SNTr0 Initial Revision Proposal

24-012r0

Technical Editor:

Curtis E. Stevens
Phone: 949-307-5050
E-Mail: Curtis.Stevens@Seagate.com

The following document looks like a realistic draft, but has not been approved as a draft at this time.

Table 1 – Revision History

Date	Rev	Description
January 8, 2024	0	Initial Draft
		1.

This proposal is for the initial draft of SNT.

Working Draft Project American NationalT10/BSR INCITS 584 Standard

Revision 0
January 8, 2024

Information technology - SCSI / NVMe Trabslation (SNT)

T10 Technical Editor: Curtis E. Stevens
Seagate Technology
47488 Kato Road
Fremont, CA 94538
USA

Telephone: (949) 307-5050
Email: Curtis.Stevens@Seagate.com

Reference number
ISO/IEC xxxx-xxx:202x
ANSI INCITS 584-20xx

Printed 5:02 PM Monday 08 January 2024

Points of Contact

InterNational Committee for Information Technology Standards (INCITS) T10 Technical Committee

T10 Chair
William Martin
Samsung Semiconductor, Inc
7213 Marblethorpe Drive
Roseville, CA 95747-5925
USA

Telephone: (916) 765-6875
Email: bill dot martin at ssi dot samsung dot com

T10 Vice-Chair
Curtis Ballard
Hewlett Packard Enterprise
3404 East Harmony Road
Fort Collins CO 80528
USA

Telephone: (970) 898-6669
Email: curtis dot ballard at hpe dot com

T10 Web Site: http://www.t10.org

INCITS Secretariat
700 K Street NW Suite 600
Washington, DC 20001
USA

Telephone: (202) 737-8888
Web site: http://www.incits.org
Email: incits@itic.org

Information Technology Industry Council
Web site: http://www.itic.org

Purchase INCITS Standards
Web site: http://www.incits.org/standards-information/purchase-standards-or-download-dpans

American National Standard
for Information Technology

T10 Style Guide 5th Edition

Secretariat
Information Technology Industry Council

Approved mm.dd.yy

American National Standards Institute, Inc.

ABSTRACT

This standard specifies something of interest, and here is where you give a brief description of that. This abstract should be confined to one or two short paragraphs.

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 towards their resolution.

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

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an 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.

The patent statement goes here.

If no patents have been disclosed, then use the no disclosed.

If any patents have been disclosed, then use the patents disclosed statement.

Published by
American National Standards Institute
25 West 43rd Street 4th floor, New York, New York 10036-7422

.

Printed in the United States of America

Editor's Note 1: Make sure the copyright year is correct.

Revision Information

R.1 Revision SNTr00 (8-Jan-2024)

This is revision 0 of this working draft.

Contents

Tables

Figures

FOREWORD (This foreword is not part of American National Standard INCITS 584.)

This foreword is not part of this standard.

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 NVMe devices and a SCSI/ NVMe Translation Layer (SNTL). The SNTL may reside in a host-based software or firmware, or it may reside in a separate component (e.g., a host bus adapter or external controller) with a separate processing unit to perform the translation. A SNTL and NVMe device combination may provide a functional subset of common SCSI capabilities. There is also a range of optional emulated SCSI capabilities that may be supported, depending on the capabilities of the SNTL.

This standard defines SNTL capabilities in terms of SCSI capabilities as defined by the applicable SCSI standards and working drafts, and defines the elements and use of the NVMe protocol to provide those SCSI capabilities and services in a consistent manner among SNT implementations that are implemented as described by this standard.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, InterNational Committee for Information Technology Standards, Information Technology Industry Council, 700 K Street NW Suite 600, Washington, DC 20001.

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 it approved this standard, INCITS had the following members:

INCITS Technical Committee T10 on SCSI Storage Interfaces, which developed and reviewed this standard, had the following members:

William Martin, Chair
Curtis Ballard, Vice-Chair
Curtis Stevens, Secretary

Organization Represented *Name of Representative*

INTRODUCTION

General

The SCSI / NVMe Trabslation (SNT) standard is divided into the following clauses and annexes:

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 NVMe protocols.
Clause 5 describes elements of SCSI / NVMe Translation that relate to the SCSI architecture model.
Clause 6 describes the mapping of command management functions in the SNTL layer.
Clause 7 provides a summary of SCSI commands mapped to NVMe in this standard.
Clause 8 describes the mapping between SCSI Primary Commands and NVMe protocol.
Clause 9 describes the mapping between SCSI Block Commands, Zoned Block Commands, and NVMe protocol.
Clause 10 describes the mapping of mode pages, log pages, and VPD page information to selected NVMe protocol elements.

