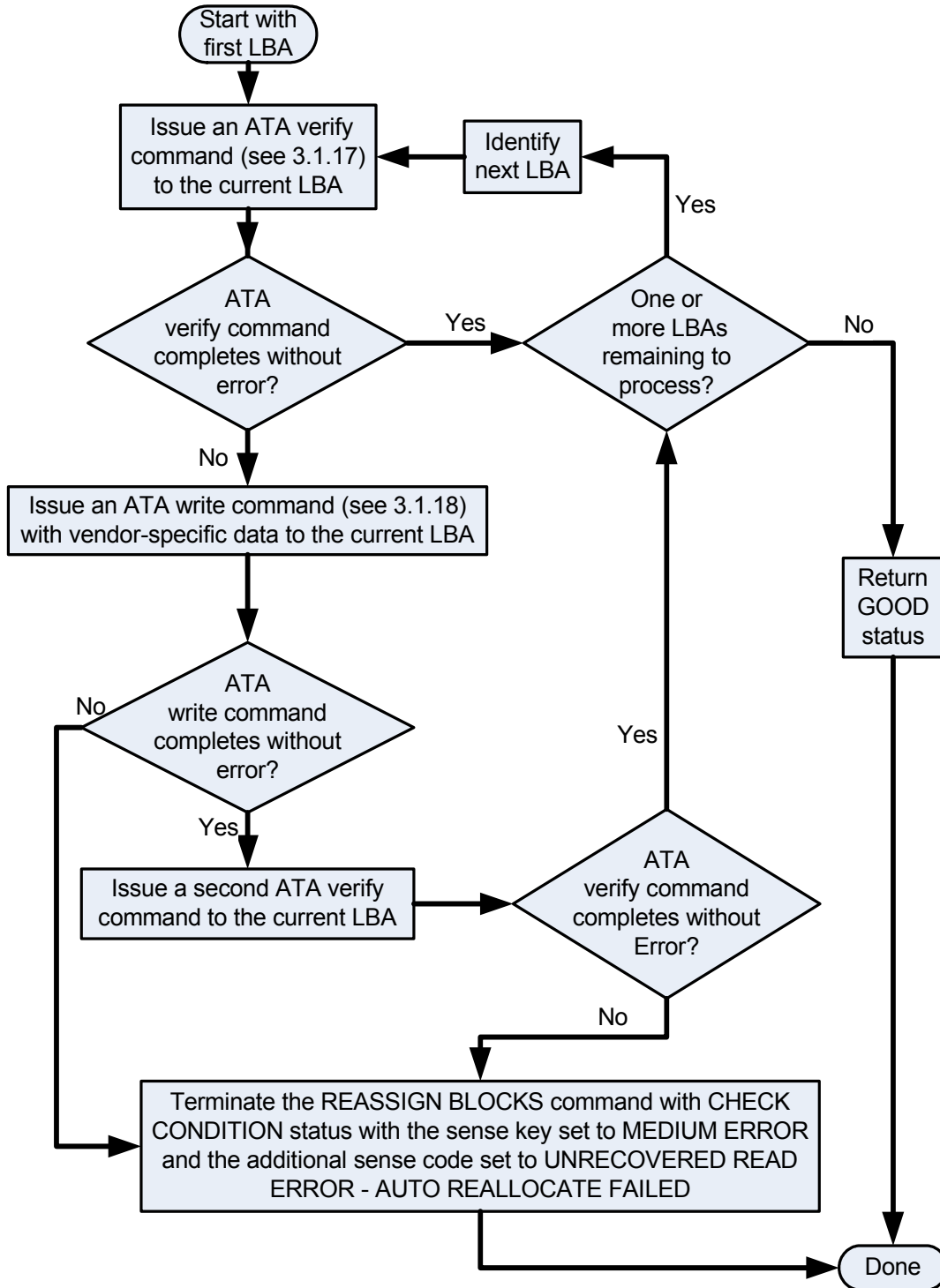


Figure 8 — REASSIGN BLOCKS command translation

## 9.11 START STOP UNIT command

### 9.11.1 START STOP UNIT command overview

The START STOP UNIT command provides a method for controlling the power state of a logical unit.

If a SATL receives a command that requires medium access while the device is in the Stopped state (see SBC-2), then the SATL shall return CHECK CONDITION status, with the sense key set to NOT READY and the additional sense code set to LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED.

Table 41 shows the translation for fields specified in the START STOP UNIT CDB.

**Table 41 — START/STOP UNIT CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 1Bh. See 9.11.2 and 9.11.3.
IMMED	The SATL shall implement this field as defined in 9.11.2 and 9.11.3.
POWER CONDITION	If the POWER CONDITION field is not set to zero, the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
LOEJ	The SATL shall implement this field as defined in 9.11.3.
START	The SATL shall implement this field as defined in 9.11.3.
CONTROL	6.4

### 9.11.2 Processing ending status if an error occurs

If an error occurs during the processing of the START STOP UNIT command and the IMMED bit is set to zero, then the SATL shall terminate the START STOP UNIT command with CHECK CONDITION status with a sense key set to ABORTED COMMAND, and the additional sense code set to the value specified for the error being reported (see table 42).

If an error occurs during the processing of the START STOP UNIT command and the IMMED bit is set to one, then the SATL shall terminate the START STOP UNIT command and return CHECK CONDITION status as a deferred error (see SPC-3) with a sense key set to ABORTED COMMAND, and the additional sense code set to the value specified for the error being reported (see table 42).

### 9.11.3 START STOP UNIT START bit LOEJ bit combinations

The SATL shall perform the actions shown in table 42 in response to a START STOP UNIT command.

**Table 42 — Definition of START and LOEJ bits in the START STOP UNIT CDB**

START	LOEJ	Definition
0	0	<p>The SATL shall:</p> <ol style="list-style-type: none"> <li>1) If the IMMED bit is set to one, then return GOOD status;</li> <li>2) Issue an ATA flush command (see 3.1.11) to the ATA device;</li> <li>3) If the ATA flush command completes with any error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR;</li> <li>4) If the ATA flush command completes without error, then issue an ATA STANDBY IMMEDIATE command to the ATA Sector Count set to zero;</li> <li>5) If the ATA STANDBY IMMEDIATE command completes with any error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and</li> <li>6) If the ATA STANDBY IMMEDIATE command completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2) <sup>a</sup>.</li> </ol>
0	1	<p>If the ATA device supports the Removable Media feature set (see ATA/ATAPI-7), then the SATL shall:</p> <ol style="list-style-type: none"> <li>1) If the IMMED bit is set to one, then return GOOD status;</li> <li>2) Issue an ATA MEDIA EJECT command to the ATA device;</li> <li>3) If the ATA MEDIA EJECT command completes with any error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to MEDIA LOAD OR EJECT FAILED; and</li> <li>4) If the MEDIA EJECT command completes without error and the IMMED bit is set to zero, then return GOOD status.</li> </ol> <p>If the ATA device does not support the Removable Media feature set, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</p>
1	0	<p>The SATL shall:</p> <ol style="list-style-type: none"> <li>1) If the IMMED bit is set to one, then return GOOD status;</li> <li>2) Issue an ATA verify command (see 3.1.23) to the ATA Sector Count set to one and the LBA set to a value between zero and the maximum LBA supported by the ATA device in its current configuration <sup>b</sup>; and</li> <li>3) If the IMMED bit is set to one, then return GOOD status when command completion is received for the ATA verify command (see 3.1.23) <sup>c</sup>.</li> </ol>
1	1	<p>The SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, with the additional sense code set to INVALID FIELD IN CDB.</p>
<p><sup>a</sup> After returning GOOD status for a START STOP UNIT command with the START bit set to zero, the SATL shall consider the ATA device to be in the Stopped power state (see SBC-2).</p> <p><sup>b</sup> An ATA device medium access occurs when an LBA is specified whose data is not contained in ATA device's cache memory. If a value in LBA is specified for an ATA verify command where the data is contained in ATA device's cache memory, then an ATA device may not be in the Active power mode (see ATA8-ACS) after completion of the ATA verify command.</p> <p><sup>c</sup> After returning GOOD status for a START STOP UNIT command with the START bit set to one, the SATL shall consider the ATA device to be in the Active power state (see SBC-2).</p>		

## 9.12 SYNCHRONIZE CACHE (10) command

The SYNCHRONIZE CACHE (10) command is used to flush the most recent data in the cache of the ATA device to physical medium.

Table 43 shows the translation for fields specified in the SYNCHRONIZE CACHE (10) CDB.

**Table 43 — SYNCHRONIZE CACHE (10) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 35h. The SATL shall issue an ATA flush command (see 3.1.11) in accordance with the constraints described in 9.1.
SYNC_NV	Unspecified (see 3.4.2)
IMMED	If the IMMED bit is set to one the SATL shall return GOOD status and then issue an ATA flush command. If the IMMED bit is set to zero the SATL shall issue an ATA flush command and return status upon completion.
LOGICAL BLOCK ADDRESS	The SATL shall ignore this field and shall process this command as though this field contained zero.
GROUP NUMBER	Unspecified (see 3.4.2)
NUMBER OF BLOCKS	The SATL shall ignore this field and shall process this command as though this field contained zero (see SBC-2).
CONTROL	6.4

## 9.13 SYNCHRONIZE CACHE (16) command

The SYNCHRONIZE CACHE (16) command is used to flush the most recent data in the cache of the ATA device to physical medium.

Table 44 shows the translation for fields specified in the SYNCHRONIZE CACHE (16) CDB.

**Table 44 — SYNCHRONIZE CACHE (16) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 91h. The SATL shall issue an ATA flush command (see 3.1.11) in accordance with the constraints described in 9.1.
SYNC_NV	Unspecified (see 3.4.2)
IMMED	If the IMMED bit is set to one the SATL shall return GOOD status and then issue an ATA flush command. If the IMMED bit is set to zero the SATL shall issue an ATA flush command and return status upon completion.
LOGICAL BLOCK ADDRESS	The SATL shall ignore this field and shall process this command as though this field contained zero.
GROUP NUMBER	Unspecified (see 3.4.2)
NUMBER OF BLOCKS	The SATL shall ignore this field and shall process this command as though this field contained zero (see SBC-2).
CONTROL	6.4

## 9.14 VERIFY (10) command

The VERIFY (10) command is used to verify data on the ATA device's medium. Table 45 describes the translation of fields in the VERIFY (10) CDB.

**Table 45 — VERIFY (10) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 2Fh. The SATL shall issue an ATA verify command (see 3.1.23) in accordance with the constraints defined in 9.1.
VRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
BYTCHK	If the SATL supports a BYTCHK bit set to one and if the BYTCHK bit is set to one, then the SATL shall perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device by the SATL, and return completion status reflecting the results of the comparison as described in (see SBC-2).
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA verify command (see 3.1.23) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
VERIFICATION LENGTH	The verification length shall be used to set the ATA Sector Count as defined in 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count (see 3.1.16) in the ATA verify command (see 3.1.23) equal to the value specified in the VERIFICATION LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2).
CONTROL	6.4

## 9.15 VERIFY (12) command

Table 46 describes the translation of fields in the VERIFY (12) CDB.

**Table 46 — VERIFY (12) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to AFh. The SATL shall issue an ATA verify command (see 3.1.23) in accordance with the constraints defined in 9.1.
VRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
BYTCHK	If the SATL supports a BYTCHK bit set to one and if the BYTCHK bit is set to one, then the SATL shall perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device by the SATL, and return completion status reflecting the results of the comparison as described in (see SBC-2).
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA verify command (see 3.1.23) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
VERIFICATION LENGTH	The verification length shall be used to set the ATA Sector Count as defined in 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count (see 3.1.16) in the ATA verify command (see 3.1.23) equal to the value specified in the VERIFICATION LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2).
CONTROL	6.4

## 9.16 VERIFY (16) command

Table 47 describes the translation of fields in the VERIFY (16) CDB.

**Table 47 — VERIFY (16) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 8Fh. The SATL shall issue an ATA verify command (see 3.1.23) in accordance with the constraints defined in 9.1.
VRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
BYTCHK	If the SATL supports a BYTCHK bit set to one and if the BYTCHK bit is set to one, then the SATL shall perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device by the SATL, and return completion status reflecting the results of the comparison as described in (see SBC-2).
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA verify command (see 3.1.23) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
VERIFICATION LENGTH	The verification length shall be used to set the ATA Sector Count as defined in 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count (see 3.1.16) in the ATA verify command (see 3.1.23) equal to the value specified in the VERIFICATION LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2).
CONTROL	6.4

## 9.17 WRITE commands overview

### 9.17.1 WRITE commands operation code translation

This subclause applies to the translation of SCSI WRITE(6), WRITE(10), WRITE(12), and WRITE(16) commands.

The SATL shall transfer the logical blocks specified in the SCSI write command (see 3.1.81) from the SCSI application client to the ATA device. The SATL shall issue ATA write commands (see 3.1.25) in accordance with the constraints specified in 9.1.

### 9.17.2 WRITE commands with FUA

This subclause applies to the translation of WRITE (10), WRITE (12), and WRITE (16) commands.

If the FUA bit is set to zero in the SCSI write command CDB, then the SATL shall process this command as described in 9.17.1.

If the FUA bit is set to one in the SCSI write command CDB, then the SATL shall issue the following, in accordance with the constraints described in 9.1:

- a) two ATA commands as follows:
  - 1) an ATA write command (see 3.1.25) excluding WRITE DMA FUA EXT, WRITE DMA QUEUED FUA EXT, WRITE MULTIPLE FUA EXT, and WRITE FPDMA QUEUE; and
  - 2) an ATA verify command (see 3.1.23);
- b) one of the following ATA commands (see ATA8-ACS):
  - A) WRITE DMA FUA EXT;
  - B) WRITE DMA QUEUED FUA EXT; or
  - C) WRITE MULTIPLE FUA EXT;or
- c) an ATA WRITE FPDMA QUEUED command (see SATA 2.5) with the FUA bit in the Device field set to one.

See 5.3 for a description of multiple command sequence error handling.

## 9.18 WRITE (6) command

The WRITE (6) command is used to request the SATL to transfer user data from the application client to the ATA device. Data may be written to the medium or to the cache of the ATA device.

Table 48 shows the translation of fields specified in the WRITE (6) CDB.

**Table 48 — WRITE (6) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 0Ah. See 9.17.1.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.25) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
TRANSFER LENGTH <sup>a</sup>	The transfer length shall be used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count (see 3.1.21) in the ATA write command (see 3.1.25) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2).
CONTROL	6.4
<sup>a</sup> A TRANSFER LENGTH field set to zero specifies a transfer of 256 logical blocks (see SBC-2).	

