Working Draft

X3T10.1/1121D

Revision 7b
July 22, 1996

Information Technology - Serial Storage Architecture - SCSI-2 Protocol (SSA-S2P)

Draft proposed American National Standard

This is a draft proposed American National Standard of Accredited Standards Committee X3. As such this is not a completed standard. The X3T10 Technical Committee may modify this document as a result of comments received during public review and its approval as a standard.

ABSTRACT

This standard describes the SCSI-2 protocol (SSA-S2P) to be used on the Serial Storage Architecture - Transport Layer 1 (SSA-TL1).

Project Leader
Charles Monia
Digital Equipment Corp.
SHR3-2/C5
334 South Street
Shrewsbury, MA 01545
Voice Voice: (508) 841-6757
FAX Fax: (508) 841-6100
EMail Email: monia@shr.dec.com

Technical Editor
John P. Scheible
IBM Corporation.
Bldg 902 Mailstop 9263
11400 Burnet Road
Austin, TX 78758
Voice (512) 823-8208
FAX (512) 838-3822
EMail scheible@vnet.ibm.com

Reference number
ANSI X3.294:1996
Printed 07/22/96

Other Points of Contact:

	X3T10.1 Chair	John P. Scheible
	Lawrence Lamers	IBM Corporation.
	Adaptec	Bldg 902 Mailstop 9263
	MS 293	11400 Burnet Road
	691 South Milpitas Blvd	Austin, TX 78758
	Milpitas CA, 95035	
Voice:	408-957-7817	(512) 823-8208
Fax:	408-957-7193	(512) 838-3822
Email:	ljlamers@aol.com	scheible@vnet.ibm.com

X3 Secretariat
Lynn Barra
Administrator Standards Processing
X3 Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005

Voice: 202-626-5738
FAX: 202-638-4922
Email: lbarra@itic.nw.dc.us

SSA Reflector

Internet address for subscription to the X3T10.1 reflector:	majordomo@symbios.com
Note should contain a line stating...	subscribe X3T10-SSA your email address
Internet address for distribution via X3T10.1 reflector:	x3t10-ssa@symbios.com
X3T10 Bulletin Board	719-533-7950
FTP Site:	ftp.symbios.com /pub/standards/io/x3t10.1
Web sites:	http://www.x3.org/x3t10 or http://www.symbios.com/x3t10 http://www.ssaia.org

Document Distribution
Global Engineering
15 Inverness Way East
Englewood, CO 80112-5704

Voice: 303-792-2181
or: 800-854-7179
FAX: 303-792-2192

PATENT STATEMENT

CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard.

As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, notice of one or more claims has been received.

By publication of this standard, no position is taken with respect to the validity of this claim or of any rights in connection therewith. The known patent holder has, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

ANSI®
X3.294:1996

American National Standard
for Information Systems -

Serial Storage Architecture - SCSI-2 Protocol (SSA-S2P)

Secretariat
Information Technology Industry Council

Approved Month dd, yy

American National Standards Institute, Inc.

Abstract

This standard describes the SCSI-2 mapping protocol (SSA-S2P) to be used on the Serial Storage Architecture - Transport Layer 1 (SSA-TL1).

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 a concerted 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.

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

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

Contents Page

Annexes Page

Tables

Figure

Foreword (This foreword is not part of American National Standard X3.294:1996.)

This standard was developed by Task Group X3T10.1 of Accredited Standards Committee X3 during 1993-96. The standards approval process started in 1995. This standard includes five annexes. Annex A is normative. Annexes B to E are informative and are not considered part of this standard.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the X3 Secretariat, Information Technology Industry Council, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, the X3 Committee had the following members:

James D. Converse, Chair

Donald C. Loughry, Vice-Chair

Joanne M. Flanagan, Secretary

Organization Represented	Name of Representative
American Nuclear Society	Geraldine C. Main
	Sally Hartzell (Alt.)
AMP, Inc	Edward Kelly
	Charles Brill (Alt.)
Apple Computer	Karen Higginbottom
Association of the Institute for Certification of Professionals (AICCP)	Kenneth Zemrowski
AT&T/NCR	Thomas W. Kern
	Thomas F. Frost (Alt.)
Boeing Company	Catherine Howells
	Andrea Vanosdoll (Alt.)
Bull HN Information Systems, Inc.	William George
Compaq Computer Corporation	James Barnes
Digital Equipment Corporation	Delbert Shoemaker
	Kevin Lewis (Alt.)
Eastman Kodak	James D. Converse
	Michael Nier (Alt.)
GUIDE International	Frank Kirshenbaum
	Harold Kuneke (Alt.)
Hewlett-Packard	Donald C. Loughry
Hitachi America, Ltd.	John Neumann
	Kei Yamashita (Alt.)
Hughes Aircraft Company	Harold L. Zebrack
IBM Corporation	Joel Urman
	Mary Anne Lawler (Alt.)
National Communication Systems	Dennis Bodson
National Institute of Standards and Technology	Robert E. Roundtree
	Michael Hogan (Alt.)
Northern Telecom, Inc	Mel Woinsky
	Subhash Patel (Alt.)
Neville & Associates	Carlton Neville
Recognition Technology Users Association	Herbert P. Schantz
	G. Edwin Hale (Alt.)
SHARE, Inc	Gary Ainsworth
	David Thewlis (Alt.)
Sony Corporation	Michael Deese
Storage Technology Corporation	Joseph S. Zajaczkowski
	Samuel D. Cheatham (Alt.)
Sun Microsystems	Scott Jameson
	Gary Robinson (Alt.)

3M Company	Eddie T. Morioka
	Paul D. Jahnke (Alt.)
Unisys Corporation	John L. Hill
	Stephen P. Oksala (Alt.)
U.S. Department of Defense	William C. Rinehuls
	C. J. Pasquariello (Alt.)
U.S. Department of Energy	Alton Cox
	Lawrence A. Wasson (Alt.)
U.S. General Services Administration	Douglas Arai
	Larry L. Jackson (Alt.)
Wintergreen Information Services	Joun Wheeler
Xerox Corporation	Dwight McBain
	Roy Peirce (Alt.)

Technical Committee X3T10 on I/O Interfaces, which reviewed this standard, had the following members:

John B. Lohmeyer, Chair

Lawrence J. Lamers, Vice-Chair

Ralph Weber, Secretary

Mr. I. Dal Allan
Mr. Tak Asami
Mr. Geoffrey Barton
Mr. Paul Boulay
Mr. John Cannon
Mr. Joe Chen
Ms. Nancy Cheng
Mr. Roger Cummings
Mr. William Dallas
Mr. Jan V. Dedek
Mr. Mark Evans
Mr. Edward Fong
Mr. Raymond Gilson
Mr. Dave Guss
Mr. Kenneth J. Hallam
Mr. Norm Harris
Mr. Mike Hetzel
Mr. Peter Johansson
Mr. Skip Jones
Mr. Thomas J. Kulesza
Mr. Edward Lappin
Mr. Robert Liu
Mr. John Masiewicz
Mr. Pete McLean
Dr. Akira James Miura
Mr. Ian Morrell
Mr. Jay Neer
Mr. Vit Novak
Mr. Duncan Penman
Mr. Kevin R. Pokorney
Mr. Ron Roberts
Mr. Steve D. Schueler
Mr. Robert N. Snively
Mr. Arlan P. Stone
Mr. Matthew Thomas
Mr. Dennis Van Dalsen
Mr. Roger Wang
Mr. Bob Whiteman
Mr. Devon Worrell
Mr. Ruben Yomtoubian
Mr. Paul D. Aloisi
Mr. Akram Atallah
Mr. Robert Bellino
Mr. Charles Brill
Mr. Ting Li Chan
Mr. Andy Chen
Mr. Mike Chennery
Mr. Chris D'Iorio
Mr. Joe Dambach
Mr. Dhiru N. Desai
Mr. Timothy Feldman
Mr. Edward A. Gardner
Mr. Chuck Grant
Mr. Peter Haas
Dr. William Ham
Mr. Edward Haske
Mr. Stephen Holmstead
Mr. Gerry Johnsen
Mr. Richard Kalish
Mr. Lawrence J. Lamers
Mr. Pat LaVarre
Mr. John Lohmeyer
Mr. David McFadden
Mr. Patrick Mercer
Mr. E.J. Mondor
Mr. John Moy
Mr. Marc A. Noblitt
Mr. Erich Oetting
Mr. George Penokie
Mr. Gary Porter
Mr. Frank Samela
Mr. J. R. Sims
Mr. Jeff Stai
Mr. Clifford E. Strang Jr.
Mr. Pete Tobias
Mr. Joseph Wach
Mr. Gary M. Watson
Mr. Jeffrey L. Williams
Mr. Anthony Yang
Mr. David Andreatta
Mr. Wayne Baldwin
Mr. Rick Bohn
Mr. Michael Bryan
Mr. Shufan Chan
Mr. Jack Chen
Mr. Dan Colegrove
Mr. Zane Daggett
Mr. Brian N. Davis
Mr. Mike Eneboe
Mr. Stephen G. Finch
Mr. John Geldman
Mr. Louis Grantham
Mr. Douglas Hagerman
Mr. Tom Hanan
Mr. Stephen F. Heil
Mr. Gerald Houlder
Mr. Brian Johnson
Mr. Greg Kapraun
Mr. Dennis Lang
Mr. Jaff Lin
Mr. Bill Mable
Mr. James McGrath
Mr. Gene Milligan
Mr. Charles Monia
Mr. S. Nadershahi
Mr. Tim Norman
Mr. Dennis Pak
Mr. Doug Piper
Mr. Robert Reisch
Mr. John P. Scheible
Mr. Michael Smith
Mr. Gary R. Stephens
Mr. Nicos Syrimis
Mr. Tokuyuki Totani
Mr. Dean Wallace
Mr. Ralph O. Weber
Mr. Michael Wingard
Mr. Danny Yeung

Task Group X3T10.1 on Serial Storage Architecture, which developed this standard, had the following members:

Lawrence J. Lamers, Chair

John Scheible, Vice-Chair

Bob Atkinson	James M. Avery	Robert Bellino
Paul Boulay	Charles Brill	Phil Bryan
Ed Carmona	Nancy Cheng	David Deming
Dhiru N. Desai	Mark DeWilde	Neil Edmunds
Richard Egan	Ken Erickson	Greg Farrin
Chuck Gibson	Chuck Grant	Ken Hallam
Bill Ham	Adge Hawes	Greg Kapraun
Sam Karunanithi	Brad Kitson	Michael Knowles
Lawrence Lamers	John Lohmeyer	Bill Mable
Gary Manchester	Charles Monia	Rich Moore
Ian Morrell	Karl Nakamura	Jay Neer
Thai Nguyen	Eric Oetting	Paul Petersen
Kevin Pokorney	Said Rahmani	Monica Roy
Sam Sanyal	John Scheible	Pete Tobias
Douglas Wagner	Michael Wingard	Kevin Ziegler

Introduction

This standard is divided into the following clauses and annexes.

Clause 1 defines the scope of the Serial Storage Architecture - SCSI-2 Protocol (SSA-S2P).

Clause 2 specifies the normative references.

Clause 3 defines the definitions, symbols and abbreviations.

Clause 4 contains the general SSA-S2P information.

Clause 5 contains the frame information.

Clause 6 contains the SSA-S2P SMS (SSA Message Structure).

Clause 7 contains transport related items.

Clause 8 contains SSA changes from parallel SCSI-2.

Annex A is normative and contains the SCSI status code values with the SCA ACTIVE value.

Annex B is informative and contains information on conversion from parallel SCSI-2 to SSA-S2P.

Annex C is informative and contains information on differences between parallel SCSI-2 to SSA-S2P.

Annex D is informative and contains the protocol services model.

Annex E is informative and contains the transport services model.

Draft American National Standard for Information Systems -

Draft Information Technology - Serial Storage Architecture - SCSI-2 Protocol (SSA-S2P)

1 Scope and family of standards

This standard describes an upper-level protocol of Serial Storage Architecture. SSA-S2P is a mapping of the existing SCSI-2 protocol, X3.131-1994, with extensions to map SCSI-2 to the SSA serial link.

Figure 1 shows the relationships of the SSA standards

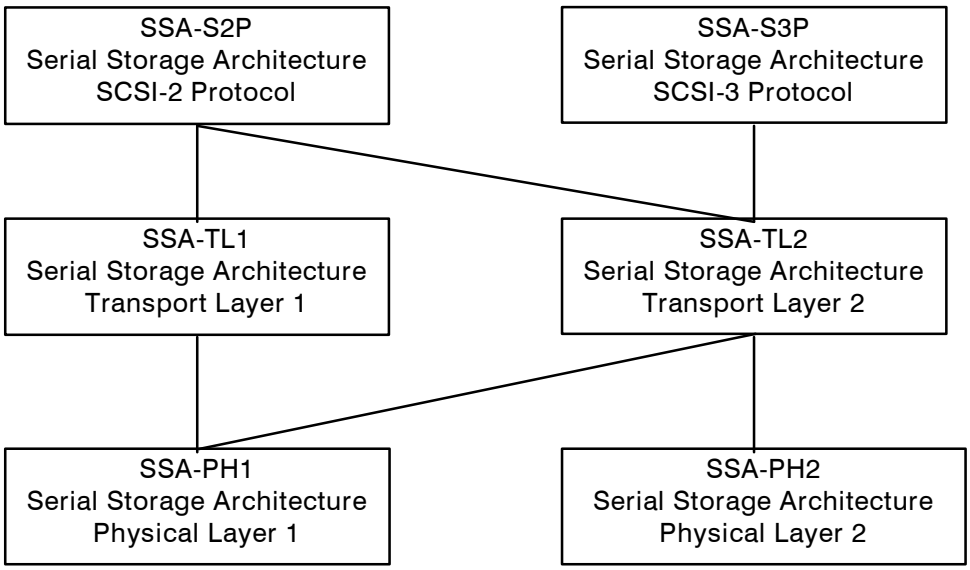

Figure 1 - Relationship of the SSA standards

1.1 SSA-S2P

Serial Storage Architecture - SCSI-2 Protocol (SSA-S2P) (Project X3T10.1/1121) defines the SCSI-2 Protocol used to runs with the SSA transport layers 1 or 2. SSA-S2P is intended to operate on the following transport and physical layers: SSA-TL1 with SSA-PH1, SSA-TL2 with SSA-PH1, and SSA-TL2 with SSA-PH2.

