Working Draft American National Standard

Project T10/XXXX-D

Revision 0.0
27 June 2004

Information technology - SCSI / ATA Translation (SAT)

T10 Technical Editor:

Robert L. Sheffield
Intel Corporation
CH6-333
5000 W. Chandler Blvd.
Chandler, AZ
85226
USA

Telephone: 480-554-8597
Facsimile: 480-554-7347
Email: robert.l.sheffield@intel.com

Reference number
ISO/IEC XXXXX-XXX : 200x
ANSI NCITS.*:200x**

Printed Sunday, June 27, 2004 4:37 PM

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: 2C6
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

T10 Reflector To subscribe send e-mail to majordomo@T10.org with ‘subscribe’ in message body
To unsubscribe send e-mail to majordomo@T10.org with ‘unsubscribe’ in message body
Internet address for distribution via T10 reflector: 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 EngineeringDocuments, an IHS Company
15 Inverness Way East
Englewood, CO 80112-5704

Telephone:303-792-2181 or
800-854-7179
Facsimile: 303-792-2192

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

ANSI (r)
NCITS.*:200x**

Draft

American National Standards
for Information Systems -

SCSI / ATA Translation (SAT)

Secretariat
National Committee for Information Technology Standards

Approved mm dd yy

American National Standards Institute, Inc.

Abstract

This standard specifies elements of translation between SCSI and ATA protocol for storage controllers that emulate SCSI capabilities using ATA devices, and a translation layer to provide capabilities defined by other SCSI standards, particularly those defined in SCSI Block Commands (SBC-2) and SCSI Primary Commands (SPC-3). For the purposes of this standard, ATA device capabilites are defined by ATA/ATAPI-7.

Special thanks to Samantha Ranaweera and Stephen Johnson, both of LSI Logic, for providing the substantive content that formed the basis of the first working draft of this standard.

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 confirming 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 and following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. 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

Printed in the United States of America

Contents

Page

Tables

Figures

Revision Information

R.1 Revision History

First release of SAT. Substantive content copied from T10/04-136r0, “SCSI to ATA Command Translations”, written by Samantha Ranaweera and Stephen Johnson of LSI Logic Corporation.

Foreword

This foreword is not part of American National Standard NCITS.***: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. In many instances the capabilities provided by the SATL and ATA device combination represent a functional subset of common SCSI capabilities due to the functional constraints of the general approach. There is also a range of optional emulated SCSI capabilities that may be supported or not, depending on the capabilites of the SATL. This standard defines only those SATL capabilities and services that require use of ATA services and capabilities in the provision of SCSI capabilities. It is recognized that there are other SCSI capabilities that may be implemented in the SATL layer that do not depend upon services available in the attached ATA device(s), but such SCSI services are beyond the scope of this standard.

This standard defines SATL capabilities in terms of SCSI capabilites 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 NCITS.***:200x, 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:

Global EngineeringTelephone:303-792-2181 or

15 Inverness Way East800-854-7179

Englewood, CO 80112-5704Facsimile: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. At the time of it approved this standard, INCITS had the following members:

<<Insert INCITS member list>>

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

Paul D. Aloisi
Dexter Anderson
Yutaka Arakawa
Dan Colegrove
Phil Colline
Roger Cummings
Zane Daggett
Claudio DeSanti
Rob Elliott
Paul Entzel
Mark Evans
Ashlie Fan
Mike Fitzpatrick
Bill Galloway
Nathan Hastad
David Hawks
Emily Hill
Gerald Houlder
Skip Jones
James A. Lott, Jr.
Kevin Marks
Ron Mathews
Shalliam P. McFerrin
Jay Neer
Terence J. Nelson
Robert H. Nixon
Vit Novak
Erich Oetting
David Peterson
Gary S. Robinson
Robert Sheffield
Robert Snively
Tim Symons
Pat Thaler
Douglas Wagner
Michael Wingard

I. Dal Allan (Alt)
Dennis Appleyard (Alt)
Charles Binford (Alt)
David Black (Alt)
Craig W. Carlson (Alt)
Doug Cole (Alt)
Jim Coomes (Alt)
Martin Czekalski (Alt)
Werner Glinka (Alt)
Shalliam Ham (Alt)
Kenneth Hirata (Alt)
Keith Holt (Alt)
Jim Jones (Alt)
Jerry Kachlic (Alt)
Robert Kando (Alt)
Tasuku Kasebayashi (Alt)
Ben-Koon Lin (Alt)
Tim Mackley (Alt)
Fabio Maino (Alt)
John Majernik (Alt)
Ron Martin (Alt)
Jeff Mastro (Alt)
Andy Nowak (Alt)
Elizabeth Rodriguez (Alt)
John P. Scheible (Alt)
Phil Shelton (Alt)
Cris Simpson (Alt)
Himmeler Themistocle (Alt)
John Tyndall (Alt)
Rudolf Vitti (Alt)
Dean Wallace (Alt)
Steven Wilson (Alt)
Steve Wong (Alt)

Introduction

The SCSI / ATA Translation (SAT) standard is divided into ten 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 page and log page information to selected ATA protocol elements.
Clause 11 describes the mapping of SMART capabilities from ATA to SCSI.
Clause 12 describes error reporting and sense data conventions for SCSI/ATA translation.
Appendix A provides informative text to assist in the implementation of SCSI/ATA translation.

American National Standard INCITS.*:200x**

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

1Scope

This standard defines the protocol requirements of the SCSI / ATA Translation Layer (SATL) to allow conforming SCSI / ATA translating elements to inter-operate with various ATA devices and various SCSI application layers. The SATL covers the range of implementations that use ATA-class devices (Paralell ATA, Serial-ATA, and ATAPI devices of both types) to emulate the behavior of SCSI devices as viewed by the SCSI application layer. The objectives of the SATL are:

a) To provide host computers with device independence with respect to the ATA devices (Serial-ATA and PATA) used to provide block storage capacity, and with respect to various implementations of the translation layer used to emulate the behavior of SCSI devices. The common elements defined in this standard represent a subset of the capabilities normally available in enterprise-class SCSI devices. In addition this standard defines common methods to mangage aspects of ATA devices that do not map directly to defined elements of SCSI when it is deemed essential to provide for such differences. Provision is made for the addition of special features and functions through the use of vendor-specific options. Reserved areas are provided for future standardization.
b) To provide consistent means for discovery and control of optional SCSI features that may or may not be emulated in SCSI/ATA translator implementations.
c) To specify handling of specific transport elements as required to represent a mixed-domain topology in a manner consistent with managemnt of devices in a SCSI domain (e.g. SATA devices attached through a SAS domain).

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

.

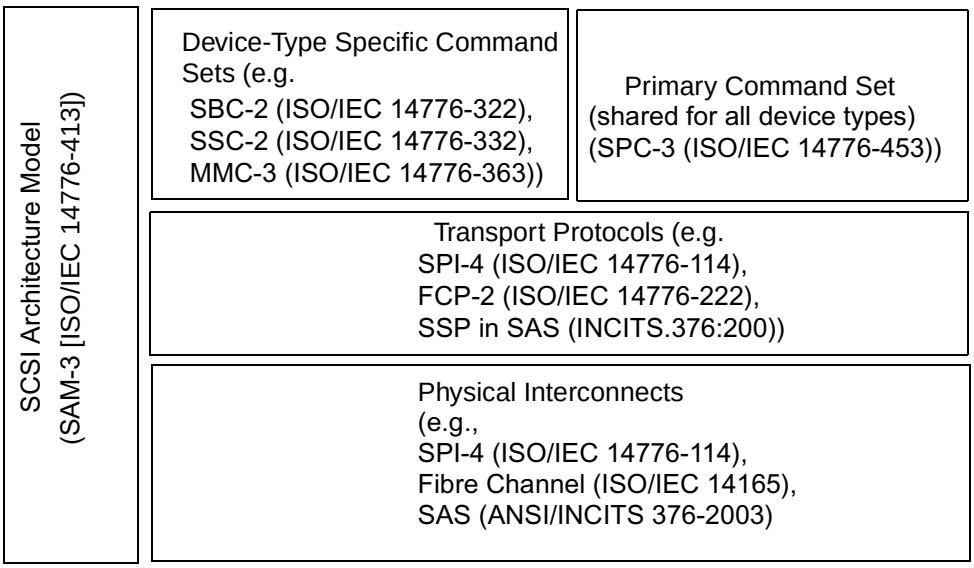

Figure 1 — General Document Structure of SCSI

The term SCSI is used wherever it is not necessary to distinguish between the versions of SCSI. The Small Computer System Interface - 2 (ANSI X3.131-1994) is referred to herein as SCSI-2.

Figure 2 shows the relationship of the ATA/ATAPI-7 documents to each other.

.

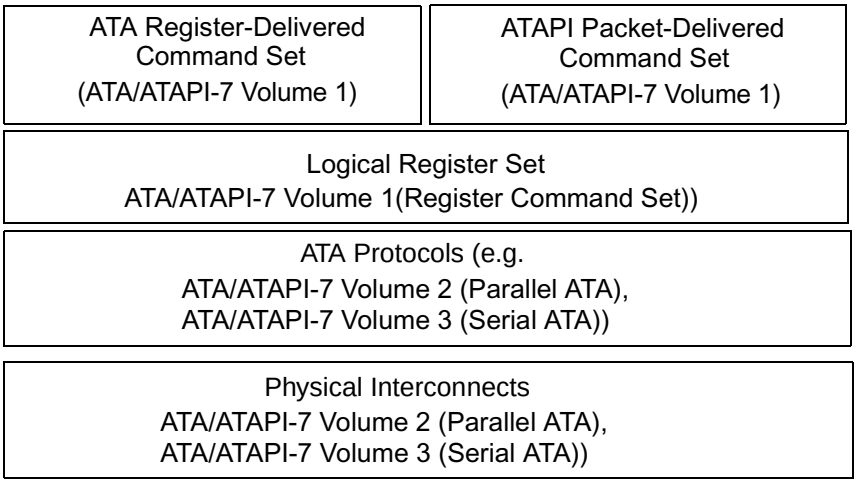

Figure 2 — ATA Document Structure

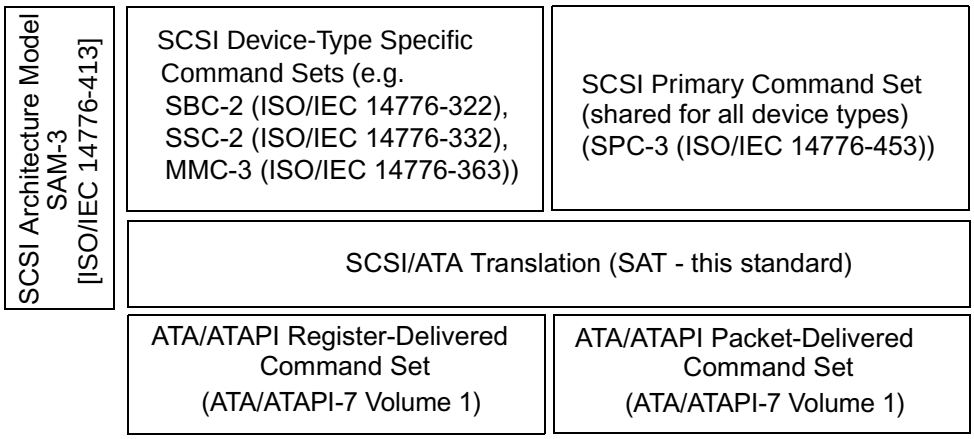

Figure 3 — SCSI/ATA Translation Document Role

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

2 Normative References

Editor's Note 1: This list shall be updated as required.

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

ID Number	Title
ISO/IEC 14776	Multipart SCSI standard
ISO/IEC 14776-113	SCSI Parallel Interface-3 standard
ISO/IEC 14776-112	SCSI Parallel Interface-2 standard
ISO/IEC 14776-411	SCSI-3 Architecture Model (SAM)
ISO/IEC 14776-311	SCSI-3 Primary Commands (SPC)
X3.270:1996 [R2001]	SCSI-3 Architecture Model (SAM)
INCITS.366:2003	SCSI Architecture Model - 2 (SAM-2)
X3.301-1997	SCSI-3 Primary Commands (SPC)
NCITS.351:2001	SCSI Primary Commands - 2 (SPC-2)
NCITS.306:1998	SCSI-3 Block Commands (SBC)
NCITS.330-2000	Reduced Block Commands (RBC)
NCITS.330-2000/AM1-2003	RBC Amendment-1
NCITS.360:2002	MultiMedia Command Set-3 (MMC-3)
INCITS.380:2003	SCSI Stream Commands - 2 (SSC-2))
INCITS.376:2003	Serial Attached SCSI (SAS)

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.

ID Number	Title
T13/1532D	AT Attachment - 7 with Packet Interface (ATA/ATAPI - 7)
T13/____D	AT Attachment - 7 with Packet Interface (ATA/ATAPI - 8)
T10/1561-D	SCSI Architecture Model - 3 (SAM-3)
T10/1683-D	SCSI Architecture Model -4 (SAM-4)
T10/1416-D	SCSI Primary Commands - 3 (SPC-3)
T10/1417-D	SCSI Block Commands - 2 (SBC-2)
T10/1545-D	MultiMedia Command Set-4 (MMC-4)
T10/1675-D	MultiMedia Command Set-4 (MMC-5)

T10/1611-D	SCSI Stream Commands - 3 (SSC-3))
T10/1601-D	Serial Attached SCSI - 1.1 (SAS-1.1)

NOTE 1 IFor more information on the current status of the document, contact the INCITS Secretariat at 202-737-8888 (phone), 202-638-4922 (fax) or via Email at 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.