## 9.19 WRITE (10) command

The WRITE (10) command is used to request the SATL to transfer user data from the application client to the ATA device. Data may be written to the medium or to the cache of the ATA device.

Table 49 shows the translation of fields specified in the WRITE (10) CDB.

**Table 49 — WRITE (10) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 2Ah. See 9.17.1 and 9.17.2.
WRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
FUA	9.17.2
FUA_NV	The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2. Note 1 - Some application clients may expect the device server to return CHECK CONDITION status if the FUA-NV bit is set to one and the Extended INQUIRY Data VPD page is not supported.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.25) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH <sup>a</sup>	The transfer length is used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count in the ATA write command (see 3.1.25) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall issue as many ATA write commands (see 3.1.25) as needed to satisfy the transfer length specified by the WRITE (10) command.
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies that a data transfer shall not take place.	

## 9.20 WRITE (12) command

The WRITE (12) command is used to request the SATL to transfer user data from the application client to the ATA device. Data may be written to the medium or to the cache of the ATA device.

Table 50 shows the translation of fields specified in the WRITE (12) CDB.

**Table 50 — WRITE (12) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to AAh. See 9.17.1 and 9.17.2.
WRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
FUA	9.17.2
FUA_NV	The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2. Note 1 - Some application clients may expect the device server to return CHECK CONDITION status if the FUA-NV bit is set to one and the Extended INQUIRY Data VPD page is not supported.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.25) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH <sup>a</sup>	The transfer length is used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count in the ATA write command (see 3.1.25) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall issue as many ATA write commands (see 3.1.25) as needed to satisfy the transfer length specified by the WRITE (12) command.
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies that a data transfer shall not take place.	

## 9.21 WRITE (16) command

The WRITE (16) command is used to request the SATL to transfer user data from the application client to the ATA device. Data may be written to the medium or to the cache of the ATA device.

Table 51 shows the translation of fields specified in the WRITE (16) CDB.

**Table 51 — WRITE (16) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 8Ah. See 9.17.1 and 9.17.2.
WRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
FUA	9.17.2
FUA_NV	The SATL may ignore the FUA_NV bit, or the SATL may implement the FUA_NV bit as defined in SBC-2. Note 1 - Some application clients may expect the device server to return CHECK CONDITION status if the FUA-NV bit is set to one and the Extended INQUIRY Data VPD page is not supported.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.25) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH <sup>a</sup>	The transfer length is used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count in the ATA write command (see 3.1.25) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall issue as many ATA write commands (see 3.1.25) as needed to satisfy the transfer length specified by the WRITE (16) command.
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies that a data transfer shall not take place.	

## 9.22 WRITE AND VERIFY commands overview

This subclause applies to the translation of WRITE AND VERIFY (10), WRITE AND VERIFY (12), and WRITE AND VERIFY (16) commands.

The SATL shall issue:

- 1) an ATA write command (see 3.1.25) in accordance with the constraints defined in 9.1; and
- 2) an ATA verify command (see 3.1.23).

## 9.23 WRITE AND VERIFY (10) command

The WRITE AND VERIFY (10) command requests that the SATL to transfer the specified logical blocks from the application client to the ATA device, and then verify that the data was written correctly to the medium of the ATA device.

Table 52 shows the translation of fields specified in the WRITE AND VERIFY (10) CDB.

**Table 52 — WRITE AND VERIFY (10) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 2Eh. See 9.22.
WRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
BYTCHK	If the SATL supports a BYTCHK bit set to one and the BYTCHK bit is set to one, then after writing the data to the medium the SATL shall read the data from the medium and perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device, and then shall return completion status reflecting the results of the comparison (see SBC-2).
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.25) and the ATA verify command (see 3.1.23) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH <sup>a</sup>	The transfer length is used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count in the ATA write command (see 3.1.25) and the ATA verify command (see 3.1.23) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall issue as many ATA write commands and ATA verify commands as needed to satisfy the transfer length specified by the WRITE AND VERIFY (10) command.
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies that a data transfer shall not take place.	

## 9.24 WRITE AND VERIFY (12) command

The WRITE AND VERIFY (12) command requests that the SATL to transfer the specified logical blocks from the application client to the ATA device, and then verify that the data was written correctly to the medium of the ATA device.

Table 53 shows the translation of fields specified in the WRITE AND VERIFY (12) CDB.

**Table 53 — WRITE AND VERIFY (12) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to AEh. See 9.22.
WRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
BYTCHK	If the SATL supports a BYTCHK bit set to one and the BYTCHK bit is set to one, then after writing the data to the medium the SATL shall read the data from the medium and perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device, and then shall return completion status reflecting the results of the comparison (see SBC-2).
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.25) and the ATA verify command (see 3.1.23) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH <sup>a</sup>	The transfer length is used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA Sector Count in the ATA write command (see 3.1.25) and the ATA verify command (see 3.1.23) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall issue as many ATA write commands and ATA verify commands as needed to satisfy the transfer length specified by the WRITE AND VERIFY (12) command.
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies that a data transfer shall not take place.	

## 9.25 WRITE AND VERIFY (16) command

The WRITE AND VERIFY (16) command requests that the SATL to transfer the specified logical blocks from the application client to the ATA device, and then verify that the data was written correctly to the medium of the ATA device.

Table 54 shows the translation of fields specified in the WRITE AND VERIFY (16) CDB.

**Table 54 — WRITE AND VERIFY (16) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 8Eh. See 9.22.
WRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
BYTCHK	If the SATL supports a BYTCHK bit set to one and the BYTCHK bit is set to one, then after writing the data to the medium the SATL shall read the data from the medium and perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device, and then shall return completion status reflecting the results of the comparison (see SBC-2).
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.25) and the ATA verify command (see 3.1.23) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH <sup>a</sup>	The transfer length is used to set the ATA Sector Count (see 3.1.21), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then direct logical block mapping the SATL shall set the ATA Sector Count in the ATA write command (see 3.1.25) and the ATA verify command (see 3.1.23) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall issue as many ATA write commands and ATA verify commands as needed to satisfy the transfer length specified by the WRITE AND VERIFY (16) command.
CONTROL	6.4
<sup>a</sup> A transfer length of zero specifies that a data transfer shall not take place.	

## 9.26 WRITE SAME (10) command

### 9.26.1 WRITE SAME (10) command overview

The WRITE SAME (10) command (see table 55) requests that the SATL transfer a single logical block from the application client and write the contents of that single logical block, with modifications based on the LBDATA bit and the PBDATA bit, to the specified range of logical block addresses on the ATA device.

**Table 55 — WRITE SAME (10) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 41h. If the ATA device supports the ATA SCT Write Same command (see ATA8-ACS), the SATL should issue the ATA SCT Write Same command to repeatedly write the data block transferred from the application client to the ATA device. If the ATA device does not implement ATA SCT Write Same command then the SATL shall issue ATA write commands as defined in 9.17.1.
WRPROTECT	Unspecified (see 3.4.2)
LBDATA	9.26.2
PBDATA	9.26.2
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the Start in the ATA SCT Write Same command or the ATA LBA in the ATA write command (see 3.1.25) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
NUMBER OF BLOCKS	A NUMBER OF BLOCKS field set to zero specifies that the SATL shall repeatedly write the logical block transferred from the application client to the range of ATA logical sectors corresponding to the range of logical block addresses specified by the LOGICAL BLOCK ADDRESS field through the LBA of the last logical block on the logical unit. If the NUMBER OF BLOCKS field is set to a value other than zero, the SATL shall repeatedly write the data block transferred from the application client to the medium of the ATA device for the number of logical blocks specified to the corresponding logical sectors on the ATA device. The SATL shall send as many ATA commands as required to satisfy the number of blocks specified by the WRITE SAME (10) command.
CONTROL	6.4

### 9.26.2 LBDATA bit and PBDATA bit

The SATL shall write data to the specified logical block addresses according to the values in the LBDATA and PBDATA bits as shown in table 56.

**Table 56 — LBDATA and PBDATA fields**

LBDATA	PBDATA	Description
0	0	The SATL shall write the block of data transferred from the application client to the range of blocks specified in LOGICAL BLOCK ADDRESS field and NUMBER OF BLOCKS field, repeatedly, on the medium of the ATA device. If the ATA device supports the ATA SCT Write Same command, then the SATL should use the ATA SCT Write Same command with the Function Code set to 002h or 004h for writing the data. Otherwise, the SATL shall use ATA write commands as defined in 9.17.2 (see SBC-2).
1	0	The SATL shall replace the first four bytes of the logical block received from the application client with the least significant four bytes of the LBA of the logical block being written to the media, ending with the least significant byte (e.g., if the LBA is 77665544_33221100h, 33221100h is written with 33h written first and 00h written last). The SATL shall use ATA write commands as defined in 9.17.2 (see SBC-2).
0	1	The SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
1	1	

## 9.27 WRITE SAME (16) command

The WRITE SAME (16) command (see table 57) requests that the SATL transfer a single logical block from the application client and write the contents of that single logical block, with modifications based on the LBDATA bit and the PBDATA bit, to the specified range of logical block addresses on the ATA device.

**Table 57 — WRITE SAME (16) CDB field translations**

Field	Description or reference
OPERATION CODE	Set to 93h. If the ATA device supports the ATA SCT Write Same command (see ATA8-ACS), the SATL should issue the ATA SCT Write Same command to repeatedly write the data block transferred from the application client to the ATA device. If the ATA device does not implement ATA SCT Write Same command then the SATL shall issue ATA write commands as defined in 9.17.1.
WRPROTECT	Unspecified (see 3.4.2)
LBDATA	9.26.2
PBDATA	9.26.2
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.14), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.33), then the SATL shall set the Start in the ATA SCT Write Same command or the ATA LBA in the ATA write command (see 3.1.25) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
NUMBER OF BLOCKS	A NUMBER OF BLOCKS field set to zero specifies that the SATL shall repeatedly write the logical block transferred from the application client to the range of ATA logical sectors corresponding to the range of logical block addresses specified by the LOGICAL BLOCK ADDRESS field through the LBA of the last logical block on the logical unit. If the NUMBER OF BLOCKS field is set to a value other than zero, the SATL shall repeatedly write the data block transferred from the application client to the medium of the ATA device for the number of logical blocks specified to the corresponding logical sectors on the ATA device. The SATL shall send as many ATA commands as required to satisfy the number of blocks specified by the WRITE SAME (16) command.
CONTROL	6.4

## 10 Parameters for SAT implementations

### 10.1 Mode parameters

#### 10.1.1 General information

SCSI mode parameters provide a mechanism to set operating parameters for SCSI devices and logical units. The MODE SENSE command obtains operating parameters while the MODE SELECT command sets operating parameters. This standard does not define the content of most operating parameters defined in mode pages due to lack of equivalent operations or features defined for ATA devices. The SATL emulates a SCSI device server for all MODE SENSE and MODE SELECT commands, and emulates the mode pages listed in 10.1.2.

The Mode Page Policy VPD page (see 10.3) should be implemented. If implemented, the MODE PAGE POLICY field in each mode page policy descriptor should be set to 00b (shared) for each mode page, and only one copy of mode page values should be maintained for all logical units within a target device (i.e., the MLUS bit is set to one in each mode page policy descriptor).

If the Mode Page Policy VPD page is not implemented, the SATL shall maintain shared mode pages for all I\_T nexuses and shall share mode pages across all logical units within a target device.

#### 10.1.2 Commonly used SCSI mode pages overview

This standard defines translations for the mode pages listed in table 58.

**Table 58 — Summary of SCSI / ATA mode page mapping**

SCSI mode page	Reference
Mode parameter header	10.1.3
Mode parameter block descriptor	10.1.4
Control (i.e., 0Ah)	10.1.5
Read-Write Error Recovery (i.e., 01h)	10.1.6
Caching (i.e., 08h)	10.1.7
Informational Exceptions Control (i.e., 1Ch)	10.1.8
All others	See SPC-3 and SBC-2 Unspecified (see 3.4.2)

### 10.1.3 Mode parameter headers

Table 59 shows the fields in the mode parameter header for the MODE SELECT (6) command and the MODE SENSE (6) command.

**Table 59 — Mode parameter header (6) fields**

Field	Description or reference
MODE DATA LENGTH	Unspecified (see 3.4.2)
MEDIUM TYPE	This field should be set to 00h. When processing a MODE SELECT command, if the MEDIUM TYPE field is set to a value other than 00h, the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.
DEVICE SPECIFIC PARAMETER	<p>Unspecified (see 3.4.2) for the MODE SELECT command.</p> <p>For the MODE SENSE command, the DEVICE SPECIFIC PARAMETER field for direct-access block devices contains the DPOFUA bit and the WP bit (see SBC-2).</p> <p>A DPOFUA bit set to zero indicates that the SATL supports neither the DPO bit nor the FUA bit. A DPOFUA bit set to one indicates that the SATL supports both the DPO bit and the FUA bit.</p> <p>A WP bit set to zero indicates that the medium is not write-protected. A WP bit set to one indicates that the medium is write-protected.</p>
BLOCK DESCRIPTOR LENGTH	This value is obtained by multiplying the number of block descriptors by eight (see SPC-3). The SATL shall support zero or one mode parameter block descriptors.

Table 60 shows the fields in the mode parameter header for the MODE SELECT (10) command and the MODE SENSE (10) command.

**Table 60 — Mode parameter header (10) fields**

Field	Description or reference
MODE DATA LENGTH	(see table 59)
MEDIUM TYPE	(see table 59)
DEVICE SPECIFIC PARAMETER	(see table 59)
LONGLBA	<p>Describes the length of the block descriptors.</p> <p>a) If set to zero, the mode parameter block descriptor is eight bytes long.</p> <p>b) If set to one, the mode parameter block descriptor is 16 bytes long.</p>
BLOCK DESCRIPTOR LENGTH	This field specifies (i.e., for a MODE SELECT command) or indicates (i.e., for a MODE SENSE command) the length of the mode parameter block descriptor. This value is obtained by multiplying the number of block descriptors by eight if LONGLBA bit is set to zero or by 16 if LONGLBA bit is set to one. The SATL shall support zero or one mode parameter block descriptors.

**10.1.4 Mode parameter block descriptor fields**

The SATL may support the direct-access short LBA mode parameter block descriptor or the long LBA mode parameter block descriptor. Table 61 describes the translation of fields in the short LBA mode parameter block descriptor and the long LBA mode parameter block descriptor supported by the SATL.