Clause 11 describes error reporting and sense data conventions for SCSI / ATA Translation.
Clause 12 describes SCSI commands and mode pages to support SCSI / ATA Translation.

MY standards family

Figure 0 shows the relationship of this standard to the other standards and related projects in the SNT family of standards as of the publication of this standard.

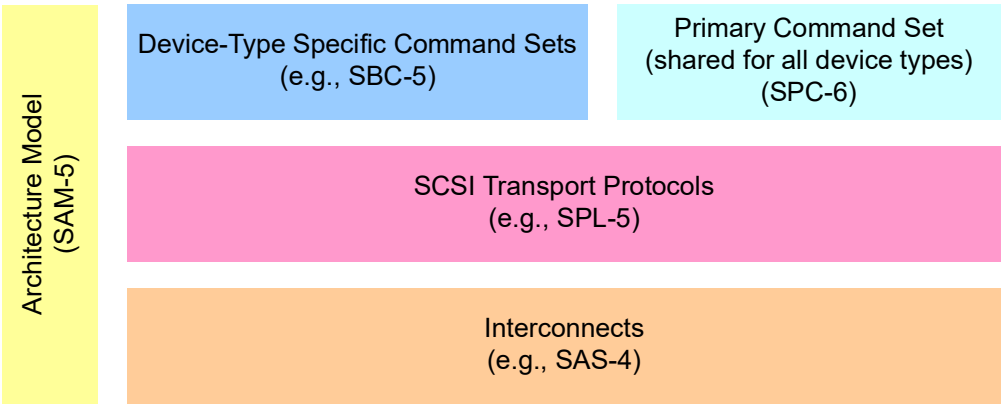

Figure 0 – SNT document relationships

The roadmap in figure 0 is intended to show the general applicability of the documents to one another. Figure 0 is not intended to imply any hierarchy, protocol stack, or system architecture relationship.

Figure 1 shows the relationship of the NVMe standard and specifications each other.

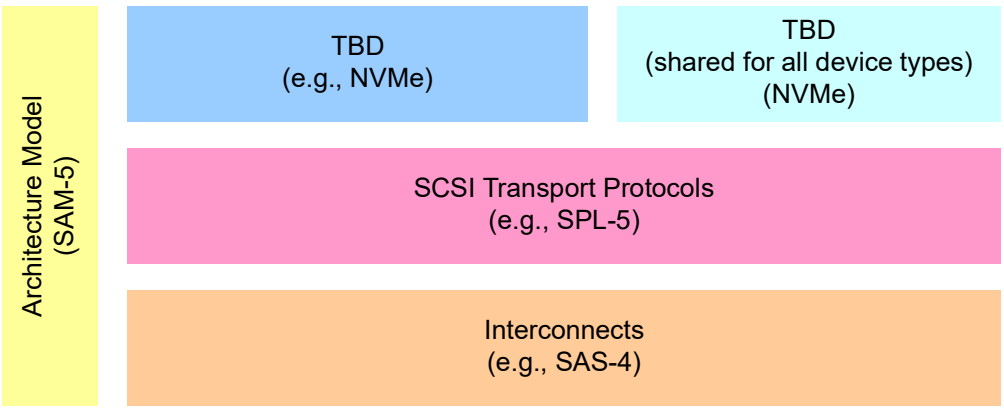

Figure 1 – SNT document relationships

This standard defines a translation between the SCSI application layer (see SAM-5) and NVMe device protocol.

Editor's Note 1: This standard has made obsolete the following:

Editor's Note 2: The special function that no one uses.

American National Standard
for Information Technology -

SCSI / NVMe Trabslation (SNT)

1 Scope

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

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

Where possible, this standard defines SCSI / NVMe Translation in a manner that is consistent with the SAM-5, SPC-6, SBC-5, and ZBC-2 standards. In some instances, the defined function of an NVMe device is different from corresponding functions defined for SCSI target devices (e.g., many NVMe devices provide no means to abort a single NVMe queued command). The translation defined in this standard, in such cases, may not be consistent with other SCSI standards. However, in such cases, this standard specifies the expected behavior, and in what manner it is inconsistent with the behavior specified in other SCSI standards.