1.2 SSA-TL1

Serial Storage Architecture - (SSA-TL1) Transport Layer 1 (Project X3T10.1/0989) defines the Transport layer that supports SSA-S2P and requires SSA-PH1.

1.3 SSA-PH1

Serial Storage Architecture - (SSA-PH1) Physical Layer 1 (Project X3T10.1/1145D) defines the Physical layer that supports SSA-TL1 and SSA-TL2, and consists of the electrical characteristics of the interface and the connectors.

1.4 SSA-S3P

Serial Storage Architecture - (SSA-S3P) SCSI-3 Protocol (Project X3T10.1/1051) defines the SCSI-3 Protocol used with the SSA transport layer 2. SSA-S3P is intended to operate on the following transport and physical layers: SSA-TL2 with SSA-PH1, and SSA-TL2 with SSA-PH2.

1.5 SSA-TL2

Serial Storage Architecture - (SSA-TL1) Transport Layer 2 (Project X3T10.1/1147) defines the Transport layer that supports SSA-S2P and SSA-S3P and requires SSA-PH1 or SSA-PH2

1.6 SSA-PH2

Serial Storage Architecture - (SSA-PH2) Physical Layer 2 (Project X3T10.1/1146) defines the Physical layer that supports SSA-TL2, and consists of the electrical characteristics of the interface and the connectors.

2 Normative references

The following standard contains provisions which, through reference in SSA-S2P, constitute provisions of this standard. At the time of publication, the edition indicated was 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 standard listed below. Members of IEC and ISO maintain registers of currently valid International Standards, and ANSI maintains registers for American National Standards.

ANSI X3.131:1994, Small Computer Systems Interface - 2 (SCSI-2)

ANSI X3.295:1996, Information Technology, Serial Storage Architecture - Transport Layer 1 (SSA-TL1)

ANSI X3.293:1996, Information Technology, Serial Storage Architecture - Physical Layer 1 (SSA-PH1)

SSA-TL refers to SSA-TL1.

All references made in this standard to a Command Descriptor Block (CDB) refer to those CDB's and CDB formats defined in the SCSI-2 standard (ANSI X3.131-1994).

3 Definitions, symbols and abbreviations

3.1 Definitions

3.1.1 Channel: The facilities in a port to receive an SMS or a single data transfer.

3.1.2 character: A sequence of 10 encoded bits that represents a data byte or a protocol character.

3.1.3 Contingent Allegiance: An SCSI-2 concept of an error condition that is cleared following the execution of the next command. It does not work well in serial interfaces, and has been replaced by the concept of Serial Contingent Allegiance.

3.1.4 Data frame: An Application frame with a non-zero channel component.

3.1.5 destination node: The node that receives a particular frame.

3.1.6 field: A group of related data characters in a frame, (e. g. the CRC field).

3.1.7 frame: A sequence of 6 or more data characters surrounded by FLAG characters.

3.1.8 I_T nexus: The specific initiator and target participating in an SCSI command.

3.1.9 I_T_L nexus: The specific initiator, target and logical unit for an SCSI command.

3.1.10 link: A serial connection between two ports.

3.1.11 logical path: A full duplex conduit for the ordered delivery of SMS and data frames between an initiator-target pair. (This includes the physical path from the initiator to the target as well as the return path from the target to the initiator. The return path need not retrace the initiator to target physical path.)

3.1.12 nexus: A relationship identifying the addressable SCSI entities associated with an SSA SMS transfer.

3.1.13 node: A system, controller or device with one or more ports.

3.1.14 port: The hardware and firmware that implements one end of a link.

3.1.15 Serial Contingent Allegiance: An error condition that holds all queued I/O processes until cleared explicitly by a reset or a CLEAR SCA CONDITION SMS and allows error recovery while the queued I/O processes are held. This concept replaces the SCSI-2 concept of Contingent Allegiance.

3.1.16 SMS: The data field portion of a frame with a Channel component of 00h. The FRAME TYPE field information shall be considered part of the SMS.

3.1.17 source: The node that originates a particular frame.

3.1.18 Web: A collection of SSA nodes that may address each other, interconnected into a dedicated connection, a loop, a string, or a complex configuration (i. e., includes a switch).

3.2 Symbols and abbreviations

CDB	Command Descriptor Block
CRC	cyclic redundancy check.
DDRM	Disable DATA READY SMSs
DMA	direct memory access
ERP	error recovery procedure
LUN	Logical Unit Number
LUNTAR	LUN or Target Routine flag
LUNTRN	LUN or Target Routine value
N/A	Not Applicable
OOT	Out of Order Transfer
POR	power-on reset
POST	power-on self-test
Qerr	A bit in the SCSI mode select pages that controls the clearing of the Queue following a SCA.
RAS	reliability, availability and serviceability
RPL	Rotational Position Locking
SCA	Serial Contingent Allegiance
SCSI	Small Computer Systems Interface
SSA	Serial Storage Architecture
SYNC	A signal (8B/10B character) that allows separate spindle motors to be synchronized.
SMS	SSA Message Structure
&	Logical AND
=	Assignment or comparison for EQUAL
≠	Comparison for NOT EQUAL
<	Comparison for LESS THAN
≤	Comparison for LESS THAN OR EQUAL TO
>	Comparison for GREATER THAN
+	ADD
-	SUBTRACT
*	MULTIPLY
±	PLUS OR MINUS
≈	APPROXIMATELY
»	MUCH GREATER THAN

4 General

4.1 Overview

It is intended that SSA-S2P should conform as closely as possible to the existing SCSI-2 logical model. This minimizes the programming changes required to convert existing systems and devices from the parallel bus to an SSA interface. Therefore the following functions of the SSA-S2P are identical to SCSI:

a) Tagged queuing,
b) Command descriptor blocks,
c) Status byte,
d) Sense bytes.

Except where necessary for clarity the above functions are not described in this standard (Please refer to the ANSI SCSI-2 standard for information). SSA-S2P concentrates on mapping the following aspects of parallel SCSI-2:

a) Bus functions,
b) Addressing,
c) Messages.

SSA-S2P supports Webs containing strings, loops and switches. The concepts of initiator, target and Logical Unit are retained although SSA-S2P supports larger configurations than parallel SCSI-2. Initiators and targets may be freely mixed throughout the Web. Each node may have from 1 to 126 physical ports.

SSA-S2P offers the following benefits compared to the parallel SCSI bus:

a) Open-ended Webs with alternative paths for availability and performance;
b) Full-duplex communication with spatial reuse on strings and loops;
c) Frame multiplexing;
d) No overhead for arbitration, disconnection or reselection;
e) Integrated spindle synchronization for array applications;
f) Fewer initiator-target exchanges;
g) Concurrent I/O processes on the same device or different devices;
h) Out-of-order data transfers.

Restrictions:

a) Untagged queuing may be simply emulated with Tagged Queuing. In SSA, the target effectively disconnects after each frame because of frame multiplexing. Hence each command shall have a tag for identification. Untagged command queuing may be simply emulated by having the initiator have only one outstanding command, re-use the same tag, and use the simple queue type;
b) SSA-S2P uses Serial Contingent Allegiance. The SCSI-2 concept of Contingent Allegiance does not work for SSA since the next command clears the Contingent Allegiance condition, and the next command may be in the SSA Web prior to the status being received by the initiator. Therefore SSA-S2P adopts the concept of Serial Contingent Allegiance where the Serial Contingent Allegiance condition shall be explicitly cleared by a CLEAR SCA CONDITION SMS.

4.2 Conventions

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined in the text with the first letter capitalized. Lower case is used for words having the normal English meaning.

Fields containing only one bit are usually referred to as the “named” bit instead of the “named” field. When a bit is set, its value is 1. When a bit is cleared, its value is 0.

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.

Decimal fractions 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).

Table 1 illustrates the bit ordering used within a byte in SSA-S2P.

Table 1 - Bit ordering in a byte

Bit 7	6	5	4	3	2	1	Bit 0
msb							lsb

Reserved bits, fields, bytes, and code values are set aside for future standardization. Their use and interpretation may be specified by future extensions to this standard. A reserved bit, field, or byte shall be set to zero, or in accordance with a future extension of this standard. Reserved bits, fields, bytes, or reserved field values shall be ignored when cut-through routing a frame. A destination node that receives a reserved bit, field, or byte that is not zero, or receives a reserved code value shall terminate the SMS as described in 6.2.

Ignored bits, fields, or bytes shall be ignored by the receiving node. Any value is considered valid.

SMS names are shown as all capital letters, such as SCSI STATUS SMS. Field names are shown as small capital letters, such as the STATUS field. Field values are shown as all capital letters, such as the STATUS field QUEUE FULL value. Variables are shown in italics.

The byte ordering convention is Big Endian (i. e., the most significant byte of a number is sent first).

5 Frames

All SSA-S2P SMSs and data are transmitted in Application frames whose format is described in SSA-TL. A Data frame is an Application frame with a non-zero Channel component.

To improve system bandwidth, individual frames are delivered without end-to-end acknowledgment (this mode of delivery is referred to as datagram service). Such acknowledgment, when required, is more efficient if performed by the higher level protocol on protocol-specific transactions, such as an S2P SCSI command or task management function.

6 SSA-S2P message structure

SMSs are used to communicate control information between the target and the initiator. They are typically used for commands, status and controlling I/O processes. Some SSA-S2P SMSs are similar in function to SCSI messages but there is not a one-to-one mapping and they should not be confused.

The ADDRESS field in an SMS contains the Path component (address to the destination node) followed by the byte 00h to select the SMS Channel component. The length of the DATA field depends on the particular SMS. Each SMS shall be fully contained in a single frame.

SSA-S2P SMS frames are identified by a Channel component value of 00h, and the first byte (byte 0) of the DATA field containing SSA-S2P SMS CODE value of 82h. The second byte (byte 1) of the DATA field of SSA-S2P SMSs contains the S2P CODE field identifying the function of SSA-S2P SMS.

The SCSI COMMAND SMS optionally contains a 2-byte CHANNEL field. This contains the Channel component for the ADDRESS field of data frames. If the Channel component is a single byte, the CHANNEL field contains the Channel component left aligned, with the second byte padded with any value by the source node and ignored by the destination node.

All SMSs contain a 2-byte TAG field that is used to relate replies to the original request. The TAG field value is assigned by the initiator and it shall be unique among all the TAG field values that are currently active from that initiator on any target or Logical Unit throughout the Web. Effectively the TAG field identifies the target, LUN and Queue-tag components of the SCSI-2 nexus. (The initiator and logical path are identified by the RETURN PATH ID field in the SMS.)

A target always replies to a given SMS using the same logical path from which it received the SMS.

Any SMS received with a valid SSA-S2P SMS code whose DATA field meets any of the error conditions in 6.2 shall cause the specified error response. An SMS has a maximum length of 32 bytes. Only the bytes defined in the appropriate SMS are required , however the SMS may be padded up to a total length of 32 bytes. The source node shall pad (if any) with zeros, and the destination node shall ignore any pad bytes.

Table 2 defines the generic structure of an SSA-S2P SMS generated by the initiator.

Table 2 - Generic SMS format for initiator generated SSA-S2P SMSs

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE							
2	TAG							
3	TAG							
4	RETURN PATH ID							
5	RETURN PATH ID							
6	RETURN PATH ID							
7	RETURN PATH ID							
8	LUNTAR	LUNTRN						
9	(Depends on the SMS)							
up to 31	(Depends on the SMS)							

All SMSs from the initiator to the target contain a RETURN PATH ID field, a value supplied by the target during configuration to identify the logical path to the appropriate initiator (see SSA-TL).

The LUNTAR bit and LUNTRN field need not be present, but if present shall reside in byte 8.

Table 3 defines the generic structure of an SSA-S2P SMS generated by the initiator.

Table 3 - Generic SMS format for target generated SSA-S2P SMSs

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE							
2	TAG							
3	TAG							
4	(Depends on the SMS)							
up to 31	(Depends on the SMS)							

The following sub clauses define the SMSs that are used by SSA-S2P.

6.1 Summary of SSA-S2P SMSs

Table 4 shows a summary of all SSA-S2P SMSs.

Table 4 - Summary of SSA-S2P SMSs

SMS name	Byte 0 SMS CODE	Byte 1 S2P CODE	SMS FRAME TYPE	Node type support	
				Sent by	Received by
SCSI RESPONSE	82h	03h	APPLICATION	target	initiator
SCSI COMMAND	82h	10h	APPLICATION	initiator	target
SCSI STATUS	82h	11h	APPLICATION	target	initiator
ABORT TAG	82h	30h	APPLICATION	initiator	target
ABORT	82h	31h	APPLICATION	initiator	target
CLEAR QUEUE	82h	32h	APPLICATION	initiator	target
DEVICE RESET	82h	33h	APPLICATION	initiator	target
CLEAR SCA CONDITION	82h	34h	APPLICATION	initiator	target

6.2 SMS validation

Prior to the SSA-S2P protocol layer receiving an SMS, the transport layer has validated the SMS CODE field to be 82h. Each SMS then undergoes three levels of validation as described in the following three clauses.

6.2.1 SMS code validation

After the transport layer checks the SMS for validity, the SSA-S2P layer shall perform validity checks on byte 1, the S2P CODE, in the following order. If the Asynchronous Alert process is invoked then the SMS is terminated without any other response:

a) If the S2P CODE is not a value supported in Table 4, then invoke the Asynchronous Alert process with an UNKNOWN SMS ALERT CODE field (see SSA-TL for a description of ASYNC ALERT SMS format).

b) If the S2P CODE is supported in Table 4, but the node does not match the corresponding type (initiator or target), then invoke the Asynchronous Alert process with an UNKNOWN SMS ALERT CODE field.

c) If Table 4 matches the S2P CODE of the SMS, but the FRAME TYPE field does not have a value of APPLICATION then invoke the Asynchronous Alert process with an SMS UNKNOWN ALERT CODE field.

d) If the S2P CODE corresponds to the ABORT, ABORT TAG, CLEAR QUEUE, DEVICE RESET, or CLEAR SCA CONDITION SMS, and one of these SMSs is already outstanding in the destination node from this Initiator, then generate an SCSI RESPONSE SMS with an OVERLAPPED SMSs ATTEMPTED RETURN CODE value.

e) If the SMS is an SCSI COMMAND SMS, then destination node shall perform the following functions:

1) If the destination node has no room to store the SMS and the destination node has not discarded the previous SCSI COMMAND SMS from this Initiator, then the destination node shall discard the SMS and generate an SCSI STATUS SMS indicating QUEUE FULL status.