**Table 61 — Mode parameter block descriptor fields**

Field	Description or reference
NUMBER OF BLOCKS <sup>a</sup>	Unspecified (see 3.4.2)
BLOCK LENGTH <sup>a</sup>	<p>When processing a MODE SELECT command, if the SATL implements direct logical block mapping (see 3.1.33) and the value of the BLOCK LENGTH field is not the same as the ATA logical sector size (see 3.1.15), then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST. If the SATL supports indirect logical block mapping this field is unspecified (see 3.4.2).</p> <p>When processing the MODE SENSE command, if the SATL implements direct logical block mapping (see 3.1.33) then the SATL shall return the same block length for the entire logical unit and the BLOCK LENGTH field shall contain the ATA logical sector size (see 3.1.15). Otherwise the BLOCK LENGTH field is unspecified (see 3.4.2).</p>
<p><sup>a</sup> The values reported in the NUMBER OF BLOCKS field and the BLOCK LENGTH field shall be such that the logical unit capacity (see 3.1.49) is less than or equal to the ATA device capacity (see 3.1.9).</p>	

**10.1.5 Control mode page**

**10.1.5.1 General translation**

The Control mode page provides controls and information about behavior of the emulated SCSI device.

Table 62 describes the translation of the Control mode page for an ATA device.

**Table 62 — Control mode page fields**

Field	Changeable	Description or reference
PS	n/a	Unspecified (see 3.4.2)
SPF	n/a	Unspecified (see 3.4.2)
PAGE CODE	no	This field shall be set to 0Ah.
PAGE LENGTH	no	Shall be set to 0Ah.
TST	no	Shall be set to 000b to indicate that a SCSI representation of an ATA device has one task set for all initiators
TMF_ONLY	no	Shall be set to zero <sup>a</sup>
D_SENSE	Unspecified	A SATL shall support a D_SENSE bit set to zero indicating that the logical unit returns the fixed sense data format, and a SATL may support a D_SENSE bit set to one indicating logical unit returns the descriptor sense data format.  The SATL shall return sense data in the format indicated when returning CHECK CONDITION status for any commands except the ATA PASS-THROUGH (12) command and the ATA PASS-THROUGH (16) command. The SATL shall return sense data for a CHECK CONDITION status returned by an ATA PASS-THROUGH (12) command or an ATA PASS-THROUGH (16) command in the descriptor format.
GLTSD	Unspecified	Unspecified (see 3.4.2)
RLEC	no	Shall be set to zero
QUEUE ALGORITHM MODIFIER	no	The QUEUE ALGORITHM MODIFIER bit shall be set to one.
QERR	no	If the SATL supports the full task management model and ATA abort retry (see 3.1.6) of ATA queued commands (see 3.1.19) aborted by ATA collateral abort (see 3.1.7), the SATL shall set this field to 00b. Otherwise, the SATL shall set this field to 01b and comply with the unit attention requirements for a task completed with CHECK CONDITION status (see SPC-3).
TAS	no	Shall be set to zero
RAC	Unspecified	Unspecified (see 3.4.2)
UA_INTLCK_CTRL	no	Shall be set to 00b
SWP	no	Shall be set to zero
ATO	Unspecified	Unspecified (see 3.4.2)
AUTOLOAD MODE	no	Shall be set to 000b
BUSY TIMEOUT PERIOD	Unspecified	The default value shall be set to FFFFh. A SATL may support variable timeout periods and allow the application client to set a new value through a MODE SELECT operation for this mode page (see SPC-3).
EXTENDED SELF-TEST COMPLETION TIME	no	10.1.5.2
<sup>a</sup> SATL implementations shall not support ACA.		

### 10.1.5.2 Extended self-test completion time

A SATL implementation shall set the EXTENDED SELF-TEST COMPLETION TIME field to 0000h unless the ATA device supports SMART self-tests and the SATL supports a value other than 000b for the SELF-TEST CODE field for a SEND DIAGNOSTIC command. The SATL determines if the ATA device supports SMART self-test by examining the value of ATA IDENTIFY DEVICE data word 84 bit 1. If ATA IDENTIFY DEVICE data word 84 bit 1 is set to one, the ATA device supports the SMART self-test and shall retrieve the ATA device SMART data structure from the ATA device by sending an ATA SMART READ DATA command to the ATA device. The SATL may cache this information for future use when a subsequent MODE SENSE command requests the control mode page. If the SATL caches such data, the SATL may reference the cached copy instead of sending a new ATA SMART READ DATA command. Then the SATL shall set the EXTENDED SELF-TEST COMPLETION TIME field as follows:

- 1) If byte 373 of the returned SMART data structure is not set to FFh, the SATL shall set the EXTENDED SELF-TEST COMPLETION TIME field to a value that is 60 times the contents of byte 373; or
- 2) If byte 373 of the returned SMART data structure is set to FFh, the SATL shall set the EXTENDED SELF-TEST COMPLETION TIME field to a value that is the lesser of FFFFh or the result of the following formula:

$$\text{EXTENDED SELF-TEST COMPLETION TIME field} = ((w \times 256) + z) \times 60;$$

where w is the contents of byte 376 and z is the contents of byte 375.

### 10.1.6 Read-Write Error Recovery mode page

The Read-Write Error Recovery mode page specifies the error recovery parameters the SATL shall use during a command that performs a read or write operation to the medium of the ATA device (see SBC-2). Table 63 defines the translation for the Read-Write Error Recovery mode page.

**Table 63 — Read-Write Error Recovery mode page fields**

Field	Changeable	Description or reference
PS	n/a	Unspecified (see 3.4.2)
PAGE CODE	no	Shall be set to 01h
PAGE LENGTH	no	Shall be set to 0Ah
AWRE	no	Shall be set to one (see SBC-2)
ARRE	no	Shall be set to zero (see SBC-2)
TB	n/a	Unspecified (see 3.4.2)
RC	no	Shall be set to zero (see SBC-2)
EER	no	Shall be set to zero (see SBC-2)
PER	no	Shall be set to zero (see SBC-2)
DTE	no	Shall be set to zero (see SBC-2)
DCR	no	Shall be set to zero (see SBC-2)
READ RETRY COUNT	n/a	Unspecified (see 3.4.2)
WRITE RETRY COUNT	n/a	Unspecified (see 3.4.2)
RECOVERY TIME LIMIT	no	Shall be set to 00h (see SBC-2)

### 10.1.7 Caching mode page

The Caching mode page defines parameters that affect the behavior of the cache in the ATA device.

Table 64 shows the translation of fields in the Caching mode page.

**Table 64 — Caching mode page fields** (part 1 of 2)

Field	Changeable	Description or reference
PS	n/a	Unspecified (see 3.4.2)
PAGE CODE	no	Shall be set to 08h
PAGE LENGTH	no	Shall be set to 12h
IC	no	Shall be set to zero
ABPF	no	Shall be set to zero
CAP	no	Shall be set to zero
DISC	no	Shall be set to zero
SIZE	no	Shall be set to zero
WCE	yes	<p>When processing a MODE SENSE command, the SATL shall determine if the write cache of the ATA device is enabled from the ATA IDENTIFY DEVICE data word 85 bit 5. If the write cache of the ATA device is enabled the SATL shall return a value of one for the WCE bit. If the write cache of the ATA device is disabled the SATL shall return a value of zero for the WCE bit.</p> <p>When processing a MODE SELECT command:</p> <ol style="list-style-type: none"> <li>if the WCE bit is set to zero, then the SATL shall disable the write cache of the ATA device by issuing an ATA SET FEATURES – Disable write cache command (i.e., with the Features register set to 82h); or</li> <li>if the WCE bit is set to one, then the SATL shall enable the write cache of the ATA device by issuing an ATA SET FEATURES – Enable write cache command (i.e., with the Features register set to 02h).</li> </ol>
MF	no	Shall be set to zero
RCD	no	Shall be set to zero
DEMAND READ RETENTION PRIORITY	no	Shall be set to zero
WRITE RETENTION PRIORITY	no	Shall be set to zero
DISABLE PRE-FETCH TRANSFER LENGTH	no	Shall be set to zero
MINIMUM PRE-FETCH	no	Shall be set to zero
MAXIMUM PRE-FETCH	no	Shall be set to zero
MAXIMUM PRE-FETCH CEILING	no	Shall be set to zero
FSW	no	Shall be set to zero
LBCSS	no	Shall be set to zero

Table 64 — Caching mode page fields (part 2 of 2)

Field	Changeable	Description or reference
DRA	yes	<p>When processing a MODE SENSE command, the SATL shall determine if the ATA device look-ahead is enabled from the ATA IDENTIFY DEVICE data word 85 bit 6. If the look-ahead is enabled the SATL shall return a value of zero for the DRA bit. If the look-ahead is disabled the SATL shall return a value of one for the DRA bit.</p> <p>When processing a MODE SELECT command,</p> <ul style="list-style-type: none"> <li>a) if the DRA bit is set to zero, the SATL shall enable the ATA device read look-ahead feature by issuing an ATA SET FEATURES – Enable read look-ahead feature command (i.e., with the Features register set to AAh); or</li> <li>b) if the DRA bit is set to one, the SATL shall disable the ATA device read look-ahead feature by issuing an ATA SET FEATURES – Disable read look-ahead feature command (i.e., with the Features register set to 55h).</li> </ul>
NV_DIS	no	Shall be set zero
NUMBER OF CACHE SEGMENTS	no	Shall be set zero
CACHE SEGMENT SIZE	no	Shall be set zero

### 10.1.8 Informational Exceptions Control mode page

#### 10.1.8.1 Informational Exceptions Control mode page overview

The Informational Exceptions Control mode page defines the methods used by the SATL to control the reporting and the operations of specific informational exception conditions. The Informational Exceptions Control mode page applies to informational exceptions that return an additional sense code of FAILURE PREDICTION THRESHOLD EXCEEDED or WARNING to the application client (see SPC-3).

Table 65 shows the translation of fields in the Informational Exceptions Control mode page.

**Table 65 — Informational Exceptions Control mode page fields**

Field	Changeable	Description or reference
PS	n/a	Unspecified (see 3.4.2)
SPF	no	Shall be set to zero
PAGE CODE	no	Shall be set to 1Ch. The SATL shall determine if the ATA SMART feature set is supported from the ATA IDENTIFY DEVICE data word 82 bit 0. If the ATA SMART feature set is not supported, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB for a MODE SENSE command or INVALID FIELD IN PARAMETER LIST for a MODE SELECT command.
PAGE LENGTH	no	Shall be set to 0Ah
PERF	no	Shall be set to zero
EBF	n/a	Unspecified (see 3.4.2)
EWASC	n/a	Unspecified (see 3.4.2)
DEXCPT	yes	Unspecified (see 3.4.2)
TEST	no	Shall be set to zero
LOGERR	n/a	Unspecified (see 3.4.2)
MRIE	no <sup>a</sup>	Should be set to 6h (see 10.1.8.2).
INTERVAL TIMER	n/a	Unspecified (see 3.4.2)
REPORT COUNT	n/a	Unspecified (see 3.4.2)
<sup>a</sup> The MRIE field should be set to 6h, however if the SATL supports other settings of the MRIE field, the SATL should permit the MRIE field to be changeable.		

#### 10.1.8.2 Method of reporting informational exceptions (MRIE)

The SATL should support 6h. Support for any other value is unspecified (see 3.4.2).

When the MRIE field is set to 6h and the SATL receives a REQUEST SENSE command, the SATL shall issue an ATA SMART RETURN STATUS command to the ATA device and return status to the application client as defined in SPC-3 (see 10.2.5.2). If the result of the ATA SMART RETURN STATUS command indicates a threshold exceeded condition the SATL shall set the additional sense code to HARDWARE IMPENDING FAILURE GENERAL HARD DRIVE FAILURE.

## 10.2 Log pages

### 10.2.1 Log pages overview

This standard defines translations for the log pages listed in table 66.

**Table 66 — Summary of SCSI / ATA log page mapping**

SCSI log page	Reference
Application Client (i.e., page code 0Fh)	10.2.2
Supported Log Pages (i.e., page code 00h)	10.2.3
Self-Test Results (i.e., page code 10h)	10.2.4
Informational Exceptions (i.e., page code 2Fh)	10.2.5
All others	Unspecified (see 3.4.2)

### 10.2.2 Application Client log page

#### 10.2.2.1 Translation Overview

The Application Client log page provides a location for application clients to store information. A SATL translates a LOG SELECT or LOG SENSE command to the application client log page to accesses to the ATA host vendor-specific log pages. Table 67 describes the translation of the general usage application client parameter data for the application client log page.

The SATL determines if the attached ATA device supports host vendor specific log pages by reading log page address 00h using READ LOG EXT, READ LOG DMA EXT, or SMART READ LOG.

If the attached ATA device:

- a) Does not support the general purpose logging feature set and the SMART feature set is disabled; or
- b) Does not support host vendor-specific log pages

Then the SATL shall complete the LOG SENSE or LOG SELECT command for the application client specific log page with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and an additional sense code of INVALID FIELD IN CDB.

**Table 67 — General usage application client parameter data fields**

<b>Field</b>	<b>Description or Reference</b>
PARAMETER CODE	10.2.2.2
DU	Shall be 1b (see SPC-4)
TSD	Shall be 0b (see SPC-4)
ETC	Shall be 0b (see SPC-4)
TMC	This field is ignored (see SPC-4)
FORMAT AND LINKING	Shall be 11b (see SPC-4)
PARAMETER LENGTH	Shall be FCh (see SPC-4)
GENERAL USAGE PARAMETER BYTES	10.2.2.2

**10.2.2.2 LOG SELECT translation**

The SATL stores the application client parameter for a LOG SELECT command in the ATA device host vendor-specific log page. The SATL stores the application client parameter data at the ATA log address as specified in table 2.

Within an ATA log address, the SATL shall store each parameter code in ascending order within the sixteen 512-byte data blocks for each ATA log address. For example, parameter code 0000h is stored at offset 0 of the first 512-byte block of data at log address 90h and parameter code 0001h is stored at offset 256 in the first 512-byte block of data at log address 90h. The SATL stores this information by issuing a SMART WRITE LOG, WRITE LOG EXT, or WRITE LOG DMA EXT command to the device.

The SATL shall ensure that any previously stored data at the log address is preserved when writing to the log address for the requested parameter data.

**Table 68 — Parameter Storage Location**

<b>Parameter Code</b>	<b>ATA Log Address</b>
0000h - 001Fh	90h
0020h - 003Fh	91h
0040h - 005Fh	92h
0060h - 007Fh	93h
0080h - 009Fh	94h
00A0h - 00BFh	95h
00C0h - 00DFh	96h
00E0h - 00FFh	97h
0100h - 011Fh	98h
0120h - 013Fh	99h
0140h - 015Fh	9Ah
0160h - 017Fh	9Bh
0180h - 019Fh	9Ch
01A0h - 01BFh	9Dh
01C0h - 01DFh	9Eh
01E0h - 01FFh	9Fh

### 10.2.2.3 LOG SENSE translation

The SATL retrieves the requested parameter data by reading the ATA log address that stores the parameter code using a SMART READ LOG, READ LOG EXT, or READ LOG DMA EXT command. The log address to read is determined by Table 2.