SFF-8485, Serial GPIO (SGPIO) Bus

NOTE 2 For more information on the current status of the document, contact the SFF committee at 408-867-6630 (phone), or 408-867-2115 (fax). To obtain copies of this document, contact the SFF committee at 14426 Black Walnut Court, Saratoga, CA 95070 at 408-867-6630 (phone) or 408-741-1600 (fax).

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 affiliation: STP target port state of limiting acceptance of connection requests to those from a single STP initiator port.

3.1.2 application client: An object that is the source of SCSI commands. Further definition of an application client is found in the SCSI Architecture Model-2 standard

3.1.3 asynchronous event notification: An optional procedure used by SCSI target devices to notify SCSI initiator devices of events that occur when a pending task does not exist for that SCSI initiator device.

3.1.4 AT Attachment (ATA): A standard for the internal attachment of storage devices to hosts.

3.1.5 ATA device: A storage peripheral (analogous to a SCSI target device).

3.1.6 ATA domain: An I/O system consisting of a set of ATA hosts and ATA devices that communicate with one another by means of a service delivery subsystem.

3.1.7 ATA host: A host device that originates requests to be processed by an ATA device (analogous to a SCSI initiator device).

3.1.8 auto-contingent allegiance: An optional condition of a task set following the return of a CHECK CONDITION status. See the SCSI Architecture Model-2 standard for a detailed definition of auto-contingent allegiance.

3.1.9 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).

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

3.1.11 character: A sequence of ten contiguous bits considered as a unit. A byte is encoded as a character using 8b10b coding (see 6.2).

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

3.1.13 confirmation: A message passed from a lower layer state machine to a higher layer state machine, usually responding to a request (see 3.1.102) from that higher layer state machine, and sometimes relaying a response (see 3.1.104) from a peer higher layer state machine.

3.1.14 contingent allegiance: An optional condition of a task set following the return of a CHECK CONDITION status. A detailed definition of contingent allegiance may be found in the SCSI Architecture Model-2 standard.

3.1.15 cyclic redundancy check (CRC): An error checking mechanism that checks data integrity by computing a polynomial algorithm based checksum.

3.1.16 deadlock: A condition in which two or more processes (e.g., connection requests) are waiting on each other to complete, resulting in none of the processes completing.

3.1.17 device server: An object within the logical unit that executes SCSI tasks according to the rules for task management as described in the SCSI Architecture Model-2 standard.

3.1.18 discover process: The algorithm used by a management application client to configure the SAS domain.

3.1.19 domain: A SCSI domain, or an ATA domain.

3.1.20 event notification: A message passed from the transport layer to the application layer notifying the application layer that a SCSI event has occurred.

3.1.21 field: A group of one or more contiguous bits.

3.1.22 frame: A sequence of data dwords between a start of frame primitive (e.g., SOF, SOAF, or SATA_SOF) and an end of frame primitive (e.g., EOF, EOAF, or SATA_EOF).

3.1.23 frame information structure (FIS): The SATA frame format.

3.1.24 hard reset sequence: A sequence that causes a hard reset (see 4.4).

3.1.25 hard reset: A SCSI target device action in response to a reset event in which a SCSI target device performs the operations described in .

3.1.26 hash function: A mathematical function that maps values from a larger set of values into a smaller set of values, reducing a long value into a shorter hashed value.

3.1.27 I_T nexus loss: A condition where a SAS port determines that another SAS port is no longer available.

3.1.28 I_T nexus: A nexus that exists between a SCSI initiator port and a SCSI target port.

3.1.29 I_T_L nexus: A nexus that exists between a SCSI initiator port, a SCSI target port, and a logical unit. This relationship extends the prior I_T nexus.

3.1.30 I_T_L_Q nexus: A nexus between a SCSI initiator port, a SCSI target port, a logical unit, and a queue tag following the successful receipt of a queue tag. This relationship replaces the prior I_T nexus or I_T_L nexus.

3.1.31 indication: A message passed from a lower layer state machine to a higher layer state machine, usually relaying a request (see 3.1.102) from a peer higher layer state machine.

3.1.32 indication: The second step of a four step confirmed service in reply to a request.

3.1.33 information unit (IU): The portion of an SSP frame that carries command, task management function, data, response, or transfer ready information.

3.1.34 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).

3.1.35 link reset sequence: For SATA, a phy reset sequence (see 3.1.87). For SAS, a phy reset sequence followed by an identification sequence (see 3.1.61), or a phy reset sequence followed by a hard reset sequence (see 3.1.53), another phy reset sequence, and an identification sequence.

3.1.36 link reset: Performing the link reset sequence (see 3.1.70).

3.1.37 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).

3.1.38 livelock: A condition where two or more processes (e.g., connection requests) continually change their state in response to changes in other processes, resulting in none of the processes completing.

3.1.39 logical unit number: An identifier for a logical unit.

3.1.40 logical unit reset event: An event that triggers a logical unit reset from a logical unit as described in SCSI Architecture Model-2.

3.1.41 logical unit reset: A logical unit action in response to a logical unit reset event in which the logical unit performs the operations described in SCSI Architecture Model-2.

3.1.42 logical unit: An externally addressable entity within a SCSI target device. See the SCSI Architecture Model-2 standard for a detailed definition of a logical unit.

3.1.43 media: Particular elements comprising the interconnect including copper cables, printed circuit boards, and other transmission line materials.

3.1.44 message: Information sent between state machines.

3.1.45 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).

3.1.46 nexus: A relationship between a SCSI initiator port and a SCSI target port that may extend to a logical unit and a queue tag.

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

3.1.48 phy: A device object that is used to interface to other devices.

3.1.49 port: A SAS port or an expander port. Each port contains one or more phys. (Editor's Note: Needs a broader definition)

3.1.50 port: A single attachment to a SCSI bus segment from a SCSI device.

3.1.51 power on: Power being applied.

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

3.1.53 receiver: The recipient of a signal.

3.1.54 request: A message passed from a higher layer state machine to a lower layer state machine, usually to initiate some action.

3.1.55 reset event: An event that triggers a hard reset (see 4.4.2) from a SAS device.

3.1.56 response: A message passed from a higher layer state machine to a lower layer state machine, usually in response to an indication (see 3.1.64).

3.1.57 SAS address: A worldwide unique name assigned to a SAS initiator port, SAS target port, expander device, SAS initiator device, or SAS target device.

3.1.58 SAS device: A SAS initiator device, SAS target device, or SAS target/initiator device.

3.1.59 SAS domain: The I/O system defined by this standard that may serve as an ATA domain and/or a SCSI domain.

3.1.60 SAS initiator device: A device containing SSP, STP, and/or SMP initiator ports in a SAS domain.

3.1.61 SAS initiator port: An SSP initiator port, STP initiator port, and/or SMP initiator port in a SAS domain.

3.1.62 SAS port: A SAS initiator port, SAS target port, or SAS target/initiator port.

3.1.63 SAS target device: A device containing SSP, STP, and/or SMP target ports in a SAS domain.

3.1.64 SAS target port: An SSP target port, STP target port, and/or SMP target port in a SAS domain.

3.1.65 SAS target/initiator device: A device that has all the characteristics of a SAS target device and a SAS initiator device.

3.1.66 SAS target/initiator port: A port that has all the characteristics of a SAS target port and a SAS initiator port in a SAS domain.

3.1.67 SATA device port: A storage device object in an ATA domain that interfaces to the service delivery subsystem with SATA (analogous to a SCSI target port).

3.1.68 SATA device: A storage device that contains a SATA device port in an ATA domain (analogous to a SCSI target device).

3.1.69 SATA host port: A host device object in an ATA domain that interfaces to the service delivery subsystem with SATA (analogous to a SCSI initiator port).

3.1.70 SATA host: A host device containing a SATA host port in an ATA domain (analogous to a SCSI initiator device).

3.1.71 SATA phy: A phy in a SATA device that interfaces to a service delivery subsystem (analogous to a SAS phy).

3.1.72 SATA port selector: 3.1.73 A device that attaches to two SATA hosts and one SATA device, and provides the means for one SATA host to access the device at any given time (see SATA II: Port Selector).

3.1.74 SATL: SCSI / ATA Translation Layer - the functional layer which provides the elements to emulate the behavior of a SCSI target device as percieved by a SCSI application client (see SAM-3), and implements the basic storage capabilities through control of attached ATA storage devices, in the manner specified by this standard.

3.1.75 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.

3.1.76 SCSI domain: An I/O system consisting of a set of SCSI devices that communicate with one another by means of a service delivery subsystem.

3.1.77 SCSI initiator device: A SCSI device containing application clients and SCSI initiator ports that originates device service and task management requests to be processed by a SCSI target device and receives device service and task management responses from SCSI target devices. See the SCSI Architecture Model-2 standard for a detailed definition of a SCSI initiator device.

3.1.78 SCSI initiator port: A SCSI initiator device object acts as the connection between application clients and the service delivery subsystem through which requests and responses are routed. See the SCSI Architecture Model-2 standard for a detailed definition of a SCSI initiator port.

3.1.79 SCSI port: A SCSI initiator port or a SCSI target port.

3.1.80 SCSI target device: A SCSI device containing logical units and SCSI target ports that receives device service and task management requests for processing See the SCSI Architecture Model-2 standard for a detailed definition of a SCSI target device.

3.1.81 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 the service delivery subsystem through which requests and responses are routed. See the SCSI Architecture Model-2 standard for a detailed definition of a SCSI target port.

3.1.82 : 3.1.83 SCSI target/initiator device: A device that has all the characteristics of a SCSI target device and a SCSI initiator device.

3.1.84 SCSI target/initiator port: A SCSI target/initiator device object that has all the characteristics of a SCSI target port and a SCSI initiator port.

3.1.85 Serial ATA (SATA): The protocol defined by ATA/ATAPI-7 V3 (see 2.4).

3.1.86 Serial ATA Tunneled Protocol (STP): The protocol defined in this standard used by STP initiator ports to communicate with STP target ports in a SAS domain.

3.1.87 Serial Attached SCSI (SAS): The set of protocols and the interconnect defined by this standard.

3.1.88 Serial Management Protocol (SMP): The protocol defined in this standard used by SAS devices to communicate management information with other SAS devices in a SAS domain.

3.1.89 Serial SCSI Protocol (SSP): The protocol defined in the SAS standard used by SCSI initiator ports to communicate with SCSI target ports in a SAS domain.

3.1.90 service delivery subsystem: The part of a SCSI I/O system that transmits information between a SCSI initiator port and a SCSI target port, or the part of an ATA I/O system that transmits information between an ATA host and an ATA device, or the part of a SAS I/O system that transmits information between a SAS initiator port and a SAS target port.

3.1.91 SMP initiator port: A SAS initiator device object in a SAS domain that interfaces to the service delivery subsystem with SMP.

3.1.92 SMP port: An SMP initiator port, SMP target port, or SMP target/initiator port.

3.1.93 SMP target port: A SAS target device object in a SAS domain that interfaces to the service delivery subsystem with SMP.

3.1.94 SMP target/initiator port: A port that has all the characteristics of an SMP initiator port and an SMP target port.

3.1.95 SSP initiator port: A SCSI initiator port in a SAS domain that implements SSP.

3.1.96 SSP port: An SSP initiator port, SSP target port, or SSP target/initiator port.

3.1.97 SSP target port: A SCSI target port in a SAS domain that implements SSP.

3.1.98 SSP target/initiator port: A port that has all the characteristics of an SSP initiator port and an SSP target port.

3.1.99 STP initiator port: A SAS initiator device object in a SAS domain that interfaces to the service delivery subsystem with STP.

3.1.100 STP port: An STP initiator port, STP target port, or STP target/initiator port.

3.1.101 STP target port: A SAS target device object in a SAS domain that interfaces to the service delivery subsystem with STP.

3.1.102 STP target/initiator port: A port that has all the characteristics of an STP initiator port and an STP target port.

3.1.103 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.

3.1.104 target: Synonymous with SCSI target port.

3.1.105 task management function: A task manager service capable of being requested by an application client to affect the processing of one or more tasks.

3.1.106 task management function: A task manager service that may be invoked by a task management message or by setting one of the task management flags in a SPI L_Q information unit to affect the execution of one or more tasks.

3.1.107 task manager: An agent within the device server that executes task management functions.

3.1.108 task set: A group of tasks within a device server, whose interaction is dependent on the task management, contingent allegiance and auto-contingent allegiance rules. See the SCSI Architecture Model-2 standard for a detailed definition of a task set.

3.1.109 task: An object within the logical unit representing the work associated with a command or group of linked commands. A task consists of one initial connection and zero or more physical or logical reconnections, all pertaining to the task.

3.1.110 transport protocol service confirmation: A message passed from the transport layer to the application layer (i.e., from the SSP initiator port to the SCSI application client) that notifies the application layer that a SCSI transport protocol service has completed.

3.1.111 transport protocol service indication: A message passed from the transport layer to the application layer notifying the application layer (i.e., from the SSP target port to the SCSI device server) to begin a SCSI transport protocol service.

3.1.112 transport protocol service request: A message passed from the SCSI application layer to the SSP transport layer (i.e., from the SCSI application client to the SCSI initiator port) to begin a SCSI transport protocol service.

3.1.113 transport protocol service response: A message passed from the application layer to the transport layer (i.e., from the SCSI device server to the SSP target port) that completes the SCSI transport protocol service.

3.1.114 vendor-specific: Something (e.g., a bit, field, code value) that is not defined by this standard and may be used differently in various implementations.

3.1.115 zero: A false signal value or a false condition of a variable.