The objective of this standard is to allow an interoperable set of SCSI functions while minimizing the complexity of the SNTL and preserving compatibility with existing SCSI application clients.

The objectives of the SNTL are:

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

2 Normative References

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14165-117, *Information technology – Fibre channel – Part 117: Methodologies for jitter and signal quality (MJSQ)*[1]

ISO/IEC 14776-261, *Information technology – Small Computer System Interface (SCSI) – Part 261*:

ISO/IEC 14776-414, *Information technology – Small Computer System Interface (SCSI) – Part 414: SCSI Architecture Model - 4 (SAM-4)*

ISO/IEC 14776-261, *Information technology – Small Computer System Interface (SCSI) – Part 261: SAS Protocol Layer (SPL)* (under consideration)

T10/BSR INCITS 515, *SCSI Architecture Model - 5 (SAM-5)* (planned as ISO/IEC 14776-415).

T10/BSR INCITS 492, *SAS Protocol Layer-3 (SPL-3)* (planned as ISO/IEC 14776-263)

ANSI INCITS 451-2008, *Information technology – AT Attachment-8 ATA/ATAPI Architecture Model (ATA8-AAM)*

For information on the current status of the listed documents, or regarding availability, contact the indicated organization.

Serial ATA Revision 3.2 (SATA). 7 August 2013

NOTE 1 - For more information on Serial ATA international Organization, see www.sata-io.org.

1.When MJSQ is referenced from this standard, the FC Port terminology used within MJSQ should be substituted with SAS phy terminology.

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 aggregation
when referring to classes, a form of association that defines a whole-part relationship between the whole (i.e., aggregate) and its parts

3.1.2 3.1.3 byte
8-bit construct

3.1.4 3.1.5 3.1.6 cyclic redundancy check
error checking mechanism that checks data integrity by computing a polynomial algorithm based checksum

3.1.7 field
group of one or more contiguous bits

3.1.8 NVMe device
device that complies with NVMe standards.

3.1.9 object
entity with a well-defined boundary and identity that encapsulates state and behavior

Note 1 to entry: All objects are instances of classes (i.e., a concrete manifestation of a class is an object).

3.2 Symbols and abbreviations

3.2.1 Abbreviations

Abbreviations used in this standard:

Abbreviation	Meaning
B	byte
CRC	Cyclic Redundancy Check (see 3.1.6)
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
SBC-2	SCSI Block Commands-2 (see clause)
SPC-3	SCSI Primary Commands-3 (see clause)
SNT	SCSI / NVMe Translation
SNTL	SCSI / NVMe Translation Layer

3.2.2 Units

Editor's Note 3: Subclause 3.2.2 is an example of what you may want in your units list. Edit as appropriate. If no units are used, then 3.2.2 should be removed from the working draft.

Units used in this standard:

Units	Meaning
µs	microsecond (i.e., 10^{-6} seconds)
m	meter
ms	millisecond (i.e., 10^{-3} seconds)
ns	nanosecond (i.e., 10^{-9} seconds)
s	second (unit of time)
V	volt
AWG	American wire gauge

3.2.3 Symbols

Symbols used in this standard:

Symbols	Meaning
Kxx.y	control character
Dxx.y	data character
®	registered trademark

3.2.4 Mathematical operators

Mathematical operators used in this standard:

Mathematical Operators	Meaning
^ or XOR	exclusive logical OR
×	multiplication
/	division
±	plus or minus
≈	approximately
~	approximately equal to
+	add
-	subtract
< or LT	less than
≤ or LE	less than or equal to
= or EQ	equal
≠ or NE	not equal

Mathematical Operators	Meaning
> or GT	greater than
≥ or GE	greater than or equal to
|v|	the absolute value (i.e., magnitude) of v
Î	set membership

3.3 Keywords

3.3.1 invalid
keyword used to describe an illegal or unsupported bit, byte, word, field, or code value

Note 1 to entry: Receipt by a device server of an invalid bit, byte, word, field, or code value shall be reported as an error.