2) If the destination node has discarded the previous SCSI COMMAND SMS from this Initiator and the SMS has the RESUME bit cleared, the destination node shall discard the SCSI COMMAND SMS even if room to store the SMS has become available. The destination node shall not generate an SCSI STATUS or SCSI RESPONSE SMS to the discarded SCSI COMMAND SMS.

3) If the destination node has no room to store the SMS, the destination node has discarded the previous SCSI COMMAND SMS from this Initiator, and the SMS has the RESUME bit set, the destination node shall discard the SMS and generate an SCSI STATUS SMS indicating a STATUS value of QUEUE FULL.

4) If the destination node has not discarded the previous SCSI COMMAND SMS from this Initiator and the SCSI COMMAND SMS has the RESUME bit set, the destination node shall discard the SMS and generate an SCSI RESPONSE SMS with a RETURN CODE value of INVALID FIELD.

5) If the destination node has room to store the SMS, the destination node has discarded the previous SCSI COMMAND SMS from this Initiator, and the SMS has the RESUME bit set, the destination node shall continue SMS validation processing as defined in 6.2.2 and 6.2.3.

6.2.2 SMS length validation

If the SMS length is shorter than that specified, then invoke the Asynchronous Alert process with an SMS TOO SHORT ALERT CODE field. If the Asynchronous Alert process is invoked the original SMS is terminated, without generating an SCSI STATUS or SCSI RESPONSE SMS.

6.2.3 SMS field validation

Each SMS listed in Table 4 shall perform the following SMS field validations in the order shown. If the Asynchronous Alert process is invoked then the SMS is then terminated, without any other response:

1) If the RETURN PATH ID field is unknown, then invoke the Asynchronous Alert process with an UNKNOWN RETURN PATH OR RETURN PATH ID ALERT CODE field value.
2) If the initiator receives an SCSI RESPONSE or SCSI STATUS SMS with an invalid TAG field, then generate an ASYNC ALERT with an SMS UNEXPECTED ALERT CODE field value.
3) If any reserved field is non-zero or valid field contains a reserved code value, then generate an SCSI RESPONSE SMS with a RETURN CODE value of INVALID FIELD. There is one exception to this case, where any improper field values within the CDB, LUNTAR, or LUNTRN fields of the SCSI COMMAND SMS shall reply with an SCSI STATUS SMS with STATUS field value of CHECK CONDITION STATUS and sense data as defined in parallel SCSI-2.

6.3 SMS buffer full condition

A destination node shall enter an SMS Buffer Full condition when an SCSI COMMAND SMS is received by the destination node and no room is available to store the SMS. Upon entering an SMS Buffer Full condition, the destination node shall discard the SCSI COMMAND SMS and generate an SCSI STATUS SMS indicating a QUEUE FULL status. The destination node shall then discard subsequent received SCSI COMMAND SMS with a RESUME bit cleared from the same Initiator even if room is available to store the SMS.

The destination node shall exit the SMS Buffer Full condition after receiving an SCSI COMMAND SMS with a RESUME bit set from the Initiator with the SMS Buffer Full condition. If the destination node still has no room to store the SMS, the destination node shall once again enter the SMS Buffer Full condition.

When an Initiator receives an SCSI STATUS SMS indicating QUEUE FULL status, the Initiator shall mark the command identified by the SMS and any subsequent commands issued to the destination node as having been discarded. The Initiator shall send the next SCSI COMMAND SMS with a RESUME bit set.

6.4 Limitation on task management SMSs

An Initiator shall only have one ABORT, ABORT TAG, CLEAR QUEUE, DEVICE RESET, or CLEAR SCA CONDITION SMS outstanding to a given target. A target shall provide storage to process one of these SMSs for each Initiator and therefore, need never discard one, provided the initiator follows the SSA-S2P requirements.

6.5 TAG field values

The TAG field values of all outstanding SSA-S2P SMSs from a given initiator shall be unique. The initiator uses the TAG field value of the SCSI STATUS SMS or SCSI RESPONSE SMS to determine both the source (target, LUN, target routine) and the SMS associated with the SCSI STATUS SMS or SCSI RESPONSE SMS. TAG field values outstanding at a target need not be unique between initiators, since the target identifies processes with both a TAG and RETURN PATH ID field.

6.6 SCSI COMMAND SMS

The SCSI COMMAND SMS is sent from the initiator to a target to initiate an I/O process or to send the next command in a series of linked commands. If the SCSI COMMAND SMS may be successfully parsed, the target may respond by initiating a data transfer and sending an SCSI STATUS SMS. See 6.2 for invalid SMS processing that may result in either an ASYNC ALERT SMS with various ALERT CODE fields or an SCSI RESPONSE SMS with a RETURN CODE of INVALID FIELD.

The contents of the DATA field in an SCSI COMMAND SMS are defined in Table 5.

Table 5 - SCSI COMMAND SMS

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE (10h)							
2	TAG							
3	TAG							
4	RETURN PATH ID							
5	RETURN PATH ID							
6	RETURN PATH ID							
7	RETURN PATH ID							
8	LUNTAR	LUNTRN						
9	reserved = 00h							
10	DDRM	OOT	RESUME	reserved = 000b			QUEUE CNTL	
11	reserved=00h							
12	CHANNEL							
13	CHANNEL							
14	reserved = 00h							
15	reserved = 00h							
16	COMMAND DESCRIPTOR BLOCK							
...	COMMAND DESCRIPTOR BLOCK							
m	COMMAND DESCRIPTOR BLOCK							

The TAG field is assigned by the initiator and used to relate subsequent SMSs to this I/O process. The value used in the TAG field becomes available for re-assignment when the I/O process completes or is terminated. All SCSI COMMAND SMSs in a linked list shall specify the same value for the TAG field.

NOTE 1 - Parallel SCSI-2 requires all TAG field values to be unique for a given I_T_L nexus, in addition SSA requires that the tag field values be unique for an initiator, regardless of the I_T_L nexus. SSA uses the TAG field value to identify the target/LUN and the SCSI COMMAND SMS the SCSI STATUS SMS is associated with. Therefore, an easy conversion from a parallel SCSI-2 device driver to SSA-S2P is to include the parallel SCSI concepts of TAG field (1 byte), target address (4 bits), and LUNTAR/LUN (1+3 bits) into the 2 byte SSA-S2P TAG field.

The RETURN PATH ID field is used by the target to locate the initiator table entry containing the return path, port and Unique ID of the initiator that issued the SCSI COMMAND SMS.. Each SCSI COMMAND SMS in a series of linked commands for a given I/O process shall use the same value for the RETURN PATH ID field.

If the LUNTAR bit is set, then the LUNTRN field refers to a Target Routine. If the LUNTAR bit is cleared, then the LUNTRN field refers to a Logical Unit.

The LUNTRN field specifies either a Logical Unit or a Target Routine, as indicated by the value of the LUNTAR bit. All SCSI COMMAND SMSs in a linked list shall specify the same values for the LUNTAR and LUNTRN fields.

If the DDRM (Disable DATA READY SMS) bit is set, DATA READY SMSs (see SSA-TL) are disabled for data transfers from the target to the initiator. If the DDRM bit is set, the initiator shall specify the channel to be used in the CHANNEL field. If the DDRM bit is cleared, then the target shall issue a DATA READY or DATA REQUEST SMS to initiate data transfer (see SSA-TL). The DDRM bit is ignored for any commands that do not transfer data to the initiator.

If the OOT (Out-of-Order Transfer) bit is set the target or Logical Unit is permitted to transfer read data to or write data from the initiator out-of-order. This enables a device to get the maximum advantage from an out-of-order read or out-of-order write. It is also useful for array controllers, e. g. RAID-5. If the OOT bit is cleared then data shall be transferred sequentially from the beginning. For control commands the OOT bit is ignored and data is always transferred in order from the beginning. If both the OOT and DDRM bits are set then the initiator shall be able to identify each logical block from a header within the block.

The RESUME bit controls the processing of queue full conditions. The initiator normally clears the RESUME bit when processing SCSI COMMAND SMSs. After receiving an SCSI STATUS SMS from a destination node indicating QUEUE FULL status, the initiator shall set the RESUME bit for the next SCSI COMMAND SMS issued to the destination node in an attempt to resume SCSI COMMAND SMS processing. Receiving an SCSI

COMMAND SMS with the RESUME bit set informs the destination node to stop discarding SCSI COMMAND SMSs due to the previously detected SMS Buffer Full condition (see 6.3).

The QUEUE CNTL field controls how the I/O process is to be queued at the specified Logical Unit or Target Routine, as defined in Table 6. For a linked list of commands the QUEUE CNTL field is ignored in all SCSI COMMAND SMSs except the first.

Table 6 - QUEUE CNTL field values

Value	Description
00b	SERIAL CONTINGENT ALLEGIANCE
01b	HEAD OF QUEUE
10b	ORDERED QUEUE
11b	SIMPLE QUEUE

The target only executes I/O processes with the SERIAL CONTINGENT ALLEGIANCE QUEUE CNTL field value while a Serial Contingent Allegiance condition exists for the same nexus (see 8.2). For example, a Request Sense command specifying a QUEUE CNTL field value of SERIAL CONTINGENT ALLEGIANCE is used to retrieve sense information after a command terminates with STATUS field of CHECK CONDITION.

The target places an I/O process with the QUEUE CNTL field value of HEAD OF QUEUE first in the queue, to be executed next when the currently active I/O process finishes.

The target shall execute a sequence of I/O processes with the QUEUE CNTL field value of ORDERED QUEUE in the strict order that they were received.

The target is permitted to reorder a sequence of I/O processes with the QUEUE CNTL field value SIMPLE QUEUE. This allows a device to optimize performance (e. g. by using an elevator seeking algorithm). All previously issued ORDERED I/O processes shall be executed prior to executing any of the SIMPLE I/O processes. Any subsequently issued ORDERED I/O process shall be executed after all the SIMPLE I/O processes have completed.

The CHANNEL field specifies the Channel component of the ADDRESS field for data frames that the target sends to the initiator (the channel to be used within the initiator). CHANNEL is ignored for any commands that do not transfer data to the initiator, or when the DDRM bit is cleared.

The COMMAND DESCRIPTOR BLOCK field is the Command Descriptor Block (CDB) as defined by SCSI-2 except that the Logical Unit Number in bits 7:5 of byte 1 of the CDB are not used. For SCSI-2 commands, the CDB is 6, 10 or 12 bytes long. For vendor-specific commands, the CDB may be up to 16 bytes long since the maximum length of an SMS frame is 32 bytes. The last byte of the CDB is always a Control byte containing the FLAG and LINK bits. The FLAG bit is returned to the initiator in the SCSI STATUS SMS corresponding to the command. When set, the FLAG bit typically causes the initiator to interrupt the application to indicate that a linked command has completed. When set, the LINK bit indicates that the I/O process is to be continued with a further command. In this case the initiator sends a further SCSI COMMAND SMS in response to the SCSI STATUS SMS for the current command, provided the STATUS field has a value of LINKED COMMAND COMPLETE or LINKED COMMAND COMPLETE, WITH FLAG.

6.7 SCSI STATUS SMS

The SCSI STATUS SMS is sent from a target to an initiator to indicate that a command and any associated data transfer have been terminated or completed. The SCSI STATUS SMS is returned using the RETURN PATH ID field specified in the SCSI COMMAND SMS.

The SCSI STATUS SMS is returned for each SCSI COMMAND SMS unless the command is rejected (with an ASYNC ALERT SMS or an SCSI RESPONSE SMS) or the command is cleared by any of the following:

a) ABORT TAG SMS
b) ABORT SMS
c) CLEAR QUEUE SMS
d) DEVICE RESET SMS

e) A Hard Reset condition (including a transport layer Total Reset or Absolute Reset frame).

The contents of the DATA field in the SCSI STATUS SMS are defined in Table 7.

Table 7 - SCSI STATUS SMS

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE (11h)							
2	TAG							
3	TAG							
4	STATUS							
5	reserved=0000 0b						FLAG	LINK

The TAG field is a copy of the TAG field in the corresponding SCSI COMMAND SMS. It allows the initiator to associate the status with the correct I/O process.

The STATUS field contains status as defined by SCSI-2 with the addition of SCA ACTIVE status (30h) (see Annex A).

SSA-S2P uses a single S2P CODE value with imbedded FLAG and LINK bits, rather than create three different completion messages as parallel SCSI-2 did. The FLAG and LINK bits indicate different completion messages as shown in Table 8.

Table 8 - Meaning of flag and link bits

FLAG	LINK	Corresponding parallel SCSI-2 completion message
0	0	Command Complete
0	1	Linked Command Complete
1	0	reserved
1	1	Linked Command Complete with Flag

6.8 ABORT TAG SMS

The ABORT TAG SMS is sent from an initiator to a target to abort a particular I/O process. Other I/O processes are not affected. The ABORT TAG SMS shall be sent over the logical path used to create the I/O process (see 8.4).

Previously established conditions including MODE SELECT parameters, reservations and the Serial Contingent Allegiance condition shall not be changed by the ABORT TAG SMS.

Before issuing the ABORT TAG SMS the initiator's S2P layer should terminate any related outbound data transfer to ensure that data is not sent to a non-existent nexus. The logical unit terminates execution of the I/O process if it has already begun or removes the I/O process from the queue if execution has not begun. The logical unit shall not send a SCSI STATUS SMS for the aborted command after the SCSI RESPONSE SMS has been sent for the ABORT TAG SMS. Upon receiving the SCSI RESPONSE SMS (see D.3.1), the initiator's S2P layer removes the aborted command if present in its Outstanding Commands Table.

The contents of the DATA field in the ABORT TAG SMS are defined in Table 9.

Table 9 - ABORT TAG SMS

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE (30h)							
2	TAG							
3	TAG							
4	RETURN PATH ID							
5	RETURN PATH ID							
6	RETURN PATH ID							
7	RETURN PATH ID							
8	TAG 2							
9	TAG 2							

The TAG field is used to relate the SCSI RESPONSE SMS to the ABORT TAG SMS.

The RETURN PATH ID field identifies the logical path, that shall be used by the target to send the associated SCSI RESPONSE SMS.