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
AWG	American wire gauge
CA	Contingent allegiance
CMOS	Complementary metal oxide semiconductor
CRC	Cyclic Redundancy Check
DUT	Device under test
EMI	Electromagnetic interference
EMC	Electromagnetic compatibility
ESD	Electrostatic discharge
ISI	Intersymbol interference
LSB	Least significant bit
LUN	Logical unit number
MSB	Most significant bit
SCSI	Small Computer System Interface
SAM-2	SCSI Architecture Model-2
SAM-3	SCSI Architecture Model-3
SCSI-2	Small Computer System Interface - 2
SCSI-3	Small Computer System Interface - 3
SPC-2	SCSI Primary Commands-2

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 to claim compliance with 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 prior SCSI standards but has been removed from this 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. 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 the phrase "it is recommended".

3.4 Conventions

Editor's Note 2: Make sure that the conventions below are true for your project or edit as appropriate.

Certain words and terms used in this American National Standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in clause 3 or in the text where they first appear. Names of signals, phases, messages, commands, statuses, SENSE KEYS, ADDITIONAL SENSE CODES, and ADDITIONAL SENSE CODE qualifiers are in all uppercase (e.g., REQUEST SENSE), names of fields are in small uppercase (e.g., STATE OF SPARE), lower case is used for words having the normal English meaning.

Fields containing only one bit are usually referred to as the name bit instead of the name field.

Numbers that are not immediately followed by lower-case b or h are decimal values.

Numbers immediately followed by lower-case b (xxb) are binary values.

Numbers immediately followed by lower-case h (xxh) are hexadecimal values.

Decimals are indicated with a comma (e.g., two and one half is represented as 2,5).

Decimal numbers having a value exceeding 999 are represented with a space (e.g., 24 255).

An alphanumeric list (e.g., a,b,c or A,B,C) of items indicate the items in the list are unordered.

A numeric list (e.g., 1,2,3) of items indicate the items in the list are ordered (i.e., item 1 must occur or complete before item 2).

In the event of conflicting information the precedence for requirements defined in this standard is:

1) text,
2) tables, then
3) figures.

3.5 Notation for Procedures and Functions

Editor's Note 3: Should the SCSI/SATA translation be specified in terms of procedure calls? If a table-format is used, the general format of the tables shall be defined here.

Procedure Name ([input:1a|input:1b|input:1c][,input:2a+input:2b]...[input:n]||
[output:1][,output:2]...[output:n])

Where

Procedure Name:	A descriptive name for the function to be performed.
"(...)":	Parentheses enclosing the lists of input and output arguments.
input:1a|input:1b|...	A number of arguments of which only one shall be used in any single procedure
input:1, input:2, ...:	A comma-separated list of names identifying caller-supplied input data objects.
output:1, output:2, ...:	A comma-separated list of names identifying output data objects to be returned by the procedure.
"||":	A separator providing the demarcation between inputs and outputs. Inputs are listed to the left of the separator; outputs, if any, are listed to the right.
"[...]":	Brackets enclosing optional or conditional parameters and arguments.
"|":	A separator providing the demarcation between a number of arguments of which only one shall be used in any single procedure.
"+":	A collection of objects presented to a single object. No ordering is implied.

4 General

TThe SCSI / ATA Translation defines standard mappings and behaviors among implementations that effect the behavior of SCSI devices as viewed by a host driver where the physical devices are ATA class devices presented to the host by applying a translation layer between the Serial ATA or Parallel ATA device and the SCSI application client interface.

The following items are defined.

1) Translation of selected SCSI commands implemented using ATA devices:
2) Specification of sense data reporting;
3) Mode and Log pages applicable to SCSI devices emulated using ATA devices;
4) Usage of Task Management functions for SCSI emulation using ATA devices;
5) Mapping of SMART functions;
6) Elements to facilitate use of SATA port selectors and SATA port multipliers;
7) Elements to provide consistent mapping of ATAPI devcies as SCSI devices;
8) Other capabilities that may fit within the scope of this project.

Editor's Note 4: This clause shall contain a description of what SAT is doing and setting the ground work for the rest of the clauses/annexes that contain the remaining normative/informative information for the project. This section may need some elaboration.

---

5 SCSI Architectural Elements

This section defines SCSI/ATA translation elements that impact the representation of the storage domain in terms of elements defined in the SCSI Architecture Model-3 (SAM-3).

5.1 Overview

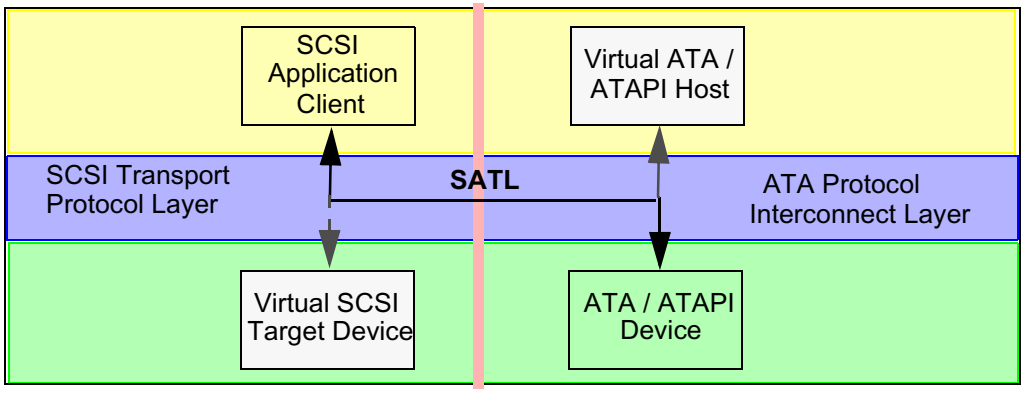

Figure 4 — SAT Functional Protocol Reference Model

Figure 4 shows the role of the SCSI/ATA Translation Layer (SATL) in connecting SCSI Application Clients to ATA/ATAPI devices by providing emulated SCSI Logical Units as percieved by the SCSI Application Clients, and emulated ATA/ATAPI Hosts as perceived by the ATA/ATAPI devices, and the necessary translation elements to link the elements together. This standard defines SCSI/ATA translation from the standpoint of the corresponding 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.

Flexibility is allowed to accomodate various mixed transport mechanisms, including the use of defined SCSI transports such as Fibre-Channel and SCSI Parallel Interface, wherein the SATL exists within an actual SCSI target comprised of ATA devices to provide the basic storage. Alternatively a system implemented in accordance to this standard could involve an ATA/ATAPI Host Bus Adapter (HBA) with ATA/ATAPI devices connected directly, but which provides instead of an ATA/ATAPI register interface to the host, instead provides SCSI Transport Protocol layer services to a SCSI Application Client in accordance with SAM-3. Yet another implementation consistent with this standard could employ theSerial Attached SCSI transport using STP protocol to connect ATA/ATAPI devices to STP initiators. The STP initiator would then provide the SATL function to present a SCSI Transport Protocol layer services to the application client as would a SCSI HBA.

5.2 Queue commands

SCSI provides for various modes of command queueing via command tags and ORDERED, SIMPLE, HEAD OF QUEUE, and ACA task attributes. SCSI devices do not report the maximum number of commands that may be queued, but report a status of BUSY or TASK SET FULL when a command is received and the queue has no space to process another command. Native command queueing (NCQ) in SATA supports up to 32 commands queued at a time.

Editor's Note 5: As the standard currently reads, use of NCQ by the SATL is prohibited, but the SATL may queue commands itself. It should at least be considered whether there is a reasonable model to optionally make use of NCQ capabilities of SATA devices that support NCQ in SAT implementations.

---

6 Task Management Model

SAT devices shall not support the full task management model, but may support the basic task management model as defined by SAM-3 (see SAM-3 8.3 Task Management Models). Elements of the basic task management model (e.g. SIMPLE task attribute and ORDERED task attribute) may be provided through the use of SATA Native Command Queueing, queueing within the SATL itself, or a combination thereof.

Editor's Note 6: The original proposal (04-136r0) forbade the use of NCQ. This could be left in as an optional capability.

6.1 Task Management Functions

This section describes the translation of SCSI task management messages to ATA or SATA equivalents.

NOTE 3 - Due to architectural differences, not all task management messages can be successfully translated to ATA commands or control operations. Please refer SPI-4 specification for specific details of task management operations.

6.1.1 ABORT TASK

ABORT TASK shall not be supported for ATA devices. The SATL shall return a status of FAILED for the ABORT TASK request. Previously established conditions such as ACA, mode parameters and reservations (supported?) shall not change. The particular task in consideration may complete, successfully or not, with or without a successful completion status.

Editor's Note 7: FIAILED is not a valid "service response" to a Task Management Request per SAM-3. Valid responses are:

FUNCTION COMPLETE:	The requested function has been completed.
FUNCTION SUCCEEDED:	The requested function is supported and completed successfully.
FUNCTION REJECTED:	The task manager does not implement the requested function.
INCORRECT LOGICAL UNIT NUMBER:	An optional task manager response indicating that the function requested processing for an incorrect logical unit number.
SERVICE DELIVERY OR TARGET FAILURE:	The request was terminated due to a service delivery failure or SCSI target device malfunction. The task manager may or may not have successfully performed the specified function.

Depending on the situation, the service response could either be FUNCTION REJECTED, or SERVICE DELIVERY OR TARGET FAILURE, depending on whether there's a transport that would recognize a similar function. What happens, for example, if the message is sent to a target device with an actual SCSI interface, but that is a "bridge" to an attached ATA devcie? Does the SCSI transport provide the means to send the FUNCTION REJECTED reply? In SAS, for example, the target would return a SAS RESPONSE information unit with DATAPRES set and a RESPONSE CODE of TASK MANAGEMENT FUNCTION NOT SUPPORTED. These are transport-specific, and should be specified in this standard using the SAM-3 "service response" values.

6.1.2 ABORT TASK SET

ABORT TASK SET shall not be supported for ATA devices. The SATL shall return a status of FAILED for the ABORT TASK SET request. Previously established conditions such as mode parameters and reservations shall not change. The particular tasks in consideration may complete, successfully or not, with or without a successful completion status.

Editor's Note 8: For SATA devices that support Native Command Queueing, outstanding commands may be aborted by issueing a NOP command.

6.1.3 CLEAR ACA

Editor's Note 9: Not sure if this is relevant. Should SAT optionally allow the SATL to implement ACA?

6.1.4 CLEAR TASK SET

CLEAR TASK SET shall not be supported for SATA devices. The SATL shall return a status of FAILED for the CLEAR TASK SET request. Previously established conditions such as mode parameters and reservations shall not change. The particular tasks in consideration may complete, successfully or not, with or without a successful completion status.

6.1.5 LOGICAL UNIT RESET

LOGICAL UNIT RESET shall not be supported for ATA devices. The SATL shall return a status of FAILED for the LOGICAL UNIT RESET request. Previously established conditions such as mode parameters and reservations shall not change. Tasks submitted for this logical unit may complete, successfully or not, with or without a successful completion status.

Editor's Note 10: Could a "Soft Reset" be issued to an ATA device in this case?

6.1.6 QUERY TASK

QUERY TASK shall not be supported for ATA devices. The SATL shall return a status of FAILED for the QUERY TASK request. Previously established conditions such as mode parameters and reservations shall not change. Tasks submitted for this logical unit may complete, successfully or not, with or without a successful completion status.

Editor's Note 11: Only possible if the SATL maintains context of outstanding queued commands. If QUERY TASK is not supported, why is there a "shall" requirement in the text?

6.1.7 TARGET RESET

TARGET RESET shall be supported by the following sequence.

All tasks in the task set for the target shall be aborted.

Any contingent allegiance condition shall be cleared.

A device reset shall be carried out for the ATA device corresponding to the specified SCSI target by performing a link reset.

The device shall be brought to its operational mode and all Mode Parameters shall be set to default values.

Editor's Note 12: TARGET RESET was made obsolete in SAM-3 (03.001r1).

6.1.8 BUS RESET

BUS RESET shall cause a link reset of all associated targets on the logical bus.

Editor's Note 13: SAM-3 does not define "BUS RESET". This is a concept carried over from earlier SPI standards. The intent is to represent a HARD RESET condition, which is generated by transport-specific means. SAS and SATA already define the behavior of HARD RESET. PATA?

6.1.9 CLEAR TASK SET

CLEAR TASK SET shall not be supported for ATA devices. The SATL shall return a status of FAILED for the CLEAR TASK SET request. Previously established conditions such as mode parameters and reservations shall not change. The particular tasks in consideration may complete, successfully or not, with or without a successful completion status.

7 Summary of SCSI / ATA command mappings

7.1 Translated and emulated commands

In the event of a discrepancy between the contents of this section and the description of individual commands, description of individual commands takes precedence.

This document describes the SCSI to SATA command mapping. No reference should be made to the handling of ATAPI devices from this document.

Editor's Note 14: Some participants in the initial SAT study group indicated a need for ATAPI support.

Only LUN 0 is supported for all SATA targets. If a command tries to access a LUN other than 0, a check condition with the sense key set to illegal request and additional sense code set to invalid field in CDB is returned.

Editor's Note 15: Might use a WLUN for ATA-specific controls (power states, password protect, ...)?

Unless otherwise noted, the IMMED bit (immediate return) is ignored. For the format unit command, this bit will be supported.

Editor's Note 16: Why not IMMED op x1B "Start Stop Unit"? Do we mean to be requiring spin up/ down to complete in, maybe, three seconds?

All ATA commands with the exception of SATA Native Command Queued Reads and Writes are single threaded per device. The translator will queue IOs as necessary to enforce this.

Table 1 shows a summary of SCSI / ATA command mappings defined in this standard.