3.3.2 mandatory
keyword indicating an item that is required to be implemented as defined in this standard

3.3.3 may
keyword that indicates flexibility of choice with no implied preference

3.3.4 may or may not
keyword that indicates flexibility of choice with no implied preference

Note 1 to entry: Significant uses of "may or may not" occur in descriptions where attention is being drawn to the "may not" case.

3.3.5 obsolete
keyword indicating that an item was defined in prior SCSI standards but has been removed from this standard

3.3.6 option, optional
keywords that describe features that are not required to be implemented by this standard

Note 1 to entry: If any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard.

3.3.7 prohibited
keyword used to describe a feature, function, or coded value that is defined in a non-SCSI standard (i.e., a standard that is not a member of the SCSI family of standards) to which this standard makes a normative reference where the use of said feature, function, or coded value is not allowed for implementations of this standard

3.3.8 reserved
keyword referring to bits, bytes, words, fields, and code values that are set aside for future standardization

Note 1 to entry: A reserved bit, byte, word, or field shall be set to zero, or in accordance with a future extension to this standard.

Note 2 to entry: Recipients are not required to check reserved bits, bytes, words, or fields for zero values.

Note 3 to entry: Receipt of reserved code values in defined fields shall be reported as error.

3.3.9 restricted
keyword referring to bits, bytes, words, and fields that are set aside for other identified standardization purposes

Note 1 to entry: A restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word, or field in the context where the restricted designation appears.

3.3.10 shall
keyword indicating a mandatory requirement

Note 1 to entry: Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

3.3.11 should
keyword indicating flexibility of choice with a strongly preferred alternative

3.3.12 vendor specific
something (e.g., a bit, field, code value) that is not defined by this standard

Note 1 to entry: Specification of the referenced item is determined by the SCSI device vendor and may be used differently in various implementations.

3.4 Editorial conventions

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in the glossary or in the text where they first appear.

Upper case is used when referring to the name of a numeric value defined in this specification or a formal attribute possessed by an entity. When necessary for clarity, names of objects, procedure calls, arguments or discrete states are capitalized or set in bold type. Names of fields are identified using small capital letters (e.g., NACA bit).

Names of procedure calls are identified by a name in bold type (e.g., **Execute Command**). Names of arguments are denoted by capitalizing each word in the name (e.g., Sense Data is the name of an argument in the **Execute Command** procedure call). For more information on procedure calls see 3.7.

Quantities having a defined numeric value are identified by large capital letters (e.g., CHECK CONDITION). Quantities having a discrete but unspecified value are identified using small capital letters. (e.g., TASK COMPLETE, indicates a quantity returned by the **Execute Command** procedure call). Such quantities are associated with an event or indication whose observable behavior or value is specific to a given implementation standard.

Lists sequenced by lowercase or uppercase letters show no ordering relationship between the listed items.

EXAMPLE 1 - The following list shows no relationship between the named items:

a) various forms of red such as:
   A) crimson; or
   B) amber;
b) blue; or
c) green.

Lists sequenced by numbers show an ordering relationship between the listed items.

EXAMPLE 2 - The following list shows an ordered relationship between the named items:

1) top;
2) middle; and
3) bottom.

Lists are associated with an introductory paragraph or phrase, and are numbered relative to that paragraph or phrase (i.e., all lists begin with an a) or 1) entry).

If a conflict arises between text, tables, or figures, then the order of precedence to resolve the conflicts is:

1) text;
2) tables; and
3) figures.

Not all tables or figures are fully described in the text.

Notes and examples do not constitute any requirements.

Notes are numbered consecutively throughout this standard.

3.5 Numeric and character conventions

3.5.1 Numeric conventions

A binary number is represented in this standard by any sequence of digits comprised of only the Arabic numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included in binary number representations to increase readability or delineate organizational boundaries (e.g., 00010101 11001110b, 00010101_11001110b, 0 0101 1010b or 0_0101_1010b).

A hexadecimal number is represented in this standard by any sequence of digits comprised of only the Arabic numerals 0 to 9 and/or the upper-case English letters A to F immediately followed by a lower-case h (e.g., FA23h). Underscores or spaces may be included in hexadecimal number representations to increase readability or delineate organizational boundaries (e.g., 3456FDCA 84BD5E7Ah, 3456FDCA_84BD5E7Ah, B FD8C FA23h, or B_FD8C_FA23h).