The TAG 2 field contains the TAG of the I/O process to be aborted. It is not an error if TAG 2 is unknown to the target since the execution of the I/O process may have already completed.

In the SCSI RESPONSE SMS, the RETURN CODE field shall have a value of REQUESTED FUNCTION COMPLETED SUCCESSFULLY, or I/O PROCESS NOT FOUND. If the I/O process has already completed then the SCSI STATUS SMS may have been sent prior to the SCSI RESPONSE SMS.

6.9 ABORT SMS

The ABORT SMS is sent from an initiator to a target to abort all I/O processes from that initiator for a selected Logical Unit or Target Routine. I/O processes from other initiators are not affected. If the initiator is using multiple logical paths to the target, the initiator shall issue an ABORT SMS over each different logical path used by I/O processes for the I_T_L nexus (see 8.4).

Previously established conditions including MODE SELECT parameters, reservations and the Serial Contingent Allegiance condition shall not be changed by the ABORT SMS.

Before issuing the ABORT SMS the initiator should terminate any related outbound data transfers to ensure that data is not sent to a non-existent nexus. A SCSI STATUS SMS shall be sent for each I/O process that completed prior to sending the SCSI RESPONSE SMS. After sending the SCSI RESPONSE SMS, the target shall not send an SCSI STATUS SMS for any aborted command. Upon receiving the SCSI RESPONSE SMS, the initiator's S2P layer removes all aborted commands from its Outstanding Commands Table (see D.3.1).

The contents of the DATA field in the ABORT SMS are defined in Table 10.

Table 10 - ABORT SMS

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE (31h)							
2	TAG							
3	TAG							
4	RETURN PATH ID							
5	RETURN PATH ID							
6	RETURN PATH ID							
7	RETURN PATH ID							
8	LUNTAR	LUNTRN						

The TAG field is used by the initiator to associate the SCSI RESPONSE SMS to the ABORT SMS.

The RETURN PATH ID field specifies the logical path, that shall be used by the target to return the associated SCSI RESPONSE SMS.

If the LUNTAR bit is set, then the LUNTRN field refers to a Target Routine. If the LUNTAR bit is cleared, then the LUNTRN field refers to a Logical Unit.

The LUNTRN field specifies either a Logical Unit or a Target Routine, as indicated by the value of the LUNTAR bit. If an unsupported or unattached LUN is specified, an SCSI RESPONSE SMS is generated with a RETURN CODE of INVALID FIELD.

In the SCSI RESPONSE SMS, the RETURN CODE field shall have a value of REQUESTED FUNCTION COMPLETED SUCCESSFULLY or INVALID FIELD.

6.10 CLEAR QUEUE SMS

The CLEAR QUEUE SMS is sent from an initiator to a target to abort all I/O processes from all initiators for a specified Logical Unit or Target Routine. If the initiator is using multiple logical paths to the target, the initiator shall issue a CLEAR QUEUE SMS over each different logical path used by I/O processes for the I_T_L nexus (see 8.4).

Before issuing the CLEAR QUEUE SMS the initiator should terminate any related outbound data transfers to ensure that data is not sent to a non-existent nexus. If the target has issued any DATA REQUEST SMSs to other initiators on behalf of the selected Logical Unit or Target Routine then the target shall receive and flush all of the requested data before aborting the corresponding I/O processes.

All pending status and data for that Logical Unit or Target Routine for all initiators shall be cleared. A SCSI STATUS SMS shall be sent for each I/O process that completed prior to the SCSI RESPONSE SMS. Once the target has sent the SCSI RESPONSE SMS for the CLEAR QUEUE request, no status shall be sent for any aborted I/O process. The target sets Unit Attention for all initiators that had an I/O process was aborted. The sense code indicates COMMANDS CLEARED BY ANOTHER INITIATOR. Upon receiving the SCSI RESPONSE SMS, the initiator's S2P layer removes all aborted commands from its Outstanding Commands Table (see annex D.3.1).

The contents of the DATA field in the CLEAR QUEUE SMS is defined in Table 11.

Table 11 - CLEAR QUEUE SMS

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE (32h)							
2	TAG							
3	TAG							
4	RETURN PATH ID							
5	RETURN PATH ID							
6	RETURN PATH ID							
7	RETURN PATH ID							
8	LUNTAR	LUNTRN						

The TAG field is used to relate the SCSI RESPONSE SMS to the CLEAR QUEUE SMS.

The RETURN PATH ID field identifies the logical path that shall be used by the target to return the associated SCSI RESPONSE SMS.

If the LUNTAR bit is set, then the LUNTRN field refers to a Target Routine. If the LUNTAR bit is cleared, then the LUNTRN field refers to a Logical Unit.

The LUNTRN field specifies either a Logical Unit or a Target Routine, as indicated by the value of the LUNTAR bit. If an unsupported or unattached LUN is specified, an SCSI RESPONSE SMS is generated with a RETURN CODE of INVALID FIELD.

In the SCSI RESPONSE SMS, the RETURN CODE field shall have a value of REQUESTED FUNCTION COMPLETED SUCCESSFULLY or INVALID FIELD.

6.11 DEVICE RESET SMS

The DEVICE RESET SMS is sent from an initiator to a target to abort all I/O processes for all initiators on all Logical Units and all Target Routines. If the initiator is using multiple logical paths to the target, the initiator shall issue a DEVICE RESET SMS over each different logical path used by I/O processes for the I_T nexus (see 8.4). The target executes a hard reset that shall perform the following.

a) Abort all I/O Processes.

b) Clear all Serial Contingent Allegiance conditions.

c) Release all SCSI device reservations.

d) Returns any device operating modes to their appropriate initial conditions similar to those that may be found following a device power-on. The MODE SELECT conditions shall be restored to their last saved values if saved values have been established. Any MODE SELECT conditions that have no saved values established shall be returned to their default values.

The SSA-TL constructs of Initiator Tables, pending asynchronous alerts, and unprocessed SMSs shall be unaffected.

Before issuing the DEVICE RESET SMS the initiator should terminate any related outbound data transfers to ensure that data is not sent to a non-existent nexus. If the target has issued any DATA REQUEST SMSs to other initiators then the target shall receive and flush all the requested data before aborting the corresponding I/O processes.

A SCSI STATUS SMS shall be sent for each I/O process that completed before the SCSI RESPONSE SMS for the DEVICE RESET was sent. Once the target has sent the SCSI RESPONSE SMS, no status shall be sent for any aborted I/O process. However the target shall set the Unit Attention condition for all initiators. The ASC/ASCQ indicates POWER-ON OR RESET OCCURRED. The target always replies to the DEVICE RESET SMS with an SCSI RESPONSE SMS. Upon receiving the SCSI RESPONSE SMS, the initiator's S2P layer shall remove all aborted commands from its Outstanding Commands Table (see annex D.3.1).

The contents of the DATA field in a DEVICE RESET SMS are as defined in Table 12.

Table 12 - DEVICE RESET SMS

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE (33h)							
2	TAG							
3	TAG							
4	RETURN PATH ID							
5	RETURN PATH ID							
6	RETURN PATH ID							
7	RETURN PATH ID							

The TAG field is used to relate the SCSI RESPONSE SMS to the DEVICE RESET SMS.

The RETURN PATH ID field identifies the logical path that shall be used by the target to return the associated SCSI RESPONSE SMS.

In the SCSI RESPONSE SMS, the RETURN CODE field shall be REQUESTED FUNCTION COMPLETED SUCCESSFULLY.

6.12 CLEAR SCA CONDITION SMS

The CLEAR SCA CONDITION SMS is sent from an initiator to a target to clear a Serial Contingent Allegiance condition for that initiator and a selected Logical Unit or Target Routine. After the Serial Contingent Allegiance condition is cleared, any suspended queued command for that initiator may become an active I/O process subject to the SCSI-2 ordering rules. Serial Contingent Allegiance conditions for other initiators, Logical Units or Target Routines are not affected. The target shall reply to the CLEAR SCA CONDITION SMS with an SCSI RESPONSE SMS.

The contents of the DATA field in the CLEAR SCA CONDITION SMS are defined in Table 13.

Table 13 - CLEAR SCA CONDITION SMS

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE (34h)							
2	TAG							
3	TAG							
4	RETURN PATH ID							
5	RETURN PATH ID							
6	RETURN PATH ID							
7	RETURN PATH ID							
8	LUNTAR	LUNTRN						

The TAG field is used to relate the SCSI RESPONSE SMS to the CLEAR SCA CONDITION SMS.

The RETURN PATH ID field identifies the logical path, that shall be used by the target to return the associated SCSI RESPONSE SMS.

If the LUNTAR bit is set, then the LUNTRN field refers to a Target Routine. If the LUNTAR bit is cleared, then the LUNTRN field refers to a Logical Unit.

The LUNTRN field specifies either a Logical Unit or a Target Routine, as indicated by the value of the LUNTAR bit. If an unsupported or unattached LUN is specified, an SCSI RESPONSE SMS is generated with a RETURN CODE of INVALID FIELD.

In the SCSI RESPONSE SMS, the RETURN CODE field shall be one of REQUESTED FUNCTION COMPLETED SUCCESSFULLY, NO SCA CONDITION EXISTED FOR THE ADDRESSED LOGICAL UNIT OR TARGET ROUTINE, or INVALID FIELD.

6.13 SCSI RESPONSE SMS

The SCSI RESPONSE SMS is used to acknowledge the following SSA-S2P SMSs.

a) ABORT TAG
b) ABORT
c) CLEAR QUEUE
d) DEVICE RESET
e) CLEAR SCA CONDITION
f) SCSI COMMAND (only if certain fields are invalid)

The contents of the DATA field in an SCSI RESPONSE SMS are defined in Table 14.

Table 14 - SCSI RESPONSE

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (82h)							
1	S2P CODE (03h)							
2	TAG							
3	TAG							
4	RETURN CODE							

The TAG field is copied from the original SMS, and identifies the SMS that is being acknowledged.

The RETURN CODE field indicates the result of the original SMS, and the values are shown in Table 15.

Table 15 - RETURN CODE values for the SCSI RESPONSE SMS

Value	Description
00h	REQUESTED FUNCTION WAS COMPLETED SUCCESSFULLY.
01h	I/O PROCESS NOT FOUND
02h-03h	reserved
04h	OVERLAPPED SMSS ATTEMPTED
05h-1Fh	reserved
20h	NO SCA CONDITION EXISTED FOR THE ADDRESSED LOGICAL UNIT OR TARGET ROUTINE
21h-FEh	reserved
FFh	INVALID FIELD

7 Transport related items

7.1 Spindle synchronization

7.1.1 SYNC character

SSA defines the 8B/10B character K28.0 as a SYNC character for rotating media as spindle sync, that may be interleaved within frames on the link subject to the rules for User Defined characters in SSA-TL.

One node in the Web should be nominated to originate the SYNC characters at the nominal rotation rate of the devices. For example, this node may be an array controller or a disk drive. The other devices decode the SYNC characters and may use the event to synchronize their spindle servos. The event replaces the Master sync pulse that was provided on a separate cable for the parallel SCSI bus. If the originating node fails, a backup node may be nominated to replace it.

Propagation of User Defined characters is controlled by the EUDC bit of the CONFIGURE PORT SMS (see SSA-TL). Dual-port and switch nodes should normally be programmed to propagate User Defined characters received on one port to the other port(s). However, to avoid indefinite circulation, one node in each Cyclic Path should be programmed to block propagation of User Defined characters.

The origination and processing of SYNC characters is controlled by the RPL field in mode page 04h of the SCSI MODE SELECT command.

7.1.2 Option Block usage of the SYNC pin

SSA-PH defines an options connector that includes a SYNC pin. This SYNC pin provides the same function as the SYNC character over the SSA links, but allows synchronization across SSA Webs.

NOTE 2 - The device should provide a way to block generation and receipt of any signal on the SYNC pin for customers who use the SYNC character on the SSA links and use a backplane or cable structure that connects the SYNC pin.

7.2 Unit attention flag in the Initiator Table

SSA-S2P adds an additional flag, UNIT ATTENTION flag, to each Initiator Table entry (see 8.5).

7.3 Tag interaction between SSA-TL and SSA-S2P

Since there is no overlap between the responses to SSA-S2P SMS, and SSA-TL SMSs (including ASYNC ALERT SMSs), there is no requirement that the TAG field values be unique across outstanding SSA-S2P and SSA-TL SMSs. However tags used by SSA-S2P shall be unique across all ULPs that use the SSA-TL layer.

7.4 Effects of reset conditions

SSA-TL defines the following types of reset: Link Reset, Total Reset, Absolute Reset, and Power On Reset.

Link Reset has no affect on the SCSI constructs.

In response to a Total Reset, Absolute Reset, or Power On Reset, the S2P layer shall clear SCSI constructs (equivalent to an SCSI-2 Hard reset). In addition to the normal clearing of all I/O processes, reservations and operating conditions shall be reset to their last saved states.

Neither SSA-TL nor SSA-S2P provide a global reset equivalent to the RST signal in the parallel SCSI bus.

8 SSA-S2P changes from parallel SCSI-2

8.1 Untagged queuing is emulated

Since SSA is frame multiplexed, a tag or channel shall be associated with each frame to indicate the proper context. For this reason, Untagged commands are not supported by SSA-S2P. Untagged command queuing may be emulated by the initiator having only one outstanding command per LUN, re-use the same tag, and use the simple queue type. This is enforced by not allowing the Untagged queue type in the QUEUE CNTL field in the SCSI COMMAND SMS of SSA-S2P.

8.2 Serial Contingent Allegiance Condition

Logical units implement the Serial Contingent Allegiance Condition in place of SCSI-2 Contingent Allegiance condition. Serial Contingent Allegiance is necessary because an SSA initiator may send SCSI COMMAND SMSs prior to receiving the CHECK CONDITION status for a previous command. This results in the Contingent Allegiance condition being cleared prior to an initiator being able to deal with it.

When a target sends CHECK CONDITION status to an initiator it creates a Serial Contingent Allegiance condition for that initiator and Logical Unit or Target Routine. The Serial Contingent Allegiance condition causes the target to either suspend or abort queued I/O processes for that nexus, depending on the value of the QERR bit in mode page 0Ah of the MODE SELECT command.