Table 1 — Summary of SCSI / ATA Command Mapping

SCSI COMMAND	SCSI Opcode	ATA Opcode	Support Mechanism		Hyper Link
			Emulate	Translate	
CHANGE DEFINITION	40h	N/A			Not Supported
COMPARE	39h	N/A			Not Supported
COPY	18h	N/A			Not Supported
COPY AND VERIFY	3Ah	N/A			Not Supported
FORMAT UNIT	04h	35h, 36h	X	X	9.1
INQUIRY	12h	ECh	X		8.1
LOCK-UNLOCK CACHE	36h	N/A			Not Supported
LOG SELECT	4Ch	N/A			Not Supported
LOG SENSE	4Dh	N/A	X		8.2
MODE SELECT(6,10)	15h, 55h	N/A	X		8.3
MODE SENSE(6,10)	1Ah, 5Ah	N/A	X		8.4

Table 1 — Summary of SCSI / ATA Command Mapping

SCSI COMMAND	SCSI Opcode	ATA Opcode	Support Mechanism		Hyper Link
			Emulate	Translate	
MOVE MEDIUM	A7h	N/A			Not Supported
PERSISTENT RESERVE IN	5Eh	N/A			Not Supported
PERSISTENT RESERVE OUT	5Fh	N/A			Not Supported
PREFETCH	34h	N/A			Not Supported
PREVENT ALLOW MEDIUM REMOVAL	1Eh	N/A			Not Supported??? Support MMC?
READ (6,10,12)	08h, 28h, A8h		X		9.2
READ (16)	88h				Not supported???
READ BUFFER	3Bh	E4h	X	X	8.5
WRITE BUFFER	3Ch	E8h	X	X	8.12
READ CAPACITY	25h	N/A	X		9.3
READ DEFECT DATA(10)	37h	N/A			Not Supported
READ DEFECT DATA (12)	B7h	N/A			Not Supported
READ ELEMENT STATUS	B4h	N/A			Not Supported
READ LONG(10)	3Eh	N/A			Not Supported
REASSIGN BLOCKS	07h	N/A			Not Supported
REBUILD	81h	N/A			Not Supported
RECEIVE DIAGNOSTIC RESULTS	1Ch	N/A			Not Supported
REGENERATE	82h	N/A			Not Supported
RELEASE AND RESERVE(6,10)	6h,17h,56h, 57h	N/A			Not Supported
REPORT DEVICE IDENTIFIER	A3h	N/A			Not Supported
REPORT LUNS	A0h	N/A	X		8.7
REQUEST SENSE	03h	N/A	X		8.8
REZERO UNIT	01h	N/A	X		9.4
SEARCH DATA EQUAL	31h	N/A			Not Supported
SEARCH DATA HIGH	30h	N/A			Not Supported
SEARCH DATA LOW	32h	N/A			Not Supported
SEEK (6,10)	0Bh, 2Bh	N/A	X		9.5
SEND DIAGNOSTIC	1Dh	90h		X	8.10
SET DEVICE IDENTIFIER	A4h	N/A			Not Supported
SET LIMITS	33h	N/A			Not Supported
START STOP UNIT	1Bh	E0h, E1h		X	9.6
SYNCHRONIZE CACHE	35h	E7h		X	9.7
TEST UNIT READY	00h	E5h		X	8.11
VERIFY (10)	2Fh 40h	42h		X	9.8
VERIFY (12,16)	AFh, 8Fh	N/A			Not Supported

Table 1 — Summary of SCSI / ATA Command Mapping

SCSI COMMAND	SCSI Opcode	ATA Opcode	Support Mechanism		Hyper Link
			Emulate	Translate	
WRITE (6,10,12)	0Ah, 2Ah, AAh	Multiple		X	9.9
WRITE (16)	8Ah	N/A			Not supported???
WRITE AND VERIFY(10)	2Eh	Multiple		X	9.10
WRITE AND VERIFY (12,16)	AEh, 8Eh	N/A			Not Supported
WRITE LONG	3Fh	N/A			Not Supported
WRITE SAME(10,12)	41h, 93h	N/A			Not Supported
XDREAD	52h	N/A			Not Supported
XDWRITE	50h	N/A			Not Supported
XDWRITE EXTENDED	80h	N/A			Not Supported
XPWRITE	51h	N/A			Not Supported

7.2 SCSI ATA Passthrough command

Editor's Note 17: The idea of defining a new SCSI opcode for ATA passthrough commands was discussed on the reflector. Discussion in SAT teleconferences indicated a preference to "spoof" as much as possible, and provide the passthrough only for specialized applications that need tighter control of ATA devices. This could, for example, be used to retrieve the IDENTIFY DEVICE information directly. But the SAT teleconference seemed to show a preference for defining a special INQUIRY VPD page for that purpose. Reflector discussion indicates there is precedent in providing a command-pair - one to issue the command, and another to retrieve the results. Perhaps this could be accomplished with a variant of the SEND DIAGNOSTIC command and the RECEIVE DIAGNOSTIC RESULTS command defined specifically for ATA command passthrough. Discussion also indicates "an ATA taskfile passthrough was released years ago". It might provide a good template for this standard if someone can dig it up. A general PIO passthrough command would handle SMART. Also consider DCO (device configuration overlay).

Editor's Note 18: Reflector discussion to support 12-byte and 16-byte variants of READ and WRITE commands.

Editor's Note 19: Is there a need to support READ(16) and WRITE(16) to handle high-capaciaty ATA devices?

Editor's Note 20: Is there a need to support PREVENT/ALLOW MEDIUM REMOVAL and MMC commands?

Editor's Note 21: Can SAT handle READ LONG and WRITE LONG for injected read errors?

8 SCSI Primary Commands (SPC) Mapping

8.1 INQUIRY command (12h)

The SCSI INQUIRY command requests general information about a target or component LUN. INQUIRY and certain vital product data pages have translations.

8.1.1 Command Summary

Table 2 — INQUIRY Command summary

Support Method	Emulated
ATA Opcodes	ECh, ATA Identify Device Command.
Supported Vital Product Data pages	00h, Supported VPD pages 80h, Unit serial number

8.1.2 SCSI INQUIRY CDB format and supported fields

A) CMDDT

This bit is now obsolete and is not supported. If set to one, return a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

B) EVPD

This bit must be set to request vital product data pages and the Page Code should indicate the page requested. Only two pages are supported currently: Supported VPD pages (00h) and Unit serial number (80h).

Editor's Note 22: Does SAT need to handle media serial numbers for removable media (e.g. compact flash)? Would this require extensions to support MMC (perhaps next generation of SAT)?

8.1.3 Data Returned

Standard inquiry data or vital product pages shall be returned depending on the request.

Editor's Note 23: At the SAT teleconference participants discussed the idea of defining a new INQUIRY VPD page to contain a direct image of IDENTIFY DEVICE information returned by the ATA device. This page would not be cached but would always be fetched from the ATA device. Fields in other INQUIRY standard and VPD pages would have specified translations from fields reported in IDENTIFY DEVICE informaiton. More detail is required in this section to fill out the required translations. Of particular interest are the firmware revision and vendor identifier information. One idea is to assign a high-order bit to identify when information is translated from ATA IDENTIFY DEVICE information, and try to include the most significant information from IDENTIFY DEVICE, up to the point where we run out of bits in the corresponding INQUIRY standard and VPD page fields.

8.1.3.1 Standard INQUIRY data

A) PERIPHERAL DEVICE TYPE

This bit field is set to 0 to indicate that a direct access device is connected.

B) PERIPHERAL QUALIFIER

This bit field is set to 0 to indicate that the specified peripheral device is currently attached to this logical unit.

C) RMB

Removable medium bit. Set the value of bit 7 of General Configuration field of IDENTIFY DEVICE information retrieved from the target.

D) VERSION

This field is set to 03h to indicate that is supports SCSI 3 SPC specification.

E) RESPONSE DATA FORMAT

Set to a value of 2. Values other than 2 are obsolete. Value of 2 indicates that data shall be in the format supported by SPC-3.

f) HISUP (hierarchical support)

This bit is set to 0 to indicate that this device shall not support hierarchical addressing model.

G) NORMAL ACA SUPPORT

This bit is set to 0 to indicate the NACA bit in the control byte of the SCSI inquiry CDB is not supported.

H) ADDITIONAL LENGTH

This field is set to indicate the length of the inquiry data that follows.

I) PROTECT

Set to zero to indicate that the device does not support protection information.

j) 3PC (Third party commands)

Set to 0 to indicate that this device does not support third party commands.

K) ALUA

These bits are set to 0 to indicate that this device does not support asymmetrical logical unit access.

L) ACC

This bit is set to 0 to indicate that no access controls coordinator may be addressed through this logical unit.

M) SCC

This bit is set to 0 to indicate that the device does not contain an embedded storage array controller component.

N) ADD16

This bit is set to 0 to indicate that this device does not support 16bit wide SCSI addresses.

O) MCHNGR

This bit is set to 0 to indicate that this device is not attached to medium transport element.

P) MULTIP

This bit is set to 0 to indicate that this device does not contain multiple ports.

Q) ENCSERV

This bit is set to 0 to indicate that this device does not contain an embedded enclosure service component.

R) BQUE

Works in combination with the CMDQUE bit. This bit is set to 0 to indicate that this device does not support basic task management model.

S) CMDQUE

This bit is set if bit 8 of the SerialATACapabilities field in the device Identify data is set and the SerialATACapabilities field represents a valid value. If set to 1 the device supports full task management model, specifically it supports command queuing. If this bit is 0, the device does not support command queuing.

T) LINKED

This bit is set to 0 to indicate that this device does not support linked commands.

U) SUNC

This bit is set to 0 to indicate that this device does not support SCSI synchronous data transfers.

V) WBUS16

This bit is set to 0 to indicate that this device does not support 16 bit transfers. SYNC/WBUS16 applies to parallel SCSI.

W) VENDOR ID

This 8 bytes field is created using the first 8 bytes of the 40-byte Model Number field from the device identify data retrieved from the device. Endianness is swapped to match the SCSI format.

Editor's Note 24: Should we define something in this field to flag that this is an emulated SCSI device with the SATL constituting part of the emulated SCSI device with an ATA device? One suggestion - use the 8-byte string, "ATA_____", (_ = space), or identify the SATL in this field, or some combination of the SATL and the attached ATA device?

X) PRODUCT ID

This 16 bytes field is created using the bytes 8 though 23 of the 40 byte Model Number field from the identify data. Endianness is swapped to match the SCSI format.

Y) PRODUCT REVISION

This field is created using the 8-byte firmware revision field of the device Identify data. The lower 4 bytes are used and the upper 4 bytes are lost since SCSI supports only 4 bytes for revision number. Endianness is swapped to match the SCSI format.

z) IUS/QAS/CLOCKING/Vendor descriptors/ and vendor specific parameters of standard inquiry data are not supported.

8.1.3.2 Vital Product data - Supported VPD pages

A) PERIPHERAL DEVICE TYPE

This bit field is set to 0 to indicate that a direct access device is connected.

B) PERIPHERAL QUALIFIER

This bit field is set to 0 to indicate that the specified peripheral device is currently attached to this logical unit.

C) PAGE CODE

This field is set to 0 to indicate 'supported VPD' pages page.

D) PAGE LENGTH

This field indicates the length of the supported VPD page list returned in number of bytes.

E) SUPPORTED VPD LIST

This list contains the page codes of the pages supported and is implemented in ascending order of page codes beginning with page code 00h.

8.1.3.3 Vital Product data - Unit serial number VPD page

A) PERIPHERAL DEVICE TYPE

This bit field is set to 0 to indicate that a direct access device is connected.

B) PERIPHERAL QUALIFIER

This bit field is set to 0 to indicate that the specified peripheral device is currently attached to this logical unit.

C) PAGE CODE

This field is set to 80h to indicate 'supported VPD' pages page.

D) PAGE LENGTH

This field indicates the length of the product serial number. Currently a 20-byte product serial number is being returned; hence the length is set to 20 bytes.

E) PRODUCT SERIAL NUMBER

The 20-byte field, serial number, found in the device identify data retrieved from the device shall be returned in this field. The serial number field shall be byte swapped to match the SCSI data format.

Editor's Note 25: What about the Device Identification VPD page (83h)? If the ATA device supports reporting an OUI should it be copied into the corresponding Device Identification VPD page?

8.1.4 Miscellaneous Notes

The LUN should be set to zero. If set, return SCSI_INQUIRY_PERIPHERAL_QUALIFIER_NOT_SUPPORTED and SCSI_INQUIRY_DEV_TYPE_UNKNOWN (7Fh).

8.2 LOG SENSE Command (4Dh)

The log sense command provides a mechanism, which an application can use to retrieve statistical or diagnostic results, or other operating data about a target or a logical unit.

NOTE 4 - Only SMART data shall be returned through this command.

8.2.1 Command Summary

Table 3 — Log Sense command summary

Support Method	For SMART data, i.e. Log Page 31h, translated to ATA SMART (B0h) command with feature register set to D0h.
ATA Opcodes	B0h

8.2.2 LOG SENSE CDB format and supported fields

a) SP (Save Parameters)

This bit is ignored. Saving of parameters is not supported.

b) PPC (Parameter Pointer Control)

This bit is ignored.

c) PC (Page Control)

Table 4 — PAGE CONTROL Value

PC	LOG Sense Parameter value
00b	Current values.
01b	Cumulative Values.
10b	Default threshold values. N/A.
11b	Default cumulative values N/A.

d) PAGE CODE

Table 5 — PAGE CODE Values

Page Code	Details
31h	The drive's SMART data with out the Page header field. Format corresponds to device's SMART data structure.
All others	A CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB shall be returned.

E) PARAMETER POINTER

This filed is ignored.

F) ALLOCATION LENGTH

Indicates the length of the use data-in buffer. Data up to or less the this size shall be transferred.

8.2.3 Data Returned

SMART data log page data shall be returned. If the PAGE CODE is another value, a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB shall be returned to the caller.