A decimal number is represented in this standard by any sequence of digits comprised of only the Arabic numerals 0 to 9 not immediately followed by a lower-case b or lower-case h (e.g., 25).

Variables (i.e., alphanumeric names that represent values in computations and other statements) are represented in the same San-serif font as other information in this standard.

This standard uses the following conventions for representing decimal numbers:

a) the decimal separator (i.e., separating the integer and fractional portions of the number) is a period;
b) the thousands separator (i.e., separating groups of three digits in a portion of the number) is a space;
c) the thousands separator is used in both the integer portion and the fraction portion of a number; and
d) the decimal representation for a year is 1999 not 1 999.

Table 1 shows some examples of decimal numbers using various conventions.

Table 1 – Numbering conventions

French	English	This standard
0,6	0.6	0.6
3,141 592 65	3.14159265	3.141 592 65
1 000	1,000	1 000
1 323 462,95	1,323,462.95	1 323 462.95

A decimal number represented in this standard with an overline over one or more digits following the decimal point is a number where the overlined digits are infinitely repeating (e.g., $666.\overline{6}$ means 666.666 666... or 666 2/3, and $12.\overline{142\ 857}$ means 12.142 857 142 857... or 12 1/7).

A range of numeric values is represented in this standard in the form “a to z”, where a is the first value included in the range, all values between a and z are included in the range, and z is the last value included in the range (e.g., the representation “0h to 3h” includes the values 0h, 1h, 2h, and 3h).

3.5.2 Units of measure

This standard represents values using both decimal units of measure and binary units of measure. Values are represented by the following formats:

a) for values based on decimal units of measure:
   1) numerical value (e.g., 100);
   2) space; and
   3) prefix symbol and unit:
      1) decimal prefix symbol (e.g., M) (see table 2); and
      2) unit abbreviation;

   and

b) for values based on binary units of measure:
   1) numerical value (e.g., 1 024);
   2) space; and
   3) prefix symbol and unit:
      1) binary prefix symbol (e.g., Gi) (see table 2); and
      2) unit abbreviation.

Table 2 compares the prefix, symbols, and power of the binary and decimal units.

Table 2 – Comparison of decimal prefixes and binary prefixes

Decimal			Binary		
Prefix name	**Prefix symbol**	**Power (base-10)**	**Prefix name**	**Prefix symbol**	**Power (base-2)**
kilo	k	10^3	kibi	Ki	2^{10}
mega	M	10^6	mebi	Mi	2^{20}
giga	G	10^9	gibi	Gi	2^{30}
tera	T	10^{12}	tebi	Ti	2^{40}
peta	P	10^{15}	pebi	Pi	2^{50}
exa	E	10^{18}	exbi	Ei	2^{60}
zetta	Z	10^{21}	zebi	Zi	2^{70}
yotta	Y	10^{24}	yobi	Yi	2^{80}

3.5.3 Byte encoded character strings conventions

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

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

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

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

Editor's Note 4: The next few subclauses define conventions for UML diagrams, state machines, and procedure calls. Any of these may be deleted if the described convention is not used in the draft standard.

3.6 Bit and byte ordering

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

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

If a table shows bit numbering (see table 3), the least significant bit (LSB) is numbered 0 and each more significant bit has the next greater number than the immediately less significant bit. If a table shows numbering of bytes or characters (see table 4), the most significant byte or character is represented at the lowest number and each less significant byte or character has the next greater number than the immediately more significant byte.

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

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

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

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

A data dword consists of 32 bits. Table 3 shows a data dword containing a single value, where the MSB is on

the upper left in bit 31 and the LSB is on the lower right in bit 0.

Table 3 – Example of ordering of bits and bytes within a data dword

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB) Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
Note - The Bit x labels in the individual table cells are for reference only and should not appear within tables that use this element format.								

Table 4 shows a data dword containing four one-byte elements, where byte 0 (the first byte) is on the top and byte 3 (the fourth byte) is on the bottom. Each byte has an MSB on the left and an LSB on the right.

Table 4 – Example of ordering of bits and bytes within a data dword element

Bit Byte	7	6	5	4	3	2	1	0
0	First byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
1	Second byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
2	Third byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
3	Fourth byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
Note - The Bit x labels in the individual table cells and the xx byte labels in the individual bytes are for reference only and should not appear within tables that use these element formats. In this example the MSB and LSB labels are for reference only. However, they may appear in multi-byte fields as described in this subclause.								