The Serial Contingent Allegiance condition also causes subsequent I/O processes received for the same nexus to be rejected with the SCSI STATUS SMS STATUS field value of SCA ACTIVE (see Annex A), unless the QUEUE CNTL field specifies SERIAL CONTINGENT ALLEGIANCE. Such an I/O process is referred to as an SCA I/O process. SCA I/O processes are executed normally during a Serial Contingent Allegiance condition.

Only one SCA I/O process may be active at a time. If a target receives an SCA I/O process from the faulting initiator while it has another active SCA I/O process then it rejects the second SCA I/O process with an SCSI STATUS SMS STATUS field value of SCA ACTIVE.

If a target receives an SCA I/O process while it does not have a Serial Contingent Allegiance condition for the same nexus, then the target returns an SCSI STATUS SMS with a RETURN CODE value of CHECK CONDITION. The sense data contains a sense key of ILLEGAL REQUEST and an additional sense code of INVALID SMS ERROR.

The Serial Contingent Allegiance condition is cleared by any of the following conditions:

a) CLEAR SCA CONDITION SMS;
b) DEVICE RESET SMS;
c) Hard Reset condition including the SSA-TL concepts of Total or Absolute Reset.

After the Serial Contingent Allegiance condition is cleared by a CLEAR SCA CONDITION SMS, the target resumes processing any remaining suspended I/O processes for that nexus.

The Serial Contingent Allegiance condition is associated with a particular initiator (the faulting initiator) and Logical Unit or Target Routine. It does not affect I/O processes for other initiators. Multiple Serial Contingent Allegiance conditions may exist for different initiators, Logical Units and Target Routines at the same time.

8.3 Concurrent data transfers from a single target

SSA allows a logical unit to have several data transfers in progress at one time. For example, a media read may be transferring over one SSA port, while a cached read (or write prefetch) is transferring over another.

When a logical unit generates a Serial Contingent Allegiance condition, any I/O processes within the Serial Contingent Allegiance domain shall be suspended. When suspending an I/O process that involves a concurrent data transfer, the data transfer burst may complete, but the target shall not initiate a new data burst (DATA READY or DATA REQUEST), nor complete the I/O process (SCSI STATUS SMS) until the Serial Contingent Allegiance has been cleared.

8.4 Multiple logical paths

As seen by S2P, a logical path is a conduit for the ordered delivery of SMS and data frames between an initiator-target pair (an I_T nexus). For a specific I_T nexus, a logical path is uniquely identified by the RETURN PATH ID field in the SCSI COMMAND and task management SMSs. Ordered delivery means that for a given logical path, data frames and SMSs are received in the same order they were sent. SSA allows an I_T nexus to be connected through multiple logical paths but does not guarantee that the order of frame and SMS delivery is preserved between them. As a result, the following conditions apply to the use of multiple logical paths between an I_T nexus:

a) For a particular I/O process, all SMSs and data frames shall use the same logical path.
b) Different I/O processes for the same I_T nexus may use different logical paths. Since delivery order is not guaranteed between logical paths, it is possible for the arrival order of frames sent along different logical paths to vary from the order that such frames were sent.
c) To ensure that all in-flight commands are aborted, the DEVICE RESET, CLEAR TASK SET, ABORT and ABORT TAG SMSs shall be sent over each different logical path that was used to start I/O processes.

8.5 Unit attention

An SSA Web may contain multiple initiators. During configuration every node receives a QUERY NODE SMS from each initiator. If an initiator intends to use alternate logical paths to the same target then it issues a QUERY NODE SMS over each logical path. A target node uses the information in QUERY NODE SMS to build an entry in its Initiator Table. The entry contains the Return Path and port for that initiator (see SSA-TL).

When a target generates a Unit Attention it sets an SSA-S2P specific UNIT ATTENTION flag in all appropriate entries of the Initiator Table (only one flag for all logical paths to a given initiator). If the Unit Attention was generated by an SMS, then the target does not set the UNIT ATTENTION flag for the associated initiator.

When a target receives an SCSI COMMAND SMS it locates the UNIT ATTENTION flag for the initiator by accessing the Initiator Table entry containing the RETURN PATH ID field value. If the UNIT ATTENTION flag is set, then the target clears the flag and terminates the SCSI COMMAND SMS by generating an SCSI STATUS SMS with a STATUS field value of CHECK CONDITION, and generates sense data. Thus the Unit Attention is presented once to each initiator.

8.6 Optional SCSI-2 features not supported in SSA-S2P

The following optional SCSI-2 features are not supported in SSA-S2P.

8.6.1 Third party commands

SSA-S2P needs more addressing information to process a third party command than SCSI-2, namely:

a) Path to the source (4 bytes);
b) Return path id to the source (4 bytes);
c) LUN of the source (1 byte);
d) Port to use to access the source (1 to 7 bits);
e) Path to the destination node (4 bytes);
f) Return path id to the destination node (4 bytes);
g) LUN of the destination node (1 byte);
h) and Port to use to access the destination node (1 to 7 bits).

Therefore, to avoid impacting the SCSI-2 commands, the following SCSI commands are not supported:

a) Copy command (18h);
b) Copy and Verify command (3Ah);
c) Release command (17h) (Third party only);
d) and Reserve command (16h) (Third Party only).

8.6.2 Asynchronous Event Notification (AEN)

SSA-S2P does not support Asynchronous Event Notification (AEN).

8.6.3 Terminate I/O process message

SSA-S2P does not support the Terminate I/O Process message, due to lack of implementation in today's parallel SCSI products.

8.7 Mode Page parameter differences

Several of the parallel SCSI-2 mode page parameters are redefined for SSA-S2P.

8.7.1 Page 02h - Disconnect/Reconnect parameters

Several parameters within the SCSI-2 Disconnect/Reconnect parameters page are not required by SSA due to the frame multiplexing nature of the link. The following parameters shall be ignored:

a) BUS INACTIVITY LIMIT;
b) DISCONNECT TIME LIMIT;
c) CONNECT TIME LIMIT;
d) MAXIMUM BURST SIZE;
e) DTDC.

The target shall return the Page 02h parameters in the Mode Sense command that were issued with the Mode Select command, according to the rules regarding current and saved values. SSA-S2P does not specify what values the parameters shall be initialized to, nor does SSA-S2P specify the parameters that are changeable.

8.7.2 Page 0Ah - Control Mode parameters

Since the optional concept of Extended Contingent Allegiance (ECA) is not supported by SSA-S2P, the following parameters are fixed in Page 0Ah (Control Mode parameters). These parameters shall be initialized to the specified values and shall not be reported as changeable:

a) EECA bit = 0b;
b) RAENP bit = 0b;
c) UAAENP bit = 0b;
d) EAENP bit = 0b;
e) READY AEN HOLDOFF PERIOD field = 0000h.

Annex A
(normative)
SCSI Status

An SCSI Status Byte is sent to the initiator in the SCSI STATUS SMS at the termination of each SCSI command unless the command is cleared by any of the following occurrences:

a) ABORT SMS
b) ABORT TAG SMS
c) CLEAR QUEUE SMS
d) DEVICE RESET SMS
e) A Hard reset condition (POR, Total Reset, and Absolute Reset).

The SCSI Status Byte is defined in Table A.1 and its values defined in Table A.2.

Table A.1 - SCSI status byte

7	6	5	4	3	2	1	0
reserved = 00b		STATUS CODE					reserved=0

Table A.2 - Status code bit definitions

Status code	7	6	5	4	3	2	1	0
GOOD	R	R	0	0	0	0	0	R
CHECK CONDITION	R	R	0	0	0	0	1	R
CONDITION MET	R	R	0	0	0	1	0	R
BUSY	R	R	0	0	1	0	0	R
INTERMEDIATE/ GOOD	R	R	0	1	0	0	0	R
INTERMEDIATE / CONDITION MET	R	R	0	1	0	1	0	R
RESERVATION CONFLICT	R	R	0	1	1	0	0	R
COMMAND TERMINATED (not used)	R	R	1	0	0	0	1	R
QUEUE FULL	R	R	1	0	1	0	0	R
SCA ACTIVE	R	R	1	1	0	0	0	R
NOTE - All reserved fields (R) shall be set to zero.								

GOOD status indicates that the target has successfully completed the SCSI command. For this status, sense is not valid and the Sense key and the Sense code have a value of zero.

CHECK CONDITION status indicates that an error, exception, or abnormal condition has caused sense data to be set. A Serial Contingent Allegiance condition exists for this initiator. The initiator should issue a REQUEST SENSE command with an QUEUE CNTL field value of SERIAL CONTINGENT ALLEGIANCE to obtain the sense data and determine the cause of the CHECK CONDITION status.

CONDITION MET status indicates that the requested operation is satisfied. For this status, sense is not valid and the Sense key and the Sense code have a value of zero.

BUSY status indicates that the logical unit shall not execute the command because it is busy performing another operation for a different initiator. For SCSI-2 compatibility, the logical unit shall return this status when a command is received while a Contingent Allegiance condition is in effect for another initiator. The recommended recovery action for the application layer is to issue the command again at a later time. For this status, sense is not valid and the Sense key and the Sense code have a value of zero.

INTERMEDIATE/GOOD status indicates that the target has successfully completed a linked command. This status is returned for every command in a series of linked commands (except the last command) unless an error, exception, or abnormal condition causes a CHECK CONDITION, BUSY, or RESERVATION CONFLICT

status to be returned. If this status is not returned, the chain of linked commands is broken. For this status, sense is not valid and the Sense key and the Sense code have a value of zero.

INTERMEDIATE/CONDITION MET status is the combination of CONDITION MET and INTERMEDIATE/GOOD. For this status, sense is not valid and the Sense key and the Sense code have a value of zero.

RESERVATION CONFLICT status shall be returned whenever an initiator attempts to access a Logical Unit or an extent within that Logical Unit that is reserved with a conflicting reservation type for another SCSI device. The recommended initiator recovery action is to issue the command again at a later time. For this status, sense is not valid and the Sense key and the Sense code have a value of zero.

COMMAND TERMINATED status is not used in SSA-S2P.

QUEUE FULL status indicates that the targets command queue is full. This status is returned when an SCSI COMMAND SMS is received and there is no room on the command queue for an I/O process from the issuing initiator. For this status, sense is not valid and the Sense key and the Sense code have a value of zero.

SCA ACTIVE status shall be returned while a Serial Contingent Allegiance condition is in effect and the command was received from the faulting initiator while one of the following was true:

a) There was another I/O process with the SCA queue type; or

b) The command did not have the SCA Queue type.

For SCSI-2 compatibility, a command received by the logical unit while a Contingent Allegiance is in effect for another initiator shall be terminated with a status of BUSY as described above.

The SCA condition is not cleared, and the command was not executed. Sense data was not modified.

Annex B
(informative)
Parallel SCSI-2 message to SSA-S2P SMS conversion

B.1 SCSI Message summary

The mapping (if any) between the functions of the SCSI-2 messages and SSA-S2P are shown in Table B.1.

Table B.1 - Mapping from parallel SCSI-2 messages to SSA-S2P SMSs

SCSI-2 message	SSA-S2P SMS
No Operation	N/A
Simple Queue Tag	SCSI COMMAND(QUEUE CNTL=11b, TAG)
Ordered Queue Tag	SCSI COMMAND(QUEUE CNTL=10b, TAG)
Head of Queue Tag	SCSI COMMAND(QUEUE CNTL=01b, TAG)
SCA Queue Tag (SCSI-3)	SCSI COMMAND (QUEUE CNTL=00b, TAG)
Identify (Out)	SCSI COMMAND (LUNTAR, LUNTRN)
Command Complete	SCSI STATUS
Linked Command Complete	SCSI STATUS(LINK=1b, FLAG = 0b)
Linked Command Complete with Flag	SCSI STATUS(LINK=1b, FLAG=1b)
Identify (In)	SCSI STATUS(TAG) DATA READY (TAG) DATA REQUEST(TAG) SCSI RESPONSE
Modify Data Pointer	DATA READY(BYTE OFFSET) DATA REQUEST(BYTE OFFSET)
Disconnect	N/A
Save Data Pointer	N/A
Restore Pointers	N/A
Initiate Recovery	N/A
Release Recovery	N/A
Terminate I/O Process	N/A
Abort	ABORT
Abort Tag	ABORT TAG
Clear Queue	CLEAR QUEUE
Clear SCA Condition (SCSI-3)	CLEAR SCA CONDITION
Bus Device Reset	DEVICE RESET
Message Reject	SCSI RESPONSE ASYNC ALERT & MASTER ALERT
Initiator Detected Error	N/A
Message Parity Error	N/A
Synchronous Transfer Request	N/A
Wide Data Transfer Request	N/A
Ignore Wide Residue	N/A
N/A	DATA REPLY

B.2 Parallel SCSI-2 message to SSA-S2P SMS Conversion Discussion

A brief description of how to map each SCSI-2 message into an SSA-S2P SMS is as follows:

(00h) Command Complete maps to the SCSI STATUS SMS with the status byte residing in the STATUS field, the LINK bit cleared, and the FLAG bit cleared.

(01h, 00h) Modify Data Pointer message maps to the SSA DATA READY and DATA REQUEST SMSs.

(01h, 01h) Synchronous Data Transfer Request message is N/A since SSA is a frame interleaved serial bus that always transfers at a fixed speed.

(01h, 02h) Extended Identify (SCSI-1) message is N/A since Extended Identify was an SCSI-1 concept and is not supported in SCSI-2, SCSI-3 or SSA.

(01h, 03h) Wide Data Transfer Request message is N/A since SSA is a serial interface, and does not operate in a wide mode.

(02h) Save Data Pointer message is N/A since a frame multiplexed protocol does an implicit disconnect after every frame.

(03h) Restore Data Pointer message is N/A since a frame multiplexed protocol does an implicit disconnect after every frame.

(04h) Disconnect message is N/A since a frame multiplexed protocol does an implicit disconnect after every frame.

(05h) Initiator Detected Error message is N/A as frame rejects are handled in the transport layer of SSA. The level of reset depends on the initiator's error recovery procedure.

(06h) Abort message maps directly to the SSA ABORT SMS.

(07h) Message Reject is N/A as frame rejects are handled in the transport layer of SSA.

(08h) No Operation is N/A for SSA. When a node does not need to send any information, FLAG characters as sent over the SSA cable.

(09h) Message Parity Error message is N/A as frame CRC checks are handled in the transport layer of SSA.