8.2.4 Miscellaneous Notes

None.

8.3 MODE SELECT(06)/(10) Command (15h)/(55h)

Mode select command provides a mechanism for application clients to change the operating parameters of the target or a logical unit. This is the complementary command for MODE SENSE. The application client is advised to send a MODE SENSE command before issuing a MODE SELECT command for the same mode page, to determine the format, length changeable field etc. For all initiators, Logical Units of a target shall share the mode parameter header and the block descriptor and mode page values.

Currently Mode Page Policy VPD page is not implemented. Therefore Mode page policy is set to 'Shared' , which indicates that only one copy of 'current' values are maintained for all logical units of a target. After a logical unit reset, these values shall reset to default values, since saved parameters are not supported. See section 4 for supported mode pages.

8.3.1 Command Summary

Table 6 — MODE SELECT(06)/(10) command summary

Support Method	Mostly emulated. Some operational parameters in individual pages are prvided via ATA. See section on Mode page for more details.
ATA Opcodes	No direct translations.

8.3.2 MODE SELECT(06) CDB format and supported fields

a) SP (Save parameters)

Save Parameter operation is not supported, there only a value of 0b is supported. If this bit is set to 1b, a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB, shall be returned.

b) PF (Page Format)

A page format of 0b indicates that modes pages are vendor specific, therefore a value of 0b is not supported. A value of 1b indicates that all page formats corresponds to SPC-3 and SBS-2 MODE PAGE formats. If this bit is set to 0b, a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB, shall be returned.

C) PARAMETER LIST LENGTH

This field should be set to the length of the mode parameter list, transferred out of the client's data-out buffer.

8.3.3 MODE SELECT (10) CDB format and supported fields

Only the size of the PARAMETER LIST LENGTH field is different in the MODE SELECT (06) command, all other details remain the same.

8.3.4 Data Transferred

The general format used for data in MODE SELECT command contains a Mode Parameter header, followed by one or more BLOCK DESCRIPTORS, followed by a variable length mode page(s). Please refer SPC-3 for more details on the format of each of these sections. They are described briefly in SCSI-SATA translation context here.

8.3.4.1 Format for MODE SELECT (06) Command

8.3.4.1.1 MODE SELECT Parameter Header (06)

A) MODE DATA LENGTH

This field is reserved for MODE SELECT command and should be set to 00h.

B) MEDIUM TYPE

For direct access devices, this field should be set to 00h.

C) DEVICE SPECIFIC PARAMETER

This field is reserved for MODE SELECT command and should be set to 00h.

D) BLOCK DESCRIPTOR LENGTH

This field indicates the length of the block descriptor section. This value is obtained by multiplying the number of block descriptors by 8, for MODE SELECT (6) commands and MODE SELECT (10) commands, if LONGLBA bit is set to 0, or by 16 for MODE SELECT (10) commands and LONGLBA bit is set to 1. Currently only one Block descriptor is supported, so this value should be set to 8 or 16.

8.3.4.1.2 MODE SELECT Parameter Header (10)

Only the length of the MODE DATA LENGTH field and the BLOCK DESCRIPTOR LENGTH field are different for MODE SELECT (10) reply parameter header, except that LONGLBA bit is defined as below.

a) LONGLBA - Describes the length of the block descriptors.
   A) If set to 0b, block descriptors are 8 bytes long.
   B) If set to 1b, block descriptors are 16 bytes long.

8.3.4.1.3 Mode Select Block Descriptor (8 byte format)

A) NUMBER OF BLOCKS

This field shall be ignored as the capacity of the devices attached cannot be changed.

B) DENSITY CODE

For direct access devices, this field should be set to 00h.

C) BLOCK LENGTH

Describes the block length for the section of the LUN described by this block descriptor. Since there is only one block descriptor, this describes the block length of the entire LUN. For direct access devices, this is set to 200h or 512 bytes per block.

8.3.4.1.4 Mode Select Block Descriptor (16 byte format)

Only the length of the NUBER OF BLOCKS field and the BLOCK LENGTH field changes between the 8-byte format and the 16-byte format, while the description of the field remains the same.

8.3.4.1.5 Mode Page format

Refer to the section on mode pages for the format of individual pages.

8.3.5 Miscellaneous Notes

Saving of parameters is not supported. Only one block descriptor shall be supported and shall describe the settings for the entire logical unit. Most MODE SELECT operations specified by the content of the mode pages concerned shall be turned to no ops due to the lack of equivalent commands or features in SATA. Refer to the section on mode pages for the details about the contents that are supported for each individual page.

8.4 Mode Sense (06)/(10) Command (1Ah)/(5Ah)

Mode sense command is used to query the device server about operational parameters of the target device or logical unit(s). The translator acts as the device server and emulates the SCSI behavior for SATA devices. The MODE SENSE command is the complementary command of the MODE SELSECT command. See clause 10.1 for supported mode pages.

8.4.1 Command Summary

Table 7 — Mode Sense command summary

Support Method	Mostly emulated. Some operational parameters in individual pages are gathered by issuing ATA commands. See section on Mode page for more details.
ATA Opcodes	No direct translations.

8.4.2 MODE SENSE (06) CDB format and supported fields

A) DBD

A value 0b indicates that one or mode Block Descriptors can be returned in Mode Sense data. Currently only one Block descriptor is supported.

b) PC (Page Control)

Table 8 — Page Control (PC) values and their descriptions

PC Value	Type of Parameter	Supported
00b	Current Values	Yes
01b	Changeable Values	No
10b	Default Values	No
11b	Saved Values	No

C) PAGE CODE

This field identifies the particular page requested. See section on Mode pages for further details.

D) SUB PAGE CODE

This field identifies the sub page code within the page code specified by PAGE CODE field that is required by the client. See section on Mode pages for further details.

8.4.3 MODE SENSE (10) CDB format and supported fields

a) LLBAA (Long LBA Accepted)
   A) If this bit is set to 0, device server is prohibited from setting LONGLBA bit in the Mode parameter header. This indicates that mode parameter block descriptor is 8 bytes in length.
   B) If this bit is set to one, the device server is allowed to set LONGLBA bit in the Mode Parameter header. If LONGLBA bit is set to 1when returning mode parameters it indicates that the block descriptors are 16 bytes in length.

8.4.3.1 Data Returned

The general format used to return data for a MODE SENSE command contains a Mode Parameter header, followed by one or more BLOCK DESCRIPTORS, followed by a variable length mode page(s). Please refer SPC-3 for more details on the format of each of these sections. They are described briefly in SCSI-SATA translation context here.

8.4.3.2 Reply for Mode Sense (06) Command

8.4.3.2.1 Mode Sense Parameter Header (06)

a) MODE DATA LENGTH

Indicates the number of bytes following this field that was transferred.

b) MEDIUM TYPE

For Direct access device types, this field is set to 00h.

c) DEVICE SPECIFIC PARAMETER

Refer SBC-2 for the format of the DEVICE SPECIFIC PARAMETER for block commands.

A) DPOFUA

If this bit is set to 0b, then it indicates that device server does not support DPO and FUA bits.

If this bit is set to 1b, then it indicates that the device server supports DPO and FUA bits.

B) WP

If this bit is set to 0b, it indicates that the device is write enabled. This is the only supported value for this bit.

If this bit is set to 1b, it indicates that the device is write protected, and this value is currently not supported.

d) BLOCK DESCRIPTOR LENGTH

This field indicates the length of the block descriptor section. This value is obtained by multiplying the number of block descriptors by 8, for MODE SENSE (6) commands and MODE SENSE (10) commands, if LONGLBA bit is set to 0, or by 16 for MODE SENSE (10) commands and LONGLBA bit is set to 1.

8.4.3.2.2 Mode Sense Parameter Header (10)

Only the length of the MODE DATA LENGTH field and the BLOCK DESCRIPTOR LENGTH field are different for MODE SENSE (10) reply parameter header, except that LONGLBA bit is defined as below.

A) LONGLBA

Describes the length of the block descriptors.

A) If set to 0b, block descriptors are 8 bytes long.
B) If set to 1b, block descriptors are 16 bytes long.

8.4.3.2.3 Mode Sense Block Descriptor (8 byte format)

A) NUMBER OF BLOCKS

This field indicates the number of blocks being described by this block descriptor. Current implementation dictates that only one block descriptor is returned, so it shall describe the blocks for the entire LUN. Therefore this field is equal to the Maximum LBA of the device minus one.

B) DENSITY CODE

For direct access devices, this field is set to 00h.

C) BLOCK LENGTH

Describes the block length for the section of the LUN described by this block descriptor. Since there is only one block descriptor, this describes the block length of the entire LUN. For direct access devices, this is set to 200h or 512 bytes per block.

8.4.3.2.4 MODE SENSE Block Descriptor (16 byte format)

Only the length of the NUBER OF BLOCKS field and the BLOCK LENGTH field changes between the 8-byte format and the 16-byte format, while the description of the field remains the same.

8.4.3.2.5 Mode page format

Refer to the section on mode pages for the format of individual pages.

8.4.4 Miscellaneous Notes

Persistent saving of parameters is not supported. Only one block descriptor shall be supported and shall describe the settings for the entire logical unit.

8.5 READ BUFFER Command (3Ch)

8.5.1 Command Summary

The READ BUFFER command is used with the write buffer command to determine the integrity of the target device's buffer memory and the physical interconnect that connects the target device and the initiator.

Table 9 — Read Buffer command summary

Support Method	Partially translated, partially emulated.		
ATA Opcodes	**MODE**	**Supported Method**	**Translated ATA Opcode**
	00h - Combined Header and Data Mode	Not Supported.	N/A
	01h - Vendor specific Mode	Not Supported.	N/A
	02h - Data Only mode	Translated to ATA Read Buffer Command	E4h
	03h - Descriptor mode	Emulated.	N/A
	0Ah - Echo Mode	Not supported.	N/A
	0Bh - Echo buffer descriptor Mode	Not Supported.	N/A
	All other MODE values	Not Supported.	N/A

8.5.2 READ BUFFER CDB format and supported fields

A) MODE

Table 9 describes MODES supported. Only data and data buffer descriptor shall be supported.

B) BUFFER ID

Only buffer 0 shall be supported.

C) BUFFER OFFSET

Refers to the offset in the buffer to start reading data from. The BUFFER OFFSET should be less than the size of the buffersize, otherwise a CHECK CONDITION shall be sent back with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

D) ALLOCATION LENGTH

Refer to individual sections for the meaning of this term.

8.5.3 Data Returned

Data returned to the requestor depends on the MODE parameter.

8.5.3.1 Data Only Mode (02h)

In this mode, data is read from the device's sector buffer and returned to the requestor. Note that sector buffer in the SATA device is being used to emulate the SCSI READ BUFFER command, so the maximum length of data that can be written is 512 bytes. Valid fields in the CDB, apart from the MODE field, are BUFFER ID, BUFFER OFFSET (< 512) and ALLOCATION LENGTH (<= 512). A write buffer command may have been sent to the same BUFFER ID before it is read.

8.5.3.2 Descriptor mode (03h)

Four bytes of information shall be returned to the requestor describing the requested buffer. These four bytes include the OFFSET BOUNDARY and the BUFFER CAPACITY. The BUFFER ID should be set to 0. For all other BUFFER ID's, all zeros shall be returned. ALLOCATION LENGTH should be set to 4.

8.5.4 Miscellaneous Notes

Sector buffer in a SATA 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.

8.6 RELEASE UNIT Command (17h) (06/10)

The RELEASE UNIT SCSI commands have been made obsolete in SBC 2 (January 12, 2004 Revision 12). There is no meaningful translation of a SCSI release unit command to an ATA command, The SATL shall return a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST (05h) and ADDITIONAL SENSE CODE set to invalid command operation code (20h).

8.6.1 Command Summary

Table 10 — Release Unit command summary

Support Method	Not supported.
ATA Opcodes	None.

8.6.2 RELEASE UNIT CDB format and supported fields

None, all parameters are ignored.

8.6.3 Data Returned

None

8.6.4 Miscellaneous Notes

None.

Editor's Note 26: Should SAS Affiliation be considered here? Editor's perspective: Affiliation does not provide the level of capability to provide this funciton. There is no persistence, and it is intended only as a fleeting mutex to arbitrate for immediate access to a disk. An acceptable emulation of Reserve/Release (prefereably the persistent variety) would require the SATL to store and manage persistent information about the current reservation status of multiple initiators.

8.7 REPORT LUNS Command (A0h)

REPORT LUNS command is used to discover the logical unit inventory of a peripheral device. Logical unit inventory is returned to the requester as a list of logical unit numbers for all logical units having a PERIPHERAL QUALIFIER of 000b. Logical units with other values for PERIPHERAL QUALIFIER are currently not included in this list.

8.7.1 Command Summary

Table 11 — REPORT LUNs command summary

Support Method	Emulated. Only LUN 0 is supported currently.
ATA Opcodes	None.

8.7.2 REPORT LUNs CDB format and supported fields

A) SELECT REPORT

This field is ignored.

B) ALLOCATION LENGTH

This field should be at least 16 bytes according to the SPC-3 specification and that should guarantee that the length of the user buffer used to DMA the report LUNs data back is not overrun.

8.7.3 Data Returned

Data returned shall be in compliance to the REPORT LUNs parameter data format specified in SPC-3. Currently the LUN LIST LENGTH, which is 8 times the number of logical units reported back, is set to 8 and only LUN 0 is reported back to the

8.7.4 Miscellaneous Notes

Only LUN zero is reported.