3.7 Notation for procedure calls

In this standard, the model for functional interfaces between entities is a procedure call. Such interfaces are specified using the following notation:

[Result =] Procedure Name (IN ([input-1] [,input-2] …), OUT ([output-1] [,output-2] …))

Where:

Result	A single value representing the outcome of the procedure or function.
Procedure Name	A descriptive name for the function to be performed.
IN (Input-1, Input-2, …)	A comma-separated list of names identifying caller-supplied input data objects.
OUT (Output-1, Output-2, …)	A comma-separated list of names identifying output data objects to be returned by the procedure.
[…]	Brackets enclose optional or conditional parameters and arguments.

This notation allows arguments to be specified as inputs and outputs. An interface between entities may require only inputs. If a procedure call has no output arguments, the word OUT, preceding comma, and associated pair of balanced parentheses are omitted.

The following is an example of a procedure call specification:

Found = **Search (**IN (Pattern, Item List), OUT ([Item Found]))

Input arguments:

Pattern: Argument containing the search pattern.

Item List: **Item<NN>** contains the items to be searched for a match.

Output arguments:

Item Found: Item located by the search procedure call. This argument is only returned if the search succeeds.

4 General

This standard defines a SCSI / NVMe Translation Layer (i.e., the SNTL) that provides a method for a SCSI application layer (see SAM-5) to access NVMe devices by representing NVMe devices as SCSI peripheral devices.

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

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

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

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

a) the consistent reporting by the SNTL of capabilities of emulated SCSI devices;
b) the consistent observed behavior for SCSI operations; and
c) the consistent identification of the attached devices by the application clients.

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

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

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

This standard refers to behaviors for SCSI devices defined in SBC-5, ZBC-2, and SPC-6. Unless otherwise specified, any behaviors that are optional in SBC-5 or SPC-6 are optional for devices implementing SCSI / NVMe Translation.

If the SNTL receives a SCSI request specifying any value in any field of the CDB that the SNTL does not support then, unless otherwise specified in the description of the command, the SNTL shall terminate the SCSI command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB (see SPC-6).

If the SNTL receives a SCSI request specifying any value in any field of the parameter data that the SNTL does not support then, unless otherwise specified in the description of the parameter, the SNTL shall terminate the SCSI command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST (see SPC-6).

5 SCSI Architecture

Need to import architecture from SAT-6.

6 Command management model

6.1 Overview

A SNTL may support the full task management model or the basic task management model as well as specific features of the task management model (e.g., SIMPLE and ORDERED task attributes) depending on the task management capabilities of the SATL and whether the SATL supports NCQ.

7 Summary of SCSI / NVMe command mappings

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

Editor's Note 5: Is device type 00h correct for SSD's?

Clause 7, clause 8, and clause 9 describe the SCSI to NVMe command mapping for NVMe devices emulating a SCSI logical unit with a peripheral device type of 00h (i.e., direct access block device) or 14h (i.e., host managed zoned device).

Table 12 lists the SCSI / ATA command mappings defined in this standard. A SNTL may implement commands defined in SPC-6, SBC-5, and ZBC-2, but not listed in table 12. Translation of commands not listed in table 12 is vendor specific. If a command is not implemented by the SNTL, then the SNTL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE.

Table 5 – Summary of SCSI / NVMe command mapping

SCSI command	NVMe commands	Reference

8 SPC-5 command mapping

8.1 INQUIRY command

8.1.1 INQUIRY command overview

...

9 SBC-5 and ZBC-2 command mapping

10 Parameters for SNT implementations

10.1 Overview

Parameters for all device types are defined in this clause as follows:

a) diagnostic parameters are defined in 10.2xxx;
b) log parameters are defined in 10.3xxx;
c) mode parameters are defined in 10.4xxx; and
d) vital product data parameters are defined in 10.5xxx.

10.2 Diagnostic parameters

10.2.1 General Information

SCSI diagnostic parameters provide a mechanism to initiate diagnostic functions and return results for those diagnostic functions. The SEND DIAGNOSTIC command is used to initiate diagnostic functions and the RECEIVE DIAGNOSTIC RESULTS command is used to obtain results from those functions.