### 10.2.3 Supported Log Pages log page

The Supported Log Pages log page (see table 69) returns the list of log pages supported by the SATL (see SPC-3).

**Table 69 — Supported Log Pages log page fields**

Field	Description or reference
PAGE CODE	Shall be set to zero
PAGE LENGTH	Unspecified (see 3.4.2)
SUPPORTED PAGE LIST	The SATL shall include log pages as follows: <ul style="list-style-type: none"> <li>a) the Informational Exceptions log page if the ATA device supports the ATA SMART feature set (i.e., ATA IDENTIFY DEVICE data word 82 bit 0 is set to one); and</li> <li>b) the Self-Test Results log page if the ATA device supports the ATA SMART self-test (i.e., ATA IDENTIFY DEVICE data word 84 bit 1 is set to one).</li> </ul>

### 10.2.4 Self-Test Results log page

#### 10.2.4.1 Self-Test Results log page overview

The Self-Test Results log page provides the results from self-test results descriptor entry pointed to by the Self-test descriptor index. Table 70 shows the Self-Test Results log page header fields.

**Table 70 — Self-Test Results log page fields**

Field	Description or reference
PAGE CODE	Shall be set to 10h
PAGE LENGTH	Shall be set to 190h

Translations of the fields for the Self-Test Results log parameters for the Self-Test Results log page are shown in table 71.

**Table 71 — Self-Test Results log parameters (part 1 of 4)**

Field	Description or reference
PARAMETER CODE	The SATL shall return log parameters with the PARAMETER CODE field set to 0001h through 0014h.
DU	Shall be set to zero
DS	Shall be set to zero
TSD	Shall be set to zero
ETC	Shall be set to zero
TMC	Shall be set to zero
LBIN	Shall be set to one
LP	Shall be set to one
PARAMETER LENGTH	Shall be set to 10h

**Table 71 — Self-Test Results log parameters (part 2 of 4)**

Field	Description or reference
SELF-TEST CODE	<p>The SATL shall read the ATA log data as defined in 10.2.4.2.</p> <p>If the SATL reads the ATA log data using the ATA READ LOG EXT command specifying the Extended SMART self-test log, then the SATL shall check if the value contained in the Self-test descriptor index field in the first block of data (i.e., bytes 2 and 3) is set to zero. If the value contained in the Self-test descriptor index field is set to zero, then the SATL shall set the SELF-TEST CODE field to zero for each of the log parameters returned. If the value contained in the Self-test descriptor index field is set to a non-zero value, then the SELF-TEST CODE field is unspecified (see 3.4.2).</p> <p>If the SATL reads the ATA log data using the ATA SMART READ LOG command specifying the SMART self-test log, then the SELF-TEST CODE field is unspecified (see 3.4.2).</p>
SELF-TEST RESULTS	<p>The SATL shall read the ATA log data as defined in 10.2.4.2.</p> <p>If the SATL reads the ATA log data using the ATA READ LOG EXT command specifying the Extended SMART self-test log, then the SATL shall check if the value contained in the Self-test descriptor index field in the first block of data (i.e., bytes 2 and 3) is set to zero. If the value contained in the Self-test descriptor index field is set to zero, then the SATL shall set the SELF-TEST RESULTS field to zero for each log parameter returned.</p> <p>If the value contained in the Self-test descriptor index field is set to a nonzero value, then the SATL shall set the SELF-TEST RESULTS field to:</p> <ul style="list-style-type: none"> <li>a) the value contained in the Self-test Execution Status bits of the content of the self-test execution status byte field of the n<sup>th</sup> descriptor entry, where n is equal to the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is greater than zero (e.g., for a log parameter with the PARAMETER CODE field of 0002h and a value contained in the Self-test descriptor index field of 6h, then the fourth descriptor entry is used); or</li> <li>b) zero, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is less than or equal to zero.</li> </ul> <p>If the SATL reads the ATA log data using the ATA SMART READ LOG command specifying the SMART self-test log, then the SATL shall set the SELF-TEST RESULTS field to the value contained in the Self-test Execution Status bits of the content of the self-test execution status byte field of the nth descriptor entry, where n is equal to the value contained in the PARAMETER CODE field for the log parameter being returned (e.g., for a log parameter with the PARAMETER CODE field of 0002h, then the second descriptor entry is used).</p>
SELF-TEST NUMBER	Unspecified (see 3.4.2)

**Table 71 — Self-Test Results log parameters (part 3 of 4)**

Field	Description or reference
TIMESTAMP	<p>The SATL shall read the ATA log data as defined in 10.2.4.2.</p> <p>If the SATL reads the ATA log data using the ATA READ LOG EXT command specifying the Extended SMART self-test log, then the SATL shall check if the value contained in the Self-test descriptor index field in the first block of data (i.e., bytes 2 and 3) is set to zero. If the value contained in the Self-test descriptor index field is set to zero, then the SATL shall set the TIMESTAMP field to zero for each log parameter returned.</p> <p>If the value contained in the Self-test descriptor index field is set to a nonzero value, then the SATL shall set the TIMESTAMP field to:</p> <ul style="list-style-type: none"> <li>a) the values contained in the Life timestamp (most significant byte) field and Life timestamp (least significant byte) field of the n<sup>th</sup> descriptor entry, where n is equal to the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one, if the result of value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is greater than zero (e.g., for a log parameter with the PARAMETER CODE field of 0002h and a value contained in the Selftest descriptor index field of 6h, then the fourth descriptor entry is used); or</li> <li>b) zero, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is less than or equal to zero.</li> </ul> <p>If the SATL reads the ATA log data using the ATA SMART READ LOG command specifying the SMART self-test log, then the SATL shall set the TIMESTAMP field to the values contained in the Life timestamp (most significant byte) field and Life timestamp (least significant byte) field of the n<sup>th</sup> descriptor entry, where n is equal to the value contained in the PARAMETER CODE field for the log parameter being returned (e.g., for a log parameter with the PARAMETER CODE field of 0002h, then the second descriptor entry is used).</p>

**Table 71 — Self-Test Results log parameters (part 4 of 4)**

Field	Description or reference
ADDRESS OF FIRST FAILURE	<p>The SATL shall read the ATA log data as defined in 10.2.4.2.</p> <p>If the SATL reads the ATA log data using the ATA READ LOG EXT command specifying the Extended SMART self-test log, then the SATL shall check if the value contained in the Self-test descriptor index field in the first block of data (i.e., bytes 2 and 3) is set to zero. If the value contained in the Self-test descriptor index field is set to zero, then the SATL shall set the ADDRESS OF FIRST FAILURE field to zero for each log parameter returned.</p> <p>If the value contained in the Self-test descriptor index field is set to a nonzero value, then the SATL shall set the ADDRESS OF FIRST FAILURE field to:</p> <p>a) the values contained in the Failing LBA (47:40) field, Failing LBA (39:32) field, Failing LBA (31:24) field, Failing LBA (23:16) field, Failing LBA (15:8) field, and Failing LBA (7:0) field of the n<sup>th</sup> descriptor entry, where n is equal to the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is greater than zero (e.g., for a log parameter with the PARAMETER CODE field of 0002h and a value contained in the Self-test descriptor index field of 6h, then the fourth descriptor entry is used); or</p> <p>b) zero, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is less than or equal to zero.</p> <p>If the SATL reads the ATA log data using the ATA SMART READ LOG command specifying the SMART self-test log, then the SATL shall set the ADDRESS OF FIRST FAILURE field to the values contained in the Failing LBA (27:24) field, Failing LBA (23:16) field, Failing LBA (15:8) field, and Failing LBA (7:0) field of the n<sup>th</sup> descriptor entry, where n is equal to the value contained in the PARAMETER CODE field for the log parameter being returned (e.g., for a log parameter with the PARAMETER CODE field of 0002h, then the second descriptor entry is used).</p>
SENSE KEY	10.2.4.3
ADDITIONAL SENSE CODE	10.2.4.3
ADDITIONAL SENSE CODE QUALIFIER	10.2.4.3

#### 10.2.4.2 A method of determining ATA command selection for field translations

To translate the SELF-TEST CODE field, the SELF-TEST RESULTS field, the TIMESTAMP field, the ADDRESS OF FIRST FAILURE field, the SENSE KEY field, the ADDITIONAL SENSE CODE field, and the ADDITIONAL SENSE CODE QUALIFIER field of Self-Test Results log parameters, the SATL shall issue an ATA IDENTIFY DEVICE command to the ATA device, and from the returned data the SATL shall determine if the ATA device supports the 48-bit Address feature set. If the 48-bit Address feature set is supported (i.e., bit 10 of word 83 of ATA IDENTIFY DEVICE data is set to one), then the SATL shall issue an ATA READ LOG EXT command with the Log address set to 07h (i.e., Extended SMART self-test log) to the ATA device. If the 48-bit Address feature set is not supported (i.e., bit 10 of word 83 of ATA IDENTIFY DEVICE data is set to zero), then the SATL shall issue

an ATA SMART READ LOG command with the Log address set to 06h (i.e., SMART self-test log) to the ATA device.

**10.2.4.3 Sense key and additional sense code**

The SATL shall determine the SENSE KEY field, the ADDITIONAL SENSE CODE field, and the ADDITIONAL SENSE CODE QUALIFIER field returned in each log parameter from the content of the self-test execution status byte returned from a ATA READ LOG EXT command or ATA SMART READ LOG command issued to the ATA device (see 10.2.4.2). The values returned in each log parameter shall be translated into sense data for the sense key, and additional sense code as shown in table 72.

**Table 72 — ATA Self-test execution status values translated to SCSI sense keys and sense codes**

ATA	SCSI		
Self-Test execution status value	Sense key	Additional sense code	NN
0	NO SENSE	NO ADDITIONAL SENSE INFORMATION	n/a
1	ABORTED COMMAND	DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	81h
2		DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	82h
3		DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	83h
4	HARDWARE ERROR	DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	84h
5		DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	85h
6		DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	86h
7	MEDIUM ERROR	DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	87h
8	HARDWARE ERROR	DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	88h
9-14	Unspecified (see 3.4.2) <sup>a</sup>		
15	NO SENSE	NO ADDITIONAL SENSE INFORMATION	n/a

<sup>a</sup> Self-Test execution status values from 9 to 14 are reserved in ATA8-ACS.

**10.2.5 Informational Exceptions log page**

**10.2.5.1 Informational Exceptions log page overview**

The Informational Exceptions log page provides detail about informational exceptions. Table 73 shows the log page header fields.

**Table 73 — Informational Exceptions log page header fields**

Field	Description or reference
PAGE CODE	Shall be set to 2Fh. This field value is specific to the Informational Exceptions log page. The SATL shall issue the ATA SMART RETURN STATUS command to the non-packet device. Data returned from the non-packet device shall be translated into the appropriate log sense parameter data (see 10.2.5.2) to be returned to the application client.
PAGE LENGTH	Unspecified (see 3.4.2)

The first log parameter is the informational exceptions general parameter shown in table 74.

**Table 74 — Informational Exceptions general parameter data**

Field	Description or reference
PARAMETER CODE	Shall be set to 0000h
DU	Shall be set to zero
DS	Shall be set to zero
TSD	Shall be set to zero
ETC	Shall be set to zero
TMC	Shall be set to zero
LBIN	Shall be set to one
LP	Shall be set to one
PARAMETER LENGTH	Unspecified (see 3.4.2)
INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE	10.2.5.2
INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER	10.2.5.2
MOST RECENT TEMPERATURE READING	10.2.5.3
Vendor Specific	Unspecified (see 3.4.2)

**10.2.5.2 Additional sense code and additional sense code qualifier translations**

Data received from a ATA device in response to an ATA SMART RETURN STATUS command shall be translated by the SATL into the informational exceptions general parameter data returned to the application client. Table 75 provides the parameter data translations for the INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE field and the INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER field.

**Table 75 — ATA SMART RETURN STATUS translations**

Data returned to SATL from the ATA device by the ATA SMART RETURN STATUS command	SMART condition	Informational exceptions general parameter data fields
LBA Mid = 4Fh LBA High = C2h	threshold not exceeded	INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE = 00h, INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER = 00h
LBA Mid = F4h LBA High = 2Ch	threshold exceeded	INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE = 5Dh, INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER = 10h

### 10.2.5.3 Most recent temperature reading translation

If the ATA device supports the SCT Feature Set (see SCT), then to translate the MOST RECENT TEMPERATURE READING field of the Informational Exceptions log page, the SATL shall Issue an SCT Status Request to the ATA device; and then:

- a) if the HDA Temp field (see SCT) is less than zero, the SATL shall set the MOST RECENT TEMPERATURE READING field to zero;
- b) if the HDA Temp field is equal to 80h, the SATL shall set the MOST RECENT TEMPERATURE READING field to FFh; or
- c) the SATL shall set the MOST RECENT TEMPERATURE READING FIELD to the value in the HDA Temp field.

If the ATA device does not support the SCT feature set, then the SATL shall set the MOST RECENT TEMPERATURE READING field to FFh.

## 10.3 Vital product data parameters

### 10.3.1 Vital product data parameters overview

Table 76 provides a summary of the VPD page translations defined in this standard.

**Table 76 — Summary of SCSI / ATA VPD page mapping**

SCSI VPD page	Reference
Supported VPD Pages VPD page (i.e., 00h)	10.3.2
Unit Serial Number VPD page (i.e., 80h)	10.3.3
Device Identification VPD page (i.e., 83h)	10.3.4
Mode Page Policy VPD page (i.e., 87h)	10.3.5
ATA Information VPD page (i.e., 89h)	12.4.2
All others	See SPC-3 and SBC-2 Unspecified (see 3.4.2)

### 10.3.2 Supported VPD pages VPD page

Table 77 shows the fields of the Supported VPD pages VPD page.

**Table 77 — Supported VPD pages VPD page fields**

Field	Description or reference
PERIPHERAL DEVICE TYPE	The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.
PERIPHERAL QUALIFIER	
PAGE CODE	The SATL shall set this field to 00h.
PAGE LENGTH	The SATL shall set this field to indicate the length of the supported VPD page list returned in number of bytes.
SUPPORTED VPD LIST	This list shall contain the page codes of the pages supported by the SATL in ascending order of page codes beginning with page code 00h.

### 10.3.3 Unit Serial Number VPD page

Table 78 defines the Unit Serial Number VPD page (see SPC-3) returned by a SATL for an ATA device.

**Table 78 — Unit Serial Number VPD page for SAT**

Bit\Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (80h)							
2	Reserved							
3	PAGE LENGTH (14h)							
4								
23	PRODUCT SERIAL NUMBER							

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.