8.8 REQUEST SENSE COMMAND (03h)

Request sense command requests that any available sense data be returned to the caller. Upon erros with auto request sense enabled, the translator shall issue a request sense command and send the sense data to the caller along with the status of the command that generated the CHECK CONDITION. This behavior is emulated for SATA devices as well. Therefore, a separate request sense command is not necessary and shall result in a sense data block with SENSE KEY set to no sense and ADDITIONAL SENSE CODE set to no additional sense data. Emulate enabling and disabling of auto request sense and cache sense data if auto request sense is disabled.

The only exception to this is when a unit format command is being executed or when SMART reporting is enabled. It is legal to send a request sense command to a target device while a device format command is outstanding. If requested, return a SENSE KEY of not ready and ADDITIONAL SENSE CODE set to logical unit not ready format in progress. The SENSE KEY specific bytes shall be set to progress indication as per SPC-3. Return sense data with SENSE KEY set to no sense, additional SENSE KEY set to general hardware failure if it detects a SMART threshold exceeding condition. Refer section 6 on SMART for further details.

8.8.1 Command Summary

Table 12 — REQUEST SENSE command summary

Support Method	Emulated. Auto request sense is always enabled, so a separate request sense command shall result in sense data with SENSE KEY set to no sense. Only exception is when a format unit command is outstanding and when SMART reporting is enabled.
ATA Opcodes	None.

8.8.2 REQUEST SENSE CDB format and supported fields

A) DESC

This bit is ignored.

B) ALLOCATION LENGTH

This field specifies how many bytes have been allocated for the returned sense data. The minimum of this value of 18 bytes shall be returned.

8.8.3 Data Returned

Sense data shall be returned with the SENSE KEY set to no sense and ADDITIONAL SENSE CODE set to no additional sense data. When a FORMAT UNIT command is outstanding, sense data indicating the progress of the format operation shall be returned. If SMART reporting is enabled, and a threshold exceeding condition is detected, the SENSE KEY set to NO SENSE and ADDITIONAL SENSE CODE set to GENERAL HARDWARE FAILURE shall be returned.

8.8.4 Miscellaneous Notes.

None.

8.9 RESERVE UNIT Command (16h)

The RESERVE UNIT, release unit SCSI commands have been made obsolete in SBC 2 (January 12, 2004 Revision 12). There is no meaningful translation of a SCSI reserve command to an ATA command, The translator shall return a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST (05h) and ADDITIONAL SENSE CODE set to INVALID COMMAND OPERATION CODE (20h).

8.9.1 Command Summary

Table 13 — RESERVE UNIT command summary

Support Method	Not supported.
ATA Opcodes	None.

8.9.2 RESERVE UNIT CDB format and supported fields

None, all parameters are ignored.

8.9.3 Data Returned

None

8.9.4 Miscellaneous Notes

None.

Editor's Note 27: Should SAS Affiliation be considered here? Editor's perspective: Affiliation does not provide the level of capability to provide this funciton. There is no persistence, and it is intended only as a fleeting mutex to arbitrate for immediate access to a disk. An acceptable emulation of Reserve/Release (prefereably the persistent variety) would require the SATL to store and manage persistent information about the current reservation status of multiple initiators.

8.10 SEND DIAGNOSTIC COMMAND (1Dh)

The send diagnostic command provides a mechanism for an application client to carry out diagnostic operations on the target, logical unit or both. Note that only drive targets can be serviced through this command. Also note that the complementary command, receive diagnostic command, is not supported.

8.10.1 Command Summary

Table 14 — SEND DIAGNOSTIC command summary

Support Method	Default Self test - Translated to ATA Execute Device Diagnostic (90h). Background tests - Not Supported.
ATA Opcodes	90h

8.10.2 SEND DIAGNOSTIC CDB format and supported fields

A) UNITOFFL

Ignored.

B) DEVOFFL

Ignored.

C) SELFTEST

A value of 1, which indicates the default self test, be returned with a status of SUCCESS if the default self test completes without any errors, or a CHECK CONDITION with SENSE KEY set to HARDWARE ERROR if an error occurs.

d) PF (Parameter format)

Ignored.

E) PARAMETER LIST LENGTH

Ignored.

F) SELF TEST CODE

If the SELFTEST bit is zero, SELF TEST CODE indicates which self-test to be carried out.

Table 15 — Send diagnostic self-test code

Value	Name of Test	Description of Test	Support
000b	None.	Used when SELFTEST bit is set to 1.	Yes
001b	Background short self test	Start a background self test. Results available in log page 10h.	No
010b	Background extended self test.	Start an extended self-test in background mode. Results available in Log page 10h.	No
011b	Reserved.		
100b	Abort background self test.	If a self test is running in background mode, abort t.	No
101b	Foreground short self test.	Start a self-test in foreground mode.	No
110b	Foreground extended self test Start	an extended self-test in foreground mode.	No

8.10.3 Data Returned

If SELFTEST bit set to 1, GOOD status shall be returned if the default self test completes without any errors. CHECK CONDITION shall be returned if an error occurred.

8.10.4 Miscellaneous Notes

NOTE 5 No mechanism is provided to pass parameters from an application data-out buffer to the target.

The RECEIVE DIAGNOSTIC command may not be not supported.

8.11 TEST UNIT READY Command (00h)

The TEST UNIT READY command is used to determine whether the device is ready or not. If the device is capable of accepting medium access commands without failing them with a CHECK CONDITION then the command should return GOOD status. However, if the device is unable to process medium access commands, Test Unit Ready Command shall return a CHECK CONDITION with SENSE KEY set to NOT READY.

8.11.1 Command Summary

Table 16 — TEST UNIT READY command summary

Support Method	Translated to ATA Check Power mode Command.
ATA Opcodes	E5h

8.11.2 TEST UNIT READY CDB format and supported fields

All fields in the CDB shall be set to zero.

8.11.3 Data Returned

If the device is capable of accepting medium access commands without failing with a CHECK CONDITION, then a STATUS of GOOD shall be returned.

If the device is not capable of accepting medium access commands, then CHECK CONDITION shall be returned with the SENSE KEY set to NOT READY and ADDITIONAL SENSE CODE set to LOGICAL UNIT NOT READY.

8.11.4 Miscellaneous Notes

a) The TEST UNIT READY command is mapped to the ATA Check Power Mode command. If the drive is at standby power mode, then it is assumed that the device is incapable of accepting medium access commands without any intervention, so a CHECK CONDITION is returned with the SENSE KEY set to NOT READY and the ADDITIONAL SENSE CODE set to LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED.

Editor's Note 28: Ah, x 2 04 02 is another option?

---

b) If the ATA Check power mode command returned a power state of active or idle, then it is assumed that the drive is ready to accept medium access commands, so a status of good shall be returned to the requester.
c) If the device is being formatted, a CHECK CONDITION is returned with the SENSE KEY set to NOT READY and the ADDITIONAL SENSE CODE set to LOGICAL UNIT NOT READY, FORMAT IN PROGRESS.

8.12 WRITE BUFFER COMMAND (3Bh)

8.12.1 Command Summary

The WRITE BUFFER command is used with the read buffer command to determine the integrity of the target device's buffer memory and the physical interconnect that connects the target device and the initiator.

Table 17 — WRITE BUFFER command summary

Support Method	Partially translated, partially emulated.		
ATA Opcodes	**MODE**	**Supported Method**	**Translated ATA Opcode**
	00h - Combined Header and Data Mode	Not supported	N/A
	01h - Vendor specific Mode	Not supported	N/A
	02h - Data Only mode	Translated to ATA Write Buffer Command	E8h
	04h - 07h Download microcode modes	Not supported	N/A
	0Ah - Echo Mode	Not supported	N/A
	All other MODE values	Not supported	N/A

8.12.2 WRITE BUFFER CDB format and supported fields

A) MODE

Table 17 describes modes supported. Only data and data buffer descriptor shall be supported.

B) BUFFER ID

Only buffer 0 shall be supported.

C) BUFFER OFFSET

Refers to the offset in the buffer to start reading data from. The BUFFER OFFSET should be less than the size of the buffersize, otherwise a CHECK CONDITION shall be returned with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

d) ALLOCATION LENGTH

Refer to individual sections for the meaning of this term.

8.12.3 Data written

Data written to the device depends on the MODE parameter.

8.12.3.1 Data only mode (02h)

In this mode, data is written to the device's sector buffer. Note that sector buffer in the SATA device is being used to emulate the SCSI WRITE BUFFER command, so the maximum length of data that can be written is 512 bytes. Valid fields in the CDB except the MODE field is 0 for BUFFER ID which should be 0, BUFFER OFFSET (< 512) and ALLOCATION LENGTH (<= 512).

8.12.4 Miscellaneous Notes

Sector buffer in a SATA device shall be used to emulate the WRITE BUFFER command, so the size of the buffer is limited to 512 bytes for data buffer and echo buffers.

9 SCSI Block Commands (SBC) Mapping

9.1 FORMAT UNIT Command (04h)

The FORMAT UNIT command verifies that all logical block addresses visible to external application clients are formatted and can be accessed. All sectors of the visible address space are written to zero.

9.1.1 Command Summary

Table 18 — FORMAT UNIT command summary

Support Method	Emulated. It shall use the same WRITE commands as in the section 3.11. Depending on if EXTENDED and QUEUED bits are set, different ATA opcodes shall be used. No optional parameters shall be implemented except for the IMMED bit.
ATA Opcodes	None. 35h shall be used to zero out all sectors in the visible address space.

9.1.2 FORMAT UNIT CDB format and supported fields

A) DEFECT LIST FORMAT

This feature shall not be supported. If a DEFECT LIST FORMAT is specified the SATL shall terminate the command with a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

B) CMPLIST

This feature shall not be supported. If CMPLIST is specified the SATL shall terminate the command with a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

C) FMTDATA

If set to 0 no data shall be transferred from the data-out buffer. If set to 1 the FORMAT UNIT PARAMETER LIST shall be transferred from the client's data out buffer. Note that only the IMMED bit in the parameter list shall be supported. All other fields shall be ignored.

D) LONGLIST

This field shall not be supported. If LONGLIST is specified the SATL shall terminate the command with a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

E) FMTPINFO

This feature shall not be supported. If FMTPINFO is specified the SATL shall terminate the command with a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

F) INTERLEAVE

This field shall be ignored.

9.1.3 Data Returned

None.

9.1.4 Miscellaneous Notes

Wile formatting a device, INQUIRY commands shall be processed as normal. REQUEST SENSE commands shall be rejected with NOT READY and FORMAT IN PROCESS SENSE KEY and ADDITIONAL SENSE CODE. All

other commands shall result in a CHECK CONDITION with the same SENSE KEY and ADDITIONAL SENSE CODE. The SATL may optionally reply with a CHECK CONDITION a SENSE KEY of ILLEGAL REQUEST if an unsupported field or a bit is set to a value other than what is defined in this document.

9.2 READ (6, 10, 12) command (08h, 28h, A8h)

The READ command is used to request the device to transfer blocks of user data to the requester. Different versions of the command support different LBA sizes and different transfer lengths. Data can be read from medium or the device cache if the most recent copy is in the cache and has not been transferred to the medium (see SBC-2).

9.2.1 Command Summary

Table 19 — READ command summary

Support Method	Translated to ATA read commands using the following criterion.			
ATA Opcodes	***Is Queuing enabled***	***Are Extended Commands Enabled***	***SCSI OP Code***	***Translated ATA Opcode***
	NO	NO	08h,28h,A8h	READ_DMA (C8h)
	NO	YES	08h,28h,A8h	READ_DMA_EXT (25h)
	YES	(Don't care)	08h,28h,A8h	READ_FPDMA_QUEUED

9.2.2 CDB format and supported fields

a) DPO (Disable page out)

DPO shall be ignored.

b) FUA (Force Unit Access)

FUA shall be supported with NCQ SATA targets.

Editor's Note 29: There are indications some key operating systems need support of DPO and FUA.

Editor's Note 30: For non NCQ drives, do we need to send a cache flush to force unit access?

C) RDPROTECT

RDPROTECT shall be ignored.

D) TRANSFER LENGTH

A) For READ(06) commands (08h), a TRANSFER LENGTH of 0 is interpreted as 256 sectors and is the same for SCSI and ATA.

B) For READ(10) commands (28h), a TRANSFER LENGTH of 0 indicates that no data transfer shall take place.

Editor's Note 31: The SATL emulates the behavior without accessing the device? Is this the same as a seek?

C) For READ(06) and READ(10) commands, if the TRANSFER LENGTH is not zero, the SATL shall transfer the specified number of sectors from the device to the application client.

E) LOGICAL BLOACK ADDRESS
   A) For READ(06) commands (08h), a 21 bit LBA is specified in bytes 3, 2, 1:5 in the CDB. Byte 3 is the LSB and the 5 bits from byte 1 are the MSB.
   B) For READ(10) commands (28h), a 32 bit LBA is specified in bytes 2 through 5. Byte 5 is the LSB and byte 2 is the MSB.
   C) For READ(10) Commands (28h), a 32 bit LBA is specified in bytes 2 through 5. Byte 5 is the LSB and byte 2 is the MSB.

9.2.3 Data Returned

Requested sector(s) shall be returned if successfully retrieved from the medium. If the LBA or the LBA plus (the TRANSFER LENGTH minus 1) is greater than the maximum sector that can be addressed in medium CHECK CONDITION shall be reported with the SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

9.2.4 Miscellaneous Notes

Protection information shall not be supported.

Editor's Note 32: Protection Information could be supported for ATA devices that support a blocksize 8-bytes larger than the blocksize specified for the emulated SCSI target. The SATL would be responsible for managing the Protection Information fields according to the specifications in SBC-2 and SPC-3.

For the READ(12) command, if the TRANSFER LENGTH field is greater than 0xFFFF the command shall terminate with a CHECK CONDITION with the SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

NOTE 6 - This is because the ATA command set supports a maximum of a 16-bit SECTOR COUNT field.