10.3 Log parameters

10.3.1 Overview

Editor's Note 5: List of translated log pages

10.4 Mode paramteters

10.4.1 General information

SCSI mode parameters provide a mechanism to set operating parameters for SCSI devices and logical units. The MODE SENSE command obtains operating parameters and the MODE SELECT command sets operating parameters. This standard does not define the content of most operating parameters defined in mode pages due to lack of equivalent operations or features defined for NVMe devices. The SATL emulates a SCSI device server for all MODE SENSE commands and MODE SELECT commands, and shall emulate the mode pages listed in 10.4.3xxx.

The Mode Page Policy VPD page (see 10.5xxx) should be implemented. If implemented, then the MODE PAGE POLICY field in each mode page policy descriptor should be set to 00b (i.e., shared) for each mode page and only one copy of mode page values should be maintained for all logical units within a target device (i.e., the MLUS bit is set to one in each mode page policy descriptor).

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

10.4.2 Common mode page structures

The format of mode parameter headers used for all pages is as described in 10.4.4xxx.The format of the optional mode parameter block descriptors used for all mode pages is as described in 10.4.5xxx.

10.5 Vital product data parameters

10.5.1 Overview

Editor's Note 6: add a table with the lists of pages

11 Translation of NVMe errors to SCSI errors

11.1 Overview

Unless otherwise specified in the subclause describing the translation of a particular SCSI command, log page, mode page, or VPD page, the SATL shall translate ATA commands that complete with an error to SCSI errors as shown in table xxx.

12 SNT specific SCSI extensions

12.1 Overview

This clause defines additional SCSI commands, mode pages, security protocols, and VPD pages that may be supported by a SNTL to provide capabilities in addition to the capabilities defined in the other SCSI command sets.

For SCSI commands specific to SNTL implementations see 12.2.

For Mode pages specific to SNTL implementations see 12.3.

For VPD pages specific to SNTL implementations see 12.4.

For Security protocols specific to SNTL implementations see 12.5.

For log pages specific to SNTL implementations see 12.6.

12.2 SNT specific SCSI extensions

Editor's Note 5: NVMe pass through goes here

12.3 SNT specific mode pages

12.4 SNT specific VPD pages

12.5 SNT specific security protocols

12.6 SNT specific log pages

Annex A
(informative)

Sample algorithms for splitting commands

A.1 TBD

Bibliography

SO/IEC 14776-223, *Information technology – Small Computer System Interface (SCSI) – Part 223: Fibre Channel-3 (FCP-3)*

T10/BSR INCITS 513, *SCSI Primary Commands-4 (SPC-4)* (planned as ISO/IEC 14776-454)

ISO/IEC 14776-333, *Information technology – Small Computer System Interface (SCSI) – Part 333: SCSI-3 Stream Commands (SSC-3)* (under consideration)

T10/BSR INCITS 465 *SCSI/ATA Translation-2 (SAT-2)* (planned as ISO/IEC 14776-922)

ANSI INCITS 452-2009, *ATA/ATAPI Command Set (ATA8-ACS)* (planned as ISO/IEC 17760-101)

IEC 60169-15:1979, *Radio-frequency connectors. Part 15: R.F. coaxial connectors with inner diameter of outer conductor 4.13 mm (0.163 in) with screw coupling – Characteristic impedance 50 Ωs (Type SMA)*

ISO 80000-2, *Quantities and units – Part 2: Mathematical signs and symbols to be used in the natural sciences and technology*

T10/BSR INCITS 514 *SCSI Block Commands-3 (SBC-3)* (planned as ISO/IEC 14776-323)

Memory Stick™ (MS). One Stop Site for Formats

NOTE 2 - Available from https://www.oss-formats.org.

MultiMediaCard (*e*•MMC), JEDEC®

NOTE 3 - Available from http://www.jedec.org.

NOTE 4 - JEDEC® is a registered trademark of JEDEC Solid State Technology Association. This information is given for the convenience of users of this standard and does not constitute an endorsement by ISO or IEC.

Editor's Note 5: This is an example of what a bibliography should look like if one is placed in a draft standard. A standard that is not required but is listed in informative text (e.g., in a note or informative annex) is required to be listed in a bibliography at the end of the draft standard.