(0Ah) Linked Command Complete message maps into the SCSI STATUS SMS with the status byte residing in the STATUS FIELD, the LINK bit set, and the FLAG bit cleared.

(0Bh) Linked Command Complete with Flag message maps into the SCSI STATUS SMS with the status byte residing in the STATUS field, the LINK bit set, and the FLAG bit set.

(0Ch) Bus Device Reset message maps to the SSA DEVICE RESET SMS.

(0Dh) Abort Tag message maps to the SSA ABORT TAG SMS.

(0Eh) Clear Queue message maps to the SSA CLEAR QUEUE SMS.

(0Fh) Initiate Recovery message is N/A since SSA does not support Extended Contingent Allegiance.

(10h) Release Recovery message is N/A since SSA does not support Extended Contingent Allegiance.

(11h) Terminate I/O Process message is N/A since SSA does not support the Terminate I/O process message.

(16h) Clear ACA Condition (SCSI-3) message maps to the SSA CLEAR SCA CONDITION SMS.

(20h, xxh) Simple Queue Tag message maps to the SCSI COMMAND SMS with a QUEUE TYPE field = 11b (Simple) and the queue tag value in the TAG field.

(21h, xxh) Head of Queue Tag message maps to the SCSI COMMAND SMS with a QUEUE TYPE field = 01b (Head) and the queue tag value in the TAG field.

(22h, xxh) Ordered Queue Tag message maps to the SCSI COMMAND SMS with a QUEUE TYPE field = 10b (Ordered) and the queue tag value in the TAG field.

(24h, xxh) ACA Queue Tag (SCSI-3) message maps to the SCSI COMMAND SMS with a QUEUE TYPE field = 00b (SCA) and the queue tag value in the TAG field.

(23h, xxh) Ignore Wide Residue message is N/A as a set of data frames may be any number of bytes, and the concept of ignoring data with a grosser granularity of one byte does not apply.

(80h-FFh) Identify message maps to the SCSI COMMAND with the LUN value indicated in the LUNTAR bit and LUNTRN field. The bit indicating the privilege of disconnection is not applicable in SSA where each frame has an implicit disconnect.

Annex C
(informative)
Some differences between SSA-S2P and parallel SCSI-2

This Annex explores some of the differences between SSA-S2P and parallel SCSI-2.

C.1 Non-interlocked

Since SSA is a frame multiplexed interface rather than an interlocked interface, it does not depend on the bus phases to indicate acceptance or completion of a message. The ACK character pair in SSA-TL only indicates that the remote port has correctly received the frame. This implies that some form of response is needed for each SMS to indicate completion (see 6.13 for more details).

C.1.1 Message Reject handling

Since SSA-S2P is not an interlocked protocol, the Message Reject condition does not exist as it does today in the parallel environment. By sending an ACK character pair, the remote port indicates acceptance of the SMS. Acceptance of the frame, does not indicate successful parsing of the frame.

However, SSA-S2P does respond to some of the SMS reject cases with an ASYNC ALERT SMS. Refer to 6.2 for the proper handling of invalid SMSs such as UNKNOWN RETURN PATH OR RETURN PATH ID, or UNSUPPORTED SMS.

C.1.2 Bus phases do not exist

Since SSA is frame multiplexed, parallel SCSI-2 bus phases do not exist. Each frame may be considered to do an arbitration, selection/reselection, and disconnect phase (without the parallel SCSI-2 bus overhead). A data frame or SCSI COMMAND SMS frame is considered to have done a data transfer phase.

C.2 Performance features

C.2.1 Less overhead for arbitration, selection, disconnection, reselection

Since SSA is frame multiplexed, each frame does implicit arbitration, selection/reselection, and disconnection with 8-12 bytes of overhead per frame (400 ns to 600 ns when running on SSA-PH1).

C.2.2 Out of order data transfers

Out of order data transfers (zero latency reads or writes) reduce latency by allowing the target to begin transfer from the current rotational position. For a full track read, this may save an average of 1/2 a revolution or 33% over a target that does not support out of order transfers. Without out of order, on average 1/2 of a rotation is used to find the beginning, and one revolution to read or write the track. With out-of-order transfers, only one revolution is used (R/W last N %, then R/W first 100- N %).

In parallel SCSI, out of order data transfers were optional by using the Modify Data Pointers message. SSA-S2P makes the support mandatory for the target, and selectable by the initiator (OOT bit). The SCSI COMMAND SMS is used along with the data transfer SMSs to activate out of order data transfers.

C.2.3 Concurrent data transfers within a Web

Since SSA is frame multiplexed with a low frame overhead and multiple channel capability, there is no penalty for transferring multiple data transfers simultaneously. This may be an advantage in a RAID application using a single SSA Web, because the data from each of the RAID components may be processed (i. e. reconstruction) as the data arrives. In other words, a parallel bus may transfer all the data from one target and then disconnect, allowing the next target to transfer a chunk of data, etc., before the data is verified. But with SSA, all RAID components may be transferring simultaneously in an interleaved fashion, allowing the data to be verified as it arrives, thereby reducing latency.

C.2.4 Minimal initiator/target exchanges

Initiator/target exchanges may be minimized by grouping the parallel SCSI-2 functions of arbitration, message out (i. e. Identify message, Queue Tag message), and command phase. into SSA-S2P SCSI COMMAND SMS. The SCSI COMMAND SMS has additional function of being able to define options not available in SCSI-2 such as the DDRM and OOT bits on a command by command basis.

C.3 Availability features

SSA has high availability by nature of its rich topologies and multiple logical paths. See 8.4 for information on multiple logical paths.

C.4 Addressing

SSA greatly enhances the maximum number of addresses on an SSA Web. This requires some changes to parallel SCSI-2.

C.4.1 Additional addressing

SCSI-2 allows addressing of 8 (8 bit SCSI) or 16 (16 bit SCSI) devices on an SCSI parallel bus. The four byte address field of SSA-TL supports approximately 250 million devices with the use of switches. However, the two byte tag value limits the number of nodes to less than 64K, since a tag is needed for a possible ASYNC ALERT SMS from each node.

C.4.2 Larger LUN field

SSA-S2P defines a 128 Logical Units (LUN) and 128 Target Routines within a target, as opposed to parallel SCSI-2, that allows 8 LUNs and 8 Target Routines.

Annex D
(informative)
Protocol services model

This annex demonstrates how the SAM Services model may map onto the SSA-S2P protocol. This annex is for informational purposes only, and does not place any requirements on the implementation, other than those requirements already specified in this standard. Since the SAM Service model is a logical model, certain assumptions were made to map it onto a more detailed implementation (see D.1.2 and D.1.3).

D.1 Introduction

D.1.2 The SAM/Protocol/Transport layers

A partial list of the major responsibilities of the SAM, Protocol, and Transport layers is as follows.

The SAM layer is responsible for SCSI Command generation, queuing, and execution. SAM also handles sense data building, Contingent Allegiance or Serial Contingent Allegiance conditions, and command termination.

The Protocol layer provides the function to map the SAM Protocol services onto the SSA-TL transport layer. In addition, the Protocol layer maintains a data structure of all outstanding commands to allow for deadlock avoidance and error handling in case of illegal operations resulting in a deadlock situation.

The Transport layer provides the function to map the Transport Services onto the SSA-PH1 physical environment. It also provides procedures for configuration, error reporting, error handling, and configuration management. The Transport layer also converts the Protocol Unique ID into an RETURN PATH ID used by some SMSs by the use of the Initiator Table. The transport layer is also responsible for all port and alternative logical path selection (The protocol layer does not know about multiple ports or paths).

D.1.3 Guidance for implementation of the SAM Services model

The SAM Services model is a logical model. In an attempt to show a more detailed implementation of the SAM Services model on SSA-S2P, certain assumptions were made as shown as follows:

a) Any implementation shall have a known parameter list rather than a set of optional parameters. Therefore, a test for null is used to determine if a parameter is included.

b) The SAM Services model (a logical model) assumes data is queued and sent with the service request or confirmation. In an actual implementation, pointers to host or device memory are sent and data is written to or read from memory below the SAM layer. Therefore the data structures are replaced with memory address pointers, and the names changed to add Address to the name.

D.1.4 Modifications to the Device Server model

SSA-S2P requires a minor modification to the device server model to control Queue Full condition handling.

Once the Device Server detects a Queue Full condition and responds to an SCSI Command with QUEUE FULL status, then all other commands received after that point shall be ignored (not processed) until an SCSI Command is received with the Resume parameter set.

More specifically, when the Device Server detects a Queue Full condition, it generates a Send Command Complete protocol service response with a Status parameter value of QUEUE FULL and sets an internal QUEUE FULL flag. All SCSI Command Received protocol service indications received with the Resume parameter cleared and the internal QUEUE FULL flag set, shall be ignored (not queued or executed). When an SCSI Command Received protocol service indication is received with the Resume parameter set, the internal QUEUE FULL flag is cleared and the command is processed (possibly causing another QUEUE FULL Status).

D.1.5 Deadlock avoidance

A potential deadlock situation may occur since resources are limited in nodes, and SMSs are unsolicited. This situation does not occur with data transfers, since they are solicited with the use of data transfer SMSs.

Deadlock avoidance consists of two parts described as follows:

a) The number of pending non-SCSI COMMAND SMSs are limited and sufficient resources exist for them. Both the Transport and Protocol layers are responsible for this. The Protocol layer simply holds any service that may send a non-SCSI COMMAND SMS while any non-SCSI COMMANDS SMSs are outstanding.

b) The SCSI COMMAND SMS resource limitations are controlled by the use of Queue Full and the protocol layers Outstanding Commands Table. When the initiator S2P protocol layer receives an SCSI STATUS SMS with STATUS field of QUEUE FULL, it generates a Command Complete Received protocol service response for all commands in its Outstanding Commands Table that were issued after the command responding with QUEUE FULL status. Since the initiator generates the QUEUE FULL status for those commands, the target is free to discard all SCSI Commands with the RESUME bit cleared, and not issue any SCSI STATUS SMSs with a QUEUE FULL status. The next SCSI Command to be sent has the RESUME bit set, causing the QUEUE FULL flag in the device to be cleared and the resumption of SCSI command processing.

D.1.6 Implementation specific options

The SSA protocol and transport services have several implementation specific features as described below:

a) The transport layer is responsible for controlling hardware and allocating data transfer resources, including CHANNEL values. If sufficient resources exist to allocate the data channel when the command is received (Send Command protocol service) rather than Data Transfer time, then overhead may be reduced during command overhead (Refer to the DDRM and CHANNEL fields of the SCSI Command SMS). If this is implemented, then the TARGET FAST READ and FAST READ DATA COMPLETED Transport services are used in place of the TARGET READ, INITIATOR READ, and READ DATA COMPLETED Transport services.

b) Accepting data transfers that are out of order (non-contiguous) for a command may improve performance significantly (Refer to the OOT bit of the SCSI COMMAND SMS in 7.4).

c) The SSA-S2P supports the concept of Serial Contingent Allegiance (SCA) that is similar to the SCSI-3 concept of Auto Contingent Allegiance (ACA). The SCSI-3 concept of Auto Contingent Allegiance is not be directly supported because the SCSI-2 CDB and Status bytes do not have the required fields and values. However, an SSA-S2P implementation is able to handle the Auto Contingent Allegiance condition by supporting the required CDB fields and Status values.

D.2 Conventions

This clause describes how to map the various SAM objects into SSA-S2P fields, and describes some of the notations used.

D.2.1 Definitions changes from SCSI-2 to SCSI-3

This standard refers to both the SCSI-2 standard, and to the SCSI-3 Architectural model (SAM). This clause gives an approximate correlation between SCSI-2 terms and SAM terms. Refer to the appropriate standard for more detailed definitions.

a) SCSI-2 I/O process relates to SCSI-3 Task.

b) SCSI-2 target portion that sits above the protocol layer is the SCSI-3 Device Server.

c) SCSI-2 initiator portion that sits above the protocol layer is the SCSI-3 Application Client

D.2.2 Service naming conventions.

The term Protocol Services refers to those services between the SSA-S2P layer and the SAM layer, and in following SAM the name contains first letter capitalized words. The term Transport Services refers to those services between the SSA-TL transport layer and the SSA-S2P protocol layer, and in following SSA-TL and SSA-S2P conventions the name is all letters capitalized.

D.2.3 Notations

The services are indicated with indented text as illustrated below.

Service-name (Input parm1, [Optional input parm2] || Output parm1, [Optional Output parm2])

The service name precedes the parentheses. The parentheses enclose the parameter lists. Any parameter enclosed in square brackets ([...]) is considered optional. The double bar (||) separates the input parameters from the output parameters.

D.3 Protocol layer data structures

D.3.1 Outstanding Commands table

The initiator's protocol layer shall maintain an Outstanding Commands Table consisting of entries for all outstanding commands as shown in Table D.1. The target does not maintain this table. The sequence of entries corresponds to the order in that the commands were sent to the target. As seen by the initiator's S2P layer, a command is outstanding after the Send Command protocol service request is received from the application client until the Command Complete Received confirmation is received from the target or the command is aborted. All fields are obtained from the Send SCSI Command protocol service.

Table D.1 - Outstanding Commands Table entry format

Field	Description
DESTINATION NODE ID	The 8 byte unique ID of the target
SOURCE ID	The 8 byte unique ID of the initiator from the Task Identifier parameter.
TAG	The TAG value of the SCSI command from the Task Identifier parameter.
LUN	The Logical Unit number from the Task Identifier parameter.
DATA-OUT BUFFER ADDRESS	The optional Buffer-Out address parameter
DATA-IN BUFFER ADDRESS	The optional Buffer In address parameter
COMMAND BYTE COUNT	The optional Command Byte Count parameter
CDB	A copy of the Command Descriptor Block

D.3.2 QUEUE FULL flag

A single bit QUEUE FULL flag for the entire data structure is maintained for deadlock avoidance.

For the target protocol layer, the Queue full flag is processed as follows:

a) The QUEUE FULL flag in the target is set when it was previously cleared and a Send Command Complete protocol service response is received with a Status parameter value of QUEUE FULL. The Send Command Complete protocol service response is processed normally.

b) Any Send Command Complete protocol service response received with a Status parameter value of QUEUE FULL while the QUEUE FULL flag is set shall be ignored.

c) Any SMS RECEIVED transport service is invoked with a valid SCSI COMMAND SMS with a RESUME bit cleared while the QUEUE FULL flag is set shall be ignored (the Return Code parameter value is FUNCTION SUCCESSFUL, but no action is taken).

d) The QUEUE FULL flag is cleared and the command is processed when an SMS RECEIVED transport service is invoked with a valid SCSI COMMAND SMS with a RESUME bit set. The SMS RECEIVED transport service executed normally.