The PAGE CODE field shall be set to 80h.

The PAGE LENGTH field shall be set to 14h.

The PRODUCT SERIAL NUMBER field contains a representation of the Serial number field in the ATA IDENTIFY DEVICE data (i.e., words 19:10) last retrieved from the ATA device. Each pair of bytes in the Serial number field shall be swapped to create a valid ASCII string format in the PRODUCT SERIAL NUMBER field as described in table 79.

**Table 79 — PRODUCT SERIAL NUMBER field**

Byte	Contents
0	IDENTIFY DEVICE word 10 bits 15:8 (i.e., byte 1)
1	IDENTIFY DEVICE word 10 bits 7:0 (i.e., byte 0)
2	IDENTIFY DEVICE word 11 bits 15:8 (i.e., byte 3)
3	IDENTIFY DEVICE word 11 bits 7:0 (i.e., byte 2)
...	...
18	IDENTIFY DEVICE word 19 bits 15:8 (i.e., byte 19)
19	IDENTIFY DEVICE word 19 bits 7:0 (i.e., byte 18)

NOTE 12 - Although SPC-3 defines the PRODUCT SERIAL NUMBER field as right-aligned, ATA8-ACS does not require its SERIAL NUMBER field to be right-aligned. Therefore, the PRODUCT SERIAL NUMBER field for SAT may not be right-aligned.

**10.3.4 Device Identification VPD page**

**10.3.4.1 Device Identification VPD page overview**

The SATL shall return the Device Identification VPD page (see SPC-3) as defined in table 80.

**Table 80 — Device Identification VPD page for SAT**

Bit\Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (83h)							
2	(MSB)	PAGE LENGTH (n-3)						(LSB)
3								
4	Identification descriptor for the logical unit (see table 81 and table 83)							
15								
16	Additional identification descriptor(s) (if any)							
n								

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.

The PAGE CODE field shall be set to 83h.

The PAGE LENGTH field contains the length of the remaining bytes of the VPD page.

One identification descriptor for a logical unit (i.e., a logical unit name) shall be included (see 10.3.4.2).

In some environments, one or more additional identification descriptors may be included (see 10.3.4.3).

**10.3.4.2 Logical unit name**

**10.3.4.2.1 Logical unit name overview**

If the ATA device returns the ATA IDENTIFY DEVICE data word 87 bit 8 is set to one indicating that the ATA device supports the World wide name field (i.e., ATA IDENTIFY DEVICE data words 111:108), the SATL shall include an identification descriptor containing a logical unit name as defined in 10.3.4.2.2.

If the ATA IDENTIFY DEVICE data returned by the ATA device word 87 bit 8 is set to zero, indicating that the ATA device does not supports the support the World wide name field (i.e., ATA IDENTIFY DEVICE data words 111:108), the SATL shall include an identification descriptor containing a logical unit name as defined in 10.3.4.2.3.

### 10.3.4.2.2 Logical unit name derived from the world wide name

Table 81 defines the logical unit name derived from the ATA device world wide name.

**Table 81 — Logical unit name derived from the world wide name**

Byte\Bit	7	6	5	4	3	2	1	0
0	PROTOCOL IDENTIFIER (0h)				CODE SET (1h)			
1	PIV (0b)	Reserved	ASSOCIATION (00b)		IDENTIFIER TYPE (3h)			
2	Reserved							
3	IDENTIFIER LENGTH (08h)							
4	NAA				(MSB)			
5	IEEE COMPANY_ID							
6								
7					(LSB)	(MSB)		
8	VENDOR SPECIFIC IDENTIFIER							
11								

The PROTOCOL IDENTIFIER field shall be set to 0h.

The CODE SET field shall be set to 1h (i.e., binary).

The PIV bit shall be set to 0b.

The ASSOCIATION field shall be set to 00b (i.e., logical unit).

The IDENTIFIER TYPE field shall be set to 3h (i.e., NAA).

The NAA field, the IEEE COMPANY\_ID field, and the VENDOR SPECIFIC IDENTIFIER field shall be based on the ATA IDENTIFY DEVICE data World wide name field as described in table 82.

**Table 82 — Fields in the logical unit name**

Field		Contents
Field name	Specific bits in table 81	
NAA	Byte 4 bits 7:4	IDENTIFY DEVICE word 108 bits 15:12 <sup>a</sup>
IEEE COMPANY_ID	Byte 4 bits 3:0	IDENTIFY DEVICE word 108 bits 11:8
	Byte 5	IDENTIFY DEVICE word 108 bits 7:0
	Byte 6	IDENTIFY DEVICE word 109 bits 15:8
	Byte 7 bits 7:4	IDENTIFY DEVICE word 109 bits 7:4
VENDOR SPECIFIC IDENTIFIER	Byte 7 bits 3:0	IDENTIFY DEVICE word 109 bits 3:0
	Byte 8	IDENTIFY DEVICE word 110 bits 15:8
	Byte 9	IDENTIFY DEVICE word 110 bits 7:0
	Byte 10	IDENTIFY DEVICE word 111 bits 15:8
	Byte 11	IDENTIFY DEVICE word 111 bits 7:0

<sup>a</sup> This 4-bit field is required to be set to 5h (i.e., IEEE Registered) by ATA8-ACS.

10.3.4.2.3 Logical unit name derived from the model number and serial number

Table 83 defines the logical unit name derived from the ATA device model number and serial number.

**Table 83 — Logical unit name derived from the model number and serial number**

Byte\Bit	7	6	5	4	3	2	1	0
0	PROTOCOL IDENTIFIER (0h)				CODE SET (2h)			
1	PIV (0b)	Reserved	ASSOCIATION (00b)		IDENTIFIER TYPE (1h)			
2	Reserved							
3	IDENTIFIER LENGTH (68)							
4	VENDOR IDENTIFICATION ('ATA-----')							
11	VENDOR IDENTIFICATION ('ATA-----')							
12	VENDOR SPECIFIC IDENTIFIER (see table 84)							
71	VENDOR SPECIFIC IDENTIFIER (see table 84)							

The PROTOCOL IDENTIFIER field shall be set to 0h.

The CODE SET field shall be set to 2h (i.e., ASCII).

The PIV bit shall be set to 0b.

The ASSOCIATION field shall be set to 00b (i.e., logical unit).

The IDENTIFIER TYPE field shall be set to 1h (i.e., T10 vendor identification).

The VENDOR IDENTIFICATION field contains the string 'ATA-----'.

The VENDOR SPECIFIC IDENTIFIER field all be set to a representation of the ATA IDENTIFY DEVICE data Model number field concatenated with a representation of the ATA IDENTIFY DEVICE data Serial number field as described in table 84.

**Table 84 — VENDOR SPECIFIC IDENTIFIER field for logical unit name**

Byte	Contents	
	Source field name	Source location
0	Model number field	IDENTIFY DEVICE word 27 bits 15:8
1		IDENTIFY DEVICE word 27 bits 7:0
2		IDENTIFY DEVICE word 28 bits 15:8
...		...
39		IDENTIFY DEVICE word 46 bits 7:0
40	Serial number field	IDENTIFY DEVICE word 10 bits 15:8
41		IDENTIFY DEVICE word 10 bits 7:0
42		IDENTIFY DEVICE word 11 bits 15:8
...		...
59		IDENTIFY DEVICE word 19 bits 7:0

NOTE 13 - The logical unit name using the T10 vendor ID based format is not guaranteed to be worldwide unique, since ATA8-ACS only requires the combination of the Model number field and Serial number field to be unique for a given manufacturer but defines no manufacturer identification field.

10.3.4.3 Examples of additional identification descriptors

10.3.4.3.1 Identification descriptors included by a SATL in an ATA host

Figure 9 shows the identification descriptor returned by a SATL in an ATA host (i.e., where the ATA device is being accessed with an ATA host port) containing a logical unit name based on ATA IDENTIFY DEVICE data (see table 81 or table 83 in 10.3.4.2).

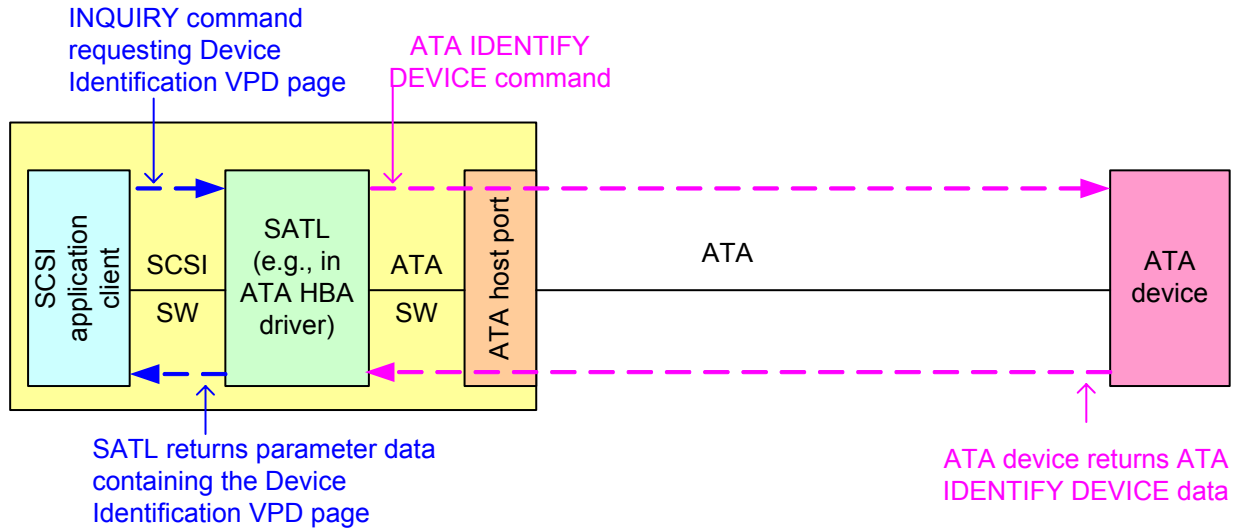


Figure 9 — Identification descriptors included by a SATL in an ATA host

10.3.4.3.2 Identification descriptors included by a SATL in a SAS initiator device

Figure 10 shows the identification descriptors returned by a SATL in a SAS initiator device (i.e., where the ATA device is being accessed by a SAS STP initiator port through an STP/SATA bridge):

- a) a logical unit name based on ATA IDENTIFY DEVICE data (see table 81 or table 83 in 10.3.4.2);
- b) a target port identifier based on the SAS STP target port SAS address (see table 85); and
- c) a relative target port identifier set to 0001h (see SPC-3).

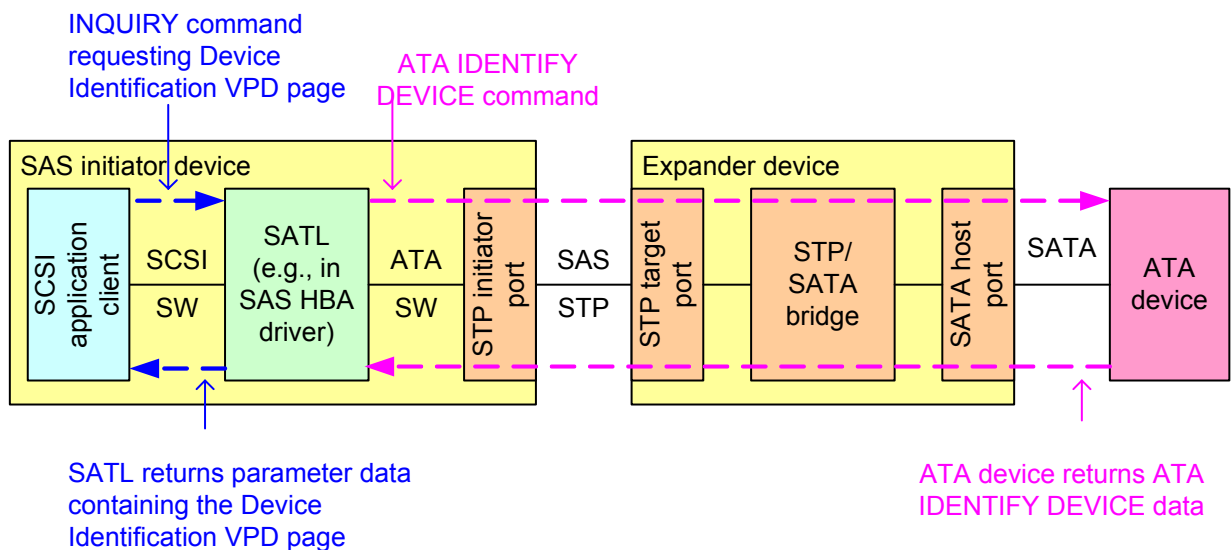


Figure 10 — Identification descriptors included by a SATL in a SAS initiator device

The SATL includes a target port identifier as defined in table 85.

**Table 85 — Target port identifier for SAS**

Byte\Bit	7	6	5	4	3	2	1	0
0	PROTOCOL IDENTIFIER (6h)				CODE SET (1h)			
1	PIV (1b)	Reserved	ASSOCIATION (01b)		IDENTIFIER TYPE (3h)			
2	Reserved							
3	IDENTIFIER LENGTH (08h)							
4	SAS ADDRESS							
11								

The CODE SET field is set to 1h (i.e., binary).

The PIV bit is set to one.

The ASSOCIATION field is set to 01b (i.e., target port).

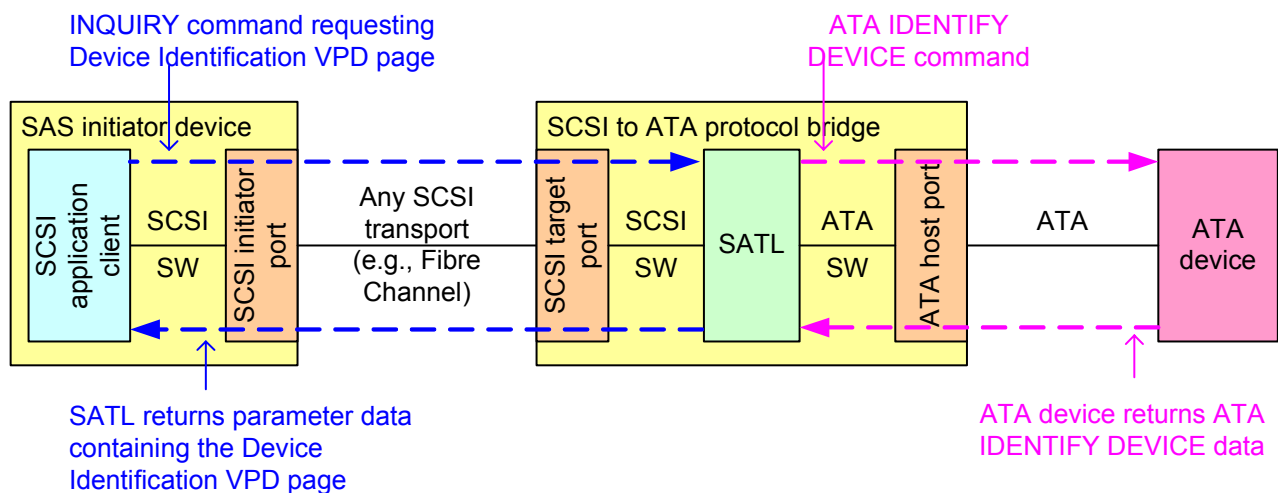
The IDENTIFIER TYPE field is set to 3h (i.e., NAA).