9.3 READ CAPACITY (10) command (25h)

The READ CAPACITY (10) command shall request information about the capacity of the block device being addressed. Information protection shall not be supported therefore the READ CAPACITY (16) command shall not supported.

9.3.0.1 Command summary

Table 20 — READ CAPACITY (10) command summary

Support method	Emulated
ATA opcodes	None. Use ATA IDENTIFY DEVICE information at power-on or link initialization to compute the maximum user addressable medium capacity.

9.3.1 READ CAPACITY (10) CDB format and supported fields

A) LOGICAL BLOCK ADDRESS

The LOGICAL BLOCK ADDRESS field works in conjunction with the PMI bit described below. Since this requires vendor specific information, this operation shall not be supported and shall be set to zero.

Editor's Note 33: What happens if the LOGICAL BLOCK ADDRESS field is not set to zero?

---

b) PMI (partial medium indicator)

The operation specified by this bit shall be ignored. The SATL shall return data for the LBA following the LBA specified in LOGICAL BLOCK ADDRESS.

Editor's Note 34: Deleted text, "...for which a vendor specific delay in data transfer may be encountered".

3.2.3 Data Returned

READ CAPACITY data, as defined by SBC-2, shall be returned to the application client.

c) RETURNED LOGICAL BLOCK ADDRESS

The SATL shall return a value one less than the value from the DEVICE IDENTIFY DATA TOTALUSERSECTORS field (words 60/61) retrieved from the device, The endianness shall be swapped to match SCSI format.

d) BLOCK LENGTH IN BYTES

This value shall be set to 512 bytes.

Editor's Note 35: The SATL could *optionally* support reporting other blocksizes reported in the SATA Identify Device information words 117 and 118 (valid bit in word 106). See ATA/ATAPI-7 Revision 4b.

9.3.2 Miscellaneous Notes.

The SATL shall ignore PMI bit and LOGICAL BLOCK ADDRESS in the INQUIRY CDB. READ CAPACITY (16) command shall not be supported.

9.4 REZERO UNIT (06) Command (01h)

The REZERO UNIT SCSI command has been made obsolete in SBC 2 (January 12, 2004 Revision 12), and the SEEK command has been made obsolete in ATA/ATAPI-7. The SATL shall successful status and shall not access the ATA device.

9.4.1 Command Summary

Table 21 — REZERO UNIT command summary

Support Method	The SCSI REZERO command has been made obsolete since SBC-2, revision 12. The command completes with successful status.
ATA Opcodes	None

9.4.2 REZERO UNIT CDB format and supported fields

Ignored.

9.4.3 Data Returned

None.

9.4.4 Miscellaneous Notes

NOTE 7 Could be supported with a READ VERIFY?

9.5 SEEK (06, 10) command (0Bh, 2Bh)

9.5.1 Command Summary

The SEEK command has been made obsolete in ATA/ATAPI-7. The SATL shall report successful completion of the SEEK command and shall not access the ATA device.

Editor's Note 36: Suggestion to execute the same as TEST UNIT READY instead, which could fail.

Table 22 — SEEK command summary

Support Method	The command is obsolete in ATA/ATAPI-7. All all SEEK or EXTENDED SEEK commands shall complete with success
ATA Opcodes	None

9.5.2 SEEK CDB format and supported fields

All fields shall be ignored.

9.5.3 Data Returned

None

9.5.4 Miscellaneous Notes

NOTE 8 Could be supported with a READ VERIFY?

9.6 START STOP UNIT command (1Bh)

The START STOP UNIT command provides a method for controlling the power state of a logical unit. For SATA devices, this implies transition between idle and standby power modes. This command is also used to load and unload medium. A LOAD MEDIUM request shall return a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

Editor's Note 37: Suggestion to execute the same as TEST UNIT READY instead, which could fail.

3.4.1 Command Summary

Table 23 shows methods to support the START/STOP UNIT command.

Table 23 — START/STOP UNIT command summary

Support Method:	Translated to:
START UNIT	Idle Immediate
STOP UNIT	Standby immediate
UNLOAD MEDIA	CHECK CONDITION with SENSE KEY ILLEGAL REQUEST and ASC INVALID FIELD IN CDB
LOAD MEDIA	CHECK CONDITION with SENSE KEY ILLEGAL REQUEST and ASC INVALID FIELD IN CDB
ATA Opcodes	E0h, E1h

9.6.1 START/STOP UNIT CDB format and supported fields

a) IMMED - Immediate bit

This bit shall be ignored.

B) START/LOEJ

Table 3.4.2.5 Interpretation of START and LOEJ bits in START/STOP UNIT CDB

LOEJ	START	Interpretation
0	0	Stop Unit command. Translated to standby immediate
0	1	Start Unit command. Translated to idle Immediate.
1	0	Not supported. CHECK CONDITION returned with SENSE KEY ILLEGAL REQUEST.
1	1	Not supported, CHECK CONDITION returned with SENSE KEY ILLEGAL REQUEST.

c) POWER CONDITIONS

Ignored.

9.6.2 Data Returned

This command does not return any data. Status or a CHECK CONDITION is returned after the completion of the command. The immediate bit in the CDB is ignored.

9.6.3 Miscellaneous Notes.

The POWER CONDITION shall be ignored.

9.7 SYNCHRONIZE CACHE (10) command (35h)

The SYNCHRONIZE CACHE (10) command is used to flush the most recent data values in the device cache to physical medium. Unlike in SCSI, ATA does not provide a way to specify a particular LBA to start flushing the device cache.

9.7.1 Command Summary

Table 24 — SYNCHRONIZE CACHE command summary

Support Method	Translated into the flush cache command.
ATA Opcodes	E7h

Editor's Note 38: Need to consider the Flush Cache Ext command for 48-bit devices (applicable to SYNCHRONIZE CACHE(16)?

9.7.2 Synchronize Cache CDB format and supported fields

a) IMMED Immediate Bit

This bit shall be ignored.

B) LOGICAL BLOCK ADDRESS

This field shall be ignored.

C) NUMBER OF BLOCKS

This field shall be ignored.

9.7.3 Data Returned

This command shall not return any data. Status shall be returned after the completion of the command.

9.7.4 Miscellaneous notes

The LBA and NUMBER OF BLOCK fields shall be ignored, as they cannot be translated to ATA command fields.

9.8 VERIFY (6,10,12) command (2Fh)

9.8.1 Command summary

The VERIFY command is used to verify data on medium which includes user data and protection data. This SCSI command is directly translated into ATA read verify sectors command or the extended version of that command.

Table 25 — VERIFY command summary

Support Method	Translated in ATA read verify sector(s) command. If the device supports48 bit mode, translated into ATA read verify extended command.
ATA Opcodes	E7h

9.8.2 VERIFY CDB format and supported fields

a) DPO (Disable pageout)

Ignored.

B) BLKVFY

Ignored.

C) BYTCHK

Ignored.

9.8.3 Data returned

None

9.8.4 Miscellaneous notes

a) If the LOGICAL BLOCK ADDRESS is larger than what can be accommodated using 28 bits and the device does not support 48-bit mode, a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to LOGICAL BLOCK ADDRESS OUT OF RANGE shall be returned.
b) If the LBA plus the VERIFICATION LENGTH is greater than the maximum sector that can be addressed on medium, return a CHECK CONDITION with SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE shall be set to LOGICAL BLOCK ADDRESS OUT OF RANGE.
c) The SATL LBA mapping algorithm shall assign the LSB of the LBA field from the SCSI CDB, i.e. byte 5, to LBALow in the ATA FIS, byte 4 from SCSI CDB to LBAMid, byte 3 from SCSI CDB to LBAHigh. If the device supports 48-bit addressing byte 2 of the SCSI CDB shall be assigned to LBALowExp.
d) The SectorCount in the ATA FIS is derived from the LSB, i.e. byte 8, of the VERIFICATION LENGTH of the SCSI CDB. If the device supports 48-bit addressing, the SectorCountExp is assigned byte 7 of the SCSI CDB, which is the MSB of the VERIFICATION LENGTH field.

e) Only READ and WRITE commands may be queued.

NOTE 9 - A performance degradation may be expected when a command such as VERIFY is issued as is must run in a single thread.

9.9 WRITE (06, 10, 12) command (0Ah, 2Ah, AAh)

The WRITE command is used to request the device to transfer user data to device medium or cache. Different versions of the command support different LBA sizes and or different transfer lengths. Data may be written to medium or the device cache.

9.9.1 Command summary

Table 26 — WRITE command summary

Support Method	Translated to ATA read commands using the following criterion.			
ATA Opcodes	***Is Queuing enabled***	***Are Extended Commands Enabled***	***SCSI OP Code***	***Translated ATA Opcode***
	NO	NO	0Ah,2Ah,AAh	WRITE_DMA (C8h)
	NO	YES	0Ah,2Ah,AAh	WRITE_DMA_EXT (25h)
	YES	(Don't care)	0Ah,2Ah,AAh	WRITE_FPDMA_QUEUED

9.9.2 CDB format and supported fields

a) DPO (disable page out)

Ignored.

b) FUA (force unit access)

Supported with NCQ capable devices.

NOTE 10 For non NCQ drives do we need to send a cache flush to force unit access?

C) WRPROTECT

Ignored.

D) TRANSFER LENGTH

A) For WRITE(06) commands (0Ah), TRANSFER LENGTH of 0 shall be interpreted as 256 sectors and is the same for SCSI and ATA.

B) For WRITE(10) commands (2Ah), TRANSFER LENGTH of 0 indicates that a data transfer shall not take place.

C) For WRITE(06) and WRITE(10) commands, if the TRANSFER LENGTH is not zero, it shall indicate the number of sectors to transfer from the device to the application client.

E) LOGICAL BLOACK ADDRESS

A) For WRITE(06) commands (0Ah), a 21 bit LBA shall be derived from bytes 3, 2, 1:5 in the CDB, with byte 3 being the LSB and the 5 bits from byte 1 being the MSB.

B) For WRITE(10) commands (2Ah), A 32 bit LBA shall be derived from bytes 2 through 5, where byte 5 is the LSB and byte 2 is the MSB.

9.9.3 Data Returned

Data blocks specified in the LOGICAL BLOACK ADDRESS field shall be transferred to the specified ATA device, and the device may transfer the data to its cache or medium. CHECK CONDITION shall be reported back if the LOGICAL BLOACK ADDRESS or the LOGICAL BLOACK ADDRESS plus one less than the TRANSFER LENGTH is greater than the maximum sector that can be addressed in medium, where the SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

9.9.4 Miscellaneous Notes

NOTE 11 - Protection information is not supported.

NOTE 12 - SCSI forced unit access mechanism is not supported.

For WRITE (12) command, if the size TRANSFER LENGTH field is greater than 16 bits, then the command shall terminate with a CHECK CONDITION where the SENSE KEY set to ILLEGAL REQUEST and ADDITIONAL SENSE CODE set to INVALID FIELD IN CDB.

NOTE 13 This is because the ATA command set supports a maximum of 16-bit sector count field.

Editor's Note 39: How will the SAT standard support ATA devices with capacities greater than 32 Mbytes?

9.10 WRITE AND VERIFY command (2Eh)

9.10.1 Command Summary

The WRITE AND VERIFY command is used to transfer application data to medium and then to verify that data was written correctly.

Table 27 — WRITE AND VERIFY command summary

Support Method	Translated to a ATA write command first and then to read verify sector(s) command.
ATA Opcodes	Clause 9.9 provides details on how the ATA write command is derived. 40h for read verify sector command if the device does not support extended commands feature set. 42h for READ VERIFY EXTENDED command if the device supports the extended commands feature set.

9.10.2 WRITE AND VERIFY CDB format and supported fields

A) BYTCHK

Ignored.

B) EBP

Ignored.

c) dpo (disable pageout)

Ignored.

D) LOGICAL BLOCK ADDRESS

Clause 9.9 provides details on how this field is translated into ATA write command LBA fields.

E) TRANSFER LENGTH

Translated into sector count of the ATA command. Note that TRANSFER LENGTH of 0 indicates that no data shall be transferred.

9.10.3 Data Returned

None

9.10.4 Miscellaneous Notes

Protection information is not supported.

10 Mode Pages and Log Pages

SCSI mode pages provide a mechanism to be used to set operating parameters for targets and logical units. MODE SENSE command is used to obtain operating parameters while MODE SELECT command is used to set operating parameters. In the context SCSI to SATA command translation most operating parameters defined by the contents of MODE PAGES are not supported due to lack of equivalent operations or features in SATA devices. The translator acts as the device server for all MODE SENSE and MODE SELECT commands and emulates the MODE PAGES listed in the table below.

In SCSI four types of mode page values are defined and they are described in Table 3.21.1.26. The 'save parameters' operation is not supported. Default values are manufacturing time initial values and are not changeable. All pages are shared between all logical units in the target. Mode Page Policy VPD is not implemented; therefore sharing of Mode pages across logical units is implied.

Fields whose contents are changeable and ignored can be changed by an application client without an error being reported back.

Currently, most mode pages can be used for information purposes only. Use of the MODE SELECT command to change the parameters are turned into no ops. Where a behavioral change takes place, this specification shall describe the ATA translation and the expected out come, where possible.

Editor's Note 40: SET FEATURES may be used to implement MODE SELECT parameters in some cases. Make this consistent throughout the document.

10.1 Commonly used SCSI Mode Pages

Table 28 lists the commonly used SCSI mode pages by operating system. This list is not exhaustive, and new entries shall be added as deemed necessary.