For the initiator protocol layer, the QUEUE FULL flag is processed as follows:

a) The QUEUE FULL flag in the target is set regardless of its previous state, and an SMS RECEIVED transport service is invoked with a valid SCSI STATUS SMS with a STATUS field value of QUEUE FULL. A Command Complete Received protocol service confirmation is then generated. The Outstanding Commands Table is scanned for any commands issued after the command returning QUEUE FULL status, and a Command Complete Received protocol service confirmation is generated and the entry cleared as if an SCSI STATUS was received with a QUEUE FULL status.

b) The QUEUE FULL flag is cleared, when it was previously set and a Send SCSI Command protocol service request is received. The resulting SCSI COMMAND SMS STRUCTURE has the RESUME bit set (see 6.6).

D.3.3 Object definitions

The Destination ID is the 64 bit Unique ID of the Destination.

The initiator Identifier is the device's 64 bit Unique ID.

The Logical Unit Number is a 7 bit value that resides in the LUNTRN field (byte 8) of those SMSs that contain a Logical Unit Number (SCSI COMMAND, ABORT, CLEAR QUEUE, and CLEAR SCA). The SCSI-2 concept of Target Routines is also included in byte 8.

The Source ID is the 64 bit Unique ID of the Source.

The Tag is a two byte value that resides in the TAG field (bytes 2 and 3) of all SSA-S2P SMSs.

The Target Identifier is the device's 64 bit Unique ID. The Unique ID is globally unique, and is the value is the same to each device in the SSA Web.

The Task Attribute is a two bit value that resides in the QUEUE CNTL field of the SCSI COMMAND SMS.

D.4 SCSI Command protocol services

Table D.2 shows the typical services activity to execute a command that does not involve data transfer, where ==> and <== indicate direction.

Table D.2 - Command execution services activity

Initiator			Cable	Target		
SAM	Protocol	Transport	Cable	Transport	Protocol	SAM
Protocol Services	Transport Services		Cable	Transport Services	Protocol Services	
==> Send SCSI Command	==> GET RETURN PATH ID ==> SEND SMS		==> SCSI COMMAND SMS	==> SMS RECEIVED <== GET UNIQUE ID	==> SCSI Command Received	
<== Command Complete Received	<== SMS RECEIVED		<== SCSI STATUS SMS	<== SEND SMS	<== Send Command Complete	

D.4.1 Send SCSI Command protocol service request

Send SCSI Command (Destination ID, Source ID, Tag, LUN, CDB, [Task Attribute], [Data-Out Buffer Address], [Data-In Buffer Address], [Command Byte Count], [Autosense Request] ||)

Upon receipt of the Send SCSI Command protocol Service request from SAM, the protocol layer builds an SCSI COMMAND SMS and instructs the transport layer to transmit it.

The SSA-S2P protocol builds an SCSI COMMAND SMS into a data structure entitled SMS STRUCTURE as follows:

a) The SMS CODE and S2P CODE fields are set to 82h and 10h respectively.

b) The Tag and LUN parameters go in the TAG and LUNTRN fields respectively.

c) The RETURN PATH ID field is the Return path id parameter from the invocation of the GET RETURN PATH ID transport service (see E.2.1).

d) The CDB parameter goes in the CDB field.

e) The Task Attribute parameter goes in the QUEUE CNTL field (the SIMPLE QUEUE value is used if the Task Attribute parameter is null).

f) The use of the DDRM, CHANNEL, and OOT fields is implementation dependent.

g) The use of the RESUME bit depends on the status of the QUEUE FULL flag condition (see D.3.2).

The Autosense parameter is ignored in the SSA-S2P protocol

The SSA-S2P protocol then invokes the transport layer service of SEND SMS as follows to send the SCSI COMMAND SMS.

SEND SMS (Control, Destination ID, Buffer Address, Length || Return Code)

a) The control parameter is always 00h.
b) The Destination ID parameter is the Destination ID from the Send SCSI Command protocol service.
c) The Buffer Address parameter is the location of the SMS STRUCTURE data structure.
d) The Length parameter is the length of the SMS STRUCTURE data structure.

If the Return Code parameter indicates FUNCTION FAILED or INVALID PARAMETER, then the protocol layer generates a Command Complete Received protocol service confirmation with a Service Response parameter value of SERVICE DELIVERY OR TARGET FAILURE.

D.4.2 SCSI Command received protocol service indication

SCSI Command Received (Source ID, LUN, Tag, Task Attribute, CDB ||)

When the target's transport layer receives an SMS, it generates an SMS RECEIVED transport service indication. If the Protocol layer validates it as an SCSI Command SMS, then an SCSI Command Received Protocol service indication is generated as follows:

a) The Source ID parameter is obtained from the Unique ID parameter returned by the invocation of the GET UNIQUE ID (Return path id || Unique ID, Return Code) transport service. The Return path id parameter is taken from the RETURN PATH ID field of the SCSI COMMAND SMS
b) The TAG field of the SCSI COMMAND SMS is returned as the Tag parameter.
c) The LUNTRN field of the SCSI COMMAND SMS is returned as the LUN parameter.
d) The QUEUE CNTL field of the SCSI COMMAND SMS is returned as the Task Attribute parameter.
e) The CDB field of the SCSI COMMAND SMS is returned as the CDB parameter.

D.4.3 Send Command Complete protocol service response

Send Command Complete (Destination ID, Source ID, Tag, LUN, Status, Service Response ||)

Upon receipt of the Send Command Compete protocol service from the device server, the protocol performs the following actions.

The SSA-S2P protocol builds an SCSI STATUS SMS into a data structure entitled SMS STRUCTURE as follows:

a) The SMS CODE and S2P CODE fields are set to 82h and 10h respectively.
b) The Tag parameter goes into the TAG field.
c) The Status parameter goes into the STATUS field.
d) The Service Response parameter causes the LINK and FLAG bits to be set as defined in Table D.3.

Table D.3 - Service Response parameter and its impact on flag and link bits

Service Response parameter	FLAG bit	LINK bit
Task Complete	0b	0b
Linked Command Complete	0b	1b
Linked Command Complete (with flag)	1b	1b
Service Delivery of Target Failure	0b	0b

The SSA-S2P protocol then invokes the transport layer service of SEND SMS as follows to send the SCSI STATUS SMS.

SEND SMS (Control, Destination ID, Buffer Address, Length || Return Code)

a) The control field is always 00h.

b) The Destination ID parameter of the Send Command Complete protocol service passed as the Destination ID parameter.

c) The Buffer Address parameter is the location of the SMS STRUCTURE data structure.

d) The Length parameter is the length of the SMS STRUCTURE data structure.

D.4.4 Command Complete Received protocol service confirmation

Command Complete Received (Destination ID, Source ID, Tag, LUN, [Data-In Buffer Address], Status, Service Response ||)

When the initiator's transport layer generates an SMS RECEIVED transport service indication, and the SMS is validated by the protocol layer as an SCSI STATUS SMS, then a Command Complete Received protocol service indication is generated as follows:

a) The TAG field of the SCSI STATUS SMS is returned as the Tag parameter.

b) The Destination ID parameter is returned from the DESTINATION NODE ID field of the Outstanding Commands Table entry identified by the Tag parameter.

c) The Source ID parameter is the initiators Unique ID.

d) The LUN parameter is returned from the LUN field of the Outstanding Commands Table entry identified by the Tag parameter.

e) The Data-In Buffer Address parameter is returned from the DATA-IN BUFFER ADDRESS field of the Outstanding Commands Table entry identified by the Tag parameter.

f) The STATUS field is returned as the Status parameter.

g) The Service Response parameter shall be returned with TASK COMPLETE.

If any of the following transport services returns a FUNCTION FAILED or INVALID PARAMETER Return Code parameter, then the protocol generates a Command Complete Received protocol service indication.

a) SEND SMS associated with the Send SCSI Command protocol service request.

b) INITIATOR READ parameters are invalid.

c) INITIATOR WRITE parameters are invalid.

In any of the previous cases, then a Command Complete Received protocol service indication is generated as follows:

a) The Tag parameter of the associated protocol service request (Send SCSI Command Protocol) or transport service (INITIATOR READ or INITIATOR WRITE) is returned as the Tag parameter.

b) The Destination ID parameter is returned from the DESTINATION NODE ID field of the Outstanding Commands Table entry identified by the Tag parameter.

c) The Source ID parameter is the initiators Unique ID.

d) The LUN parameter is returned from the LUN field of the Outstanding Commands Table entry identified by the Tag parameter.

e) The Data-In Buffer parameter is not used.

f) The Status parameter is not used.

g) The Service Response parameter shall be returned with DELIVERY OR TARGET FAILURE.

D.5 Data transfer protocol services

Table D.4 shows the typical services activity to transmit Data-In, where ==> or <== indicates direction.

Table D.4 - Data-In services activity

Initiator			Cable	Target		
SAM	Protocol	Transport	Cable	Transport	Protocol	SAM
Protocol Services	Transport Services		Cable	Transport Services	Protocol Services	
	Initiator Read <==		DATA READY <== ==> DATA REPLY data<==	Target Read <==	Send Data-In <==	
				Read Data Completed ==>	Data Sent ===>	

Table D.5 shows the typical services activity to transmit Data-Out, where ==> or <== indicates direction.

Table D.5 - Data-Out services activity

Initiator			Cable	Target		
SAM	Protocol	Transport	Cable	Transport	Protocol	SAM
Protocol Services	Transport Services		Cable	Transport Services	Protocol Services	
	Initiator Write <==		DATA REQUEST<== ==>data	Target Write <==	Receive Data Out <==	
				Write Data Completed ==>	Data Out Received ==>	

D.5.1 Send Data-In protocol service request

Send Data-In (Destination ID, Source ID, Tag, LUN, Device Server Buffer, Application Client Buffer Offset, Request Byte Count ||)

Upon receipt of the Send Data-In protocol service request, the protocol invokes the following transport service.

TARGET READ (Destination ID, Tag, Buffer Address, Byte Offset, Byte Count, Start Count, Threshold Count || Return Code)

a) The Destination ID parameter of the Send Data In protocol service request becomes the Destination ID parameter of the TARGET READ protocol service.

b) The Tag parameter of the Send Data In protocol service request becomes the Tag parameter of the TARGET READ protocol service.

c) The Device Server Buffer parameter of the Send Data In protocol service request becomes the Buffer Address parameter of the TARGET READ protocol service.

d) The Application Client Buffer Offset parameter of the Send Data In protocol service request becomes the Byte Offset parameter of the TARGET READ protocol service.

e) The Request Byte Count parameter of the Send Data In protocol service request becomes the Byte Count parameter of the TARGET READ protocol service.

f) The use of the Start Length and Threshold Length parameters are implementation dependent. These parameters may be used when the target does not have Byte Count worth of data in its buffer at the time of invocation (but it does have Start Count), and intends to use the transport layer to throttle the data transfer. Refer to the Target Read service in the SSA-TL standard for more information on this function. Normal operation may set the Start Length parameter to the Byte Count parameter, and the Threshold Length parameter to zero.

If the Return Code parameter indicates FUNCTION FAILED, then the protocol layer generates a Command Complete Received protocol service confirmation with a Service Response parameter value of SERVICE DELIVERY OR TARGET FAILURE.

If the Return Code parameter indicates FUNCTION SUCCESSFUL, then the invocation of the TARGET READ transport service causes the Initiator protocol to receive an INITIATOR READ transport service request, by sending one or more DATA READY SMS(s).

D.5.2 Data Sent protocol service confirmation

Data Sent (Destination ID, Source ID, Tag, LUN ||)

When the data transfer initiated by the Send Data-In protocol service request completes, the transport layer informs the protocol layer with a READ DATA COMPLETE transport service confirmation.

READ DATA COMPLETED (|| Tag, LUN, Destination ID, Return Code)

Upon receipt of the READ DATA COMPLETED transport service, the protocol layer generates a Data Sent protocol service confirmation as follows:

Data Sent (Destination ID, Source ID, Tag, LUN ||)

a) The READ DATA COMPLETED Destination ID parameter is copied into the Destination ID parameter.
b) The target's Unique ID is returned as the Source ID parameter.
c) The READ DATA COMPLETED Tag parameter is copied into the Tag parameter.
d) The READ DATA COMPLETED LUN parameter is copied into the LUN parameter.

D.5.3 Receive Data-Out protocol service request

Receive Data-Out (Destination ID, Source ID, Tag, LUN, Application Client Buffer Offset, Request Byte Count, Device Server Buffer ||)

Upon receipt of the Receive Data-Out protocol service request, the protocol invokes the following transport service:

Target Write (Destination ID, Tag, Buffer Address, Byte Offset, Byte Count || Return Code)

a) The Receive Data-Out Destination ID parameter becomes the Destination ID parameter.
b) The Receive Data-Out Tag parameter becomes the Tag parameter.
c) The Receive Data-Out Device Server Buffer parameter becomes the Buffer Address parameter.
d) The Receive Data-Out Application Client Buffer Offset parameter becomes the Byte Offset parameter.
e) The Receive Data-Out Request Byte Count parameter becomes the Byte Count parameter.

If the Return Code parameter indicates FUNCTION FAILED, then the protocol layer generates a Send Command Complete protocol service request with a Service Response parameter value of SERVICE DELIVERY OR TARGET FAILURE.

The invocation of the Target Write transport service causes the Initiator protocol to receive an Initiator Write transport service request.

D.5.4 Data-Out Received protocol service confirmation

Data-Out Received (Destination ID, Source ID, Tag, LUN ||)

When the data transfer initiated by the Send Data-Out protocol service request completes, the transport layer informs the protocol layer with a WRITE DATA COMPLETE transport service confirmation.

WRITE DATA COMPLETED (|| Tag, LUN, Destination ID, Return Code)

Upon receipt of the READ DATA COMPLETED transport service, the protocol layer generates a Data-Out Received protocol service confirmation as follows:

Data-Out Received (Destination ID, Source ID, Tag, LUN ||)

a) The WRITE DATA COMPLETED Destination ID parameter is copied into the Destination ID parameter.
b) The target's Unique ID is returned as the Source ID parameter.
c) The WRITE DATA COMPLETED Tag parameter is copied into the Tag parameter.
d) The WRITE DATA COMPLETED LUN parameter is copied into the LUN parameter.