The SAS ADDRESS field is set to the SAS address of the STP target port providing the STP/SATA bridge functionality (i.e., the SAS address of the SATA device).

**10.3.4.3.3 Identification descriptors included by a SATL in a SCSI to ATA protocol bridge**

Figure 11 shows the identification descriptors returned by a SATL in a SCSI to ATA protocol bridge, where the ATA device is being accessed by an ATA host port, and the SATL is being accessed with a SCSI target port using a SCSI transport protocol (e.g, FCP-3 or iSCSI):

- a) a logical unit name based on ATA IDENTIFY DEVICE data (see table 81 or table 83 in 10.3.4.2);
- b) any target port identifiers specified by the SCSI transport protocol standard (e.g., for FCP-3, the SATL includes an identifier with identifier type 3h (i.e., NAA) containing the Port\_Name, and may include an identifier with identifier type 4h (i.e., relative target port identifier)); and
- c) any other identification descriptors supported by the protocol bridge (e.g., a target device name).



**Figure 11 — Identification descriptors included by a SATL in a SCSI to ATA protocol bridge**

**10.3.5 Mode Page Policy VPD page**

The SATL should implement the Mode Page Policy VPD page for each logical unit emulated (see SPC-3). Table 86 defines the Mode Page Policy VPD page (see SPC-3) returned by the SATL.

**Table 86 — Mode Page Policy VPD page for SAT**

Byte/Bit	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (87h)							
2	PAGE LENGTH (n-3)							
3	Mode page policy descriptor list							
4	Mode page policy descriptor (first)							
7	Mode page policy descriptor (first)							
•	•							
•	•							
•	•							
n-3	Mode page policy descriptor (last)							
n	Mode page policy descriptor (last)							

The PERIPHERAL QUALIFIER FIELD and the peripheral device type field shall be set as described in 8.1.2.

The PAGE CODE field shall be set to 87h.

The PAGE LENGTH FIELD is unspecified (see 3.4.2).

If the SATL implements the Mode Page Policy VPD page, then the SATL shall include at least one mode page policy descriptor (see table 87).

**Table 87 — Mode policy descriptor for SAT**

Byte/Bit	7	6	5	4	3	2	1	0
0	Reserved		POLICY PAGE CODE					
1	POLICY SUBPAGE CODE							
2	MLUS	Reserved					MODE PAGE POLICY	
3	Reserved							

The POLICY PAGE CODE field, the POLICY SUBPAGE CODE field, the multiple logical units share (i.e., MLUS) bit, and MODE PAGE POLICY field are unspecified (see 3.4.2 and SPC-3).

## 11 Error and sense reporting

### 11.1 Error translation – ATA device error to SCSI error map

ATA device errors are translated to the appropriate SCSI errors. The ATA Status register and Error register bit settings provide the information to be translated into SCSI sense key, additional sense code, and additional sense code qualifier for error reporting. Unless otherwise specified in the subclause describing the translation of a particular SCSI command, log page, mode page or VPD page, the SATL shall translate ATA device errors to SCSI errors as shown in table 88.

**Table 88 — Translation of ATA errors to SCSI errors**

ATA Error		SCSI Error	
Register			
Status	Error <sup>a</sup>	Sense key	Additional sense code
DF <sup>b</sup>	n/a	HARDWARE ERROR	INTERNAL TARGET FAILURE
ERR	NM	NOT READY	MEDIUM NOT PRESENT
ERR	UNC	MEDIUM ERROR	UNRECOVERED READ ERROR
ERR	WP	DATA PROTECT	WRITE PROTECTED
ERR	IDNF	MEDIUM ERROR	RECORD NOT FOUND
ERR	ABRT <sup>c</sup>	ABORTED COMMAND	NO ADDITIONAL SENSE INFORMATION
ERR	MC	UNIT ATTENTION	NOT READY TO READY CHANGE, MEDIUM MAY HAVE CHANGED
ERR	MCR	UNIT ATTENTION	OPERATOR MEDIUM REMOVAL REQUEST
ERR	ICRC	ABORTED COMMAND	INFORMATION UNIT iuCRC ERROR DETECTED
CORR	n/a	This condition is not considered an error.	

<sup>a</sup> If the Error register has an obsolete bit set to one, the SATL may return a vendor-specific additional sense code (e.g., if the AMNF bit is set to one, return MEDIUM ERROR - ADDRESS MARK NOT FOUND FOR DATA FIELD).

<sup>b</sup> After an ATA device returns a DF bit set to one, the SATL processes any subsequent commands received for the emulated logical unit corresponding to the ATA device by terminating the command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to INTERNAL TARGET FAILURE.

<sup>c</sup> The ABRT bit is ignored if any other ATA error bit is set.

## 12 SAT-specific SCSI extensions

### 12.1 SAT-specific SCSI extensions overview

This subclause defines additional SCSI commands, mode pages and VPD pages that may be supported by a SATL to provide capabilities beyond those defined in the other SCSI command sets.

SCSI commands defined for SATL implementations include:

- a) ATA PASS-THROUGH (12) command (see 12.2.2); and
- b) ATA PASS-THROUGH (16) command (see 12.2.3).

Mode pages defined for SATL implementations include:

- a) PATA Control mode page (see 12.3.2).

Vital Product Data pages defined for SATL implementations include:

- a) ATA Information VPD page (see 12.4.2).

## 12.2 ATA PASS-THROUGH commands

### 12.2.1 ATA PASS-THROUGH commands overview

ATA PASS-THROUGH commands provide a method for:

- a) an application client to transmit an ATA command to an ATA device;
- b) optionally, transferring data between an application client and an ATA device; and
- c) an ATA device to transfer completion status through the SATL.

This is accomplished by defining:

- a) CDBs containing ATA command information (see 12.2.2 and 12.2.3); and
- b) specific SCSI status and sense data usage for returning the results of an ATA command (see 12.2.5).

12.2.2 ATA PASS-THROUGH (12) command

Table 89 shows the CDB for the ATA PASS-THROUGH (12) command.

Table 89 — ATA PASS-THROUGH (12) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (A1h)							
1	MULTIPLE_COUNT			PROTOCOL				Reserved
2	OFF_LINE	CK_COND	Reserved	T_DIR	BYTE_BLOCK	T_LENGTH		
3	FEATURES (7:0)							
4	SECTOR_COUNT (7:0)							
5	LBA_LOW (7:0)							
6	LBA_MID (7:0)							
7	LBA_HIGH (7:0)							
8	DEVICE							
9	COMMAND							
10	Reserved							
11	CONTROL (see 6.4)							

12.2.4 describes the mapping between the fields in the ATA PASS-THROUGH (12) CDB to corresponding ATA command fields (see ATA8-ACS).

If the SATL receives an ATA PASS-THROUGH (12) command, then the SATL shall check the PROTOCOL field (see table 90) to determine the type of action requested.

Table 90 — PROTOCOL field

Code	Description
0	ATA hardware reset
1	SRST
2	Reserved
3	Non-data
4	PIO Data-In
5	PIO Data-Out
6	DMA
7	DMA Queued
8	Device Diagnostic
9	DEVICE RESET
10	UDMA Data In
11	UDMA Data Out
12	FPDMA <sup>a</sup>
13, 14	Reserved
15	Return Response Information
<sup>a</sup> See SATA 2.5.	

The `PROTOCOL` field specifies the protocol to use when the ATA device executes the command. ATA8-AAM defines the meaning of protocol values ranging from 0 to 11.

If the `PROTOCOL` field specified is in the range from 3 to 12, the SATL shall send an ATA command to the ATA device.

If the `PROTOCOL` field contains 15 (i.e., Return Response Information), then the SATL shall:

- a) if the transport is SATA, read the current Shadow Command Block registers; or
- b) if the transport is PATA, read the current Command Block registers;

and return the contents in the ATA Status Return Descriptor as defined in 12.2.6. The SATL shall ignore all other fields in the CDB.

If the value in the `PROTOCOL` field is inappropriate for the command specified in the `COMMAND` field (see ATA8-ACS), then the SATL may lose communication with the ATA device. This standard does not specify the SATL behavior if this occurs.

If the value in the `PROTOCOL` field is set to zero (i.e., ATA Hardware Reset) and the device is a PATA device, then the SATL shall assert `RST-` (see ATA8-APT). If the value in the `PROTOCOL` field is set to zero (i.e., ATA Hardware Reset) and the device is a SATA device, then the SATL shall issue a `COMRESET` to the SATA device. When this protocol is selected, only the `PROTOCOL` field and the `OFF_LINE` field are valid. The SATL shall ignore all other fields in the CDB.

If the `PROTOCOL` field is set to one, then the SATL shall issue a software reset to the ATA device (see ATA8-AAM). When this protocol is selected, only the `PROTOCOL` field and the `OFF_LINE` field are valid. The SATL shall ignore all other fields in the CDB.

If the value in the `PROTOCOL` field requests the SATL to send a command to the ATA device, then the SATL shall set the fields in the ATA command using fields in the ATA PASS-THROUGH CDB as shown in table 94.

The SATL shall determine if a data transfer is necessary and how to perform the data transfer by examining values in the `MULTIPLE_COUNT` field, `PROTOCOL` field, `OFF_LINE` field, `T_DIR` bit, `BYTE_BLOCK` bit, and `T_LENGTH` field. The SATL shall ignore the `COMMAND` field in the CDB except to copy the `COMMAND` field in the CDB to the Command field in the Register – Host to Device FIS or to the ATA Command register. If the ATA command completes with an error, then the SATL shall return the Error Output fields (see ATA8-ACS) in the ATA Return descriptor (see 12.2.6).

The SATL shall configure the ATA host and the ATA device for the PIO, DMA, and UDMA transfer rates that both the SATL and ATA device support. The SATL should set the transfer rates to the maximum supported by both the SATL and the ATA device. The `COMMAND` field of the CDB may specify the ATA SET FEATURES command. The ATA PASS-THROUGH (12) command should not be used to issue an ATA SET FEATURES command that changes the PIO/DMA/UDMA or other transfer modes of the ATA device. The result of a SET FEATURES command that changes the PIO/DMA/UDMA or other transfer modes of the ATA device is outside the scope of this standard and may cause communication to be lost with the ATA device; preventing the SATL from performing any action based on the contents of the CDB.

The `BYTE_BLOCK` (Byte/Block) bit specifies whether the transfer length in the location specified by the `T_LENGTH` field specifies the number of bytes to transfer or the number of blocks to transfer. If the value in the `BYTE_BLOCK` bit is set to zero, then the SATL shall transfer the number of bytes specified in the location specified by the `T_LENGTH` field. If the value in the `BYTE_BLOCK` bit is set to one the SATL shall transfer the number of blocks specified in the location specified by the `T_LENGTH` field. The SATL shall ignore the `BYTE_BLOCK` bit when the `T_LENGTH` field is set to zero.

The `CK_COND` (Check Condition) bit may be used to request the SATL to return a copy of ATA register information in the sense data upon command completion. If the `CK_COND` bit is set to one the SATL shall return a status of CHECK CONDITION when the ATA command completes, even if the command completes successfully, and return the ATA Normal Output fields (see ATA8-ACS) in the sense data using the ATA Return descriptor (see 12.2.6). If the `CK_COND` bit is set to zero, then the SATL shall terminate the command with CHECK CONDITION status only if an error occurs in processing the command. See clause 11 for a description of ATA error conditions.

The DEVICE field specifies a value for the SATL to load into the ATA Device field. Table 91 shows the bits in the DEVICE field.

**Table 91 — ATA PASS-THROUGH (12) command and ATA PASS-THROUGH (16) command DEVICE field**

Bit							
7	6	5	4	3	2	1	0
Obsolete	Command Specific	Obsolete	DEV	Command Specific			

The SATL shall ignore the DEV bit in the DEVICE field of the CDB.

The SATL shall set the value of the DEV bit in the ATA device register based upon the SATL mapping of ATA devices to I\_T\_L nexuses.

If the PROTOCOL field specifies a PIO data transfer, the SATL shall perform a PIO type transfer. The MULTIPLE\_COUNT field specifies the logarithm base 2 of the number of logical sectors an ATA host shall transfer per DRQ Data Block (e.g, if the field is set to 4, the SATL shall transfer  $2^4$  (i.e., 16) logical sectors of data in each DRQ Data Block). If the MULTIPLE\_COUNT field is nonzero and the COMMAND field is not a READ MULTIPLE command, a READ MULTIPLE EXT command, a WRITE MULTIPLE command, a WRITE MULTIPLE EXT command, or a WRITE MULTIPLE FUA EXT command, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

The OFF\_LINE field specifies the time period during which the ATA Status register and the ATA Alternate Status register may be invalid after command acceptance. In a SATL with a PATA device attached, some commands may cause the PATA device to place the ATA bus in an indeterminate state. This may cause the ATA host to see command completion before the command is completed. When the application client issues a command that is capable of placing the bus in an indeterminate state, it shall set the OFF\_LINE field to a value that specifies the maximum number of seconds from the time a command is issued until the ATA Status register is valid. The SATL shall not use the ATA Status register or ATA Alternate Status register to determine ATA command completion status until this time has elapsed. The valid status is available ( $2^{\text{off\_line}+1} - 2$ ) seconds (i.e., 0, 2, 6, and 14 seconds) after the command register is stored.

NOTE 14 - If the application client specifies an off\_line value that is too small, the results are indeterminate and may compromise the integrity of the data.

If the Transfer Direction (T\_DIR) bit and the direction of the data transfer specified in the PROTOCOL field do not match, the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

If the T\_DIR bit is set to zero, then the SATL shall transfer data from the application client to the ATA device. If the T\_DIR bit is set to one, then the SATL shall transfer data from the ATA device to the application client. The SATL shall ignore the T\_DIR bit if the T\_LENGTH field is set to zero.

The Transfer Length (T\_LENGTH) field specifies where in the CDB the SATL shall locate the transfer length for the command (see table 92).