Table 28 — Commonly used SCSI mode pages

Mode Page	
Page Code	**Page Name**
01h	Read Write Error Recovery
02h	Disconnect-Reconnect
03h	Format Device
04h	Rigid Disk Geometry
08h	Caching
0Ah	Control
1Ch	Informational Exception Control
3Fh	Return All Pages

10.1.1 Read-Write Error Recovery mode page (01h)

This page contains the parameters used by the device server during error condition when read write commands are executed. Read write operations include all variants of the READ command, the WRITE command and WRITE AND VERIFY commands. Note that an ignored bit may be changed, unless otherwise stated, but the change shall have no effect on the operation of the target.

10.1.1.1 Read-Write Error Recovery Mode Page format and supported fields

a) PS (parameters savable)

This bit shall be set to 0b to indicate that parameters cannot be saved in a non-volatile location. It is reserved for MODE SELECT commands.

Editor's Note 41: Should it be left up to the SATL implemention to decide if it has the resources to save mode parameters or not?

---

b) DCR (disable correction)

This bit shall be set to 0b and ignored as data correction is done at drive level.

c) DTE (disable transfer on error)

This bit shall be ignored and set to 0 to indicate that data transfers shall not be terminated at the detection of a recovered error.

d) PER (Post Error)

This bit is ignored and set to 0 to indicate that recovered errors not reported to the user.

e) EER (Enable Early Recovery)

This bit is ignored set to 0 to indicate that the device server shall use the error recovery procedure that minimizes the risk of mis-detection or mis-correction.

f) RC (Read Continuous)

This bit is ignored and set to 0 to indicate that error recovery delays are acceptable and that Data shall not be fabricated.

g) TB (Transfer Block)

This bit is ignored and set to 0 to indicate that a logical block that is not recovered within the recovery limits specified shall not be transferred to the application client.

h) ARRE (Automatic Read Reallocation)

This bit is ignored and shall be set to 1 to indicate that automatic read reallocation of defective blocks shall always be enabled at drive level.

i) AWRE (Automatic Write Reallocation)

This bit is ignored and shall be set to 1 to indicate that automatic Write reallocation of defective blocks shall always be enabled at drive level.

j) READ RETRY COUNT

This field shall be set to 00h and ignored. This field identifies the maximum number of times the operation shall be retried by the device server in its recovery algorithm.

k) WRITE RETRY COUNT

This field shall be set to 00h and ignored. This field identifies the maximum number of times the operation shall be retried by the device server in its recovery algorithm.

l) RECOVERY TIME LIMIT

This field is ignored and set to 00h. This field specifies in increments of one millisecond, the maximum duration the device server shall use for error recovery.

10.1.2 Format Mode Page (03h)

(Made obsolete in SPI-3, Do we want this?)

a) TRACKS PER ZONE

This value would be set to 128. It specifies the number of tracks per zone in dividing the capacity of the drive for the purpose of allocating alternate sectors.

b) ALTERNATE SECTORS PER ZONE

This field shall be set to 00h and is ignored. This specifies the number of sectors the device server shall reserve per zone for defect handling.

c) ALTERNATE TRACKS PER ZONE

This field shall be set to 00h and is ignored. This specifies the number of the tracks device server shall reserve per zone for defect handling.

d) ALTERNATE TRACKS PER LOGICAL UNIT

This field shall be set to 00h and is ignored. This specifies the number of the tracks device server shall reserve per logical units for defect handling.

e) SECTORS PER TRACK

This value shall be set to 128. Self-explanatory.

f) BLOCK SIZE

This value shall be set to 512 bytes. Self-explanatory.

Editor's Note 42: ATA/ATAPI-7 IDENTIFY DEVICE words 117-118 allow an ATA device to report other sector sizes. Should this be reflected here?

---

g) INTERLEAVE

This value shall be set to 00h. This specifies the same value from the last format operation. Only a value of 00h is supported in the original format command, which implies that the default interleave mechanism for the drive shall be used.

h) TRACK SKEW FACTOR

This value shall be set to 01h. specifies the number of physical sectors between the last logical block of one cylinder and the first logical block the next sequential cylinder.

i) SSEC (Soft Sector Formatting)

Set to 0 and ignored. Mutually exclusive with HSEC bit.

j) HSEC (Hard Sector formatting)

This bit shall be set to 1b to indicate that hard sector formatting shall be used.

k) RMB (Removable media)

This bit shall be set to 0 and ignored.

l) SURF (Surface)

This bit shall be set to 0 and ignored.

10.1.3 Rigid Disk Device Geometry Page (04h)

a) NUMBER OF CYLINDERS

This field shall be set to the number of physical cylinders used for data storage.

b) NUMBER OF HEADS

This field shall be set to the number of heads used for data storage.

c) START CYLINDER FOR WRITE PRECOMPENSATION

This value shall be set to 0 and ignored.

d) START CYLINDER REDUCED WRITE CURRENT

This value shall be set to 0 and ignored.

e) DEVICE STEP RATE

This value shall be set to 0 and ignored.

f) LANDING ZONE CYLINDER

This value shall be set to 0 and ignored.

g) ROTATIONAL OFFSET

This value shall be set to 0 and ignored.

h) MEDIUM ROTATION RATE

This value shall be set to the rotation rate of the device. Units are rotation per minute.

Editor's Note 43: Is the medium rotation rate available from an ATA/ATAPI device?

10.1.4 Caching Mode Page (08h)

Caching Mode page (08h) defines parameters that affect the behavior of the device cache and the caching policy used.

a) PS (Parameters Savable)

This bit shall be set to 0b to indicate that parameters cannot be save in a non-volatile location. It is reserved for MODE SELECT commands.

b) RCD (Read Cache Disable)

This bit shall be set to 0 and ignored. Set DRA bit to prevent read ahead, which is the equivalent of enabling and disabling of read cache in SATA.

Editor's Note 44: I believe an RCD bit of 1 disables the cache. Is there a reason not to use the ATA device cache if one is available?

---

c) MF (Multiplication Factor)

This bit is set to 0 and is ignored.

d) WCE (Write Cache Enable)

By default, this bit shall be set to 1 to indicate that write caching is enabled.

If this bit is set to 0 using a MODE SELECT command, it shall be translated into a SET FEATURES command 82h to disable write cache.

If this bit is set to 1 using a MODE SELECT command, it shall be translated into a SET FEATURES command 02h to enable write cache.

This bit it set to the default values after a logical unit reset or a target reset.

e) SIZE

This bit is set to 0 and is ignored. This bit indicates whether CACHE SEGMENT SIZE field or NUMBE ROF CACHE SEGMENTS field to use for caching algorithms.

f) DISC (Discontinuity)

This bit is set to 0 and is ignored. This bit defines whether prefetches can be continued across discontinuities.

g) CAP (Caching analysis permitted)

This bit is set to 0 to indicate that caching analysis is disabled and ignored for a MODE SELECT command.

h) ABPF (Abort Prefetch)

This bit is set to 0 and ignored. This bit is used in conjunction with the DRA bit and defines how prefetch operation is controlled when a new command is received.

i) IC (initiator Control)

This bit is set to 0 and ignored. Initiator control bit set to 1 specifies that the device server use the NUMBER OF CACHE SEGMENTS or the CACHE SEGMENT SIZE, depending on the SIZE bit to control the caching algorithm.

j) WRITE RETENTION PRIORITY

This field is set to zero and ignored.

k) DEMAN READ RETENTION PRIORITY

This field is set to zero and ignored.

l) DISABLE PRE-FETCH TRANSFER LENGTH

This field is set to zero and ignored. This field specifies the cut off transfer length in number blocks to disable prefetch for long transfers.

m) MINIMUM PRE-FETCH

This field is set to zero and ignored.

n) MAXIMUM PRE-FETCH

This field is set to zero and ignored.

o) MAXIMUM PRE-FETCH CEILING

This field is set to zero and ignored.

p) DRA (Disable Read Ahead)

By default, this bit shall be set to 0 to indicate that read look-ahead is enabled.

If this bit is set to 1 using a MODE SELECT command, it shall be translated into a SET FEATURES command 55h to disable read look-ahead.

If this bit is set to 0 using a MODE SELECT command, it shall be translated into a SET FEATURES command AAh to enable read look-ahead.

This bit it set to the default values after a logical unit reset or a target reset.

q) LBCSS (Logical block cache segment size)

This field is set to zero and ignored. This bit specifies the units to be used for CACHE SEGMENT SIZE field.

r) FSW (Force Sequential Writes)

This field is set to zero and ignored.

s) NUMBER OF CACHE SEGMENTS

This field is set to zero and ignored.

t) CACHE SEGMENT SIZE

This field is set to zero and ignored.

u) NON-CACHE SEGMENT SIZE

This field is set to zero and ignored.

10.1.5 Informational Exceptions Control Mode Page (1Ch)

This page allows the initiator to specify the methods used to control the reporting and operation under specific informational exception conditions. This page applies to informational exceptions that report an ADDITIONAL SENSE CODE of FAILURE PREDICTION THRESHOLD EXCEEDED or WARNING to the application client.

a) PS (Parameters Savable)

This bit shall be set to 0b to indicate that parameters cannot be save in a non-volatile location. It is reserved for MODE SELECT commands.

b) LOGERR (Log Error)

This bit is set to 0 and ignored. Indicates that logging of informational exceptions is vendor specific.

c) TEST

This bit is set to 0b to indicate reporting false device failure notifications is not supported.

d) DEXCPT (Disable Exception Control)

Set to 0, enables informational exceptions reporting using the method defined by MRIE. This is the default value. SMART is enabled if this bit is 0b.

Set to 1, disables informational exceptions reporting using the method defined by MRIE. SMART is disabled if this bit is 0b.

e) EWASC (enable Warning)

This bit is set to 0 and is ignored. Reporting of warning shall be disabled.

f) EBF (Enable Background Functions)

This bit is set to 0 and is ignored. Enabling of background functions shall be disabled. PERF

g) (Performance)

This bit is set to 0 and is ignored.

h) MIRE (Method of reporting Informational Exceptions)

Only a value of 6h (report on request) is supported.

i) INTERVAL TIMER

This bit is set to 0 and is ignored. This field defines the interval in 100 milliseconds that the device shall be polled for SMART threshold conditions. Conditions shall be reported after the interval has elapsed. Check section 6 for more details on SMART.

j) REPORT COUNT

This field is set to 0 and ignored. The number of times a condition can be reported is unlimited.

10.2 Log Pages

Editor's Note 45: This clause shall describe any log pages applicable to SCSI/ATA Translation.

11 SMART HANDLING

This section describes the SMART features used by most drive targets to monitor the health of the drives by application software.

Needs work. TBD

11.1

11.2 6.1 General Description of SMART

Provide a method for an application client to query for SMART attribute threshold exceeding conditions. The host shall be notified in a manner specified by the Informational Exception Mode page parameters. This page is also used to enable or disable SMART feature in full. Once the application software becomes aware of a threshold exceeding condition, it is the duty of application software to retrieve SMART data, analyze it and take necessary actions. A simple method is provided to retrieve SMART data. Following sub sections address these aspects in more details.

11.3 6.2 Host Notification

The method used to notify the host of a SMART threshold exceeding condition is specified in the Informational Exceptions Mode Page, using the MRIE field. Only a value of 06h is supported for this field as described in section 4.6. A value of 06h indicates that the application client shall send unsolicited request sense commands described in 3.6, to monitor SMART errors. When an unsolicited request sense command is recieved, issue an ATA SMART RETURN STATUS command if the target device is SMART capable. (ATA opcode B0h with features register set to DAh). Upon receiving results, if a threshold exceeding condition has been detected and met, sense data with the SENSE KEY set to no sense and ADDITIONAL SENSE CODE set to general hardware failure shall be returned. (ASC 5Dh, ASCQ 10h).

11.4 6.3 Disabling, enabling SMART

To enable or disable SMART reporting use the DEXCPT bit in the Informational Exceptions Mode Page as described in section 4.6, using a mode select command. A value of 0, which is the default, indicates to enable informational exceptions reporting while a value of 1 indicates to disable informational exceptions reporting. Do not issue SMART RETURN STATUS command, if DEXCPT bit is set to 1.

11.5 6.4 Retrieving SMART data from targets

Application software should use LOG SENSE command described in section 3.20 (Update Reference), to retrieve 512 byte SMART unaltered data from drives. Only page 31h shall be supported under LOG SENSE command and only current values as specified by the PC field (00b) shall be reported to the client. The page header shall not be appended to SMART data. Also the LOG SELECT command is not supported. Transfer length can be controlled using the ALLOCATION LENGTH field in the LOG SENSE CDB.

12 Error Handling and Sense Reporting

12.1 Example of a 6 byte CDB format

An example 6 byte CDB is shown in table 1. An example data structure is shown in table 2. Edit, cut, copy, paste as desired. These examples are also included in Annex B with additional examples.

Table 29 — Example 6 byte CDB

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (03h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	ALLOCATION LENGTH							
5	CONTROL							

12.2 Example data structure

Example data structure that you can cut, copy, paste, etc.

Table 30 — Example data structure

Bit Byte	7	6	5	4	3	2	1	0
0	VALID	RESPONSE CODE (70h or 71h)						
1	Obsolete							
2	FILEMARK	EOM	ILI	Reserved	SENSE KEY			
3	(MSB)	INFORMATION						
6								(LSB)
7	ADDITIONAL SENSE LENGTH (n-7)							
8	(MSB)	COMMAND-SPECIFIC INFORMATION						
11								(LSB)
12	ADDITIONAL SENSE CODE							
13	ADDITIONAL SENSE CODE QUALIFIER							
14	FIELD REPLACEABLE UNIT CODE							
15	SKSV	SENSE-KEY SPECIFIC						
17								
18	Additional sense bytes							
n								

Annex A
(Informational)

Guides to Implementation of SCSI / ATA Translation

A.1 General

This annex describes methods used to implement various SCSI features which may be implemented in the SATL, but are not specifically dictated in the normative text of the document.