D.6 Task management functions

This clause describes the common actions of all Task Management functions except for the Terminate Task function, that always returns a Service Response of FUNCTION REJECTED.

Upon receipt of the Task Management Function protocol Service request, the protocol layer builds an SMS structure and instructs the transport layer to transmit it. The type of SMS depends on the Task Management function, and is detailed in each Task Management clause.

In all cases, the SSA-S2P protocol then invokes the transport layer service of SEND SMS as follows:

SEND SMS (Control, Destination ID, Buffer Address, Length || Return Code)

a) The Control parameter is always 00h
b) The Destination ID parameter is taken from the Send Task Management Request Destination ID parameter.
c) The Buffer Address parameter is the location of the SMS STRUCTURE data structure.
d) The Length parameter is the length of the SMS STRUCTURE data structure.

If the Return Code parameter indicates FUNCTION FAILED or INVALID FIELD, then the protocol layer Service Response value is SERVICE DELIVERY OR TARGET FAILURE. If the Return Code parameter indicates FUNCTION SUCCESSFUL, then the protocol layer Service Response returns a value of FUNCTION COMPLETE.

It is the responsibility of the initiator's S2P protocol layer to make sure that only one Task Management request for a given target is outstanding. Additional Task Management requests received from an application client while another Task Management request is active for an I_T nexus are held until the outstanding Task Management function completes.

D.6.1 Abort Task

Service Response = ABORT TASK (Destination ID, LUN, Tag ||)

The SSA-S2P protocol builds an ABORT TAG SMS into a data structure entitled SMS STRUCTURE as follows:

a) The SMS CODE and S2P CODE fields are set to 82h and 30h respectively.
b) The TAG field is set to 00h.
c) The RETURN PATH ID field is set to the Return path id parameter returned by GET RETURN PATH ID (see E.2.1).
d) The TAG2 field is set to the Tag parameter.

D.6.2 Abort Task Set

Service Response = ABORT TASK SET (Destination ID, LUN ||)

The SSA-S2P protocol builds an ABORT SMS into a data structure entitled SMS STRUCTURE as follows:

a) The SMS CODE and S2P CODE fields are set to 82h and 31h respectively.
b) The TAG field is set to 00h.
c) The RETURN PATH ID field is set to the Return path id parameter from the invocation of the GET RETURN PATH ID (see E.2.1).
d) The LUNTRN field is set to the LUN parameter.

D.6.3 Clear SCA

Service Response = CLEAR SCA (Destination ID, LUN ||)

The SSA-S2P protocol builds a CLEAR SCA SMS into a data structure entitled SMS STRUCTURE as follows:

a) The SMS CODE and S2P CODE fields are set to 82h and 34h respectively.
b) The TAG field is set to 00h.
c) The Logical Unit Identifier parameter is passed in the LUNTRN field.

D.6.4 Clear Task Set

Service Response = CLEAR TASK SET (Destination ID, LUN ||)

The SSA-S2P protocol builds a CLEAR QUEUE SMS into a data structure entitled SMS STRUCTURE as follows:

a) The SMS CODE and S2P CODE fields are set to 82h and 32h respectively.
b) The TAG field is set to 00h.
c) The RETURN PATH ID field is set to the Return path id parameter from the invocation of the GET RETURN PATH ID (see E.2.1).
d) The LUNTRN field is set to the LUN parameter.

D.6.5 Target Reset

Service Response = TARGET RESET (Destination ID ||)

The SSA-S2P protocol builds a TARGET RESET SMS into a data structure entitled SMS STRUCTURE as follows:

a) The SMS CODE and S2P CODE fields are set to 82h and 33h respectively.
b) The TAG field is set to 00h.
c) The RETURN PATH ID field is set to the Return path id parameter from the invocation of the GET RETURN PATH ID (see E.2.1).

D.6.6 Terminate Task

Service Response = TERMINATE TASK (Destination ID, LUN, Tag ||)

Implementation of the terminate Task protocol service is optional for a logical unit in SAM and is not supported in SSA-S2P. The Service Response is always unconditionally returned as FUNCTION REJECTED.

D.7 Task management protocol services

D.7.1 Send Task Management Request

Send Task Management Request (Object Address, Function Identifier ||)

Upon receipt of the Send Task Management Request protocol service, the protocol layer performs the actions defined in the appropriate Task Management Function clause.

D.7.2 Task Management Request Received

Task Management Request Received (Object Address, Function Identifier ||)

Upon receipt of an SMS RECEIVED transport service indication and validation of SMS contents, the target's S2P layer generates a Task Management Request Received protocol service indication with the appropriate parameters.

D.7.3 Task Management Function Executed

Task Management Function Executed (Object Address, Service Response ||)

Upon receipt of a Task Management Function Executed protocol service, the protocol layer builds a partial SCSI RESPONSE SMS and instructs the transport layer to transmit it.

The SSA-S2P protocol builds an SCSI Response SMS into a data structure entitled SMS STRUCTURE as follows:

a) The SMS CODE and S2P CODE fields are set to 82h and 03h respectively.
b) The TAG field is set to 00h.
c) The return code field shall be set to value based on the value of Service Response as shown in Table D.6.

Table D.6 - RETURN CODE field value based on Service Response

Service Response parameter value	RETURN CODE field value
FUNCTION REJECTED	INVALID PARAMETER
FUNCTION COMPLETE	FUNCTION SUCCESSFUL
SERVICE DELIVERY OR TARGET FAILURE	FUNCTION FAILED

The SSA-S2P protocol then invokes the transport layer service of SEND SMS as follows:

SEND SMS (Control, Destination ID, Buffer Address, Length || Return Code)

a) The control field is always 00h.
b) The Destination ID parameter is the Destination ID parameter of the Task Function.
c) The Buffer Address parameter is the location of the SMS STRUCTURE data structure.
d) The Length parameter is the length of the SMS STRUCTURE data structure.

D.7.4 Received Function-Executed

Received Function-Executed (Object Address, Service Response ||)

Upon receipt of an SMS RECEIVED transport service indication and validation of SMS contents, the initiator's S2P protocol layer generates a Received Function Executed protocol service confirmation with the appropriate parameters. The return code field maps into the Service Response parameter as defined in Table D.6.

Annex E
(Informative)
Transport service handling

This clause describes how the protocol layer handles Transport Services. Refer to the SSA-TL standard for more information on the transport layer services.

E.1 Transport services invoked by the Transport Layer

The following clauses describe:

a) Transport Service requests directed to the Transport Layer from the S2P layer;
b) Transport Service indications received by the S2P layer from the Transport Layer;
c) Transport Service confirmations received by the S2P layer from the transport layer,
d) The responsibilities of the S2P layer.

E.1.1 Object definitions

The parameters for the Transport services are as follows (see the SSA-TL standard for a detailed description):

a) The Buffer Address parameter is a memory location where data is to be stored to or retrieved from.
b) The Byte Offset parameter is a 4 byte value indicating the offset of this data transfer based on the Buffer Address parameter.
c) The Byte Count parameter is a 4 byte value indicating the number of bytes for a data transfer. In the case of a read or write that transfers only a part of the data for an SCSI command, the byte count field only indicates the amount of data to be transferred by the request.
d) The Control parameter indicates the CONTROL field for or from the SMS frame as described in the SSA-TL standard.
e) The Length parameter is a 1 byte count of the number of bytes in the DATA field of the frame to be sent or received.
f) The Channel parameter is a 2 byte value indicating the initiator channel for the target to use as the channel component in the ADDRESS field of the read Data frame(s).
g) The Return Path ID parameter identifies a logical path and is used to set the value of the RETURN PATH ID field when required in the SMS.
h) The In Order parameter is a 1 bit flag that indicates that the data is written into the buffer space as received (In Order set to 0b), or written into the buffer space in order with the Buffer Address location containing the byte of data with an offset of zero (In order set to 1b). The In Order parameter allows the transport layer to use split data transfers even if the Protocol layer does not support splits.
i) The Replace4-7 parameter is a flag to indicate whether the transport layer replaces bytes 4 through 7 of the SMS with the Return Path ID associated with the Unique ID parameter.
j) The Return Code parameter is an indicator of the success or failure of the service to execute properly. Valid Return Code values are listed in SSA-TL.

E.1.2 SMS RECEIVED

SMS RECEIVED (Control, Buffer Address, Length ||)

The SMS RECEIVED transport service input parameters are Control, Buffer Address and Length. There are no output parameters. It is initiated by the Transport layer when an SMS is received. The contents of the SMS are stored in the location designated by the Buffer Address parameter. The Length parameter specifies the length of the SMS payload.

In response to an SMS RECEIVED transport service indication, the protocol shall perform the SMS Validation process defined in 7.2. If the SMS is valid, it is processed as defined in the clauses shown in Table E.1.

Table E.1 - SMS Processing Clause References

SMS Received	Clause
SCSI COMMAND	D.4.2 SCSI Command received protocol service indication
SCSI STATUS	D.4.4 Command Complete Received protocol service confirmation
ABORT TAG	D.6.1 Abort Task
ABORT	D.6.2 Abort Task Set
CLEAR SCA	D.6.3 Clear SCA
CLEAR QUEUE	D.6.4 Clear Task Set
DEVICE RESET	D.6.5 Target Reset
SCSI RESPONSE	D.7.4 Received Function-Executed

E.1.3 INITIATOR READ

INITIATOR READ (Tag, Byte Offset, Byte Count || Return path id, Buffer Address, Start Count, Threshold Count, Return Code)

When the initiator transport layer receives a DATA READY SMS, it generates the INITIATOR READ transport service indication. This causes the Initiator protocol layer to perform the following actions.

a) If the Command Byte Count or Buffer Address entries referenced by the Tag parameter in the OUTSTANDING COMMANDS data structure are null, then return FUNCTION FAILED in the Return Code parameter.

b) If the Byte Offset plus the Byte Count exceeds the Command Byte Count entry referenced by the Tag parameter in the OUTSTANDING COMMANDS data structure, then return FUNCTION FAILED in the Return Code parameter.

c) If neither of the conditions in a) nor b) occurred, then set up the data transfer to begin at the location pointed to by the Buffer Address entry in the OUTSTANDING COMMANDS data structure plus the Byte Offset parameter. The length of the transfer shall be the Byte Count parameter.

d) The Protocol layer returns the Initiator Read Buffer Address parameter from the Data-In Buffer Address parameter of the appropriate entry in the OUTSTANDING COMMANDS data structure.

e) The use of the Start Length and Threshold Length parameters are implementation dependent. These parameters may be used when the initiator does not have Byte Count worth of buffer space at the time of receipt of this service (but it does have Start Count), and intends to use the transport layer to throttle the data transfer. Refer to the Initiator Read service in the SSA-TL standard for more information on this function.

E.1.4 READ DATA COMPLETED

READ DATA COMPLETED (|| Tag, Unique ID, LUN, Return Code)

The receipt of a READ DATA COMPLETED transport service confirmation by the target's S2P layer results in the generation of a Data Sent protocol service confirmation (see annex D.5.2).

E.1.5 INITIATOR WRITE

INITIATOR WRITE (Tag, Byte Count, Byte Offset || Unique ID, Buffer Address, Return Code)

The invocation of the Target Write transport service causes the Initiator protocol to receive an Initiator Write transport service indication. This causes the Initiator protocol to perform the following functions.

a) If the Command Byte Count or Buffer Address entries referenced by the Tag parameter in the OUTSTANDING COMMANDS data structure are null, then return FUNCTION FAILED in the Return Code parameter.

b) If the Byte Offset plus the Byte Count exceeds the Command Byte Count entry referenced by the Tag parameter in the OUTSTANDING COMMANDS data structure, then return FUNCTION FAILED in the Return Code parameter.

c) The Protocol layer returns the Initiator Write Buffer Address parameter from the Data-Out Buffer Address parameter of the appropriate entry in the OUTSTANDING COMMANDS data structure.

d) If neither of the conditions in a) nor b) occurred, then set up the data transfer to begin at the location pointed to by the Initiator Write Buffer Address plus the Byte Offset parameter. The length of the transfer shall be the Byte Count parameter.

E.1.6 WRITE DATA COMPLETED

WRITE DATA COMPLETED (|| Tag, Unique ID, LUN, Return Code)

The receipt of a WRITE DATA COMPLETED transport service confirmation by the target's S2P layer results in the generation of a Data-Out Received protocol service confirmation (see annex D.5.4).

E.1.7 QUIESCE

QUIESCE (Source ID || Return Code)

The receipt of a Quiesce transport service indication causes the following actions to occur:

a) Generate a Abort Task Set protocol service indication for all Logical Units supported by the Target.

b) If any Service Response was SERVICE DELIVERY OR TARGET FAILURE, then return a FUNCTION FAILED Return Code value, otherwise return a FUNCTION SUCCESSFUL Return Code value.

E.2 Transport Services invoked by the Protocol Layer

E.2.1 GET RETURN PATH ID

GET RETURN PATH ID (Source ID || Return path id, Return Code)

The GET RETURN PATH ID transport service request is used during the construction of SMS data structures to map a Source ID to an Return path id.

E.2.2 GET UNIQUE ID

GET UNIQUE ID (Return Path ID || Unique ID, Return Code)

The GET UNIQUE ID transport service request is used to map the Return path id in a received SMS to a Source ID.

E.2.3 SEND SMS

SEND SMS (Control, Unique ID, Buffer Address, Length, Replace4-7 || Return Code)

The SEND SMS request is used to send SMS data structures.

E.2.4 TARGET READ

TARGET READ (Unique ID, Tag, Buffer Address, Byte Offset, Byte Count, Start Count, Threshold Count || Return Code)

The TARGET READ transport service request is used to initiate read data transfers initiated by the Send Data-In protocol service request.

E.2.5 TARGET FAST READ

TARGET FAST READ (Unique ID, Byte Count, Channel || Return Code)

The TARGET FAST READ request is only used by those implementations that support it (see annex D.1.5).

E.2.6 TARGET WRITE

TARGET WRITE (Unique ID, Tag, Buffer Address, Byte Offset, Byte Count || Return Code)

The TARGET WRITE transport service request is used to request write data transfers by the Receive Data-Out protocol service request.