**Table 92 — T\_LENGTH field**

Code	Description
00b	No data is transferred
01b	The transfer length is an unsigned integer specified in the FEATURES (7:0) field.
10b	The transfer length is an unsigned integer specified in the SECTOR_COUNT (7:0) field.
11b	The transfer length is an unsigned integer specified in the TPSIU (see 3.1.95).

12.2.4 describes the mapping from the FEATURES (7:0) field, the SECTOR\_COUNT (7:0) field, the LBA\_LOW (7:0) field, the LBA\_MID (7:0) field, the LBA\_HIGH (7:0) field, the DEVICE field, and the COMMAND field in the ATA PASS-THROUGH (12) CDB to corresponding ATA command fields (see ATA8-ACS).

### 12.2.3 ATA PASS-THROUGH (16) command

Table 93 shows format of the ATA PASS-THROUGH (16) command.

If the EXTEND bit is set to zero, then the FEATURES (15:8) field, the SECTOR\_COUNT (15:8) field, the LBA\_LOW (15:8) field, the LBA\_MID (15:8) field, and the LBA\_HIGH (15:8) field shall be ignored by the SATL, and the SATL shall process this command as specified in 12.2.2.

If the EXTEND bit is set to one, then the FEATURES (15:8) field, the SECTOR\_COUNT (15:8) field, the LBA\_LOW (15:8) field, the LBA\_MID (15:8) field, and the LBA\_HIGH (15:8) field are valid, and the SATL shall process this command as specified in 12.2.2 except as described in the remainder of this subclause.

**Table 93 — ATA PASS-THROUGH (16) command**

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (85h)							
1	MULTIPLE_COUNT			PROTOCOL				EXTEND
2	OFF_LINE	CK_COND	Reserved	T_DIR	BYTE_BLOCK	T_LENGTH		
3	FEATURES (15:8)							
4	FEATURES (7:0)							
5	SECTOR_COUNT (15:8)							
6	SECTOR_COUNT (7:0)							
7	LBA_LOW (15:8)							
8	LBA_LOW (7:0)							
9	LBA_MID (15:8)							
10	LBA_MID (7:0)							
11	LBA_HIGH (15:8)							
12	LBA_HIGH (7:0)							
13	DEVICE							
14	COMMAND							
15	CONTROL (see 6.4)							

If the EXTEND bit is set to one and the value in the PROTOCOL field requests the SATL to send an ATA command to the device, then the SATL shall send a 48 bit ATA command to the ATA device.

12.2.4 describes the mapping from the FEATURES (15:8) field, the FEATURES (7:0) field, the SECTOR\_COUNT (15:8) field, the SECTOR\_COUNT (7:0) field, the LBA\_LOW (15:8) field, the LBA\_LOW (7:0) field, the LBA\_MID (15:8) field, the LBA\_MID (7:0) field, the LBA\_HIGH (15:8) field, the LBA\_HIGH (7:0) field, the DEVICE field, and the COMMAND field in the ATA PASS-THROUGH (16) CDB to corresponding ATA command fields (see ATA8-ACS).

### 12.2.4 Mapping of ATA PASS-THROUGH CDB field translations

Table 94 shows the mapping between the fields in the ATA PASS-THROUGH (12) CDB and the the ATA PASS-THROUGH (16) CDB to corresponding ATA command fields (see ATA8-ACS).

**Table 94 — Mapping of ATA PASS-THROUGH (16) CDB fields to ATA command fields**

CDB field	48-bit ATA command field <sup>a</sup>	28-bit ATA command field <sup>b</sup>
FEATURES (15:8)	Features (15:8)	n/a
FEATURES (7:0)	Features (7:0)	Features (7:0)
SECTOR_COUNT (15:8)	Count (15:8)	n/a
SECTOR_COUNT (7:0)	Count (7:0)	Count (7:0)
LBA_LOW (15:8)	LBA (31:24)	n/a
LBA_LOW (7:0)	LBA (7:0)	LBA (7:0)
LBA_MID (15:8)	LBA (39:32)	n/a
LBA_MID (7:0)	LBA (15:8)	LBA (15:8)
LBA_HIGH (15:8)	LBA (47:40)	n/a
LBA_HIGH (7:0)	LBA (23:16)	LBA (23:16)
DEVICE (7:4)	Device (7:4)	Device (7:4)
DEVICE (3:0)	Device (3:0)	LBA (27:24)
COMMAND	Command	Command

<sup>a</sup> The 48-bit ATA command translation applies only to the ATA PASS-THROUGH (16) command, and not to the ATA PASS-THROUGH (12) command.

<sup>b</sup> The 28-bit ATA command translation may apply to either the ATA PASS-THROUGH (12) command or the ATA PASS-THROUGH (16) command.

See 12.2.2 for a description of the MULTIPLE\_COUNT field, the PROTOCOL field, the OFF\_LINE field, the CK\_COND bit, the T\_DIR bit, and the BYTE\_BLOCK bit.

The SATL shall determine the transfer length by the method specified in the T\_LENGTH field (see table 95).

**Table 95 — EXTEND bit and T\_LENGTH field**

EXTEND	T_LENGTH	Description
0	00b	No data is transferred.
	01b	The transfer length is an unsigned integer specified in the FEATURES (7:0) field.
	10b	The transfer length is an unsigned integer specified in the SECTOR_COUNT (7:0) field.
	11b	The transfer length is an unsigned integer specified in the TPSIU (see 3.1.95).
1	00b	No data is transferred.
	01b	The transfer length is an unsigned integer specified in the FEATURES (7:0) field and the FEATURES (15:8) field.
	10b	The transfer length is an unsigned integer specified in the SECTOR_COUNT (7:0) field and the SECTOR_COUNT (15:8) field.
	11b	The transfer length is an unsigned integer specified in the TPSIU (see 3.1.95) STPSIU field.

**12.2.5 ATA PASS-THROUGH status return**

Table 96 shows the possible results of ATA PASS-THROUGH (12) command or ATA PASS-THROUGH (16) command processing depending on the value of the CK\_COND bit in the CDB, as reflected in the ERR bit and the DF bit in the ATA Status field.

**Table 96 — ATA command results**

CK_COND	Status field		Sense data returned
	ERR	DF	
0	0	0	No error, successful completion or command in progress. The SATL shall respond to a REQUEST SENSE command and shall return sense data with the sense key set to NO SENSE with the additional sense code set to NO ADDITIONAL SENSE INFORMATION.
1			No error, successful completion or command in progress. The SATL shall terminate the command with CHECK CONDITION status with the sense key set to RECOVERED ERROR with the additional sense code set to ATA PASS-THROUGH INFORMATION AVAILABLE (see SPC-4). The sense data shall include the ATA Status Return Descriptor (see 12.2.5) <sup>a</sup> .
n/a	n/a	1	The ATA command completed with an error. The SATL shall terminate the command with CHECK CONDITION status with the sense key and additional sense code set as described in clause 11 and the sense data shall include the ATA Status Return Descriptor (see 12.2.6).
	1	0	

<sup>a</sup> This capability allows the host to retrieve the ATA register or field information with successful command completion by returning data in the ATA registers or fields.

ATA commands may return information in the ATA registers or the Shadow Command Block. The current ATA register information may be retrieved by requesting the ATA Status Return Descriptor issuing the ATA PASS-THROUGH (12) command or ATA PASS-THROUGH (16) command with the PROTOCOL field set to 15 (i.e., Return Response Information).

### 12.2.6 ATA Return descriptor

Table 97 shows the format of the ATA Return descriptor returned in the sense data (see SPC-3 and SAM-3). The SATL shall return the ATA Return descriptor if the PROTOCOL field in the ATA PASS-THROUGH (12) command or ATA PASS-THROUGH (16) command is set to 15 (i.e., Return Response Information).

The SATL shall support the ATA Return descriptor if the SATL supports the ATA PASSTHROUGH (12) command or the ATA PASS-THROUGH (16) command. Each time the ATA Return descriptor is requested; the SATL shall read the ATA registers and return those values in the sense data as shown in table 97. If the sense data is for an ATA PASS-THROUGH (12) command or for the ATA PASS-THROUGH (16) command with the EXTEND bit set to zero the SATL shall return the 28-bit extended status and shall set the EXTEND bit to zero.

If the sense data is for an ATA PASS-THROUGH (16) command with the EXTEND bit set to one the SATL shall return the 48-bit extended status and shall set the EXTEND bit to one.

**Table 97 — ATA Return descriptor**

Byte\Bit	7	6	5	4	3	2	1	0
0	DESCRIPTOR CODE (09h)							
1	ADDITIONAL DESCRIPTOR LENGTH (0Ch)							
2	Reserved							EXTEND
3	ERROR							
4	SECTOR_COUNT (15:8)							
5	SECTOR_COUNT (7:0)							
6	LBA_LOW (15:8)							
7	LBA_LOW (7:0)							
8	LBA_MID (15:8)							
9	LBA_MID (7:0)							
10	LBA_HIGH (15:8)							
11	LBA_HIGH (7:0)							
12	DEVICE							
13	STATUS							

If the EXTEND bit is set to one, then the SECTOR\_COUNT (7:0) field and SECTOR\_COUNT (15:8) field specify the ATA Sector Count. If the EXTEND bit is set to zero, then the SECTOR\_COUNT (7:0) field specifies the ATA Sector Count and SECTOR\_COUNT (15:8) field shall be ignored.

If the EXTEND bit is set to one, then the LBA\_LOW (7:0) field, LBA\_MID (7:0) field, LBA\_HIGH (7:0) field, LBA\_LOW (15:8) field, LBA\_MID (15:8) field, and LBA\_HIGH (15:8) field specify the ATA LBA. If the EXTEND bit is set to zero, then the LBA\_LOW (7:0) field, LBA\_MID (7:0) field, and LBA\_HIGH (7:0) field specify the ATA LBA, and the LBA\_LOW (15:8) field, LBA\_MID (15:8) field, and LBA\_HIGH (15:8) field shall be ignored.

## 12.3 SAT-specific mode pages

### 12.3.1 SAT-specific mode pages overview

This subclause describes mode pages that the SATL may implement that are unique to the SCSI / ATA Translation standard. These mode pages are for use by the SATL, are shown in table 98, and are described in this subclause.

Support for these mode pages is optional. A SATL shall support the appropriate mode page for the attached ATA environment (e.g., PATA).

**Table 98 — SCSI / ATA Translation specific mode pages**

PAGE CODE	SUB PAGE CODE	Mode page name
0Ah	F1h	PATA Control Mode Page
0Ah	F2h	Reserved

### 12.3.2 PATA Control mode page

The PATA Control mode page provides PATA specific controls for a SATL to configure the underlying PATA host and to understand what parameters are communicated to the PATA device to ensure proper communication for specific transfer rates. This standard specifies the mode parameters that are provided for this mode page.

SATL implementations that support the attachment of PATA devices shall support this mode page. The SATL should allow application clients to configure alternate PATA timings using the MODE SELECT command.

Table 99 shows the PATA Control mode page.

**Table 99 — PATA Control mode page**

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (0Ah)					
1	SUBPAGE CODE (F1h)							
2	(MSB)	PAGE LENGTH (0004h)						
3	(LSB)							
4	Reserved	MWDMA <sup>a</sup> bits			Reserved		PIO <sup>b</sup> bits	
		MWD2	MWD1	MWD0			PIO4	PIO3
5	Reserved	UDMA <sup>c</sup> bits						
		UDMA6	UDMA5	UDMA4	UDMA3	UDMA2	UDMA1	UDMA0
6	Reserved							
7	Reserved							

<sup>a</sup> The Multi-Word Direct Memory Access (MWDMA) bits specify a number of hardware-assisted data transfer modes defined in ATA8-APT.  
<sup>b</sup> PIO stands for Programmed Input and Output and the PIOx bits specify transfer modes performed under program control defined in ATA8-APT.  
<sup>c</sup> The Ultra Direct Memory Access (UDMA) bits represent a number of hardware-assisted data transfer modes defined in ATA8-APT.

The parameters saveable (PS) bit is defined in SPC-3.

The SPF bit (see SPC-3) shall be set to one to access this mode page.

The PAGE CODE field shall be set to 0Ah.

The PAGE LENGTH field shall be set to 0004h.

The SUBPAGE CODE field shall be set to F1h.

SATL implementations may save the state of the timing parameters defined in this mode page.

Application clients may use the MODE SENSE command for changeable values to determine the underlying ATA host support for a given ATA timing mode. The SATL shall support changeable mode parameters for this mode page.

When processing a MODE SENSE command, the SATL shall set the PIO3 bit and PIO4 bit as shown table 100 to identify the configured PIO mode.

**Table 100 — PIO modes**

PIO4	PIO3	PIO mode
0	0	Reserved
0	1	The ATA host shall use PIO mode 3 transfers.
1	0	The ATA host shall use PIO mode 4 transfers.
1	1	Reserved

When changeable values are requested, the PIO3 bit and the PIO4 bit indicate if the underlying ATA host supports those transfer modes. The PIO3 bit shall be set to one if the ATA host supports PIO mode 3. The PIO3 bit and the PIO4 bit shall be set to one if the ATA host supports PIO mode 4.

If the SATL receives a MODE SELECT command and the PIO bits specify a change from the current setting, the SATL shall configure the ATA host to use the new PIO transfer rate, if supported. If the application client requests a PIO setting that the ATA device does not support, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The MWD0 bit, the MWD1 bit, and the MWD2 bit are collectively referred to as the MWDMA bits. If the ATA host in the SATL is currently configured to use multiword DMA (MWDMA), then the MWDMA bits are used to determine what mode is currently being used, what modes are supported by the ATA host, and control of the MWDMA mode.

If the SATL receives a MODE SENSE command requesting the current values of the PATA Control mode page, the MWD0 bit shall be set to one by the SATL when the host and device are configured to use MWDMA mode 0. The MWD1 bit shall be set to one by the SATL when the host and device are configured to use MWDMA mode 1. The MWD2 bit shall be set to one by the SATL when the host and device are configured to use MWDMA mode 2.

If the SATL receives a MODE SENSE command requesting the changeable values of the PATA Control mode page, the MWD0 bit shall be set to one if the ATA host supports MWDMA mode 0. The MWD1 bit and MWD0 bit shall each be set to one if the ATA host supports MWDMA mode 1. The MWD2 bit, the MWD1 bit, and the MWD0 bit shall be each be set to one if the ATA host supports MWDMA mode 2.

Table 101 specifies values set by the SATL in the MWD0 bit, the MWD1 bit, and the MWD2 bit for current and changeable MWDMA settings.

**Table 101 — MWDMA modes reported by MODE SENSE**

MWDMA <sup>a</sup> bits			ATA host and device shared configuration settings returned as current values	ATA host support returned as changeable values
MWD2	MWD1	MWD0		
0	0	0	Configured not to use multiword DMA	Illegal combination
1	0	0	Configured to use MWDMA mode 1	
0	1	0	Configured to use MWDMA mode 2	
1	1	0	Configured to use MWDMA modes 1 and 2	
0	0	1	Configured to use MWDMA mode 0	MWDMA mode 0 supported
1	0	1	Configured to use MWDMA modes 0 and 2	Illegal combination
0	1	1	Configured to use MWDMA modes 0 and 1	MWDMA mode 1 supported
1	1	1	Configured to use MWDMA modes 0, 1 and 2	MWDMA mode 2 supported
<sup>a</sup> If the application client attempts to set a MWDMA mode that is not supported by the ATA host environment, the SATL shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.				

If the SATL receives a MODE SELECT command and the MWDMA bits specify a change from the current settings, then the SATL shall issue an ATA SET FEATURES - Set Transfer Mode (i.e., Features register set to 03h) command to the ATA device to set the MWDMA mode on the ATA device to the requested state, and then:

- a) if the ATA SET FEATURES command completes with an error, then the SATL shall:
  - 1) not change any host transfer modes;
  - 2) complete the MODE SELECT command with a CHECK CONDITION status with the sense key set to ABORTED COMMAND with the additional sense code set to ATA DEVICE FAILED SET FEATURES; and
  - 3) take no further action regarding this request to change the MWDMA transfer rate; or
- b) if the SET FEATURES command completes without error, then the SATL shall:
  - 1) configure the ATA host to communicate with the device at the requested MWDMA transfer rate; and
  - 2) complete the MODE SELECT command with GOOD status.

The MWDMA bits values used to configure ATA hosts and ATA devices using the MODE SELECT command have the same meaning as the MWDMA bits values returned by the MODE SENSE command when current values are requested as shown in table 101.

If the SATL receives a request to set a MWDMA mode that is not supported by the ATA host or the attached PATA device, the SATL shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The UDMA0 bit, the UDMA1 bit, the UDMA2 bit, the UDMA3 bit, the UDMA4 bit, the UDMA5 bit, and the UDMA6 bit are collectively referred to as the UDMA bits, and are used to determine support for, current use of, and control of Ultra DMA (UDMA) transfer rates on the ATA host and device. The SATL shall determine the highest UDMA mode supported as being the lower of the ATA host maximum transfer mode and the device maximum transfer mode.

NOTE 15 - The ATA device returns the UDMA transfer mode specified in ATA IDENTIFY DEVICE data, word 88 bits 6:0 (see ATA8-ACS).

If the SATL receives a MODE SENSE command requesting the changeable values of the PATA Control mode page, the UDMA bits shall be set according to table 102.

**Table 102 — UDMA bits requirements for changeable MODE SENSE parameters**

UDMA6	UDMA5	UDMA4	UDMA3	UDMA2	UDMA1	UDMA0	Highest UDMA mode supported
0	0	0	0	0	0	0	UDMA Unsupported
0	0	0	0	0	0	1	0
0	0	0	0	0	1	1	1
0	0	0	0	1	1	1	2
0	0	0	1	1	1	1	3
0	0	1	1	1	1	1	4
0	1	1	1	1	1	1	5
1	1	1	1	1	1	1	6

If the SATL receives a MODE SENSE command requesting the current values of the PATA Control mode page, then the SATL shall set the UDMA bits as defined in table 103. Only one of the UDMA bits shall be set to one at any time for such a request. If UDMA is not the current DMA transfer mode, then all the UDMA bits shall be set to zero. If a UDMA transfer mode is being used, then all of the MWDMA bits shall be set to zero.

**Table 103 — UDMA for current MODE SENSE settings**

UDMA bit	Value	Description
udma0	0	ATA host and device are not communicating using UDMA Mode 0
	1	ATA host and device are communicating using UDMA Mode 0
udma1	0	ATA host and device are not communicating using UDMA Mode 1
	1	ATA host and device are communicating using UDMA Mode 1
udma2	0	ATA host and device are not communicating using UDMA Mode 2
	1	ATA host and device are communicating using UDMA Mode 2
udma3	0	ATA host and device are not communicating using UDMA Mode 3
	1	ATA host and device are communicating using UDMA Mode 3
udma4	0	ATA host and device are not communicating using UDMA Mode 4
	1	ATA host and device are communicating using UDMA Mode 4
udma5	0	ATA host and device are not communicating using UDMA Mode 5
	1	ATA host and device are communicating using UDMA Mode 5
udma6	0	ATA host and device are not communicating using UDMA Mode 6
	1	ATA host and device are communicating using UDMA Mode 6

When the SATL receives a MODE SELECT command and the UDMA bits request a change in the UDMA transfer rate, then the SATL shall:

- 1) if the SET FEATURES command completes with an error, then the SATL shall:
  - A) not change any host transfer modes;
  - B) complete the MODE SELECT command with a CHECK CONDITION status with the sense key set to ABORTED COMMAND with the additional sense code set to ATA DEVICE FAILED SET FEATURES; and
  - C) take no further action regarding this request to change the UDMA transfer rate;and
- 2) if the SET FEATURES command completes without error, then the SATL shall:
  - A) configure the ATA host to communicate with the device at the requested UDMA transfer rate; and
  - B) complete the MODE SELECT command with GOOD status.

If the application client attempts to set a mode that the ATA host or the ATA device does not support, then the SATL shall terminate the MODE SELECT command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

## 12.4 SAT-specific VPD pages

### 12.4.1 SAT-specific VPD pages overview

This subclause defines VPD pages specific to SAT implementations.

### 12.4.2 ATA Information VPD page

#### 12.4.2.1 ATA Information VPD page overview

The ATA Information VPD page shall contain:

- a) information about the SATL;
- b) Signature of the ATA or ATAPI device; and
- c) ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data from the ATA or ATAPI device.

Some SATLs may modify ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data. If a SCSI application client requires the unmodified ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data, then the ATA PASS-THROUGH command should be used to retrieve the ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data.

Table 104 defines the ATA Information VPD page.

**Table 104 — ATA Information VPD page**

Byte\Bit	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (89h)							
2	(MSB)							
3	PAGE LENGTH (238h)							(LSB)
4	Reserved							
7	Reserved							
8	Reserved							
15	SAT VENDOR IDENTIFICATION							
16	Reserved							
31	SAT PRODUCT IDENTIFICATION							
32	Reserved							
35	SAT PRODUCT REVISION LEVEL							
36	Reserved							
55	ATA device signature (see 12.4.2.2)							
56	COMMAND CODE							
57	Reserved							
59	Reserved							
60	ATA IDENTIFY DEVICE data or							
571	ATA IDENTIFY PACKET DEVICE data (see 12.4.2.3)							

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.

The PAGE CODE field shall be set to 89h.

The PAGE LENGTH field shall be set to 238h.

The SAT VENDOR IDENTIFICATION field shall contain an 8-byte ASCII string identifying the vendor of the SATL. The data shall be left aligned within the field. The vendor identification string shall be one assigned by INCITS for use in the Standard INQUIRY data VENDOR IDENTIFICATION field. A list of assigned vendor identification strings is in SPC-3 and on the T10 web site (<http://www.t10.org>).

The SAT PRODUCT IDENTIFICATION field shall contain sixteen bytes of ASCII data as defined by the vendor of the SATL. The data shall be left-aligned within the field.

The SAT PRODUCT REVISION LEVEL field shall contain four bytes of ASCII data as defined by the vendor of the SATL. The data shall be left-aligned within the field.

The ATA device signature is described in 12.4.2.2.

The COMMAND CODE field contains the of the ATA command used to retrieve the data in the IDENTIFY DEVICE or IDENTIFY PACKET DEVICE DATA field. The possible command codes are:

- a) ECh for an IDENTIFY DEVICE command (i.e., for an ATA device);
- b) A1h for an IDENTIFY PACKET DEVICE command (i.e., for an ATAPI device); or
- c) 00h for other device types.

The ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data is described in 12.4.2.3.

**12.4.2.2 ATA device signature**

The ATA device signature shall contain the contents of the task file registers after the last power-on reset, hardware reset, software reset, or EXECUTE DEVICE DIAGNOSTIC command. The ATA device signature shall follow the format of the initial SATA Device-to-Host Register FIS (see SATA 2.5). Table 105 shows the ATA device signature.

**Table 105 — ATA device signature**

Byte\Bit	7	6	5	4	3	2	1	0
0	TRANSPORT IDENTIFIER							
1	Reserved	INTERRUPT/ Reserved <sup>a</sup>	Reserved		PM PORT / Reserved <sup>a</sup>			
2	STATUS <sup>b</sup>							
3	ERROR <sup>b</sup>							
4	LBA LOW <sup>b</sup>							
5	LBA MID <sup>b</sup>							
6	LBA HIGH <sup>b</sup>							
7	DEVICE <sup>b</sup>							
8	LBA LOW EXP <sup>b</sup>							
9	LBA MID EXP <sup>b</sup>							
10	LBA HIGH EXP <sup>b</sup>							
11	Reserved							
12	SECTOR COUNT <sup>b</sup>							
13	SECTOR COUNT EXP <sup>b</sup>							
14	Reserved							
19	Reserved							
<sup>a</sup> The INTERRUPT bit and the PM PORT field are defined only if the TRANSPORT IDENTIFIER field is set to 34h (see SATA 2.5). Otherwise the INTERRUPT field and the PM PORT field are reserved. <sup>b</sup> These fields are fields with the same names defined in ATA8-ACS.								

The TRANSPORT IDENTIFIER field may contain the values shown in table 106.

**Table 106 — TRANSPORT IDENTIFIER field values**

Code	Transport
00h	PATA (see ATA8-APT)
34h	SATA (see SATA 2.5)
All others	Reserved

The INTERRUPT bit corresponds to the “I” bit (i.e., bit 14 of dword 0) of the Register Device-to-Host FIS (see SATA 2.5).

All the remaining fields within the ATA device signature are defined in ATA8-APT and SATA 2.5.

### 12.4.2.3 ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data

If the command is an ATA IDENTIFY DEVICE command, and the command completes without error, then the IDENTIFY DEVICE OR IDENTIFY PACKET DEVICE DATA field shall contain the ATA IDENTIFY DEVICE data (ATA8-ACS).

If the command is an ATA IDENTIFY PACKET DEVICE command, and the command completes without error, then the IDENTIFY DEVICE OR IDENTIFY PACKET DEVICE DATA field shall contain the IDENTIFY PACKET DEVICE data (see ATA8-ACS).

The IDENTIFY DEVICE OR IDENTIFY PACKET DEVICE DATA field shall contains 512 bytes of 00h if:

- a) the command is an IDENTIFY DEVICE command or an IDENTIFY PACKET DEVICE command and the command completes with an error; or
- b) the command code is 00h (i.e., some other device type).

The data shall be presented with byte preservation (i.e., ATA byte n maps to SCSI byte n), as shown in table 107.

**Table 107 — ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data**

Byte	Contents
0	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 0 bits 7:0 (i.e., byte 0)
1	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 0 bits 15:8 (i.e., byte 1)
2	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 1 bits 7:0 (i.e., byte 2)
3	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 1 bits 15:8 (i.e., byte 3)
...	...
510	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 255 bits 7:0 (i.e., the signature byte of the Integrity word, see ATA8-ACS)
511	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 255 bits 15:8 (i.e., the checksum byte of the Integrity word, see ATA8-ACS)

NOTE 16 - Although the Serial number field (i.e., words 19:10), Firmware revision field (i.e., words 26:23), and Model number field (i.e., words 46:27) contain ASCII characters, every other byte is swapped within them (see ATA8-ACS) (e.g., the Serial number field is interpreted as: {word 10 bits 15:8, word 10 bits 7:0, word 11 bits 15:8, word 11 bits 7:0, ...}, which corresponds to these bytes in the IDENTIFY DEVICE OR IDENTIFY PACKET DEVICE DATA field: {byte 21, byte 20, byte 23, byte 22, etc.}).

Since some of the fields within the ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data may change depending on the state of the ATA device, the SATL shall reissue the ATA IDENTIFY DEVICE command or ATA IDENTIFY PACKET DEVICE command to retrieve updated data whenever the ATA Information VPD page is requested.

## **Annex A** (Informational)

### **INQUIRY command translation for ATAPI devices**

#### **A.1 INQUIRY command overview**

This annex describes elements of SCSI / ATA Translation for the INQUIRY command that are specific to ATAPI devices.

For ATAPI devices, the SATL may support the ATA Information VPD page (see A.3 and 12.4.2) to provide information about the SATL and provide the ATA IDENTIFY PACKET DEVICE data from the ATAPI device.

If the SATL does not support the ATA Information VPD page, then the SATL shall use the ATA PACKET Command feature set to pass all INQUIRY commands and parameter data to the ATAPI device without altering the INQUIRY commands or the parameter data.

If the SATL supports the ATA Information VPD page, then the SATL shall:

- a) use the ATA PACKET Command feature set to pass all INQUIRY commands requesting standard INQUIRY data to an ATAPI device without altering the INQUIRY commands;
- b) use the ATA PACKET Command feature set to pass all INQUIRY commands requesting VPD pages, other than the Supported VPD Pages VPD page and the ATA Information VPD page, to an ATAPI device without altering the INQUIRY commands;
- c) process INQUIRY commands requesting the Supported VPD Pages VPD page (see SPC-3) as described in A.2; and
- d) process INQUIRY commands requesting the ATA Information VPD page (see 12.4.2) as described in A.3.

#### **A.2 Supported VPD Pages VPD page**

If the SATL supports the ATA Information VPD page, then the SATL shall use the ATA PACKET Command feature set to pass all INQUIRY commands requesting the Supported VPD Pages VPD page (see SPC-3) to the ATAPI device without altering the INQUIRY commands.

If the ATAPI device returns CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB, and the field pointer in the sense data points to the EVPD bit or the PAGE CODE field, the SATL shall return GOOD status and return parameter data containing the Supported VPD Pages VPD page with only two supported VPD page codes:

- a) 00h (i.e., Supported VPD Pages); and
- b) 89h (i.e., ATA Information).

If the ATAPI device returns GOOD status, the SATL shall modify the parameter data containing the Supported VPD Pages VPD page returned by the ATAPI device to add 89h (i.e., the ATA Information VPD page) into the supported VPD page code list.

#### **A.3 ATA Information VPD page**

If the SATL supports the ATA Information VPD page (see 12.4.2), and the SATL receives an INQUIRY command requesting the ATA Information VPD page, then the SATL shall process the command (i.e., return parameter data and status) itself and shall not pass the INQUIRY command through to the ATAPI device.